

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมตำแหน่งลูกบอลบนระนาบ

BALL PLATE CONTROL SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
คณะวิศวกรรมศาสตร์
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2550

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

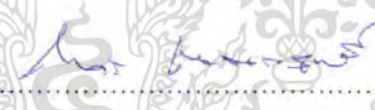
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

ระบบควบคุมตำแหน่งลูกบอลบนระนาบ
BALL PLATE CONTROL SYSTEM

ผู้จัดทำ

นางสาวกนก ทาสี	47010003
นายสฤทธิภัทร์ บู่ทอง	47010375
นายวศกร นาวีกิจ	47010674


.....อาจารย์ที่ปรึกษา
(ผศ.ดร. ดาวร เบนยงนราสุทธิ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมตำแหน่งลูกบอลบนระนาบ

โดย

นางสาวกนก	ทาสี	47010003
นายสฤทธิภัสร์	นุ่ทอง	47010375
นายวศกร	นาวีกิจ	47010674

อาจารย์ที่ปรึกษา

ศ.ดร. ถาวร เบญจนราสุทธ

ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอทฤษฎี และการออกแบบระบบควบคุม เพื่อควบคุมตำแหน่งลูกบอลบนระนาบ โดยโครงสร้างของระบบประกอบด้วย คอมพิวเตอร์ กล้องดิจิตอลวีดีโอ และวงจรรีเลย์ทรอนิกส์ที่เกี่ยวข้อง เป้าหมายคือบังคับลูกบอลให้ไปยังตำแหน่งที่ต้องการตามที่กำหนดบนระนาบและรักษาตำแหน่งของบอลนั้น โดยการควบคุมมอเตอร์กระแสตรงที่ต่อกับระนาบ

ขั้นตอนดำเนินการ เริ่มจากการออกแบบและประกอบ โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ศึกษาและออกแบบวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง รวมทั้งการทำงานของไมโครคอนโทรลเลอร์ แล้วจึงเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาซีเพื่อทำการตรวจจับตำแหน่งของลูกบอลจากข้อมูลภาพจากกล้องดิจิตอลวีดีโอ หาค่าสัญญาณควบคุมที่เหมาะสม โดยอาศัยแนวคิดระบบควบคุมพีไอดี ระบบควบคุมแบบมุนำ และระบบควบคุมป้อนกลับสเตอร์ท่วมกับตัวสังเกตสเตรต ผลจากการทดลองพบว่าระบบควบคุมที่ได้ออกแบบนั้นสามารถควบคุมระบบให้ลูกบอลไปอยู่ ณ ตำแหน่งที่ต้องการได้ โดยมีค่าความผิดพลาดอยู่ในระดับที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BALL PLATE CONTROL SYSTEM

By

Miss Kanok Thasi 47010003

Mr. Saritthiphath Boothong 47010375

Mr. Wasakorn Naweekit 47010674

Advisor

Asst. Prof. Dr. Taworn Benjanarasuth

Academic Year 2007

ABSTRACT

This thesis presents theories and implementation procedures of a ball plate control system using a digital video camera as a visual feedback sensor. The system composes of the ball and plate structure, computer, digital video camera and interfacing circuits. The goal is to move the ball to any desired position on the plate and maintain the ball at that position by controlling the DC motors connected to the plate.

The project has been conducted as in the following steps. First, the structure of the ball plate control system is designed and constructed. Second, necessary electronic circuits are studied and implemented. Then, the computer program written in Visual Basic language is composed. Its tasks are to detect the ball position from digital video camera data and to compute suitable control signal based on PID control system, lead control system and state feedback control system with state observer. Lastly, the experiments are conducted. The results show that the ball position can be controlled to the desired set point with acceptable error.

กิตติกรรมประกาศ

การจัดทำปฏิญญาพันธบัตรฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก ผศ.ดร. ถาวร เบญจนราสุทธิ ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเชื้อเพื่ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆ ที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ รศ.ดร. จงกต งามวิวิทย์ ที่คอยถามความคืบหน้าอยู่ตลอดเวลา อาจารย์สองเมื่อนันทขว้าง ที่ให้คำปรึกษาในส่วนของ การปรับปรุงโครงสร้างของโครงการ อาจารย์รัชวิชชัย คำศรี ที่ให้คำปรึกษาในส่วนของ การเขียนโปรแกรม และภาควิชาวิศวกรรมระบบควบคุม ที่ให้การสนับสนุนด้านงบประมาณในการทำโครงการ

ขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน และถามไถ่ความคืบหน้าของโครงการอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดมาเป็นแรงบันดาลใจที่ดีที่สุดทำให้โครงการนี้สำเร็จสมบูรณ์ได้

ผู้จัดทำ

นางสาวอนก

ทาสี

นายสุฤษธิ์ภัสร์

บุ๋ทอง

นายวศกร

นาวิกิจ

สารบัญ

	หน้าที่
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของการจัดทำโครงการ	2
1.2 เนื้อหาในปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	3
2.1 ทฤษฎีการประมวลผลภาพ	3
2.1.1 การกำหนดสี	3
2.1.2 องค์ประกอบของสี	4
2.1.3 ภาพดิจิทัล	5
2.1.3.1 การแทนภาพด้วยข้อมูลดิจิทัล	5
2.1.3.2 ลักษณะการจัดเก็บข้อมูลภาพดิจิทัล	6
2.1.4 การประมวลผลภาพเชิงตัวเลข	7
2.1.5 สัญญาณข้อมูลภาพจากดิจิทัลวิดีโอ	7
2.1.6 ข้อมูลภาพชนิดบิตแมป	7
2.2 ตัวควบคุมแบบพีไอดี	8
2.2.1 ตัวควบคุมแบบพี	8
2.2.2 ตัวควบคุมแบบไอ	11
2.2.3 ตัวควบคุมแบบดี	13
2.2.4 ระบบควบคุมแบบพีไอ	15
2.2.5 ระบบควบคุมแบบพีดี	17
2.2.6 ตัวควบคุมแบบพีไอดี	18
2.4 ตัวควบคุมแบบป้อนกลับสเตทและตัวสังเกตสเตท	19
2.4.1 ตัวควบคุมแบบป้อนกลับสเตทเวลาต่อเนื่อง	19
2.4.2 ตัวสังเกตสเตทเวลาต่อเนื่อง	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ระบบเซอร์โวป้อนกลับสเตทเวลาต่อเนื่อง	22
2.4.4 ระบบเซอร์โวป้อนกลับสเตทร่วมกับตัวสังเกตสเตทเวลาดิสรีต	25
บทที่ 3 หลักการออกแบบ	28
3.1 กล้องดิจิทัลวีดีโอ	29
3.2 คอมพิวเตอร์	29
3.3 ไมโครคอนโทรลเลอร์	30
3.4 โครงสร้างทางกายภาพของระบบ	34
บทที่ 4 ผลการทดลอง	36
4.1 การระบุตำแหน่งมุมของระนาบ โดยเทียบกับค่าการแปลงค่าระดับอะนาลอกเป็นดิจิทัล	36
4.2 การควบคุมตำแหน่งมุมของระนาบ โดยอาศัยตัวควบคุมแบบพีไอดี	36
4.3 การประมาณฟังก์ชันถ่ายโอนของระบบควบคุมมุมของระนาบ	39
4.4 การระบุตำแหน่งของลูกบอล โดยเทียบพิกเซล	40
4.5 ผลการทดลองควบคุมตำแหน่งลูกบอล โดยอาศัยตัวควบคุมแบบตัวชดเชยชนิดมุนนำ	40
4.6 ผลการทดลองควบคุมตำแหน่งลูกบอล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	44
4.6.1 การทดลองควบคุมตำแหน่งลูกบอล โดยใช้ตัวควบคุมแบบป้อนกลับสเตท	46
4.6.2 การเขียน โปรแกรมชดเชยผลกระทบของแรงเสียดทานสถิต	49
4.6.3 การเขียน โปรแกรมกำหนดช่วงค่าผิดพลาดที่ยอมรับได้	53
บทที่ 5 บทวิจารณ์และสรุปผล	61
5.1 สรุปผลการทดลอง	61
5.2 ปัญหาที่พบและแนวทางการแก้ไข	61
5.3 ข้อเสนอแนะ	62
ภาคผนวก ก โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม	63
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	86
ข-1 เอกสารคู่มือการใช้งาน PIC 16F877A	86
ข-2 เอกสารคู่มือการใช้งาน IC MAX 232	87
ข-3 เอกสารคู่มือการใช้งาน L293D	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้า
1.1 โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนระนาบ	1
2.1 กล้องสี RGB	4
2.2 การกำหนดช่วงจำกัดของเอาต์พุต	9
2.3 ผลตอบสนองของตัวควบคุมแบบพี	10
2.4 บล็อกไดอะแกรมของระบบควบคุมแบบพี	11
2.5 ผลตอบสนองของตัวควบคุมแบบไอ	12
2.6 บล็อกไดอะแกรมของระบบควบคุมแบบไอ	12
2.7 ผลตอบสนองของตัวควบคุมแบบดี	13
2.8 บล็อกไดอะแกรมของระบบควบคุมแบบดี	14
2.9 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอ	15
2.10 การตอบสนองของตัวควบคุมแบบพีไอ	15
2.11 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี	17
2.12 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอดี	18
2.13 บล็อกไดอะแกรมของระบบป้อนกลับสเตท	19
2.14 บล็อกไดอะแกรมของระบบวงปิดแบบตัวสังเกตสเตท	21
2.15 บล็อกไดอะแกรมของระบบเซอร์โวป้อนกลับสเตท	23
2.16 บล็อกไดอะแกรมของระบบเซอร์โวป้อนกลับสเตทและตัวสังเกตสเตทเวลา discrete	26
3.1 หลักการทำงานของระบบควบคุมตำแหน่งลูกบอลบนระนาบ	28
3.2 กล้องดิจิทัลวีดีโอ	29
3.3 โพลีชาร์ตการควบคุมตำแหน่งมุมของระนาบโดยไม่โครคอนโทรลเลอร์	31
3.4 โพลีชาร์ตการหาตำแหน่งลูกบอลบนระนาบ	32
3.5 โพลีชาร์ตการควบคุมตำแหน่งมุมระนาบในไมโครคอนโทรลเลอร์	33
3.6 โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนระนาบ	35
3.7 การยัดคอมพิวเตอร์	35
3.8 การยัดโพแทสเซียมไอเมเตอร์	35
4.1 ผลตอบสนองของระบบควบคุมมุมระนาบโดยใช้ตัวควบคุมแบบ พี พีไอ พีดี และพีไอดี	37
4.2 ผลตอบสนองระบบควบคุมมุมระนาบจาก 0 องศาไปยังมุมบวกโดยอาศัยตัวควบคุมแบบพีไอดี	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา VI และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3	ผลตอบสนองของระบบควบคุมมูรณะจาก 0 องศาไปยังมุมลบโดยอาศัยตัวควบคุมแบบพีไอดี	38
4.4	การประมาณแบบจำลองมอเตอร์จากผลตอบสนองของระบบควบคุมมูรณะจาก -2 องศา ไปยังมุม 2 องศา	30
4.5	ผลตอบสนองของการจำลองระบบควบคุมตำแหน่งลูกบอลบนระนาบ จากตำแหน่ง 0 พิกเซล ไปยัง ตำแหน่ง 148 พิกเซล โดยอาศัยตัวชดเชยชนิดมูรณะ	40
4.6	สัญญาณควบคุมการจำลองระบบควบคุมตำแหน่งลูกบอลบนระนาบ จากตำแหน่ง 0 พิกเซล ไปยัง ตำแหน่ง 148 พิกเซล โดยอาศัยตัวชดเชยชนิดมูรณะ	41
4.7	ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยัง ตำแหน่ง 148 พิกเซล โดยอาศัยตัวชดเชยชนิดมูรณะ	42
4.8	สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จาก ตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 148 พิกเซล โดยอาศัยตัวชดเชยชนิดมูรณะ	42
4.9	ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 128 พิกเซล โดยอาศัยตัวชดเชยชนิดมูรณะ	43
4.10	สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จาก ตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 128 พิกเซล โดยอาศัยตัวชดเชยชนิดมูรณะ	43
4.11	การจำลองผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	44
4.12	การจำลองสัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	45
4.13	การจำลองสัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	45
4.14	ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	46
4.15	สัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	47
4.16	สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จาก	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา VII และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	
4.17 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	48
4.18 สัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	48
4.19 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	49
4.20 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	50
4.21 สัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	50
4.22 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	51
4.23 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	51
4.24 สัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	52
4.25 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	52
4.26 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท	53
4.27 สัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	54
4.28 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	54
4.29 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท	55

4.30	สัญญาณควบคุมตัวแปรสแตกความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบ ป้อนกลับสแตก	55
4.31	สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จาก ตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสแตก	56
4.32	ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไป ยังตำแหน่งเป้าหมายต่างๆแบบต่อเนื่องในแนวแกน X โดยตัวควบคุมแบบ ป้อนกลับสแตก	57
4.33	สัญญาณควบคุมตัวแปรสแตกความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน X โดยตัวควบคุมแบบป้อนกลับสแตก	57
4.34	สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ ไปยังตำแหน่งเป้าหมายต่างๆแบบต่อเนื่องในแนวแกน X โดยอาศัยตัวควบคุม แบบป้อนกลับสแตก	58
4.35	ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไป ยังตำแหน่งเป้าหมายต่างๆแบบต่อเนื่องในแนวแกน Y โดยอาศัยตัวควบคุมแบบ ป้อนกลับสแตก	58
4.36	สัญญาณควบคุมตัวแปรสแตกความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน Y	59
4.37	สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ ไปยังตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน Y โดยอาศัยตัวควบคุมแบบ ป้อนกลับสแตก	59
4.38	ผลตอบสนองระบบควบคุมตำแหน่งลูกบอลบนระนาบในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล และมีการรบกวนระบบจากภายนอกโดย ตัวควบคุมแบบป้อนกลับสแตก	60
4.39	ผลตอบสนองระบบควบคุมตำแหน่งลูกบอลบนระนาบในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซลและมีการรบกวนระบบจากภายนอกโดย ตัวควบคุมแบบป้อน กลับสแตก	60
ก.1	หน้าต่างเชื่อมต่อกับผู้ใช้ในโปรแกรมการตรวจสอบตำแหน่งลูกบอลบนระนาบ	63

สารบัญตาราง

ตารางที่

หน้า

4.1 ผลทดลองหาความสัมพันธ์ระหว่างค่าระดับสัญญาณดิจิทัลกับมุมของระนาบ

36



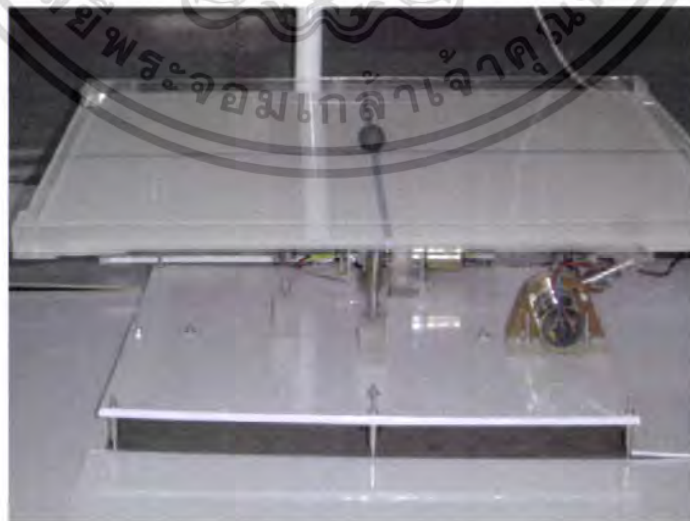
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา **X** และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

การศึกษาในสาขาวิชาวิศวกรรมระบบควบคุม เป็นการศึกษาและประยุกต์ทฤษฎีต่างๆ เพื่อการออกแบบระบบควบคุมให้มีเสถียรภาพ และมีสมรรถนะตามความต้องการหรือให้เป็นไปตามข้อกำหนด อาทิ ระบบควบคุมตำแหน่งลูกบอลบนจาน ระบบควบคุมความเร็วและตำแหน่งของมอเตอร์ไฟฟ้ากระแสตรง และระบบควบคุมการไหลและความสูงของระดับน้ำ จากโครงการที่ผ่านมามีการศึกษาระบบควบคุมตำแหน่งของลูกบอลบนจานซึ่งเป็นการควบคุมในหนึ่งมิติ ดังนั้นเพื่อเป็นการพัฒนาความรู้ด้านระบบควบคุม จึงคิดที่จะศึกษาการควบคุมในระบบสองมิติ เพื่อให้เป็นไปตามวัตถุประสงค์ที่ต้องการนี้จึงจำเป็นต้องมีการศึกษาการจำลองระบบควบคุมด้วยสมการทางคณิตศาสตร์ การออกแบบตัวควบคุมหรือตัวชดเชยแบบต่างๆ การศึกษาวงจรถอนิกส์และอิเล็คทรอนิกส์กำลังรวมถึงศึกษาการเขียนโปรแกรมคอมพิวเตอร์การศึกษาและการเลือกอุปกรณ์วัดและแปลงสัญญาณ ตลอดจนการบูรณาการเรื่องที่ศึกษาเหล่านี้ในการประยุกต์ใช้กับระบบควบคุมทางกายภาพจริงซึ่งนับเป็นสิ่งจำเป็นในการศึกษาในสาขาวิชานี้

ระบบควบคุมตำแหน่งลูกบอลบนระนาบ เป็นตัวอย่างหนึ่งของระบบควบคุมที่นิยมในการศึกษา เพราะเป็นระบบวงเปิดที่ไม่มีเสถียรภาพ จึงจำเป็นต้องมีการควบคุมเพื่อให้มีเสถียรภาพ โดยเลือกใช้การควบคุมแบบป้อนกลับอัตโนมัติ ซึ่งมีหลายแบบ เช่น การป้อนกลับสเตต (State Feedback) การควบคุมแบบพีไอดี และแบบอื่นๆ เป้าหมายของการควบคุมคือการทำให้ลูกบอลหยุดนิ่ง ณ ตำแหน่งใดตำแหน่งหนึ่งบนระนาบได้ตามต้องการ



รูปที่ 1.1 โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของบทนี้จะกล่าวถึงวัตถุประสงค์ของการจัดทำโครงการ และรวมถึงขั้นตอน การศึกษาและการจัดทำโครงการ โดยภาพรวม ตลอดจนรายละเอียดต่างๆที่จะกล่าวถึงในบทต่อไป

1.1 วัตถุประสงค์ของการจัดทำโครงการ

1. ศึกษาทฤษฎีระบบควบคุมเพื่อออกแบบตัวควบคุมแบบต่างๆและสามารถนำมาใช้งานจริง
2. ศึกษาและทดลองการใช้งานอุปกรณ์ต่างๆในทางระบบควบคุม
3. ศึกษาและสร้างระบบควบคุมตำแหน่งลูกบอลบนระนาบ ซึ่งประกอบด้วยอุปกรณ์ที่ ประกอบเป็นโครงสร้างทางกล อุปกรณ์ทางอิเล็กทรอนิกส์ของระบบควบคุมอุปกรณ์วัด ตำแหน่งของลูกบอลบนระนาบ ซึ่งในโครงการนี้ใช้กล้องดิจิทัลวิดีโอ (Webcam หรือ Digital Video) รวมถึงส่วนของการควบคุมและประมวลผล โดยโครงการนี้ใช้คอมพิวเตอร์และโปรแกรม วิชาลเบสิก (Visual Basic)

1.2 เนื้อหาในปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ ขั้นตอนการศึกษา การจัดทำโครงการ พร้อมทั้งรายละเอียดของแต่ละบทในปริญญานิพนธ์ บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้องในการจัดทำโครงการ เช่น การประมวลผลภาพ การออกแบบระบบ ควบคุมแบบพีไอดี การออกแบบระบบควบคุมป้อนกลับสเตตและตัวสังเกตสเตต และการ ประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ บทที่ 3 หลักการออกแบบ นำเสนอการประกอบ โครงสร้าง ของระบบ รวมถึงแนวคิดในการออกแบบระบบควบคุม บทที่ 4 การทดลอง เป็นส่วนการทดสอบ องค์ประกอบต่างๆ ในระบบ ตลอดจนการทดลองระบบควบคุมตำแหน่งลูกบอลบนระนาบ บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนา โครงการนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีและความรู้ที่เกี่ยวข้อง

การออกแบบและสร้างระบบควบคุมตำแหน่งลูกบอลบนระนาบนั้น ต้องศึกษาองค์ความรู้และทฤษฎีซึ่งจะนำไปใช้ในโรงงาน โดยในระบบควบคุมตำแหน่งลูกบอลบนระนาบนั้น มีส่วนที่สำคัญหลายส่วน ดังนั้นในบทนี้จะกล่าวถึงองค์ประกอบต่างๆ ที่จะนำไปใช้งานจริง ประกอบด้วย การประมวลผลภาพ ไมโครคอนโทรลเลอร์ และตัวควบคุมแบบต่างๆ

2.1 ทฤษฎีการประมวลผลภาพ

ความรู้เกี่ยวกับการประมวลผลภาพ นับเป็นส่วนสำคัญที่จำเป็นจะต้องศึกษาเพื่อให้สามารถระบุตำแหน่งของลูกบอลที่อยู่บนระนาบได้ โดยต้องทำความเข้าใจเกี่ยวกับหลักการของการประมวลผลภาพเบื้องต้น ซึ่งประกอบไปด้วยหัวข้อต่างๆ ดังนี้

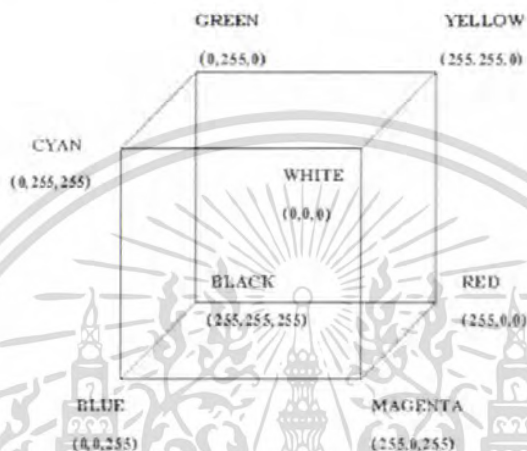
2.1.1 การกำหนดสี

ในคอมพิวเตอร์ สีทุกสีจะประกอบด้วย 3 สีพื้นฐาน คือ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) หรืออาจเรียกว่า RGB ซึ่งสามารถที่จะสังเคราะห์สีอื่นๆ ได้เกือบทุกสีโดยการผสมสีพื้นฐานทั้งสามนี้ในสัดส่วนต่างๆ การผสมสีตั้งอยู่บนพื้นฐานของความเข้มขององค์ประกอบของ RGB ซึ่งเรียกว่า RGB model

สีแดงสีน้ำเงินและสีเขียว โดยสัดส่วนของสีพื้นฐานข้างต้นแสดงในลักษณะ (Red,Green,Blue) โดยในแต่ละสีนิยมนำเสนอด้วยค่าขนาด 1 ไบต์หรือเท่ากับ 8 บิต โดยค่าที่เล็กที่สุดคือค่า 0 สอดคล้องกับการไม่มีสีพื้นฐานสีนั้นเลย และค่าที่ใหญ่ที่สุดคือค่า 255 บ่งบอกถึงความเข้มสูงสุดของสีพื้นฐานสีนั้น อาทิเช่น สัดส่วน (0,0,0) สอดคล้องกับสีดำ เพราะสีพื้นฐานทั้งสามสีไม่มีความเข้มเลย ในขณะที่สัดส่วน (255,255,255) สอดคล้องกับสีขาว ส่วนสีอื่นๆ ซึ่งเกิดจากการรวมกันนั้นมีได้มากมาย เช่น (255,0,0) จะเป็นสีแดงบริสุทธิ์ (0,255,255) เป็นสีน้ำเงินเข้ม และ (0,128,128) เป็นสีน้ำเงินอ่อนขึ้น ดังนั้นการรวมกันที่เป็นไปได้ของสีพื้นฐานในกรณีนี้จะมีขนาด 24 บิต หรือมีค่าได้ $256 \times 256 \times 256$ ซึ่งเท่ากับ 16,777,216 ระดับสี จึงนิยามจัดเก็บค่าสีในรูปแบบตัวแปร Long Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการของการสร้างสีซึ่งมีสีพื้นฐาน 3 สีนี้ ตั้งอยู่บนพื้นฐานของกล่องสี RGB ดังแสดงในรูปที่ 2.1 โดยมุมของกล่องสีนี้จะสอดคล้องกับสีต่างๆดังรูป ปริมาณของสีพื้นฐานทั้งสามสีในสีใดๆจะมีค่าตามแกน x y และ z ตามลำดับ สีที่เป็นส่วนกลับ (Complementary Colors) สามารถคำนวณได้อย่างง่ายๆ โดยการลบค่าของสีจาก 255 ตัวอย่างเช่นสี (0,0,255) เป็นสีน้ำเงินบริสุทธิ์เป็นสีส่วนกลับของสีเหลืองบริสุทธิ์ที่สอดคล้องกับค่า (255,255,0) ซึ่งทั้งสองสีดังกล่าวอยู่ที่มุมตรงข้ามกันของกล่องสี ส่วนมุมอื่นๆ ก็เช่นเดียวกัน



รูปที่ 2.1 กล่องสี RGB

องค์ประกอบของสี ที่มุมของกล่องสีจะมีความเข้มข้นสูงที่สุด หรือ ไม่มีความเข้มข้นเลยเท่านั้น เมื่อเคลื่อนที่จากมุมหนึ่งไปยังอีกมุมหนึ่งตามขอบที่เหมือนกัน จะมีเพียงแค่องค์ประกอบเดียวเท่านั้นที่เปลี่ยนแปลง ยกตัวอย่างเช่นเมื่อเคลื่อนที่จากมุมสีเขียวไปยังมุมสีเหลือง องค์ประกอบของสีแดงจะเปลี่ยนแปลงจาก 0 ถึง 255 โดยเมื่อเคลื่อนที่ระหว่างมุมของสีคู่นี้ จะได้ทุกโทนสี จากเขียวไปเหลือง

2.1.2 องค์ประกอบของสี

ในการประมวลผลภาพบางประเภท อาจต้องการอ่านค่าขององค์ประกอบสีของแต่ละพิกเซล แล้วแยกองค์ประกอบของ RGB เพื่อนำไปใช้งานแยกกัน ดังนั้นจึงจำเป็นต้องมีฟังก์ชันซึ่งทำหน้าที่แยกองค์ประกอบของสีทั้งสามซึ่งถูกเก็บในตัวแปร Long Integer ดังที่กล่าวไว้ในหัวข้อ 2.1.1 ในตัวแปรดังกล่าวจะประกอบด้วย 4 ไบต์ โดยไบต์แรก (The Most Significant Byte) เก็บ 0 ต่อมาเป็นองค์ประกอบของสีน้ำเงิน เขียว และแดงตามลำดับ ค่าซึ่งสอดคล้องกับสัดส่วน RGB (64,32,192) มีค่าเป็น 12,591,168 ซึ่งค่านี้ดูไม่เหมือนกับจำนวนเต็มเลข ถ้านำเสนอองค์ประกอบของสีด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขฐานสิบหก (Hexadecimal Format) ซึ่งในตัวอย่างนี้มีค่าเป็น C02040h จะเป็นเลขฐานสิบหกของ 12,591,168 ในเลขฐานสิบ จะเห็นว่าสองหลักสุดท้ายของฐานสิบหกสอดคล้องกับองค์ประกอบสีแดง ($40h = 64$) สองหลักถัดมาสอดคล้องกับองค์ประกอบสีเขียว ($20h = 32$) และสองหลักที่มีความสำคัญสูงสุดสอดคล้องกับค่าสีน้ำเงิน ($C0h = 192$) ด้วยเหตุนี้เลขฐานสิบหกจึงถูกใช้มากในการกำหนดค่าสี

2.1.3 ภาพดิจิทัล

ในการที่จะนำภาพหนึ่งมาประมวลด้วยเครื่องคอมพิวเตอร์นั้น ภาพดังกล่าวจำเป็นต้องถูกนำเสนอหรือแทนด้วยตัวเลข แต่ภาพที่ได้มาโดยส่วนมากจะเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชัน $f(x,y)$ ที่ต่อเนื่องในระนาบสองมิติ โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพที่ตำแหน่ง (x,y) ซึ่งเรียกว่าระดับสีเทา (Gray Level) สำหรับภาพขาวดำและจะเป็นสัดส่วนกับระดับค่า RGB สำหรับภาพสี

2.1.3.1 การแทนภาพด้วยข้อมูลดิจิทัล

ภาพแบบดิจิทัลเป็นภาพที่ถูกแปลงมาจากแบบอนาลอกเพื่อให้อยู่ในรูปของตัวเลข โดยภาพจะถูกแบ่งพื้นที่เป็นสี่เหลี่ยมเล็กๆ ที่เรียกว่าจุดภาพ หรือ พิกเซล (Pixel) ซึ่งถูกระบุตำแหน่งของพิกเซลโดยพิกัด (x,y) และในแต่ละพิกเซลจะระบุข้อมูลระดับสีหรือค่า RGB สำหรับภาพสี และระดับความเข้มสำหรับระดับสีเทา โดยการแปลงข้อมูลแบบดิจิทัลสามารถทำได้ดังขั้นตอนต่อไปนี้

เมื่อนำภาพแบบอนาลอกที่ต้องการประมวลผลมาผ่านส่วนที่เรียกว่าดิจิติเซอร์ (Digitizer) ซึ่งมีหน้าที่ในการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิทัล อุปกรณ์ประเภทนี้ได้แก่ กล้องดิจิติเซอร์ ซึ่งจะทำให้ภาพที่นำเสนอด้วยฟังก์ชันที่ต่อเนื่อง $f(x,y)$ ให้กลายเป็นฟังก์ชันที่ไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเรียกว่าการสุ่ม (Sampling) และผ่านการควอนไทซ์ (Quantization) ก็จะได้ข้อมูลที่เป็นดิจิทัล

สมมติว่าสัญญาณภาพต่อเนื่อง $f(x,y)$ ได้ถูกแปลงให้อยู่ฟังก์ชันที่ไม่ต่อเนื่องซึ่งจะสามารถนำเสนอในรูปแบบดิทช์ และสมมติว่ามีขนาด $N \times N$ ได้ตามสมการ (2.1)

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & f(N-1,2) & \dots & f(N-1,N-1) \end{matrix} \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในห้องปฏิบัติการเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยฟังก์ชัน (x,y) ทางขวาของสมการจะเรียกว่า ภาพดิจิทัล และทุกสมาชิกของเมตริกซ์จะเรียกว่า พิกเซล จากกระบวนการสร้างภาพดิจิทัลข้างต้นจะเห็นได้ว่าเราสามารถทราบขนาดของความละเอียดของภาพเป็น $N \times N$ พิกเซล และในกรณีภาพขาวดำ ขนาดของข้อมูลภาพที่เป็นดิจิทัลจะมีขนาดดังสมการ (2.2)

$$B = N \times N \times M \text{ บิต} \quad (2.2)$$

เมื่อ B คือขนาดของข้อมูลภาพที่เป็นดิจิทัล (บิต)

M คือจำนวนบิตที่ใช้ในการแทนข้อมูลภาพ 1 พิกเซล สามารถหาได้จากสมการที่ (2.3)

$$G = 2^M \quad (2.3)$$

เมื่อ G คือจำนวนระดับของสีเทา (Gray Scale) ที่ต้องการใช้ในการเก็บภาพ

2.1.3.2 ลักษณะการจัดเก็บข้อมูลภาพดิจิทัล

โดยทั่วไปแล้วข้อมูลภาพสีหรือภาพขาวดำจะมีค่าตั้งแต่สองระดับขึ้นไป โดยเฉพาะในกรณีภาพขาวดำ นิยมใช้ระดับความเข้มของจุดภาพระดับของสีเทาท่ากับ 256 ระดับ ซึ่งทำให้จุดภาพมีค่าอยู่ในช่วง 0 ถึง 255 โดยใช้พื้นที่เก็บข้อมูล 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ($M=8$) ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มหรือระดับสีจำนวนสูงๆ อาจจะต้องการจำนวนบิตสำหรับเก็บข้อมูลมากกว่า 8 บิต อาจเป็น 16 หรือ 24 บิต โดยที่ความเข้มของจุดภาพเท่ากับ 65,536 และ 16,777,216 โดยจะแยกให้ชัดเจนดังนี้

1. ภาพ 2 ระดับคือมีเพียงแต่จุดขาวกับจุดดำเท่านั้น โดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต
2. ภาพ 16 ระดับ คือในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต สามารถทำให้แสดงได้ 16 ระดับความเข้มหรือระดับสี ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือภาพขาวดำ
3. ภาพ 256 ระดับ คือในแต่ละจุดภาพจะมีขนาดของข้อมูลขนาด 8 บิต ซึ่งสามารถทำให้แสดงได้ 256 ระดับความเข้มหรือระดับสี ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ภาพทิวทัศน์ (True Color) ในแต่ละจุดภาพจะมีขนาดของข้อมูล 24 บิต สามารถทำให้แสดงภาพสีได้เหมือนจริงที่สุดเพราะสามารถแสดงสีได้ถึง 16,777,216 ระดับสี

2.1.4 การประมวลผลภาพเชิงตัวเลข

โดยทั่วไปวิธีการประมวลผลภาพเชิงตัวเลขโดยคอมพิวเตอร์นั้น อาจแบ่งโดยคร่าวเป็นสองระดับด้วยกัน คือ การประมวลผลภาพในระดับต่ำ (Low Level Processing) และการประมวลผลภาพในระดับสูง (High Level Processing) การประมวลผลภาพในระดับต่ำจะเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมดเพื่อหาตัวแปรต่างๆ มาอธิบายข้อมูลภาพ มีวัตถุประสงค์เพื่อนำตัวแปรเหล่านั้นไปใช้ในการประมวลผลในระดับสูงต่อไป

ส่วนการประมวลผลภาพในระดับสูง เป็นการนำผลลัพธ์ที่ได้จากการประมวลผลภาพในระดับต่ำมาตีความหรือเพื่อส่งข้อมูลให้คอมพิวเตอร์สามารถรู้และเข้าใจได้ ดังนั้นความแตกต่างที่สำคัญของการประมวลผลภาพระดับต่ำและระดับสูง คือ ข้อมูลที่นำเข้ามาในการประมวลผล โดยการประมวลผลภาพในระดับต่ำจะใช้เวลาของจุดโดยตรง ส่วนการประมวลผลระดับสูงนั้น ข้อมูลที่นำมาประมวลผลจะถูกแสดงในรูปของสัญลักษณ์ ซึ่งจะแสดงถึงสิ่งต่างๆ ที่อยู่ในภาพ เช่นขนาดหรือ รูปร่างของวัตถุในภาพ

2.1.5 สัญญาณข้อมูลภาพจากดิิจิตอลวิดีโอ

การส่งสัญญาณข้อมูลจากภาพวิดีโอมีลักษณะการส่งที่เป็นลำดับภาพเดี่ยวหรือเฟรม (Frame) ที่ฉายต่อเนื่องกัน เช่นระบบวิดีโอ NTSC จะส่งด้วยอัตราเร็ว 30 เฟรมต่อวินาที โดยดิิจิตอลวิดีโอแต่ละเฟรมจะเป็นข้อมูลดิิจิตอลในลักษณะเมตริกซ์ซึ่งแต่ละจุดจะเรียกว่าพิกเซล มีค่าระดับของความเข้มสี โดยทั่วไปจะใช้ระดับของสีเทาที่มีค่าตั้งแต่ 0 ถึง 255 โดย 0 จะแทนความมืดมากที่สุดและ 255 จะแทนความสว่างมากที่สุด

2.1.6 ข้อมูลภาพชนิดบิตแมป

รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป (Bitmap) เป็นรูปแบบไฟล์มาตรฐานสำหรับภาพกราฟิกบนวินโดวส์ซึ่งจะใช้ในการตัดต่อ หรือทำสำเนาภาพต่างๆ ลงบนคลิปบอร์ด (Clipboard) เมื่อเวลาจัดเก็บไฟล์ที่มีนามสกุล .BMP เป็นฟอร์แมตที่สามารถใช้สร้างเป็นวอลล์เปเปอร์ (Wallpaper) ได้อีกด้วย

โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมปจะประกอบด้วย 3 ส่วน คือ
เอกสารนี้เป็นเอกสารทศงานวิเศษหรือการเชิงงานเพื่อการศึกษเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ข้อมูลเฮดเดอร์ (Header)

2. ข้อมูลจานสี (Palette)

3. ข้อมูลภาพ (Data)

1. ข้อมูลเฮดเดอร์ คือข้อมูลที่อยู่บริเวณส่วนหัวของไฟล์ ซึ่งประกอบด้วยข้อมูลที่บอกรายละเอียดต่างๆ ของภาพ เช่น ความกว้าง ความยาวของภาพ จำนวนสี จำนวนบิต ความละเอียดของภาพ
2. ข้อมูลจานสี เป็นค่าแม่สี RGB ของภาพ
3. ข้อมูลภาพ เป็นส่วนเก็บข้อมูลสีของแต่ละพิกเซลของภาพ โดยข้อมูลแรกจะเป็นค่าสีของพิกเซลที่อยู่แถวล่างสุดที่ตำแหน่งซ้ายสุดข้อมูลลำดับต่อไปจะเรียงจากทางขวาจากแถวล่างจนถึงแถว

2.2 ตัวควบคุมแบบพีไอดี

ตัวควบคุม (Controller) เป็นส่วนประกอบหนึ่งในระบบควบคุมแบบป้อนกลับ ซึ่งมีสัญญาณความผิดพลาดเป็นอินพุตและมีเอาต์พุตเป็นอินพุตของอุปกรณ์ขับเคลื่อนสำหรับพลานท์ (Plant) ที่ต้องการควบคุม ในโครงงานนี้จะใช้ตัวควบคุมแบบพีไอดี ซึ่งระบบควบคุมแบบพีไอดีที่ใช้นั้นประกอบด้วยตัวควบคุมแบบต่างๆดังต่อไปนี้

1. ตัวควบคุมแบบพี
2. ตัวควบคุมแบบไอ
3. ตัวควบคุมแบบดี

ในระบบบางประเภท มีความจำเป็นที่ต้องปรับปรุงสมรรถนะของระบบควบคุม ซึ่งสามารถทำได้โดยการใช้ส่วนประกอบอื่นเพิ่มเติมเข้ากับระบบควบคุม การที่ปรับเปลี่ยนสมรรถนะของระบบควบคุมป้อนกลับแบบนี้เรียกว่า การชดเชย (Compensation)

2.2.1 ตัวควบคุมแบบพี

ในตัวควบคุมแบบพี เอาต์พุตของตัวควบคุม $u(t)$ จะเป็นสัดส่วนกับอินพุตของตัวควบคุม และถ้ากำหนดสัญญาณอินพุตให้กับตัวควบคุมเป็นค่าความผิดพลาด $e(t)$ ซึ่งเป็นฟังก์ชันของเวลา จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

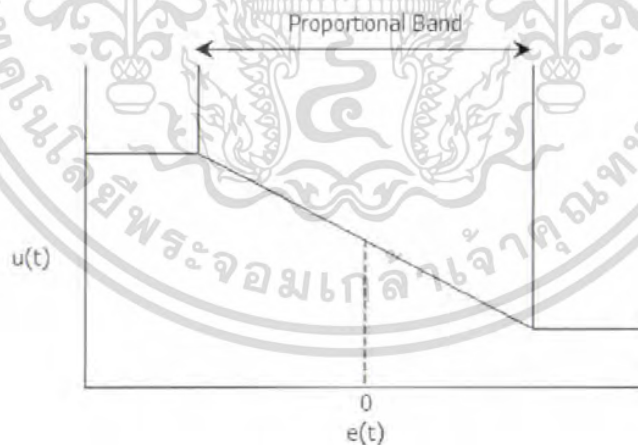
$$u(t) = K_p e(t) \quad (2.4)$$

เมื่อ K_p เป็นค่าคงที่เรียกว่า อัตราขยายแบบสัดส่วน (Proportional Gain)

จะพบว่าเอาต์พุตที่ออกจากตัวควบคุมแบบพี เป็นสัดส่วน โดยตรงกับขนาดของความผิดพลาด ในขณะที่กำลังพิจารณา ทำให้ฟังก์ชันถ่ายโอนของตัวควบคุม $G_c(s)$ มีค่าเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \quad (2.5)$$

ดังนั้นการควบคุมด้วยตัวควบคุมแบบนี้จะเป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้น การที่ได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง ทำให้เกิดเอาต์พุตที่มีขนาดใหญ่จากตัวควบคุมในเวลาสั้น อย่างไรก็ตามการที่ให้อัตราขยายของฟังก์ชันในทางปฏิบัติอาจกำหนดไว้ในบางช่วงของสัญญาณความผิดพลาดเท่านั้น หรืออาจกำหนดให้ตัวควบคุมมีค่าเอาต์พุตจะมีลักษณะดังรูปที่ 2.2 และการกำหนดเอาต์พุตของตัวควบคุมแบบพี ช่วงที่มีการกำหนดสัดส่วนนี้ จะเรียกว่า Proportional Band



รูปที่ 2.2 การกำหนดช่วงจำกัดของเอาต์พุต

การกำหนด Proportional Band นี้จะช่วยให้สัญญาณเอาต์พุตมีค่าจำกัดไม่ไปสู่ค่าอนันต์ ทั้งทางด้านบวกและทางด้านลบ และเมื่อตัวควบคุมมีเอาต์พุตสูงที่สุดที่เป็นไปได้ค่าหนึ่ง ก็นิยมที่จะกำหนดเอาต์พุตค่าใดๆ เป็นร้อยละของค่าสูงสุดที่เป็นไปได้ ดังนั้นการเปลี่ยนแปลงค่าเอาต์พุตของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุม 100% ก็หมายถึงว่า เอาท์พุทจะเปลี่ยนจากค่าต่ำสุดที่เป็นไปได้ ไปเป็นค่าสูงสุดที่เป็นไปได้ ซึ่งจะได้ว่า

$$K_p = \frac{100}{\text{ProportionalBand}} \quad (2.6)$$

เนื่องจากเอาท์พุทของตัวควบคุมจะเป็นสัดส่วนกับอินพุท ดังนั้น ถ้าหากอินพุทมีลักษณะเป็นสัญญาณระดับ (Step) เอาท์พุทที่ได้ก็จะมีลักษณะเป็นสัญญาณระดับ เช่นกัน โดยลักษณะของกราฟแสดงอินพุทและเอาท์พุทจะมีสัดส่วนที่แน่นอนค่าหนึ่ง แสดงดังรูปที่ 2.3 โดยรูปนี้แสดงถึงการตอบสนองของตัวควบคุม เมื่ออินพุทอยู่ในช่วง Proportional Band



รูปที่ 2.3 ผลตอบสนองของตัวควบคุมแบบพี

ในทางปฏิบัติตัวควบคุมแบบพีนี้ มีลักษณะเหมือนกับเครื่องขยายสัญญาณรูปแบบหนึ่ง ซึ่งอาจจะเป็นในลักษณะของอุปกรณ์ทางไฟฟ้า หรืออาจจะเป็นเครื่องขยายสัญญาณเชิงกล เช่น กานรณะนาบ ก็ได้ ลักษณะของตัวควบคุมแบบพี ได้ตั้งบล็อกไดอะแกรมแสดงในรูปที่ 2.4 และได้ฟังก์ชันถ่ายโอนระบบเป็น

$$G_o(s) = K_p G_p(s) \quad (2.7)$$

เมื่อ $G_p(s)$ เป็นฟังก์ชันถ่ายโอนของพลานท์ที่ต้องการควบคุม

ข้อเสียของระบบที่มีตัวควบคุมแบบพี คือ ไม่มีการเพิ่มอินทิเกรตหรือ $1/s$ ในส่วน Forward Path ซึ่งหมายความว่า ถ้าพลานท์เป็นชนิด 0 (Type-0) ตัวควบคุมจะไม่เปลี่ยนแปลงชนิด ทำให้ระบบเป็นเอกสิทธิ์แบบเอกสิทธิ์ถึงแม้จะมีค่าที่ตรงกันเพื่อที่จะทำการแก้ไขที่นั้น เมื่อผู้ดูแลที่นั้นใช้ระบบจะเอื้อมหน้าในการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิด 0 เหมือนเดิม ซึ่งจะทำให้เกิดความผิดพลาดที่สถานะคงตัว (Steady State Error) จากรูปที่ 2.4 ระบบควบคุมแบบพี ซึ่งมีการป้อนกลับหนึ่งหน่วย มีฟังก์ชันถ่ายโอนของระบบเป็น

$$G(s) = \frac{K_p G_p(s)}{1 + K_p G_p(s)} \quad (2.8)$$

พหุนามคุณลักษณะเป็น $1 + K_p G_p(s)$ ซึ่งมีรากเปลี่ยนไปตามค่าของ K_p



รูปที่ 2.4 บล็อกไดอะแกรมของระบบควบคุมแบบพี

2.2.2 ตัวควบคุมแบบไอ

ในระบบควบคุมแบบไอ เอาท์พุทของตัวควบคุมจะเป็นสัดส่วนกับการอินทิเกรตสัญญาณของค่าผิดพลาดเทียบกับเวลา หรือ

$$u(t) = K_i \int_0^t e(t) dt \quad (2.9)$$

เมื่อ K_i เป็นค่าคงที่ เรียกว่า อัตราขยายแบบอินทิเกรต (Integral Gain) ซึ่งจะมีหน่วยเป็น sec^{-1} รูปที่ 2.5 แสดงลักษณะการตอบสนองของตัวควบคุมแบบไอ เมื่อได้รับสัญญาณอินพุตแบบระดับค่า $u(t)$ ณ เวลา t ใดๆ หมายถึงพื้นที่ใต้กราฟของสัญญาณอินพุตจากเวลา 0 ถึง t คูณด้วยอัตราขยายอินทิเกรต ดังนั้นเนื่องจากเริ่มมีสัญญาณอินพุตแบบระดับ เอาท์พุทที่ออกจากตัวควบคุมจึงมีค่ามากขึ้นเรื่อยๆ ด้วยอัตราที่คงที่ในลักษณะรูปสัญญาณความชัน

จากสมการที่ (2.9) ได้ฟังก์ชันถ่ายโอนของตัวควบคุมแบบไอเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ระบบควบคุมแบบโอทีแสดงใน รูปที่ 2.5 มีฟังก์ชันถ่ายโอน Forward-Path เป็น $\frac{K_i}{s} G_p(s)$ และมีฟังก์ชันถ่ายโอนวงปิดเป็น

$$G(s) = \frac{\frac{K_i}{s} G_p(s)}{1 + \frac{K_i}{s} G_p(s)} \quad (2.11)$$



รูปที่ 2.6 บล็อกไดอะแกรมของระบบควบคุมแบบโอ

ดังนั้นสามารถพิจารณาถึงข้อได้เปรียบของการควบคุมแบบโอ ได้จากฟังก์ชันถ่ายโอน Forward - Path ซึ่งจะเห็นว่า ระบบควบคุมแบบโอ จะเพิ่มชนิดของระบบขึ้นมา 1 ระดับ เช่น หากพลานท์เดิมเป็นชนิด 0 ระบบควบคุมแบบโอจะเป็น ชนิด 1 ซึ่งทำให้ระบบมีความผิดพลาดที่สถานะคงตัวเป็นศูนย์เทียบต่อสัญญาณอินพุตแบบระดับ อย่างไรก็ตาม การเพิ่มโพลที่ $s = 0$ ในระบบวงเปิดและไม่มีการเพิ่มซีโรให้กับระบบควบคุม ทำให้ความแตกต่างระหว่างจำนวนโพล (n) และจำนวนซีโร (m) เพิ่มขึ้นอีก 1 ซึ่งมีผลให้มุมของเส้นอะซิมีโทค (Asymptotic Angles) ของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นทางรากลดลง และจุดตัดเส้นอะซิมโททบนแกนจริง (Centroid) เคลื่อนไปทางครึ่งขวาของระนาบ s มากขึ้นมีผลทำให้ความเสถียรสัมพัทธ์ของระบบลดลง

2.2.3 ตัวควบคุมแบบดี

ตัวควบคุมแบบดี นี้เอาที่พุดของตัวควบคุมจะเป็นสัดส่วนกับอนุพันธ์ของอินพุตความผิดพลาด เทียบต่อเวลา นั่นคือ

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.12)$$

เมื่อ K_d คือ อัตราขยายอนุพันธ์ (Derivative Gain) และมีหน่วยเป็นวินาที



รูปที่ 2.7 ผลตอบสนองของตัวควบคุมแบบดี

จากรูปที่ 2.7 แสดงผลตอบสนองของตัวควบคุมแบบดี เมื่อสัญญาณอินพุตเป็นสัญญาณแบบความชัน จากสมการที่ (2.12) จะเห็นว่าตัวควบคุมแบบดีจะให้สัญญาณส่งออกจกตัวควบคุมมีค่ามากเมื่อค่าอินพุตความผิดพลาดมีแนวโน้มเพิ่มในอัตราที่สูงก่อนที่จะเกิดความผิดพลาดขึ้นมากจริงๆ

อย่างไรก็ตามหากความผิดพลาดมีค่าคงที่ก็จะไม่มีการสะสมค่าความผิดพลาดแม้ว่าค่าความผิดพลาดจะมีมากก็ตาม ทำให้การควบคุมแบบดีนี้ ไม่เปลี่ยนแปลงต่อค่าความผิดพลาดที่คงที่หรือ

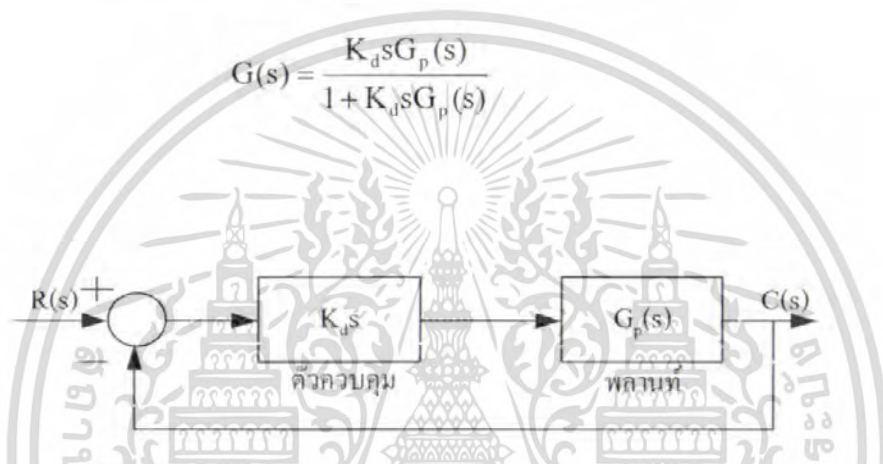
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงอย่างช้าๆ ดังนั้นการควบคุมแบบนี้จึงไม่นิยมใช้เพียงลำพัง แต่มักจะใช้ควบคู่กับการควบคุมแบบอื่น ฟังก์ชันถ่ายโอนของตัวควบคุมแบบดี จะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_d s \tag{2.13}$$

ดังนั้น สำหรับระบบควบคุมแบบดีจะได้ฟังก์ชันถ่ายโอนวงปิดเป็น

$$G(s) = \frac{K_d s G_p(s)}{1 + K_d s G_p(s)} \tag{2.14}$$



รูปที่ 2.8 บล็อกไดอะแกรมของระบบควบคุมแบบดี

ถ้าหากว่าระบบเป็นแบบชนิด 1 หรือสูงกว่า การควบคุมแบบดีจะลดเทอม s ในพหุนามส่วนของฟังก์ชันถ่ายโอนของ Forward - Path ลง ซึ่งจะเปลี่ยนชนิดของระบบลง 1 อย่างไม่รู้ก็ตามดังที่ได้กล่าวก่อนหน้านี้แล้วว่า การควบคุมแบบดีนี้จะไม่ใช่เพียงลำพังแต่จะใช้ร่วมกับตัวควบคุมแบบอื่น เพราะเมื่อใช้การควบคุมแบบดี ทำให้เพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้น

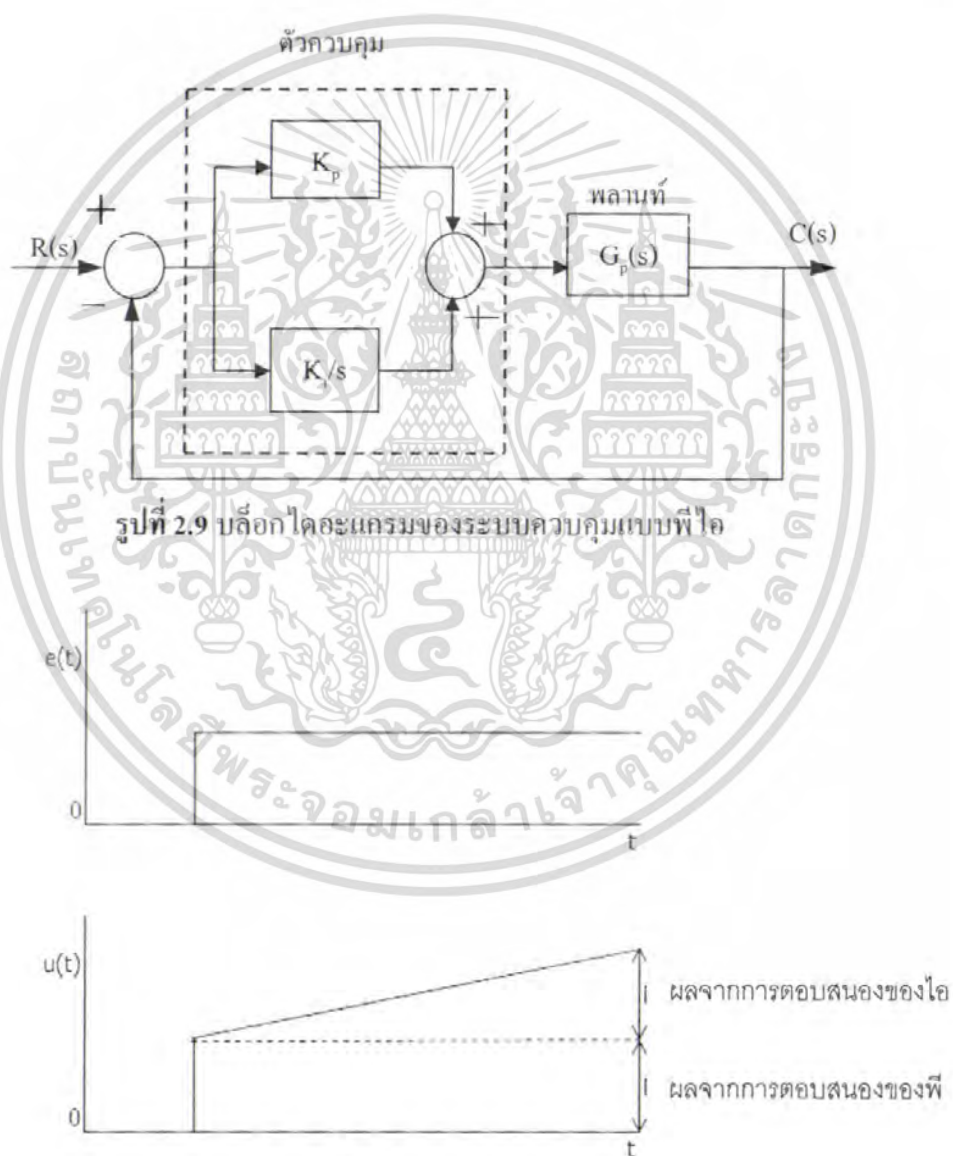
ในทางปฏิบัติการนำการควบคุมแบบดีไปใช้นั้นค่อนข้างจะลำบาก ดังนั้น ในทางปฏิบัติโดยทั่วไปจะเป็นการประมาณการควบคุมแบบดีโดยใช้ตัวชดเชยแบบมูมนำ (Lead Compensator)

2.2.4 ระบบควบคุมแบบพีไอ

การที่ระบบควบคุมมีความเสถียรสัมพัทธ์ลดลงเมื่อใช้การควบคุมแบบไอ สามารถที่จะแก้ไขได้ในระดับหนึ่งโดยการใช้การควบคุมแบบพีร่วมกับแบบไอเรียกว่า การควบคุมแบบพีไอ (Proportional plus Integral Control) ซึ่งลักษณะของระบบควบคุมเป็นตามรูปที่ 2.9

สำหรับระบบดังกล่าวจะมีเอาต์พุตของตัวควบคุมเป็น

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (2.15)$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.10 แสดงเอาต์พุตของตัวควบคุมที่ได้รับเมื่อมีอินพุตเป็นแบบระดับ จากสมการที่ (2.15) ฟังก์ชันถ่ายโอนของตัวควบคุมแบบพีไอเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s}$$

$$= \frac{sK_p + K_i}{s}$$

$$= \frac{K_i}{s} + \frac{K_p}{s}$$
(2.16)

นิยามค่าเวลาคงตัวอินทิกรัล (Integral Time Constant) เป็น

$$\tau_i = \frac{K_p}{K_i}$$
(2.17)

ดังนั้นจะได้

$$G_c = \frac{K_p[s + (1/\tau_i)]}{s}$$
(2.18)

และจะทำให้ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) = G_c(s)G_p(s)$$
(2.19)

$$= \frac{K_p[s + (1/\tau_i)]G_p(s)}{s}$$

จะเห็นว่ามีซีโรที่ $s = -1/\tau_i$ และ โพลที่ $s = 0$ เพิ่มให้กับฟังก์ชันถ่ายโอนของระบบเมื่อใช้การ

ควบคุมแบบพีไอ การที่เพิ่มตัวประกอบ s เข้ากับพหุนามส่วนของฟังก์ชันถ่ายโอนวงเปิดเป็นการ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มชนิดของระบบขึ้นไป 1 จึงทำให้ระบบควบคุมแบบพีไอนี้ไม่มีความผิดพลาดที่สภาวะคงตัว สำหรับอินพุตแบบระดับ นอกจากนั้นการที่เพิ่มซีโรให้กับระบบไปพร้อมๆกัน ก็ทำให้ความแตกต่างระหว่างจำนวนโพล (n) และจำนวนซีโร (m) มีค่าคงที่ ดังนั้นมุมของเส้นอะซิมโทตสำหรับเส้นทางรากมีค่าคงเดิม แต่จุดตัดของเส้นอะซิมโทตบนแกนจริง จะเคลื่อนที่เข้าหาจุดกำเนิดมากขึ้น เนื่องจาก

$$\text{จุดตัดของเส้นอะซิมโทตบนแกนจริง} = (\text{ผลรวมของโพล} - \text{ผลรวมของซีโร}) / (n - m)$$

ดังนั้นจุดตัดของเส้นอะซิมโทตจะเปลี่ยนไปเท่ากับ $-(1/\tau_i)/(n - m)$ ส่งผลให้คุณภาพของระบบลดลง อย่างไรก็ตาม การลดลงของความเสถียรสัมพัทธ์นี้จะน้อยกว่าการที่ใช้การควบคุมแบบไอเพียงอย่างเดียว

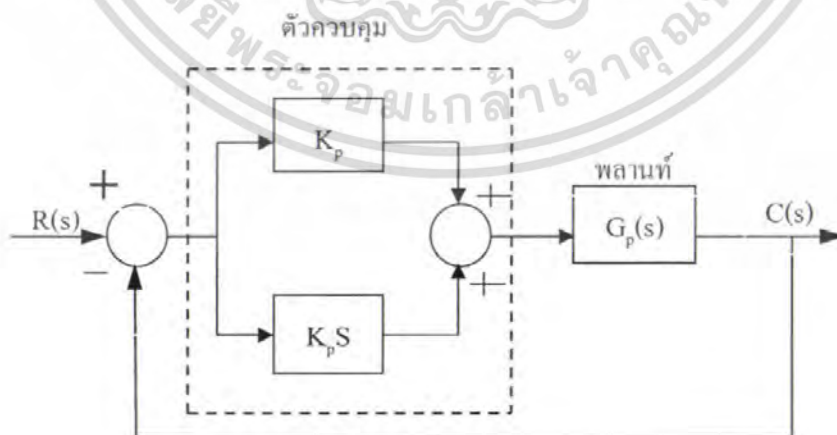
2.2.5 ระบบควบคุมแบบพีดี

การควบคุมแบบดีใช้ร่วมกับการควบคุมแบบพี หรือเรียกว่าการควบคุมแบบพีดี (Proportional plus Derivative Control) ดังที่แสดงใน รูปที่ 2.11 ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) = (K_p + K_d s) G_p(s)$$

$$G_o(s) = K_p (1 + \tau_d s) G_p(s)$$

(2.20)

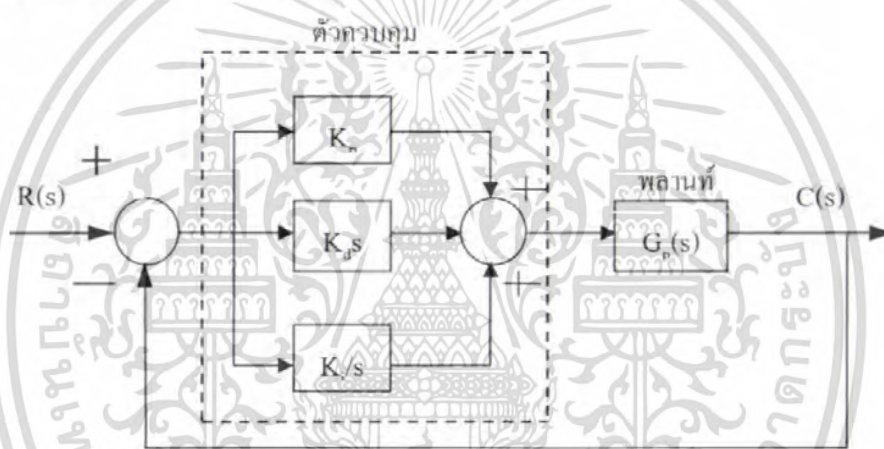


รูปที่ 2.11 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี

เมื่อ $\tau_d = \frac{K_p}{K_d}$ คือ ค่าเวลาคงตัวอนุพันธ์ (Derivative Time Constant) ซึ่งในการควบคุมแบบนี้ จะมี จีโรเพิ่มขึ้นที่ $s = -\tau_d$ และจะเห็นว่าไม่มีการเปลี่ยนแปลงชนิดของระบบ ทำให้ไม่มีการเปลี่ยนแปลงค่าความผิดพลาดที่สภาวะคงตัว

2.2.6 ตัวควบคุมแบบพีไอดี

ตัวควบคุมแบบพีไอดีนี้เป็นการรวมตัวควบคุมแบบพีรวมกับแบบไอและรวมกับแบบดี (Proportional plus Integral plus Derivative Control) หรือที่เรียก การควบคุมแบบ 3 เทอม (Three – term Control) ระบบจะมีลักษณะดัง รูปที่ 2.12 ทำให้เอาที่พูดของตัวควบคุมเมื่อรับอินพุตเป็น ความผิดพลาด $e(t)$ ดังนี้



รูปที่ 2.12 บล็อกโคะแกรมของระบบควบคุมแบบพีไอดี

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.21)$$

ฟังก์ชันถ่ายโอนของตัวควบคุม จะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2.22)$$

ซึ่งสามารถจัดรูปได้เป็น

$$G_c(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2.23)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันถ่ายโอนระบบแบบเปิดของระบบควบคุมแบบฟีดแบ็คที่แสดงใน รูปที่ 2.12 จะเป็น

$$G_o(s) = G_c(s)G_p(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s\right) G_p(s)$$

$$G_o(s) = \frac{K_p (\tau_i \tau_d s^2 + \tau_i s + 1)}{\tau_i s} G_p(s) \quad (2.24)$$

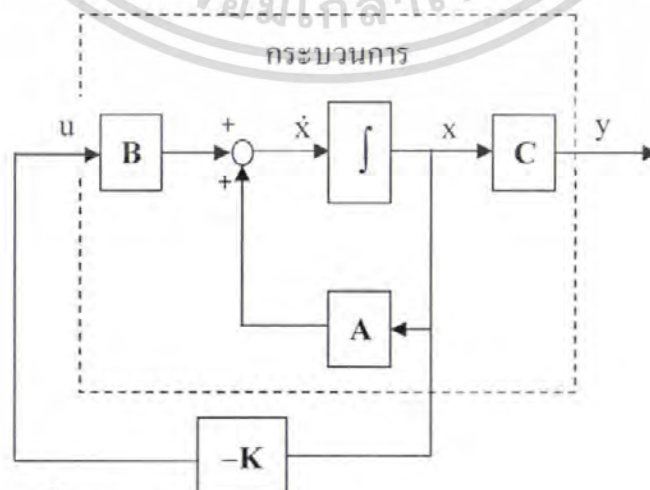
ดังนั้น การควบคุมแบบฟีดแบ็ค จะเพิ่มจำนวนขั้วโข่งให้กับระบบเท่ากับ 2 ตัวและเพิ่มจำนวนโพล 1 ตัว และทำให้ชนิดระบบเพิ่มขึ้น 1

2.4 ตัวควบคุมแบบป้อนกลับสเตตและตัวสังเกตสเตต

ตัวควบคุมแบบป้อนกลับสเตตและตัวสังเกตสเตตเป็นการควบคุมที่นิยมใช้กันอยู่แพร่หลายอีกประเภทหนึ่ง ซึ่งสามารถใช้ได้กับระบบควบคุมเวลาต่อเนื่องและระบบควบคุมเวลาไม่ต่อเนื่อง ซึ่งจะอธิบายดังต่อไปนี้

2.4.1 ตัวควบคุมแบบป้อนกลับสเตตเวลาต่อเนื่อง

โครงสร้างของระบบป้อนกลับสเตตเวลาต่อเนื่องแสดงได้ดังบล็อกไดอะแกรมใน รูปที่ 2.13 ซึ่งประกอบด้วยกระบวนการที่นำเสนอด้วยแบบจำลองสเตตเวลาต่อเนื่อง (Continuous-time State Space Model) ดังสมการที่ (2.25) และ (2.26)



รูปที่ 2.13 บล็อกไดอะแกรมของระบบป้อนกลับสเตต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\dot{x} = Ax + Bu \quad (2.25)$$

$$y = Cx \quad (2.26)$$

เมื่อ $x = [x_1 \ x_2 \ \dots \ x_n]^T$ คือ ตัวแปรสถานะของระบบ และตัวควบคุมการป้อนกลับสเตตซึ่งมีกฎการควบคุมดังอธิบายได้โดยสมการที่ (2.27)

$$u = -Kx = -[k_1 \ k_2 \ \dots \ k_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.27)$$

เมื่อ $K = [k_1 \ k_2 \ \dots \ k_n]$ คือ เมตริกซ์อัตราขยายป้อนกลับสเตต (State Feedback Gain Matrix) ดังนั้นจะได้ระบบวงปิดดังสมการที่ (2.28)

$$\begin{aligned} \dot{x} &= Ax + B(-Kx) \\ &= (A - BK)x \end{aligned} \quad (2.28)$$

ซึ่งมีสมการคุณลักษณะของระบบวงปิดเป็น

$$\det(sI - (A - BK)) = 0 \quad (2.29)$$

การออกแบบเมตริกซ์อัตราขยายป้อนกลับสเตต K อาจออกแบบโดยวิธีการวางโพลโดยหากกำหนดให้โพลวงปิดที่ต้องการอยู่ที่ s_1, s_2, \dots, s_n จะได้สมการคุณลักษณะของระบบที่ต้องการออกแบบคือ

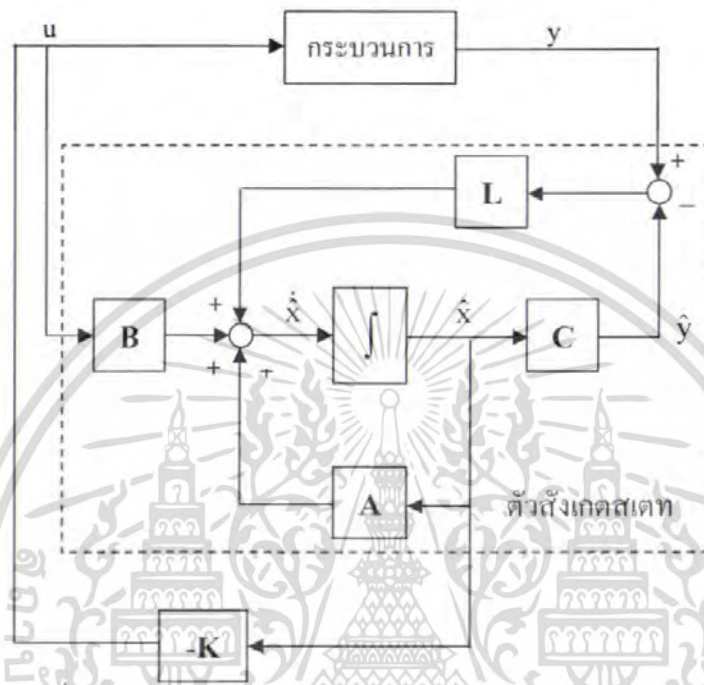
$$\alpha_c(s) = (s - s_1)(s - s_2) \dots (s - s_n) = 0 \quad (2.30)$$

ซึ่งสามารถหาแก้หาค่า K ได้จากความสัมพันธ์ของสมการที่ (2.29) และ (2.30)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ตัวสังเกตสถานะเวลาต่อเนื่อง

เนื่องจากการออกแบบระบบจริงนั้นตัวแปรสถานะ x ไม่สามารถวัดค่าเพื่อนำมาป้อนกลับได้ จึงต้องใช้วิธีการออกแบบตัวสังเกตสถานะ (State Observer หรือ State Estimator) เพิ่มขึ้นมาในระบบ ป้อนกลับสถานะ แสดงได้ดังบล็อกไดอะแกรมใน รูปที่ 2.14



รูปที่ 2.14 บล็อก ไดอะแกรมของระบบวงปิดแบบตัวสังเกตสถานะ

ซึ่งตัวสังเกตสถานะเวลาต่อเนื่องสามารถอธิบายด้วยสมการที่ (2.31) และ (2.32)

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (2.31)$$

$$\hat{y} = C\hat{x} \quad (2.32)$$

เมื่อ $L = [l_1 \ l_2 \ \dots \ l_n]^T$ คือ เมตริกซ์อัตราขยายตัวสังเกตสถานะ (State Observer Gain Matrix)

นิยามค่าผิดพลาดตัวสังเกต (Error in Estimator) \tilde{x} โดย

$$\tilde{x} = x - \hat{x} \quad (2.33)$$

จากสมการที่ (2.31) (2.32) และ (2.33) จะได้สมการสถานะของค่าผิดพลาดสังเกต ดังสมการที่ (2.34) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\dot{\bar{x}} = (A - LC)\bar{x} \quad (2.34)$$

โดยการออกแบบ ต้องการให้ค่า \bar{x} ที่ประมาณนั้นเข้าสู่ค่า x ของระบบจริงได้อย่างรวดเร็ว นั่นคือ \bar{x} เข้าสู่ศูนย์ได้อย่างรวดเร็ว ซึ่งสามารถทำได้โดยการปรับค่าเมตริกซ์อัตราขยายตัวสังเกตสเกต L ให้สมการสเกตของค่าผิดพลาดสังเกตในสมการที่ (2.34) มีเสถียรภาพและสมรรถนะตามต้องการ ซึ่งสามารถใช้วิธีการวางโพลได้เช่นกัน จากสมการที่ (2.34) สมการคุณลักษณะของสมการสเกตของค่าผิดพลาดเป็น

$$\det(sI - (A - LC)) = 0 \quad (2.35)$$

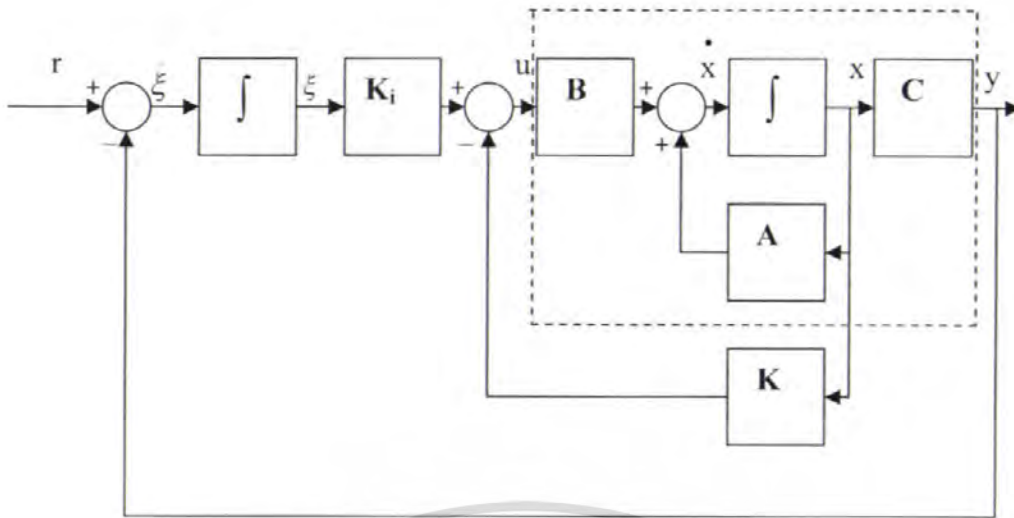
หากกำหนดให้ s_1, s_2, \dots, s_n คือโพลของตัวสังเกตสเกต ซึ่งเลือกตามข้อกำหนดเสถียรภาพและสมรรถนะที่ต้องการ จะได้สมการคุณลักษณะของตัวสังเกตสเกตที่ต้องการออกแบบเป็น

$$\alpha_c(s) = (s - s_1)(s - s_2) \cdots (s - s_n) = 0 \quad (2.36)$$

ดังนั้นสามารถหาค่า L ได้จากความสัมพันธ์ของสมการที่ (2.35) และ (2.36)

2.4.3 ระบบเซอร์โวป้อนกลับสเกตเวลาต่อเนื่อง

เนื่องจากการออกแบบระบบป้อนกลับสเกตที่ได้กล่าวมาข้างต้นนั้น ไม่ได้พิจารณาถึงค่าอ้างอิงภายนอก แต่ในทางปฏิบัติระบบที่ใช้งานจริงนั้นต้องการค่าอ้างอิงจากภายนอก ดังนั้นจึงต้องออกแบบระบบให้เป็นระบบเซอร์โว (Servo System) เพื่อรองรับค่าอ้างอิงที่จะป้อนเข้ามาในระบบ



รูปที่ 2.15 บล็อกไดอะแกรมของระบบเซอร์โวป้อนกลับสแตต

จากรูปที่ 2.15 สามารถอธิบายความสัมพันธ์ของสัญญาณควบคุมด้วยสมการดังต่อไปนี้

$$u = +Kx + K_i \xi \quad (2.37)$$

$$\xi = r - y = r - Cx \quad (2.38)$$

เมื่อ K_i คือ อัตราขยายแบบอินทิกรัล

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (2.39)$$

$$\begin{bmatrix} x(\infty) \\ \xi(\infty) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(\infty) \\ \xi(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u(\infty) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(\infty) \quad (2.40)$$

โดยให้ $r(t)$ เป็นสัญญาณอินพุตแบบสัญญาณขั้นบันได (Step Signal) ดังนั้นจะได้ $r(0) = r(\infty) = r$ ดังนั้นจากสมการที่ (2.39) – (2.40) และกำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$x(t) - x(\infty) = x_e(t) \quad (2.41)$$

$$\xi(t) - \xi(\infty) = \xi_e(t) \quad (2.42)$$

$$u(t) - u(\infty) = u_e(t) \quad (2.43)$$

จะได้

$$\begin{bmatrix} \dot{x}_e(t) \\ \dot{\xi}_e(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x_e(t) \\ \xi_e(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_e(t) \quad (2.44)$$

เมื่อ

$$u_e(t) = -Kx_e(t) + K_\xi \xi_e(t) \quad (2.45)$$

ถ้ากำหนดให้

$$e(t) = \begin{bmatrix} x_e(t) \\ \xi_e(t) \end{bmatrix} \quad (2.46)$$

ดังนั้น จากสมการที่ (2.44) จะได้

$$\dot{e} = Ae + Bu_e \quad (2.47)$$

โดยที่

$$\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \quad (2.48)$$

$$\hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (2.49)$$

และจากสมการที่ (2.45) จะได้

$$u_e(t) = -\hat{K}e \quad (2.50)$$

เมื่อ

$$\hat{K} = [K; K_\xi] \quad (2.51)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจากสมการที่ (2.47) และ (2.50) จะได้

$$\dot{e} = (\hat{A} - \hat{B}\hat{K})e \quad (2.52)$$

ดังนั้นจากสมการที่ (2.52) จะได้สมการคุณลักษณะที่สอดคล้องเป็น

$$\det(sI - (\hat{A} - \hat{B}\hat{K})) = 0 \quad (2.53)$$

ซึ่งสามารถออกแบบค่า \hat{K} ได้โดยวิธีการวางโพลเช่นเดียวกับที่ได้กล่าวมาแล้ว

2.4.4 ระบบเซอร์โวป้อนกลับสแตทร่วมกับตัวสังเกตสถานะเวลาดีสครีต

การออกแบบข้างต้นที่ได้กล่าวมานั้นเป็นการออกแบบระบบเซอร์โวป้อนกลับสแตทและตัวสังเกตสแตทบนโดเมนเวลาต่อเนื่อง แต่การใช้งานจริงนั้นต้องออกแบบให้อยู่บนโดเมนเวลาดีสครีต

จากรูปที่ 2.16 พิจารณาระบบบนโดเมนเวลาดีสครีตได้ดังนี้

$$x[k+1] = Fx[k] + Gu[k] \quad (2.54)$$

$$y[k] = Hx[k] \quad (2.55)$$

จากรูปที่ 2.16 ระบบที่มีเซอร์โวบนเวลาดีสครีต ซึ่งได้อธิบายมาแล้วในส่วนของเวลาต่อเนื่องจะสามารถหาสมการคุณลักษณะของระบบในช่วงเวลาดีสครีตได้เป็น

$$\det(ZI - (\hat{F} - \hat{G}\hat{K})) = 0 \quad (2.56)$$

เมื่อ

$$\hat{F} = \begin{bmatrix} F & 0 \\ -H & 1 \end{bmatrix} \quad (2.57)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ

$$x[k] = \begin{bmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_i[k] \end{bmatrix} \quad (2.62)$$

ในการออกแบบระบบป้อนกลับสแตท สามารถหาค่า เมตริกซ์อัตราขยายป้อนกลับสแตทที่เหมาะสมได้จากการพิจารณาค่าดัชนีสมรรถนะควอดราติกซ์ (Quadratic Cost Function) ดังนี้

$$J = \sum_{n=1}^{\infty} (x[k]^T Q x[k] + u[k]^T R u[k] + 2x[k]^T N u[k]) \quad (2.63)$$

จะได้ว่าค่าเกน K ที่ทำให้ J มีค่าน้อยที่สุดสามารถหาได้จาก

$$K = (G^T S G + R)^{-1} (G^T S F + N^T) \quad (2.64)$$

เมื่อ S คือคำตอบของสมการริคคาตี (Riccati Equation)

$$F^T S F - S - (F^T S G + N)(G^T S G + R)^{-1} (G^T S F + N^T) + Q = 0 \quad (2.66)$$

โดยมีเงื่อนไขดังนี้

1. คู่เมตริกซ์ F และ G ต้องสามารถทำให้ระบบเสถียร (Stabilizable)
2. $R > 0$ และ $Q - N R^{-1} N^T \geq 0$
3. $Q - N R^{-1} N^T, F - G R^{-1} N^T$ ต้องเป็นตัวสังเกตค่าได้

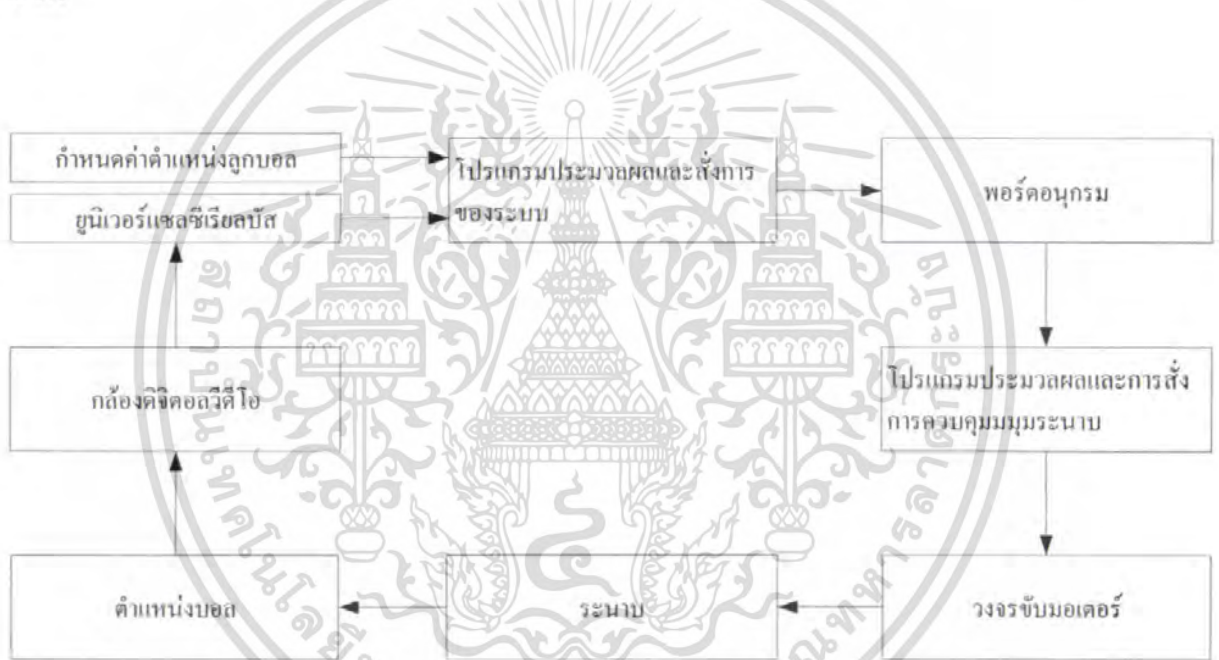
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการออกแบบ

การออกแบบระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในเบื้องต้นนั้นต้องเข้าใจหลักการทำงานในภาพรวมของระบบควบคุมก่อนว่ามีหลักการทำงานอย่างไร และจึงพิจารณาส่วนย่อยต่างๆ อันประกอบไปด้วยโครงสร้างของระบบควบคุม ส่วนที่เป็นวงจรถิเล็กทรอนิกส์ และส่วนของโปรแกรมที่ใช้ในการประมวลผลและควบคุมระบบ

หลักการทำงานในภาพรวมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ สามารถแสดงดังรูปที่ 3.1



รูปที่ 3.1 หลักการทำงานของระบบควบคุมตำแหน่งลูกบอลบนระนาบ

จากรูปที่ 3.1 จะเห็นว่าระบบควบคุมตำแหน่งลูกบอลบนระนาบประกอบด้วยส่วนสำคัญดังนี้

- กล้องดิจิทัลวีดีโอ
- คอมพิวเตอร์
- โครงสร้างทางกายภาพของระบบ
- ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 กล้องดิจิทัลวีดีโอ

กล้องดิจิทัลวีดีโอ เป็นอุปกรณ์ที่สำคัญสำหรับ โครงการนี้ เนื่องจากเป็นอุปกรณ์ที่ใช้สำหรับ ตรวจสอบตำแหน่งลูกบอลบนระนาบ เพื่อทำการประมวลผลหาค่าสัญญาณควบคุมที่เหมาะสม การเลือกใช้กล้องดิจิทัลวีดีโอรุ่น PC – 01 ซึ่งมีคุณสมบัติดังต่อไปนี้



รูปที่ 3.2 กล้องดิจิทัลวีดีโอ

คุณสมบัติของกล้องดิจิทัลวีดีโอ

- ให้ภาพละเอียดสูงสุดถึง 640 x 480 พิกเซลในภาพสี 24 บิต
- อัตราการส่งภาพ ด้วยความเร็วสูงสุดถึง 30 ภาพต่อวินาที
- มีระบบปรับความสว่างอัตโนมัติ
- ปรับโฟกัสได้ตั้งแต่ 5 เซนติเมตร ถึงอนันต์
- ใช้งานร่วมกับคอมพิวเตอร์ผ่านพอร์ตยูนิเวอร์แซลซีเรียลบัส

3.2 คอมพิวเตอร์

คอมพิวเตอร์เป็นอุปกรณ์ที่ทำหน้าที่รับข้อมูลจากกล้องดิจิทัลวีดีโอผ่านทางยูนิเวอร์แซลซีเรียลบัส โดยมีโปรแกรมซึ่งทำหน้าที่ประมวลผลเพื่อหาค่าตำแหน่งของลูกบอล และจึงประมวลผลสัญญาณควบคุมที่เหมาะสม จากนั้นจึงส่งค่าสัญญาณควบคุมที่คำนวณได้ผ่านทางพอร์ตอนุกรมไปยังไมโครคอนโทรลเลอร์ เพื่อควบคุมมุมของระนาบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากการควบคุมในโครงการนี้ใช้คอมพิวเตอร์เป็นอุปกรณ์ในการประมวลผลหลัก ดังนั้น การเขียน โปรแกรมจึงเป็นการดำเนินการบนเวลาสี่สัปดาห์ แทนการควบคุมแบบระบบเวลาต่อเนื่อง ดังนั้นการเขียน โปรแกรมต้องรัดกุม และทำงานได้รวดเร็วซึ่งคอมพิวเตอร์ที่ใช้ต้องมีความเร็วพอที่จะทำให้การคำนวณสัญญาณมีค่าน้อยเพียงพอจนไม่มีผลต่อการควบคุมระบบทั้งหมดด้วย สำหรับโครงการนี้เลือกใช้การเขียนโปรแกรมโดยภาษาซี

ในส่วนของโปรแกรมควบคุมระบบ เริ่มจากเขียนโปรแกรมรับภาพจากกล้องดิจิตอลวีดีโอ และหาตำแหน่งลูกบอลบนระนาบ จากนั้นเปรียบเทียบตำแหน่งมุมระนาบที่ต้องการให้ลูกบอลเคลื่อนไป โปรแกรมส่วนควบคุมจะประมวลผลเพื่อหาสัญญาณควบคุมที่เหมาะสม ซึ่งอาศัยตัวควบคุมแบบมูมนำและตัวควบคุมแบบป้อนกลับสเกต ดังที่ได้กล่าวไว้ในบทที่ 2 แล้ว จึงสั่งให้คอมพิวเตอร์ส่งค่าสัญญาณควบคุมออกทางพอร์ตอนุกรม เพื่อให้ไมโครคอนโทรลเลอร์ทำการควบคุมตำแหน่งมุมระนาบต่อไป สามารถเขียนโฟลว์ชาร์ต (Flowchart) แสดงการทำงานของโปรแกรมดังรูปที่ 3.3 และรูปที่ 3.4

3.3 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ที่ทำหน้าที่รับคำสั่งสัญญาณควบคุมของระบบ เพื่อทำการควบคุมมุมของระนาบผ่านวงจรมอเตอร์ต่อไป

เนื่องจากไมโครคอนโทรลเลอร์เป็นอุปกรณ์หลักในการควบคุมตำแหน่งมุมของระนาบ ดังนั้นจึงเลือกใช้ไมโครคอนโทรลเลอร์ ที่มีความสามารถในการประมวลผลเร็วพอต่อการคำนวณสัญญาณ การเขียนโปรแกรมต้องมีการรัดกุมและง่ายต่อความเข้าใจ สำหรับโครงการนี้เลือกใช้การเขียนโปรแกรมโดยภาษาซี

การเขียนโปรแกรมเริ่มจากการรับข้อมูลค่าพารามิเตอร์ของตัวควบคุมและตำแหน่งมุมของระนาบที่ต้องการผ่านทางพอร์ตอนุกรม และประมวลผลข้อมูลที่ได้ว่าเป็นข้อมูลสำหรับการเปลี่ยนตำแหน่งมุมของระนาบ การปรับเปลี่ยนค่าพารามิเตอร์ของตัวควบคุมแบบ พีไอดี หรือสั่งหยุดการเคลื่อนที่ของระนาบ จากนั้นโปรแกรมส่วนควบคุมจะรับค่าตำแหน่งมุมปัจจุบันของระนาบและประมวลผล เพื่อหาสัญญาณควบคุมที่เหมาะสม จากนั้นจะส่งสัญญาณควบคุมซึ่งอยู่ในลักษณะสัญญาณแบบพัลส์วิดโมดูเลชัน (Pluse Wide Modulation: PWM) ไปยังมอเตอร์ไฟฟ้ากระแสตรงเพื่อควบคุมมุมระนาบ โดยสามารถเขียนโฟลว์ชาร์ต แสดงการทำงานของโปรแกรมดัง รูปที่ 3.5



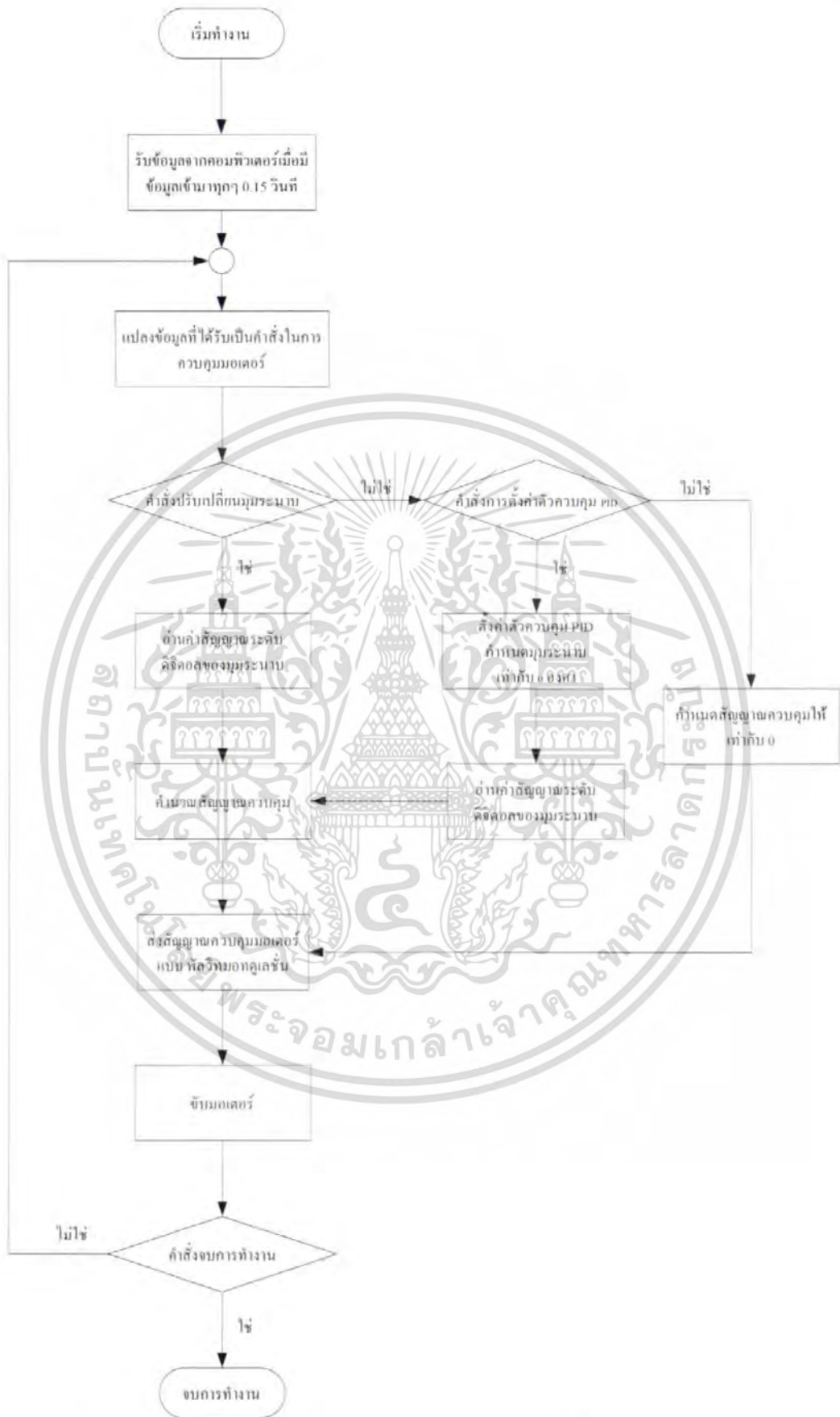
รูปที่ 3.3 โฟลว์ชาร์ตการควบคุมตำแหน่งมุมของระนาบโดยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการแข่งขันเพื่อการแข่งขันเท่านั้น เมื่อเผยแพร่ให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 โพลีชาร์ตการหาตำแหน่งลูกบอลบนระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 โฟลว์ชาร์ตการควบคุมตำแหน่งมุมระนาบในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการควบคุมตำแหน่งลูกบอลบนระนาบนี้เลือกใช้ไมโครคอนโทรลเลอร์เบอร์ 16F877A เหตุที่เลือกเพราะมีคุณสมบัติและฟังก์ชันการทำงานต่างๆที่ค่อนข้างครบถ้วน และมีโครงสร้างการทำงานที่ไม่ยุ่งยากมากนัก สามารถศึกษาไว้เป็นพื้นฐานสำหรับการใช้งานไมโครคอนโทรลเลอร์ในเบอร์อื่นๆ ที่สูงขึ้นได้เป็นอย่างดี ไมโครคอนโทรลเลอร์ชนิดนี้เป็นแบบ 14 บิต คือ มีโครงสร้างของคำสั่งเป็นแบบ 14 บิต และมีคุณสมบัติที่เด่น และเหมาะสมกับการใช้งานดังนี้

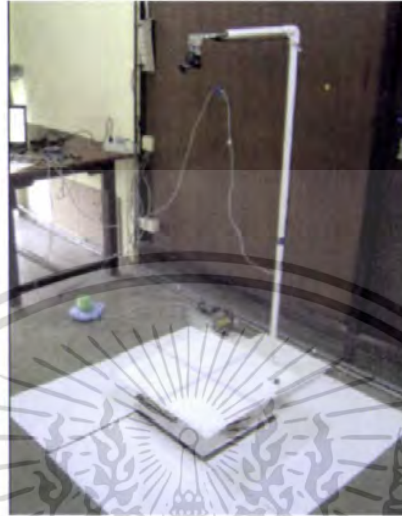
1. การง่าย และการรับกระแสของขาสัญญาณอินพุตพอร์ตต่างๆ ที่สามารถง่ายและรับกระแสสูงสุดได้ถึง 25 มิลลิแอมป์ ทำให้สามารถขับโหลดต่างๆได้เป็นอย่างดีโดยไม่จำเป็นต้องต่อบัฟเฟอร์ หรือวงจรขยายกระแสอื่นๆ แต่อย่างใด
2. มีฟังก์ชันการทำงานต่างๆ มากมาย เช่น Power – on Reset , Power – up Timer , Timer/Counter , A/D , PWM เป็นต้น
3. มีโมดูลการใช้งานให้เลือกมากมายเช่น โมดูลการสื่อสาร USART , โมดูลตรวจจับระดับไฟเลี้ยง Brown – out reset (BOR) , โมดูล Capture/Compare/PWM จำนวน 2 ชุด
4. สามารถเลือกใช้งานโหมดของสัญญาณนาฬิกาได้หลายโหมด และมีโหมดประหยัดพลังงาน
5. สามารถโปรแกรมให้ใช้ได้กับแรงดันขนาด +5 โวลต์ และทำงานได้ที่ 2.0 ถึง 5.5 โวลต์
6. สามารถตรวจสอบสัญญาณรบกวนได้ถึง 14 แหล่ง
7. มีฟังก์ชันทรีเซต
 - วงจรเพาเวอร์อ้อนรีเซตภายใน
 - วงจรเพาเวอร์อัพไทม์เมอร์
 - วงจรออสซิลเลเตอร์สควร์ทไทม์เมอร์
 - เบริาน – เติวรีเซต

3.4 โครงสร้างทางกายภาพของระบบ

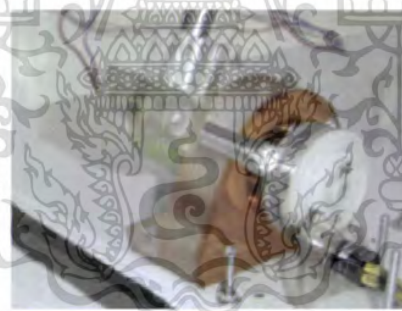
โครงสร้างทางกายภาพของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ประกอบด้วยโครงสร้างหลัก เช่น ขาดังกล้อง มอเตอร์ โฟเทินทีโอมิเตอร์ กล้องดิจิตอลวีดีโอ เป็นต้น

จากรูปที่ 3.6 โครงสร้างหลักนั้นประกอบด้วย ขาดังกล้องดิจิตอลวีดีโอ ที่ทำมาจากท่อพีวีซี ซึ่งมีความสูง 1.5 เมตร และในส่วนโครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ซึ่งประกอบด้วยแผ่นระนาบการติดตั้งมอเตอร์แสดงดัง รูปที่ 3.7 และโฟเทินทีโอมิเตอร์แสดงดัง รูปที่ 3.8 ทำมาจากอะคริลิก โดยส่วนฐานมีน๊อต ใช้เป็นตัวปรับมุมของโครงสร้างเพื่อให้สามารถใช้งานได้ในกรณีที่พื้นที่ที่ใช้วางโครงสร้างมีความเอียง

ในโครงการนี้ใช้โพเทนทิโอมิเตอร์ขนาด 10 กิโลโอห์ม เพื่อใช้ในการวัดมุมของระนาบได้โดยไมโครคอนโทรลเลอร์จะแปลงระดับสัญญาณไฟฟ้า ที่เปลี่ยนแปลงตามค่าความต้านทานของโพเทนทิโอมิเตอร์ ให้เป็นระดับสัญญาณดิจิทัล



รูปที่ 3.6 โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนระนาบ



รูปที่ 3.7 การขันมอเตอร์



รูปที่ 3.8 การขันโพเทนทิโอมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้กล่าวถึงการทดลองต่างๆ ของระบบควบคุมตำแหน่งลูกบอลบนระนาบ โดยทำการทดลองควบคุมตำแหน่งมุมระนาบ ควบคุมตำแหน่งของมอเตอร์ และการควบคุมตำแหน่งลูกบอลบนระนาบ โดยมีรายละเอียดของการทดลองดังนี้

4.1 การระบุตำแหน่งมุมของระนาบโดยเทียบกับค่าการแปลงค่าระดับอะนาลอกเป็นดิจิทัล

การระบุตำแหน่งมุมของระนาบโดยการเปรียบเทียบค่ามุมของระนาบเป็นระดับสัญญาณดิจิทัล โดยการให้ระดับนำวัดมุมของระนาบที่ระดับ 0 องศา และอ่านค่าระดับดิจิทัลจากไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม สามารถสรุปได้ดังตารางที่ 4.1 พบว่า เมื่อค่ามุมของระนาบเปลี่ยนไป 1 องศา ค่าระดับดิจิทัล มีการเปลี่ยนแปลงไป 49 ระดับในแกน X และ 47 ระดับในแกน Y ซึ่งค่าดังกล่าวนี้จะนำไปใช้ในการคำนวณระดับมุมจากค่าวัดที่เป็นระดับดิจิทัล

X		Y	
มุม (องศา)	ระดับสัญญาณดิจิทัล	มุม (องศา)	ระดับสัญญาณดิจิทัล
+2	595	+2	597
+1	546	+1	550
0	497	0	503
-1	448	-1	456
-2	399	-2	409

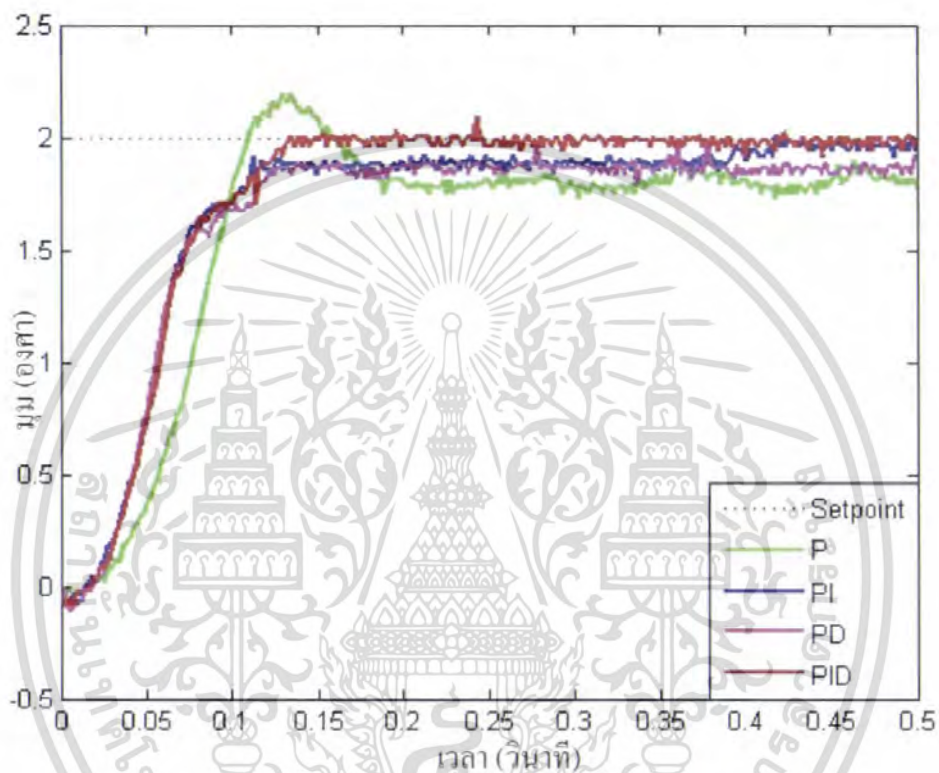
ตารางที่ 4.1 ผลทดลองหาความสัมพันธ์ระหว่างค่าระดับสัญญาณดิจิทัลกับมุมของระนาบ

4.2 การควบคุมตำแหน่งมุมของระนาบโดยอาศัยตัวควบคุมแบบพีไอดี

ในการทดลองนี้ได้ทำการควบคุมตำแหน่งมุมระนาบ โดยอาศัยตัวควบคุมแบบพีไอดีร่วมกับแอนติ-ไวคัพ (Anti-windup) ซึ่งส่วนแอนติ-ไวคัพ เป็นวิธีการที่ใช้เพื่อลดปัญหาการเกิดค่าพุ่ง (Overshoot) ซึ่งอาจเกิดขึ้น เมื่อเอาต์พุตของตัวควบคุมซึ่งคือสัญญาณควบคุมเกิดการอิ่มตัว ซึ่งสามารถทำได้โดยการหยุดการเพิ่มค่าในส่วนการอินทิเกรตของการควบคุมแบบพีไอ โดยในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการนี้มอเตอร์ไฟฟ้ากระแสตรงที่ใช้ในโครงการจะรับค่าแรงดันไฟฟ้าในย่านจำกัด -5 ถึง 5 โวลต์

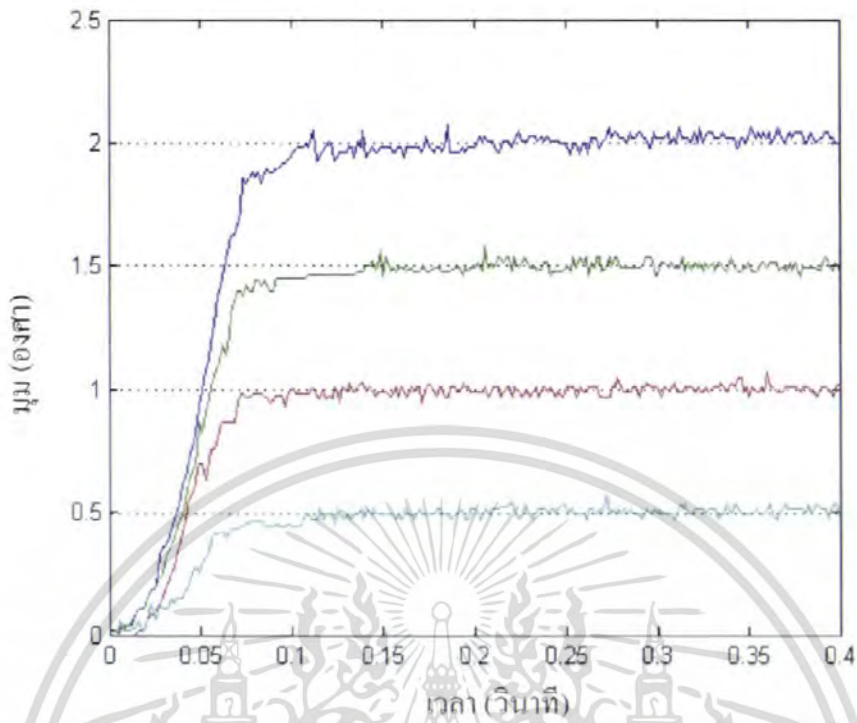
การควบคุมตำแหน่งระนาบโดยไมโครคอนโทรลเลอร์ ในโครงการนี้เลือกใช้คาบการสุ่มสัญญาณ (Sampling Period) ที่ 1.6 มิลลิวินาที ซึ่งเป็นช่วงเวลาที่มากพอในการทำงานแต่ละรอบการควบคุมของไมโครคอนโทรลเลอร์



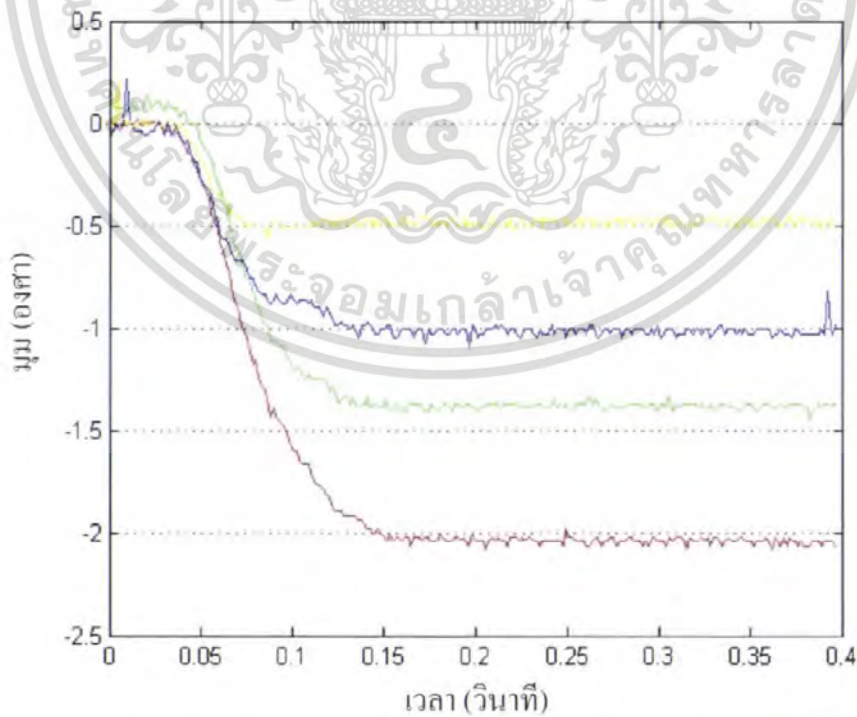
รูปที่ 4.1 ผลตอบสนองของระบบควบคุมมุมระนาบโดยใช้ตัวควบคุมแบบ พีพีไอ พีดี และพีไอดี

จากการทดลองใช้ตัวควบคุมแบบพีไอดี ในการควบคุมตำแหน่งมุมของระนาบจากตำแหน่งมุม 0 องศาไปยังตำแหน่งมุม 2 องศา โดยใช้เกน $K_p=99$, $K_i=20$, $K_d=8$ พบว่าตัวควบคุมแบบพีไอดี ให้ผลตอบสนองได้ดีกว่าตัวควบคุมแบบ พี พีไอ และ พีดี ในโครงการนี้จึงเลือกใช้ตัวควบคุมแบบพีไอดี ในการควบคุมตำแหน่งมุมของระนาบ โดยการทดลองควบคุมตำแหน่งมุมของระนาบ จากตำแหน่ง 0 องศา ไปถึง 2 และ -2 องศา โดยเพิ่มและลดครั้งละ 0.5 องศา ตามลำดับ ได้ผลการทดลองดังรูปที่ 4.2 และรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ผลตอบสนองของระบบควบคุมมุมระนาบจาก 0 องศาไปยังมุมบวก โดยอาศัยตัวควบคุมแบบฟuzzy



รูปที่ 4.3 ผลตอบสนองของระบบควบคุมมุมระนาบจาก 0 องศาไปยังมุมลบ โดยอาศัยตัวควบคุมแบบฟuzzy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การประมาณฟังก์ชันถ่ายโอนของระบบควบคุมของระนาบ

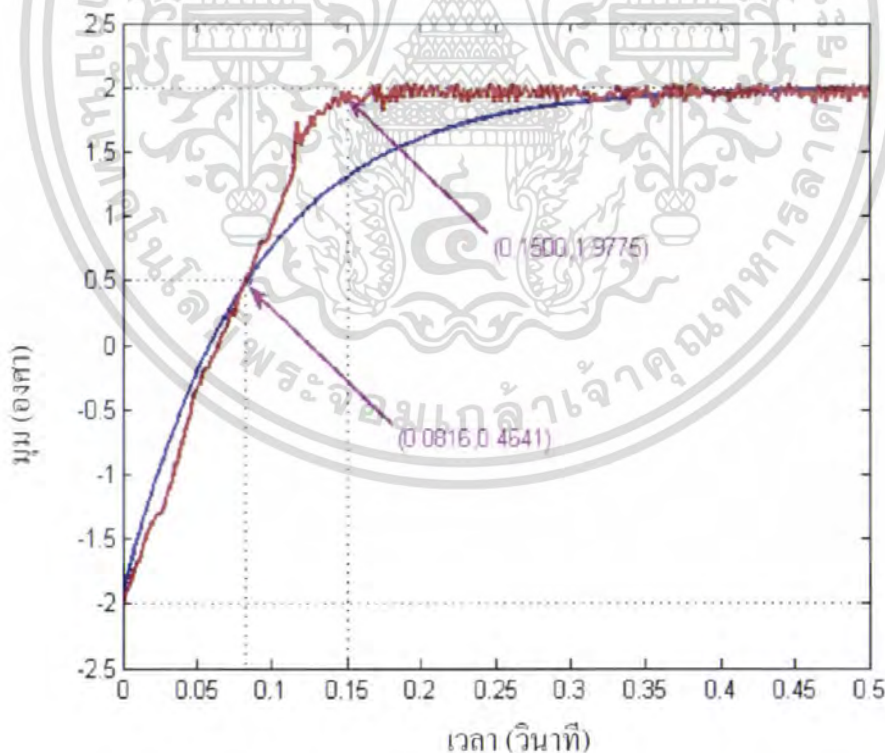
จากการทดลองควบคุมมุมของระนาบ ได้ผลตอบสนองมีรูปร่างคล้ายผลตอบสนองต่อเวลาของฟังก์ชันถ่ายโอนแบบระบบควบคุมอันดับหนึ่ง ดังรูปที่ 4.4

ฟังก์ชันถ่ายโอนระบบควบคุมอันดับหนึ่งเป็น

$$\frac{C(s)}{R(s)} = \frac{1}{Ts+1} \quad (4.1)$$

จากรูปที่ 4.4 แสดงให้เห็นว่าค่า T มีค่าประมาณ 0.0816 วินาที ดังนั้นจากสมการที่ (4.1) จึงประมาณฟังก์ชันถ่ายโอนของระบบควบคุมวงปิดเพื่อใช้ในการจำลองระบบต่อไป ดังสมการที่ (4.2)

$$\frac{C(s)}{R(s)} = \frac{1}{0.0816s+1} \quad (4.2)$$



รูปที่ 4.4 การประมาณแบบจำลองมอเตอร์จากผลตอบสนองของระบบควบคุมมุมระนาบจาก -2 องศา ไปยังมุม 2 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การระบุตำแหน่งของลูกบอลโดยเทียบฟิกเซล

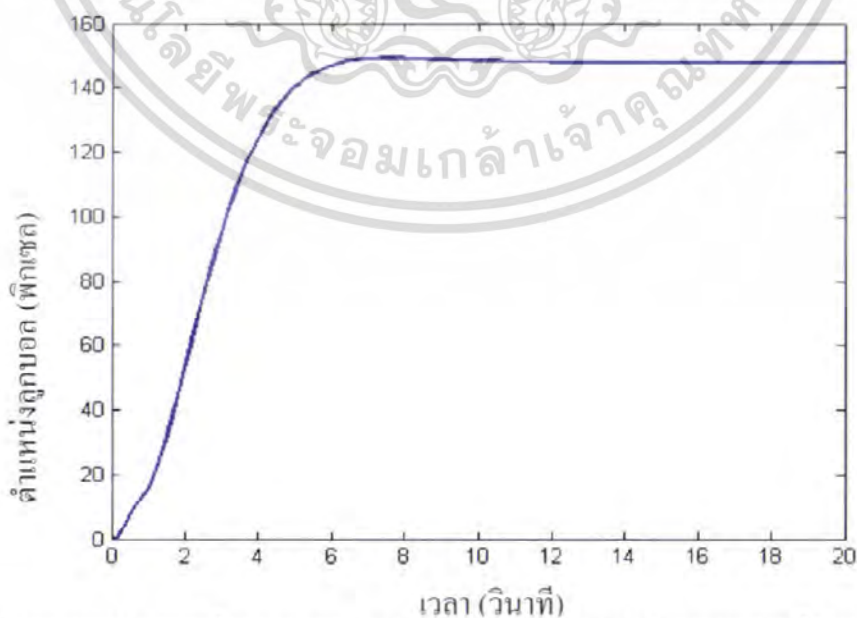
โครงการนี้ติดตั้งกล้องดิจิทัลวีดีโอสำหรับตรวจจับตำแหน่งของลูกบอลบนระนาบ ณ ความสูง 1.5 เมตรจากระนาบ ค่าตำแหน่งของลูกบอลที่อ่านได้จากหน้าจอคอมพิวเตอร์ จะอยู่ในรูปของฟิกเซล จึงต้องทำการแปลงโดยเลือกใช้ค่าความกว้างของระนาบมีค่าเท่ากับ 0.5 เมตร และที่ระยะความกว้าง 0.5 เมตร มีค่าเท่ากับ 217 ฟิกเซล ความสัมพันธ์ดังกล่าวสามารถเขียนได้ดังสมการ (4.3)

$$A = \frac{B * 0.5}{217} \quad (4.3)$$

เมื่อ A คือ ตำแหน่งจริงของลูกบอลบนระนาบที่หน่วยเป็นเมตร B คือ ตำแหน่งของลูกบอลที่อ่านได้จากหน้าจอคอมพิวเตอร์มีหน่วยเป็นฟิกเซล เมื่อติดตั้งกล้องสูงจากระนาบ 1.5 เมตร

4.5 ผลการทดลองควบคุมตำแหน่งลูกบอลโดยตัวควบคุมแบบตัวชดเชยชนิดมูมนำ

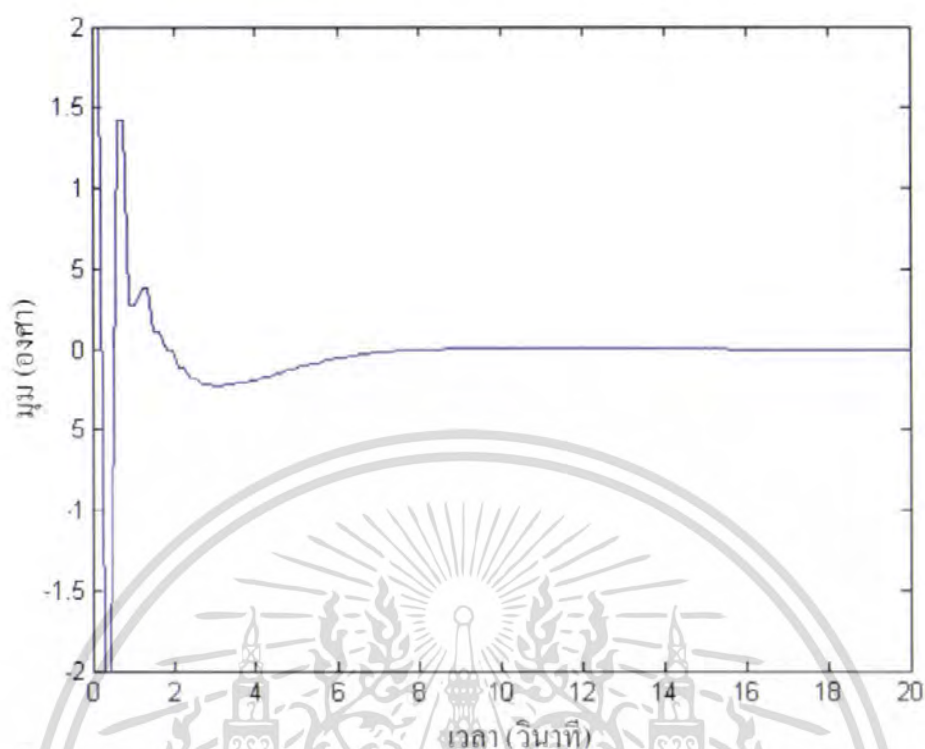
การควบคุมตำแหน่งลูกบอลไปยังตำแหน่งเป้าหมายซึ่งใช้ตัวควบคุมแบบชดเชยชนิดมูมนำ เวลาดีสครีต นำการออกแบบตัวควบคุมโดยใช้โปรแกรม Matlab ช่วยในการออกแบบ ได้โพลของตัวชดเชยที่ -0.335 และซีโร ของตัวชดเชยที่ 0.888 โดยควบคุมจากตำแหน่ง 38 ฟิกเซล ไปยังตำแหน่ง 148 ฟิกเซล ซึ่งการจำลองผลตอบสนองและศึกษาควบคุมของระบบควบคุมลูกบอลบนระนาบ ได้ผลการจำลองดังรูปที่ 4.5



รูปที่ 4.5 ผลตอบสนองการจำลองระบบควบคุมตำแหน่งลูกบอลบนระนาบ จากตำแหน่ง 0 ฟิกเซล

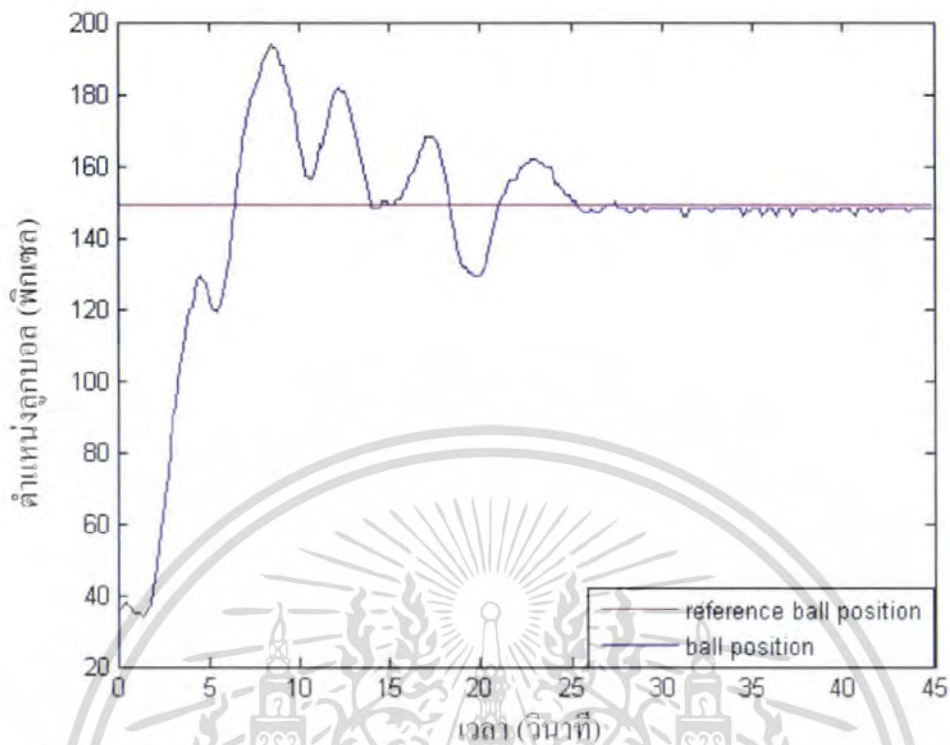
ไปยัง ตำแหน่ง 148 ฟิกเซลโดยอาศัยตัวชดเชยชนิดมูมนำ

เอกสารนี้เป็นเอกสารที่ลงนามในหนังสือพิมพ์ ซึ่งไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

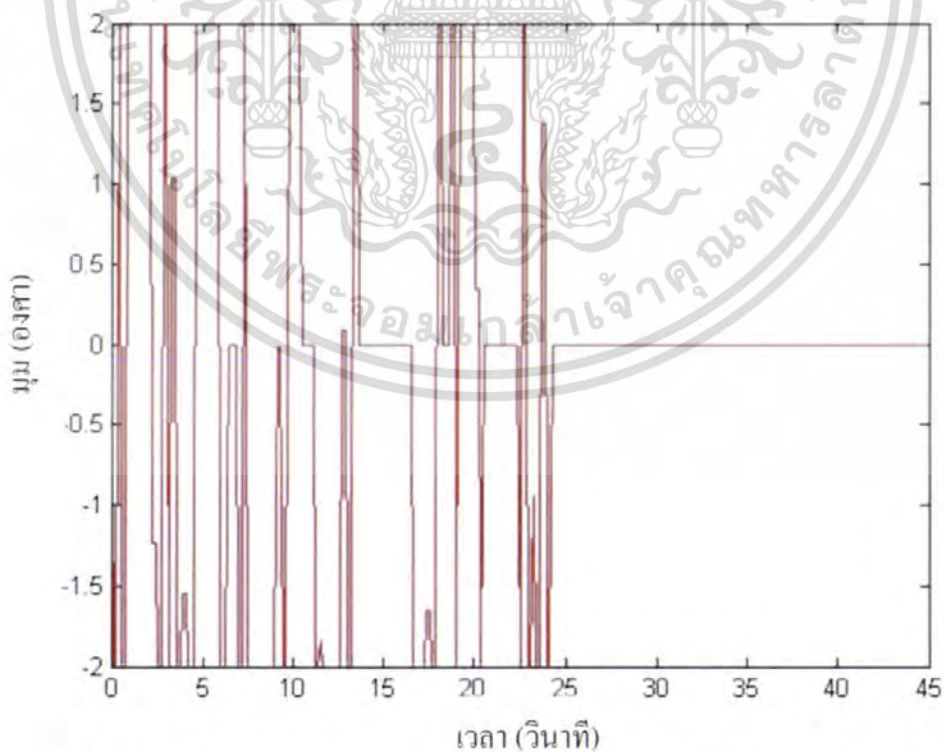


รูปที่ 4.6 สัญญาณควบคุมการจำลองระบบควบคุมตำแหน่งลูกบอลบนระนาบ จากตำแหน่ง 0 พิกเซลไปยัง ตำแหน่ง 148 พิกเซลโดยอาศัยตัวชดเชยชนิดมูมนำ

จากนั้นเมื่อนำตัวควบคุมชนิดนี้ไปทดลองใช้ควบคุมระบบควบคุมลูกบอลบนระนาบ พบว่าเมื่อควบคุมบอลไปยังตำแหน่งเป้าหมายโดยใช้ตัวควบคุมแบบตัวชดเชยชนิดมูมนำจากตำแหน่ง (38.8) พิกเซล ไปยังตำแหน่ง (148.128) พิกเซล สามารถควบคุมตำแหน่งลูกบอลไปยังบริเวณตำแหน่งเป้าหมายได้ แต่มีกรแกว่งรอบตำแหน่งเป้าหมายเกิดขึ้นในช่วงเริ่มต้นหลังจากนั้นลูกบอลจึงเข้าสู่ตำแหน่งเป้าหมายได้ ระบบเข้าสู่สภาวะคงตัวที่เวลาประมาณ 25 วินาที ในแนวแกน X และ ระบบเข้าสู่สภาวะคงตัวที่เวลาประมาณ 27 วินาที และมีค่าผิดพลาดเกิดขึ้นเล็กน้อยในแนวแกน Y ได้ผลการทดลองดังรูปที่ 4.7 และรูปที่ 4.9



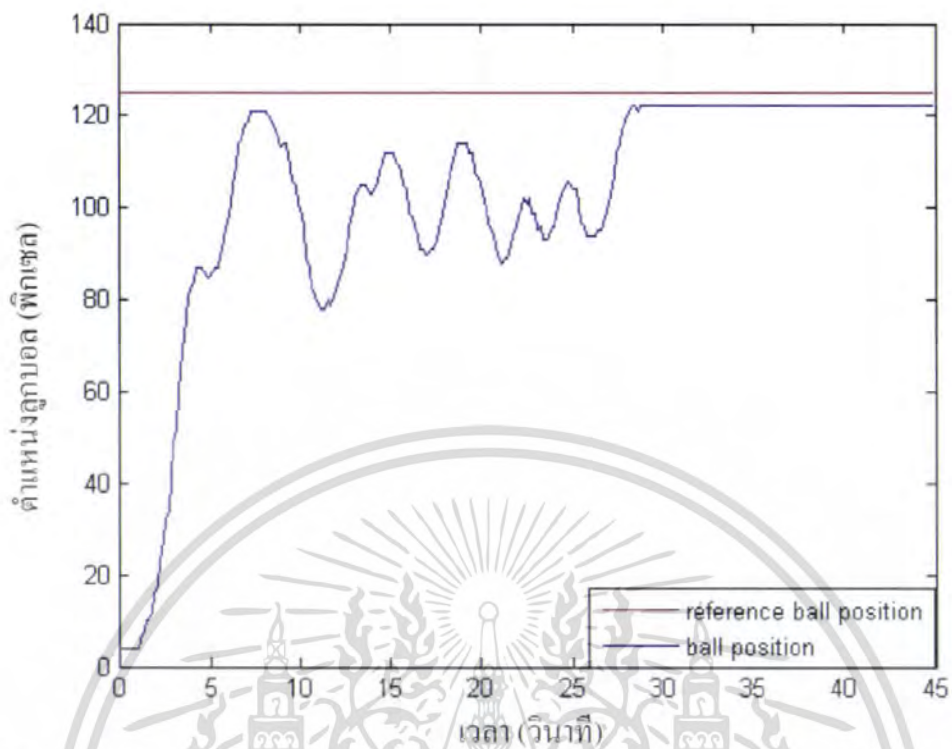
รูปที่ 4.7 ผลตอบสนองระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซลไปยัง ตำแหน่ง 148 พิกเซล โดยอาศัยตัวชดเชยชนิดมูมน้ำ



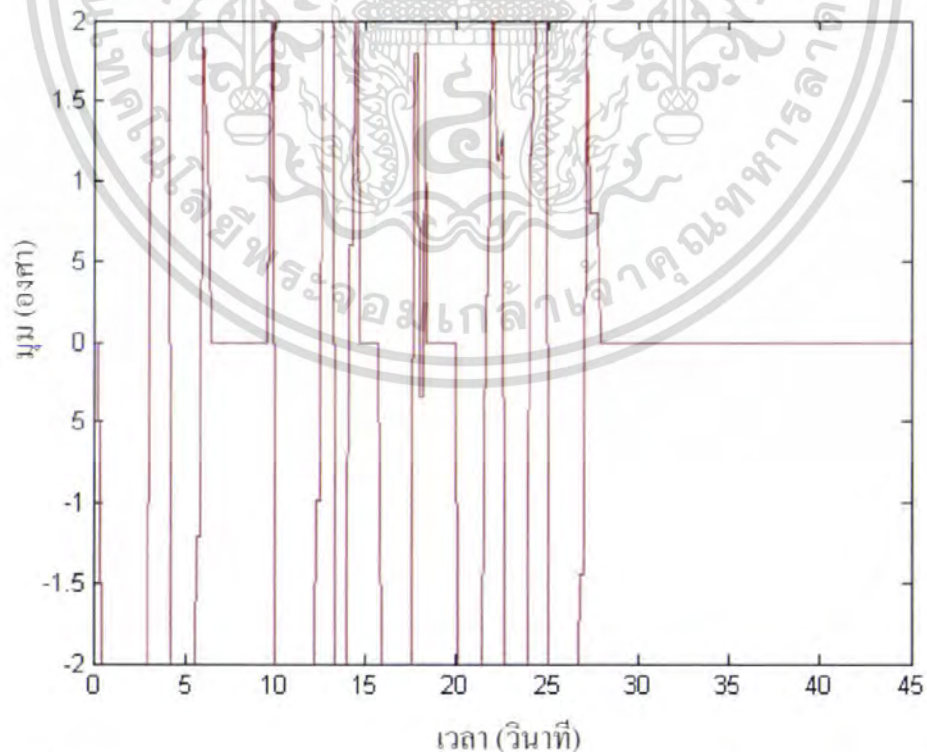
รูปที่ 4.8 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X

จากตำแหน่ง 38 พิกเซลไปยังตำแหน่ง 148 พิกเซล โดยอาศัยตัวชดเชยชนิดมูมน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 128 พิกเซล โดยอาศัยตัวควบคุมชนิดมูบนำ

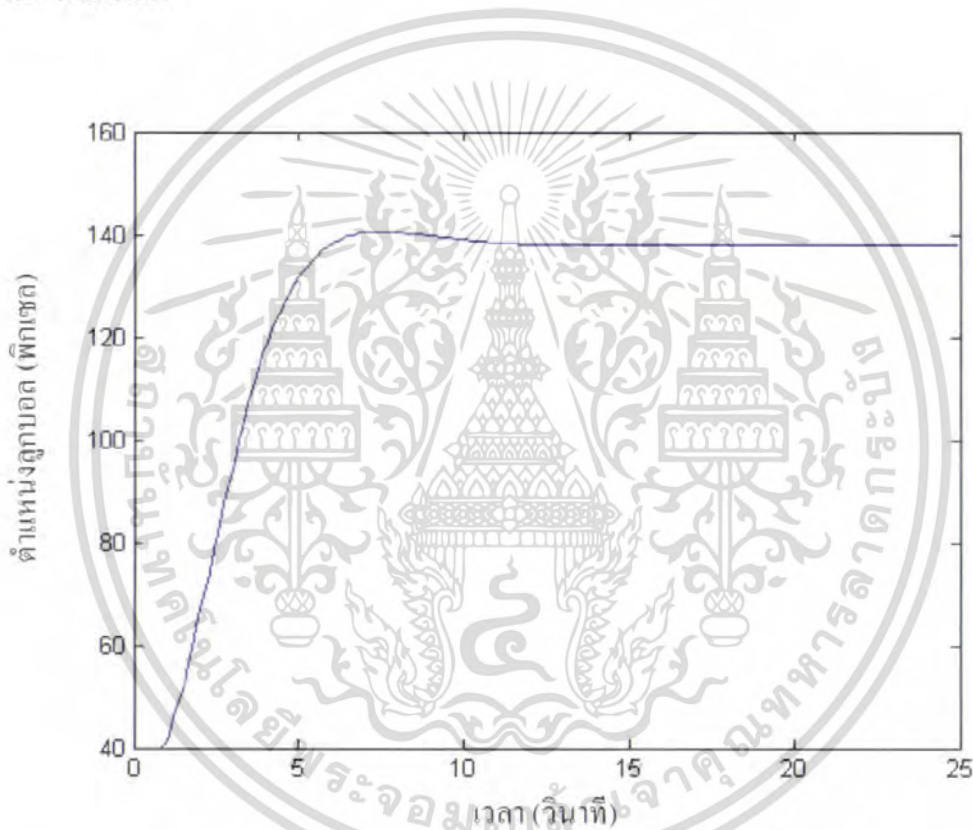


รูปที่ 4.10 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จาก

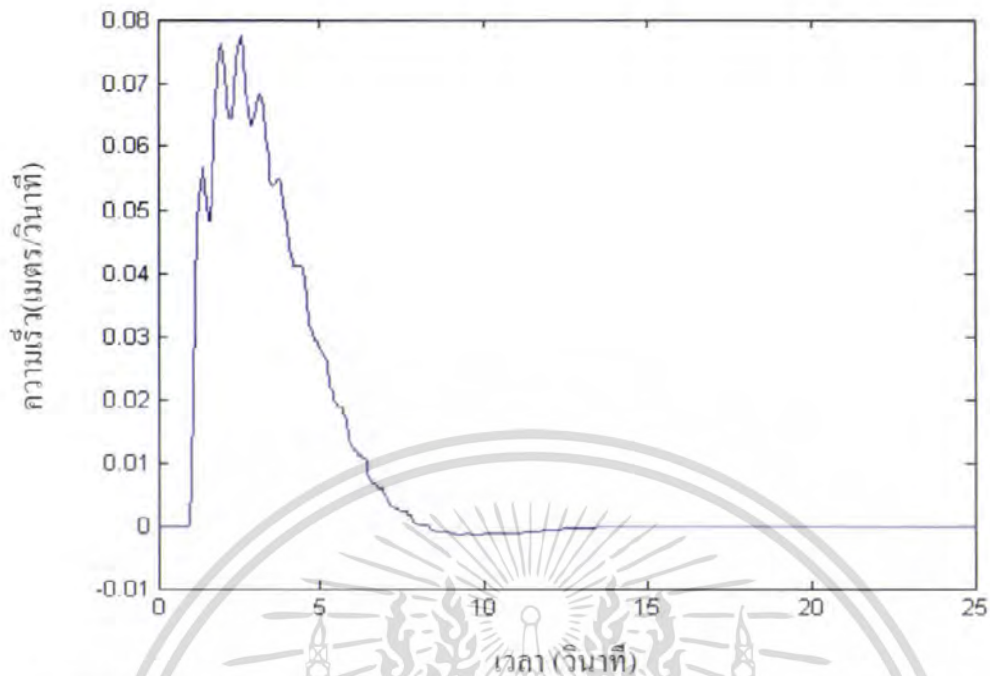
ตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 128 พิกเซล โดยอาศัยตัวควบคุมชนิดมูบนำ ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดลองควบคุมตำแหน่งลูกบอลโดยอาศัยตัวควบคุมแบบป้อนกลับสเตท

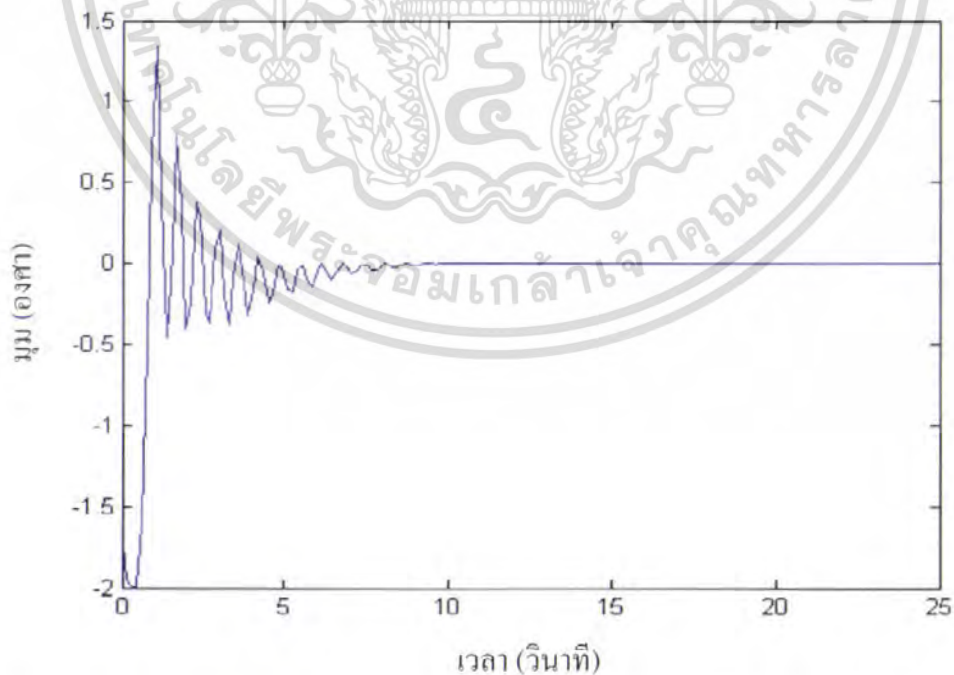
การควบคุมตำแหน่งลูกบอลไปยังตำแหน่งเป้าหมายโดยใช้ตัวควบคุมแบบป้อนกลับสเตทแบบ เซอร์โว ทำการออกแบบตัวควบคุมโดยใช้โปรแกรม Matlab ช่วยในการออกแบบ เมื่อเลือกความถี่ธรรมชาติเท่ากับ 0.7 และ อัตราหน่วงเท่ากับ 0.7 สามารถคำนวณได้ค่าเกน $K_1=1.5727$, $K_2=1.2986$ และ $K_i=0.0865$ ทำการจำลองการควบคุมจากตำแหน่ง (38.8) พิกเซล ไปยังตำแหน่ง (138,108) พิกเซล ซึ่งได้ผลตอบสนองตำแหน่งลูกบอล มีค่าพุ่งเกินเพียงเล็กน้อย และระบบเข้าสู่สภาวะคงตัวที่เวลาประมาณ 12 วินาที ดังรูปที่ 4.11 ความเร็วลูกบอล และสัญญาณควบคุม ดังรูปที่ 4.12 และ รูปที่ 4.13 ตามลำดับ



รูปที่ 4.11 การจำลองผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซลไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท



รูปที่ 4.12 การจำลองสัญญาณความถี่แปรสเกลความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท



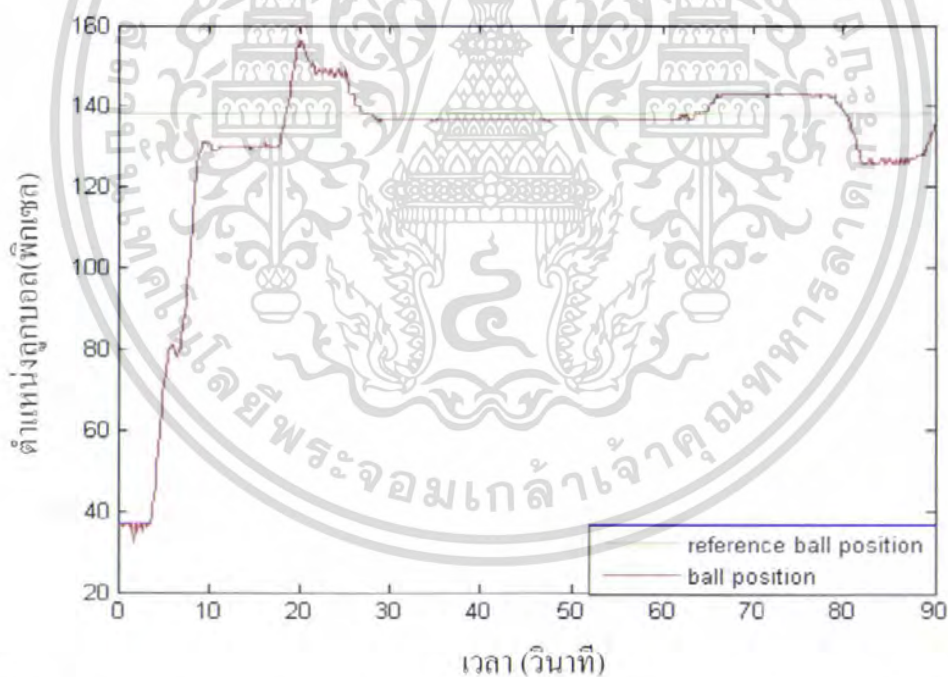
รูปที่ 4.13 การจำลองสัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

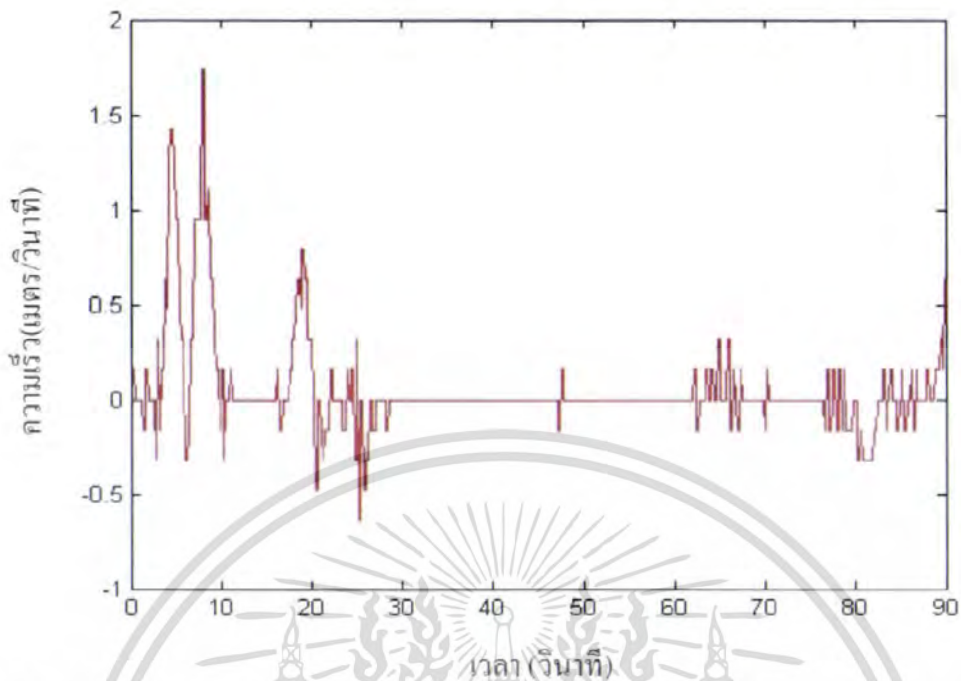
4.6.1 การทดลองควบคุมตำแหน่งลูกบอลโดยใช้ตัวควบคุมแบบป้อนกลับสเตท

การทดลองควบคุมตำแหน่งลูกบอลจากตำแหน่ง (38,8) พิกเซล ไปยังตำแหน่ง (138,108) พิกเซล ได้ผลการทดลองดังรูปที่ 4.14 ซึ่งผลตอบสนองตำแหน่งในแนวแกน X มีค่าพุ่งเกินเล็กน้อยและระบบเข้าสู่สภาวะคงตัวที่เวลาประมาณ 13 วินาที แต่มีค่าผิดพลาดเกิดขึ้นระบบจึงมีการสะสมค่าผิดพลาดในตัวควบคุมแบบอินทิเกรต จึงทำให้เกิดค่าพุ่งเกินเนื่องจากแรงเสียดทานสถิตย์ และมีการแกว่งเกิดขึ้นเป็นบางช่วง ความเร็วในแนวแกน X สัญญาควบคุมแกน X ผลตอบสนองในแนวแกน Y มีค่าพุ่งเกินเล็กน้อยและระบบเข้าสู่สภาวะคงตัวที่เวลาประมาณ 12 วินาที แต่มีค่าผิดพลาดเกิดขึ้นระบบจึงมีการสะสมค่าผิดพลาดในตัวควบคุมแบบอินทิเกรต จึงทำให้เกิดค่าพุ่งเกินเนื่องจากแรงเสียดทานสถิตย์ และมีการแกว่งเกิดขึ้นเป็นบางช่วง ความเร็วในแนวแกน Y สัญญาควบคุมแกน Y ได้ผลการจำลองดังรูปที่ 4.14 รูปที่ 4.15 รูปที่ 4.16 รูปที่ 4.17 รูปที่ 4.18 และรูปที่ 4.19

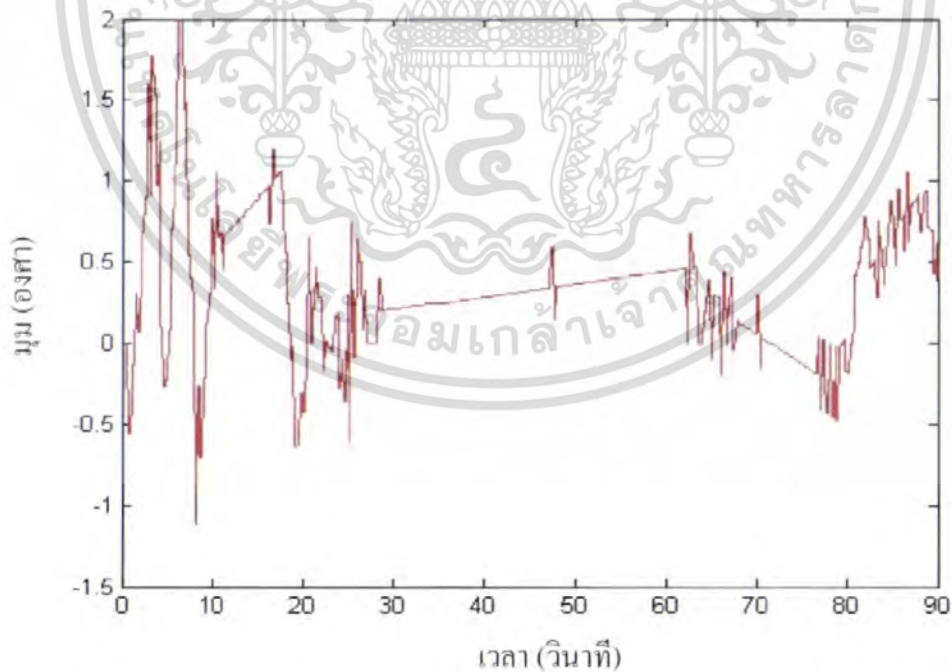
จากการทดลองพบว่า สามารถควบคุมตำแหน่งลูกบอลไปยังตำแหน่งเป้าหมายได้ แต่มีการแกว่งรอบตำแหน่งเป้าหมายเกิดขึ้นเป็นช่วงๆ



รูปที่ 4.14 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท

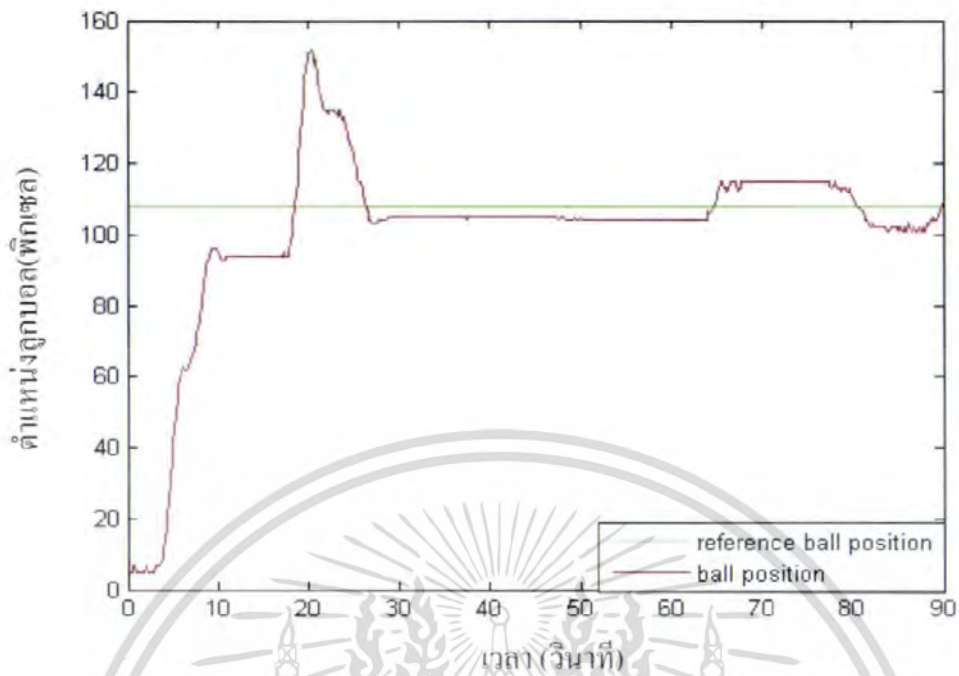


รูปที่ 4.15 สัญญาณควบคุมตัวแปรสแตกความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสแตก

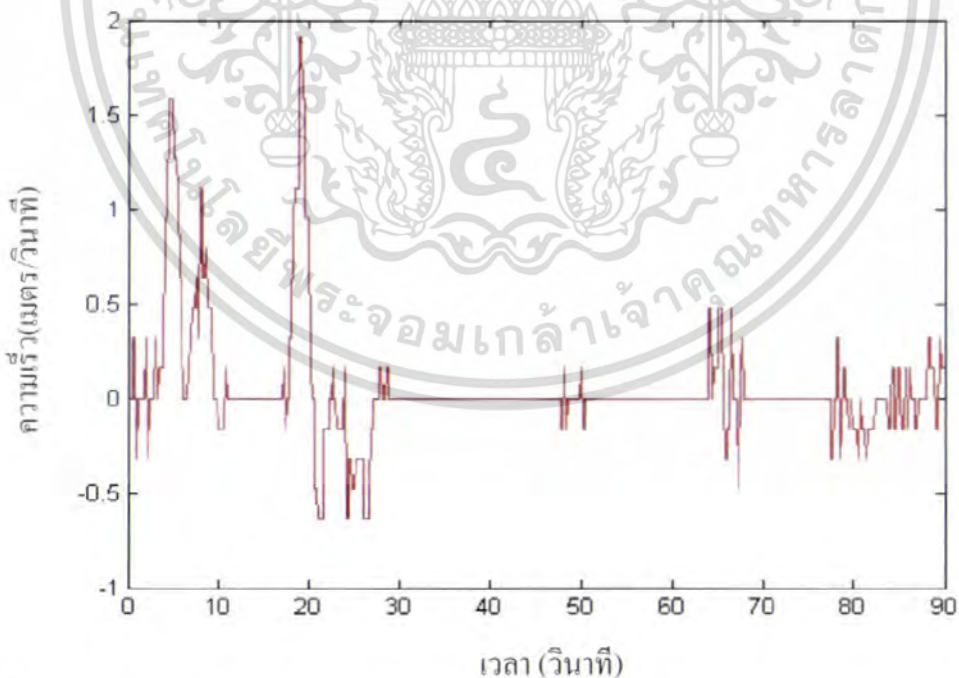


รูปที่ 4.16 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสแตก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

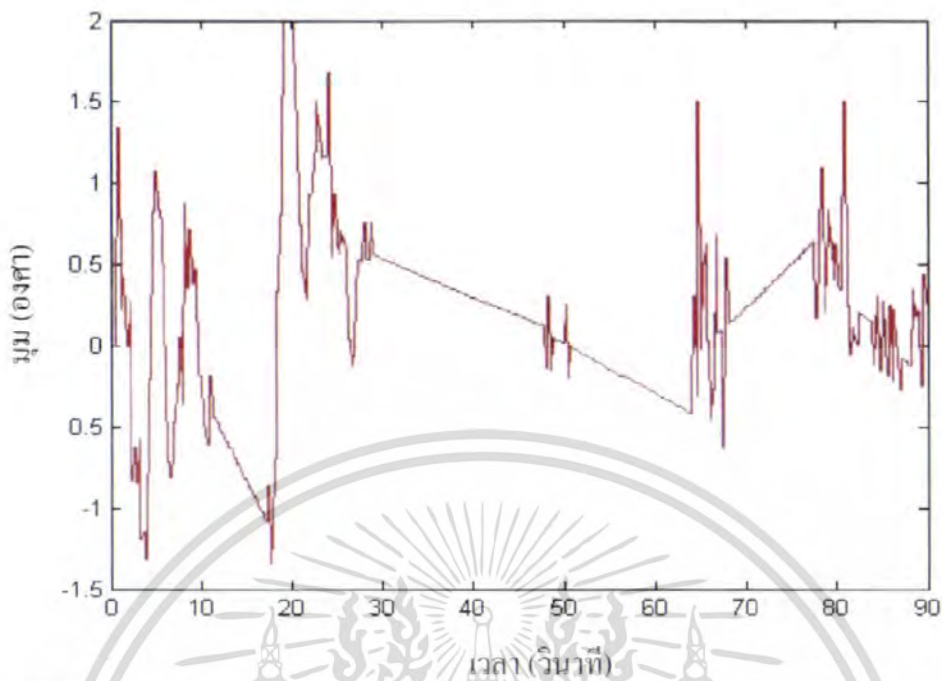


รูปที่ 4.17 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซลไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสแตท



รูปที่ 4.18 สัญญาณควบคุมตัวแปรสแตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซลไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสแตท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

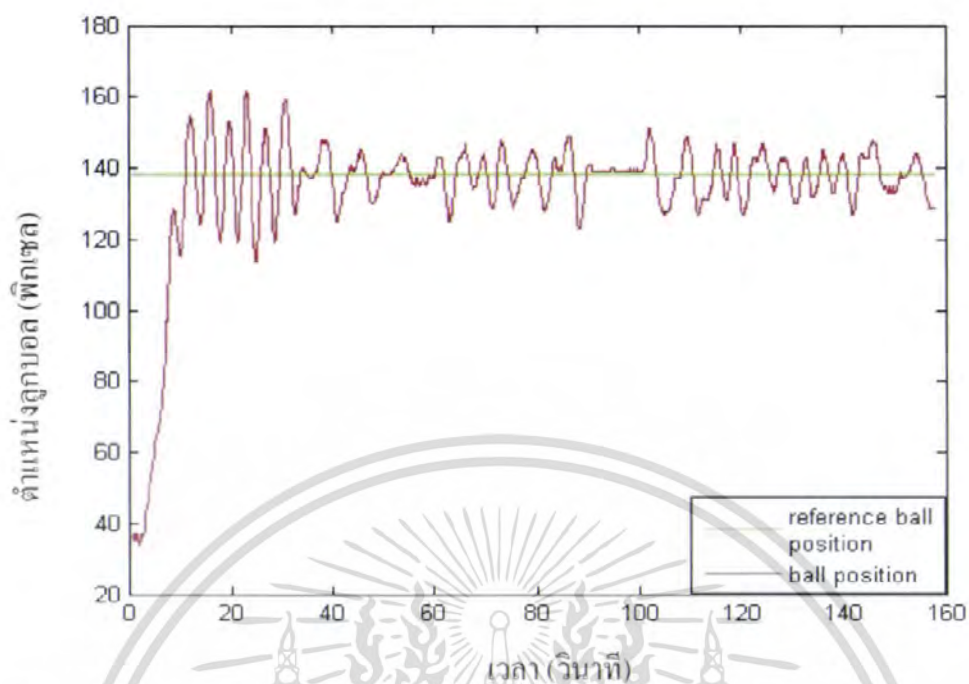


รูปที่ 4.19 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซลไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสแตท

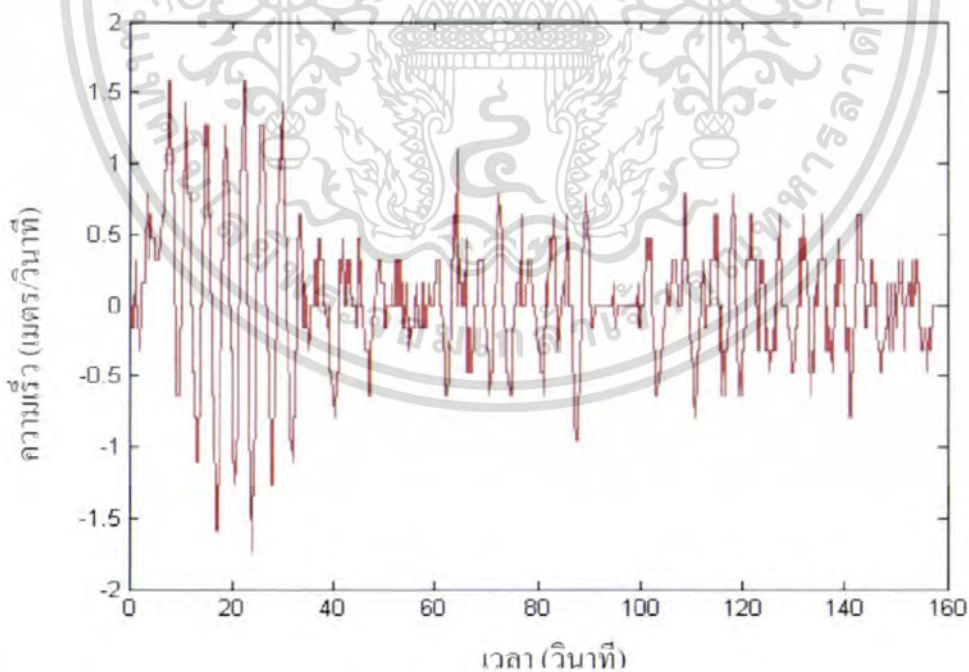
4.6.2 การเขียนโปรแกรมชดเชยผลกระทบของแรงเสียดทานสถิต

จากการทดลอง พบว่าเมื่อระนาบมีการเอียงเกิดขึ้นเล็กน้อย ลูกบอลยังไม่มีการเคลื่อนที่ จึงเขียนโปรแกรมเพื่อชดเชยแรงเสียดทานสถิต โดยโปรแกรมมีเงื่อนไขว่า ให้ระบบสั่งมุมของระนาบให้เอียงเพิ่มขึ้นอีกโดยประมาณ 0.5 องศา ในทิศทางขึ้นหรือลงจากค่ามุมเดิม เมื่อคำสั่งค่ามุมไม่เท่ากับศูนย์แต่ความเร็วของลูกบอลเท่ากับศูนย์ โดยถ้าคำสั่งค่ามุมมากกว่า 0 และความเร็วของลูกบอลเท่ากับ 0 จะได้ว่า คำสั่งค่ามุมใหม่เท่ากับคำสั่งค่ามุมเดิมบวกค่าชดเชย 0.5 องศา หรือถ้าคำสั่งค่ามุนน้อยกว่า 0 และความเร็วของลูกบอลเท่ากับ 0 จะได้ว่าคำสั่งค่ามุมใหม่เท่ากับคำสั่งค่ามุมเดิมลบค่าชดเชย 0.5 องศา

จากการทดลองโดยใช้เงื่อนไขชดเชยผลกระทบของแรงเสียดทานสถิตได้ผลการทดลองดังรูปที่ 4.20 รูปที่ 4.21 รูปที่ 4.22 รูปที่ 4.23 รูปที่ 4.24 และรูปที่ 4.25 พบว่าทำให้ลูกบอลสามารถเคลื่อนที่เข้าหาตำแหน่งเป้าหมายได้เร็วขึ้น แต่มีการแกว่งรอบตำแหน่งเป้าหมายมากขึ้น

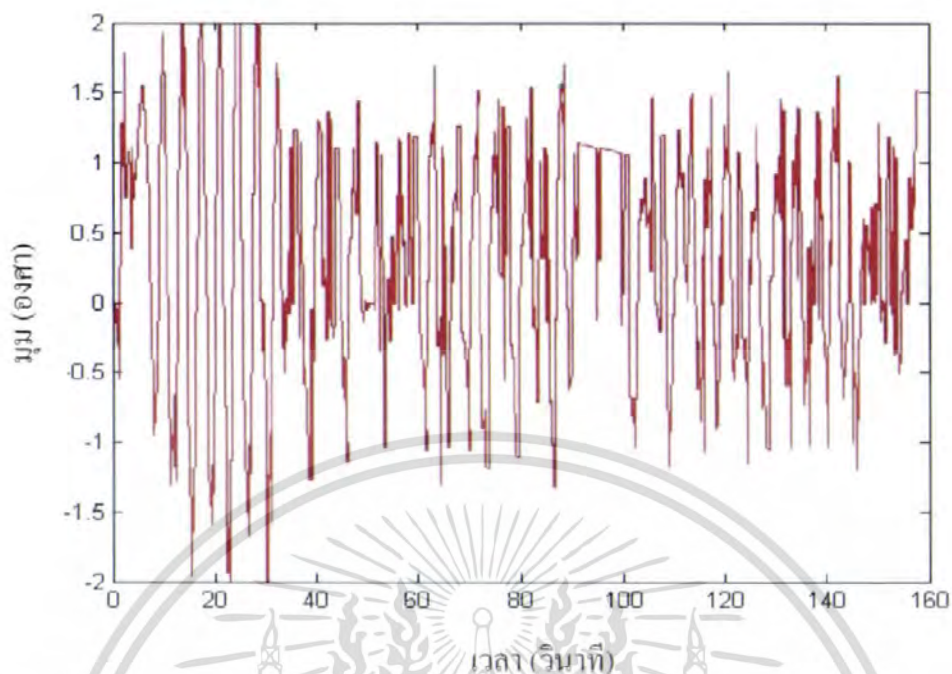


รูปที่ 4.20 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตรต

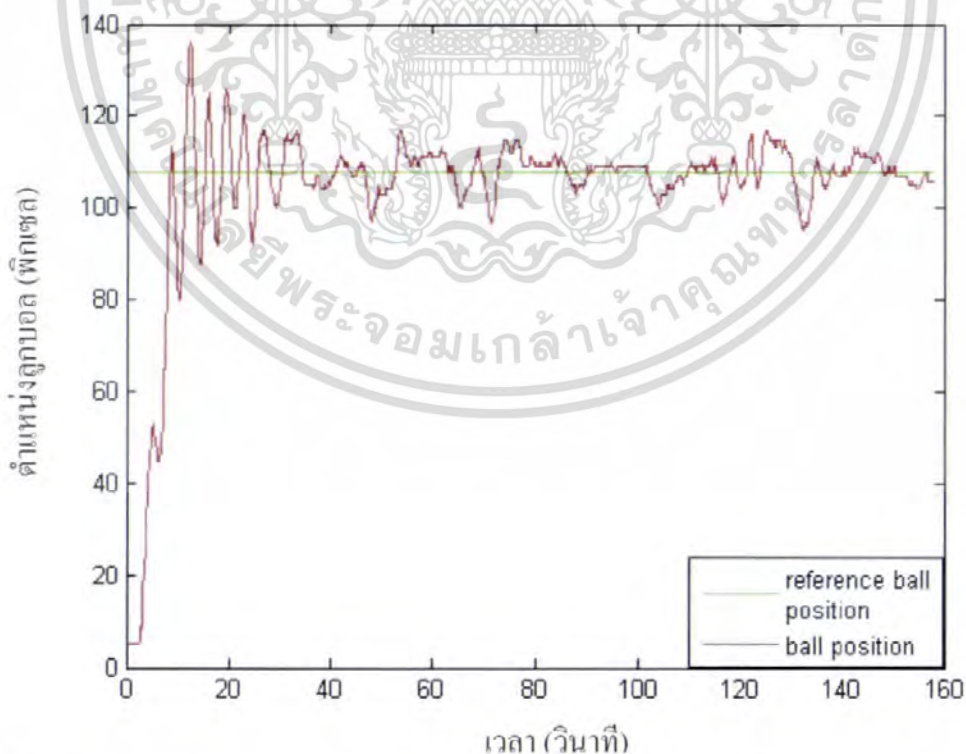


รูปที่ 4.21 สัญญาณควบคุมตัวแปรสเตรตความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตรต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



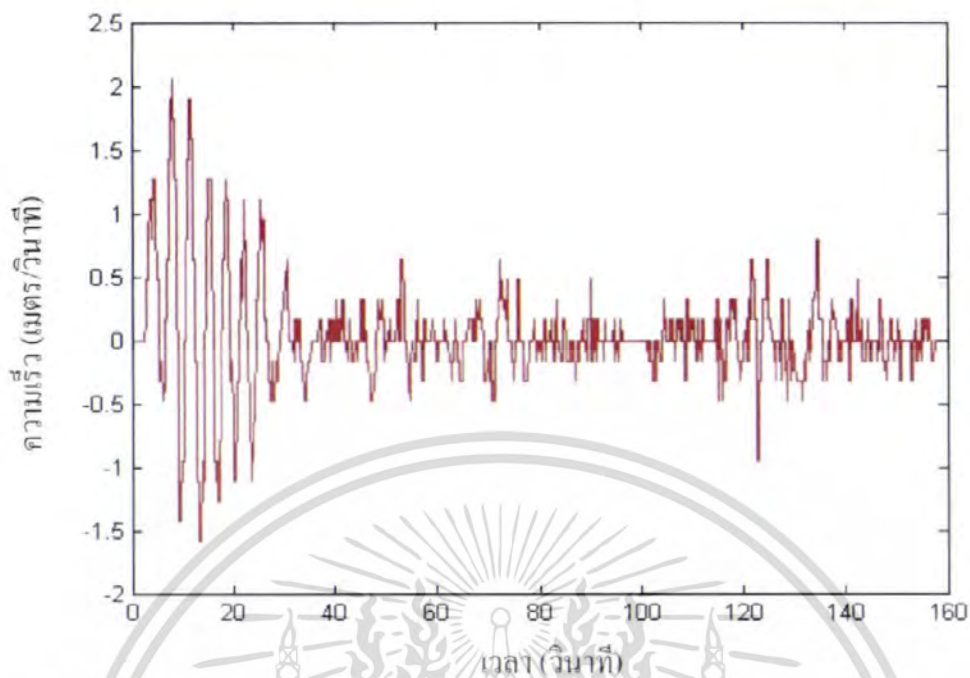
รูปที่ 4.22 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตต



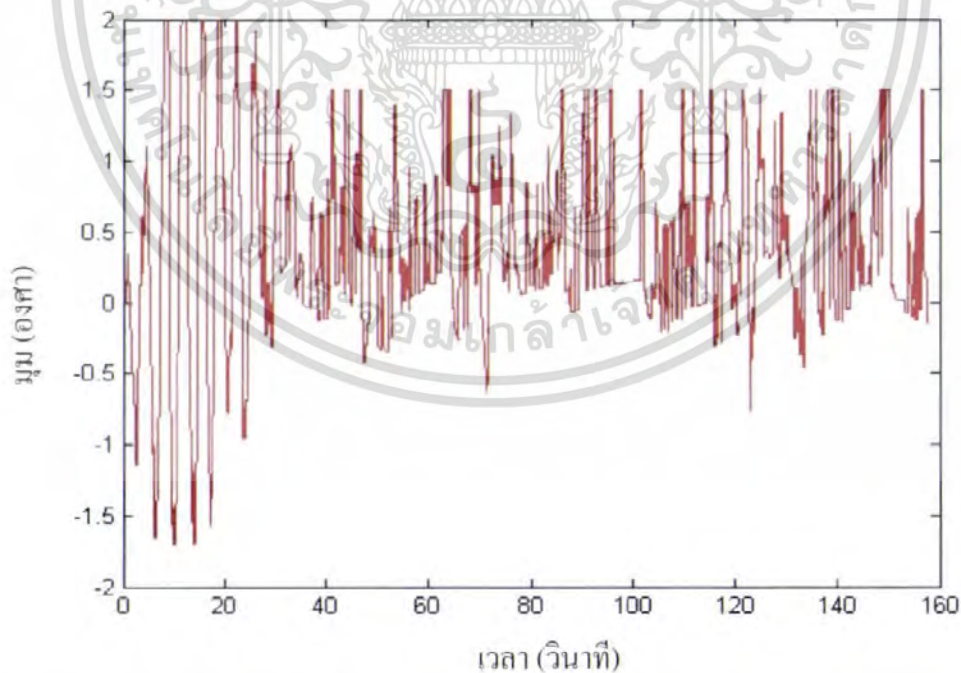
รูปที่ 4.23 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง

8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 สัญญาณควบคุมตัวแปรสแตกความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท



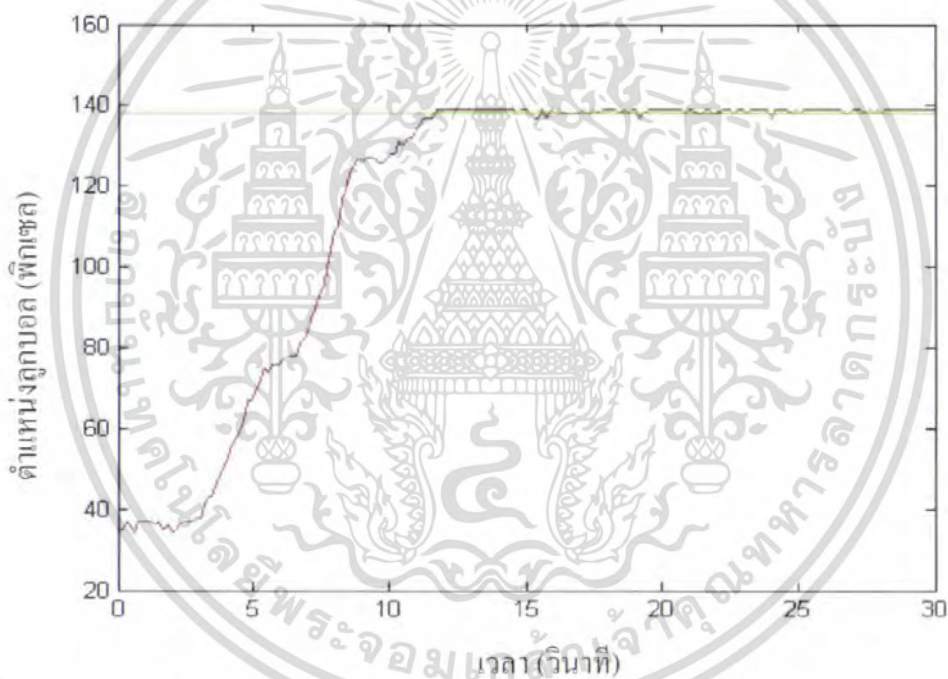
รูปที่ 4.25 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

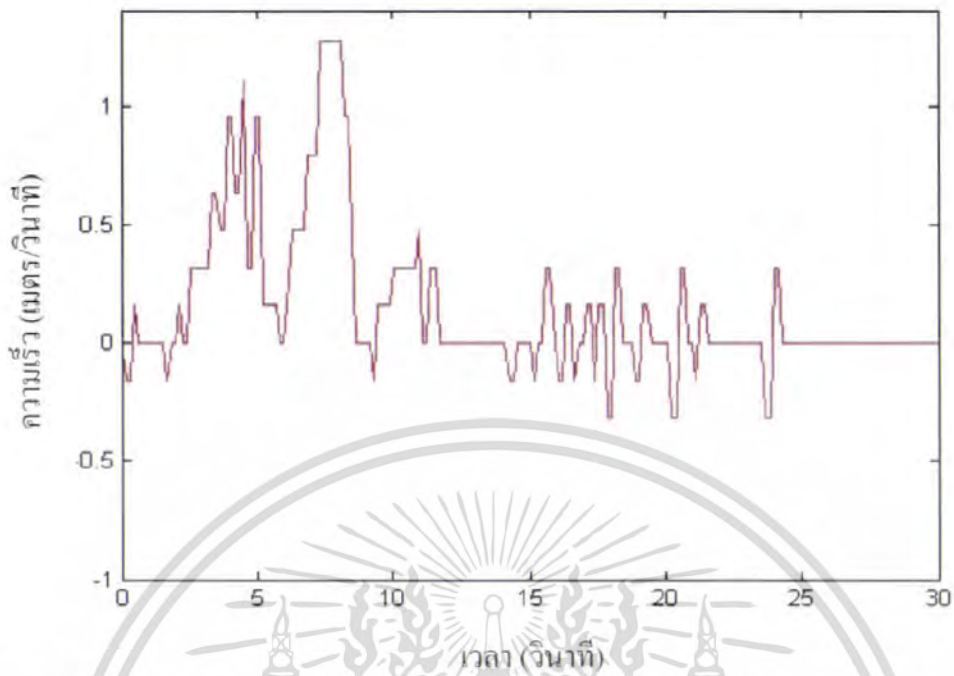
4.6.3 การเขียนโปรแกรมกำหนดช่วงค่าผิดพลาดที่ยอมรับได้

จากการทดลองที่ผ่านมา พบว่าระบบไม่สามารถควบคุมตำแหน่งของลูกบอลให้เข้าสู่ตำแหน่งเป้าหมายที่กำหนดได้ เนื่องจากมีการแกว่งรอบตำแหน่งเป้าหมายจะได้เขียนโปรแกรมขึ้นมาใหม่ โดยเพิ่มเงื่อนไขว่าขณะที่ลูกบอลอยู่ในช่วงของค่าผิดพลาดที่กำหนดไว้ ให้มุมของระนาบมีค่าเป็น 0 องศา โดยกำหนดค่าความผิดพลาดที่ยอมรับได้ที่ 3 พิกเซล หรือประมาณ 0.75 เซนติเมตร

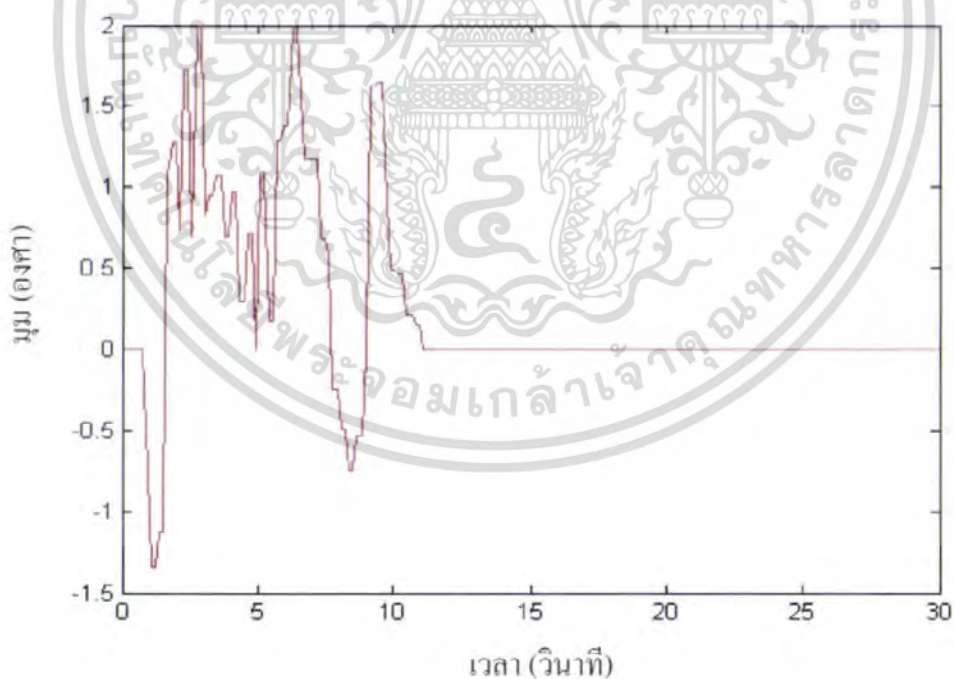
จากการทดลอง โดยใช้เงื่อนไขกำหนดช่วงค่าผิดพลาดที่ยอมรับได้ พบว่าระบบสามารถควบคุมลูกบอล ให้เคลื่อนที่ไปยังบริเวณเป้าหมายที่กำหนด ภายในช่วงค่าผิดพลาดที่ยอมรับได้ และหยุดนิ่ง ทำให้ปัญหาในการแกว่งรอบตำแหน่งเป้าหมายหมดไป โดยมีผลการทดลองดังรูปที่ 4.26 และรูปที่ 4.29



รูปที่ 4.26 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซลไปยังตำแหน่ง 138 พิกเซล โดยอาศัยตัวควบคุมแบบป้อนกลับสเตท

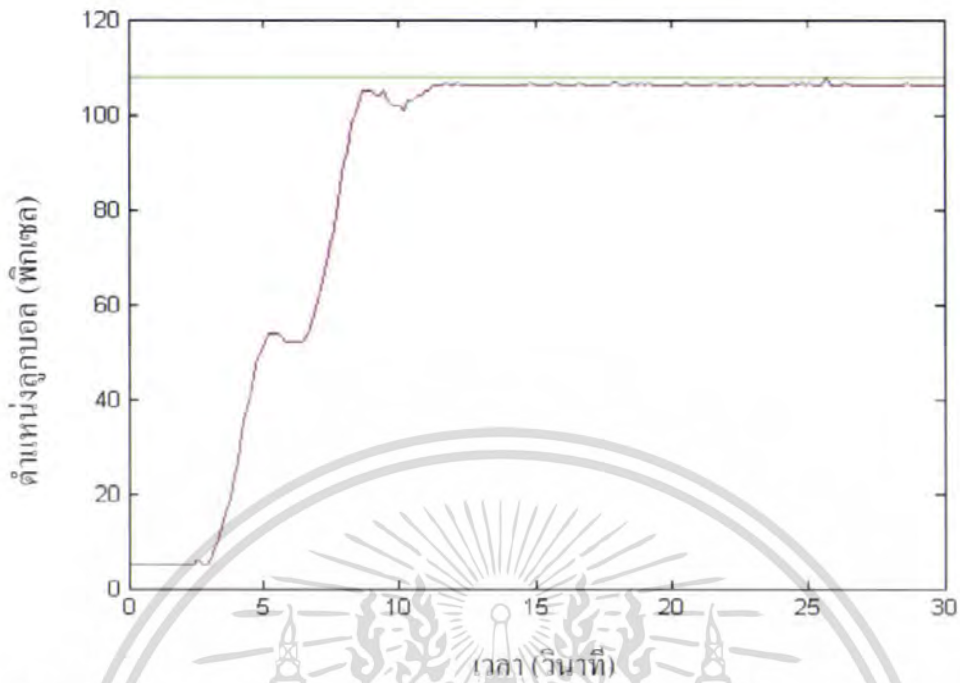


รูปที่ 4.27 สัญญาณควบคุมตัวแปรสเกลความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท

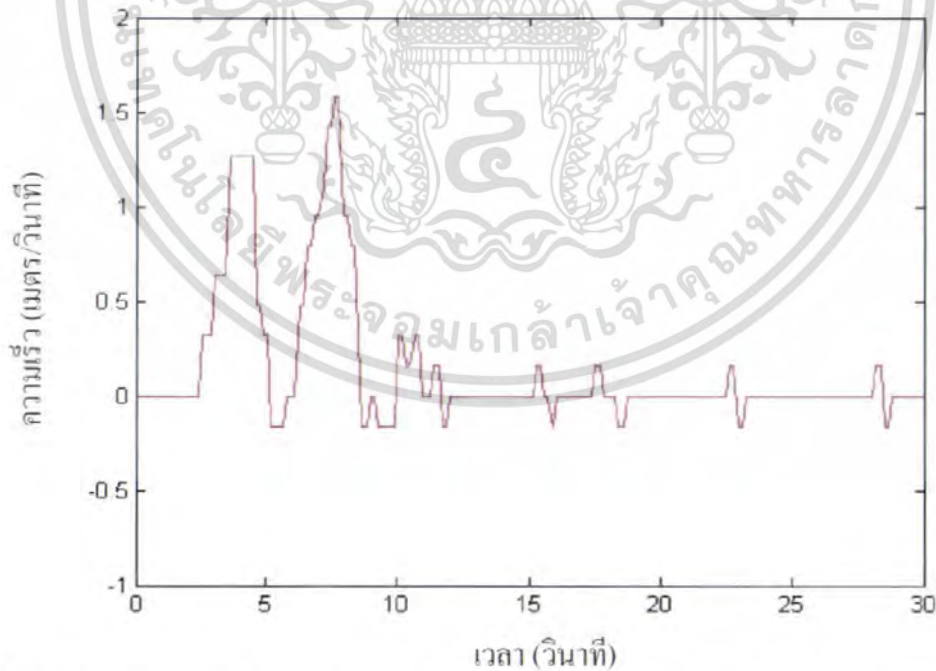


รูปที่ 4.28 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

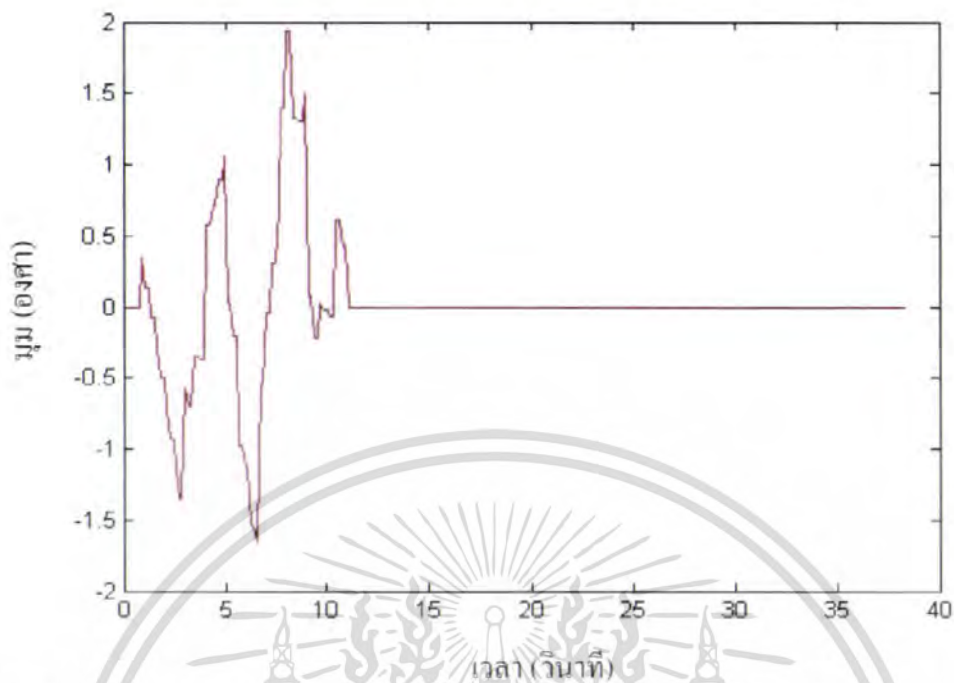


รูปที่ 4.29 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระบบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท



รูปที่ 4.30 สัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระบบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท

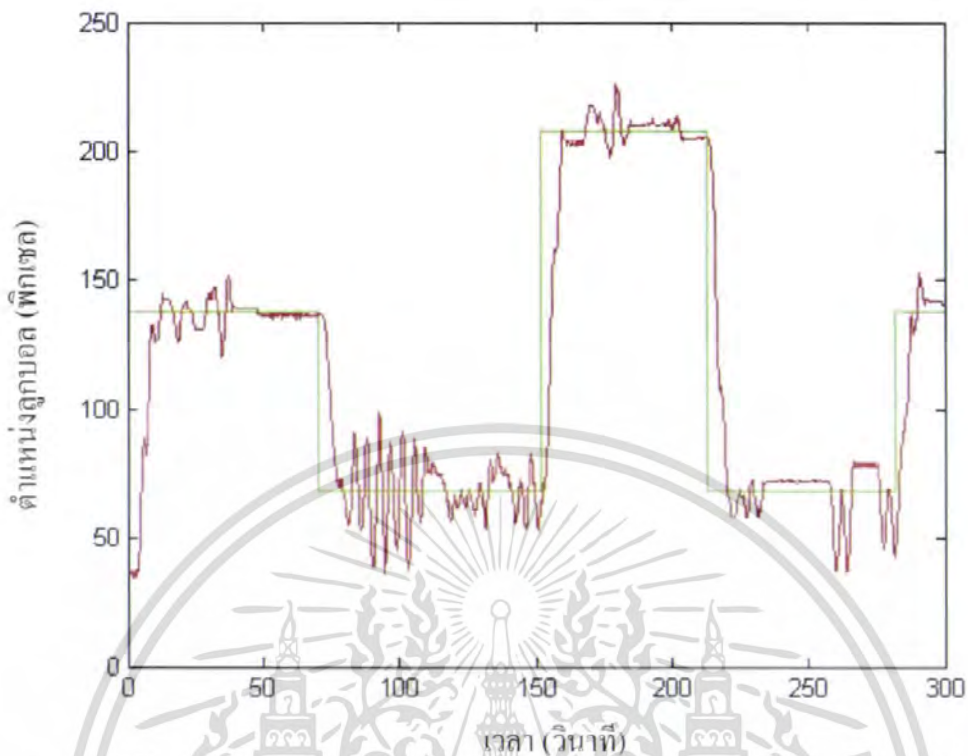
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



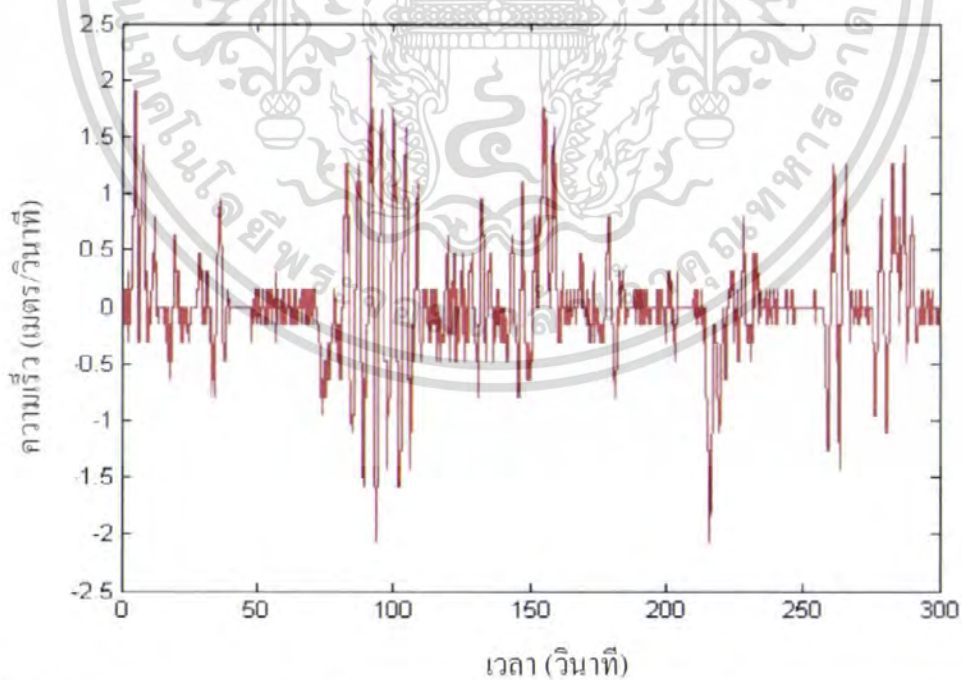
รูปที่ 4.31 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ ในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล โดยตัวควบคุมแบบป้อนกลับสเตท

จากการทดลองข้างต้นที่ผ่านมาพบว่าระบบสามารถควบคุมตำแหน่งลูกบอลให้เคลื่อนที่ไปยังตำแหน่งเป้าหมายบนระนาบได้ แล้ว จึงทำการทดลองเปลี่ยนตำแหน่งเป้าหมายไปยังตำแหน่งต่างๆบนระนาบอย่างต่อเนื่อง พบว่าระบบสามารถควบคุมลูกบอลให้เคลื่อนที่ไปยังตำแหน่งเป้าหมายที่เปลี่ยนไปได้เช่นกัน รวมทั้งทำการเชื่อมลูกบอลเพื่อรับกวนระบบ พบว่าระบบก็ยังสามารถควบคุมลูกบอลให้กลับมาอยู่ที่ตำแหน่งเป้าหมายเดิมได้ โดยใช้เวลาประมาณ 5 วินาที

ในโครงการนี้ได้ทดลองใช้ตัวควบคุมแบบป้อนกลับสเตทร่วมกับตัวสังเกตสเตท เมื่อเลือกความถี่ธรรมชาติเท่ากับ 0.7 และอัตราหน่วงเท่ากับ 0.6 ได้ค่า $11=2$ และ $12=6.67$ สามารถประมาณค่าตัวสังเกตสเตทได้ใกล้เคียงความจริง เมื่อทดลองใช้กับระบบควบคุมลูกบอลบนระนาบ พบว่าผลการทดลองใกล้เคียงกับตัวควบคุมแบบป้อนกลับสเตท

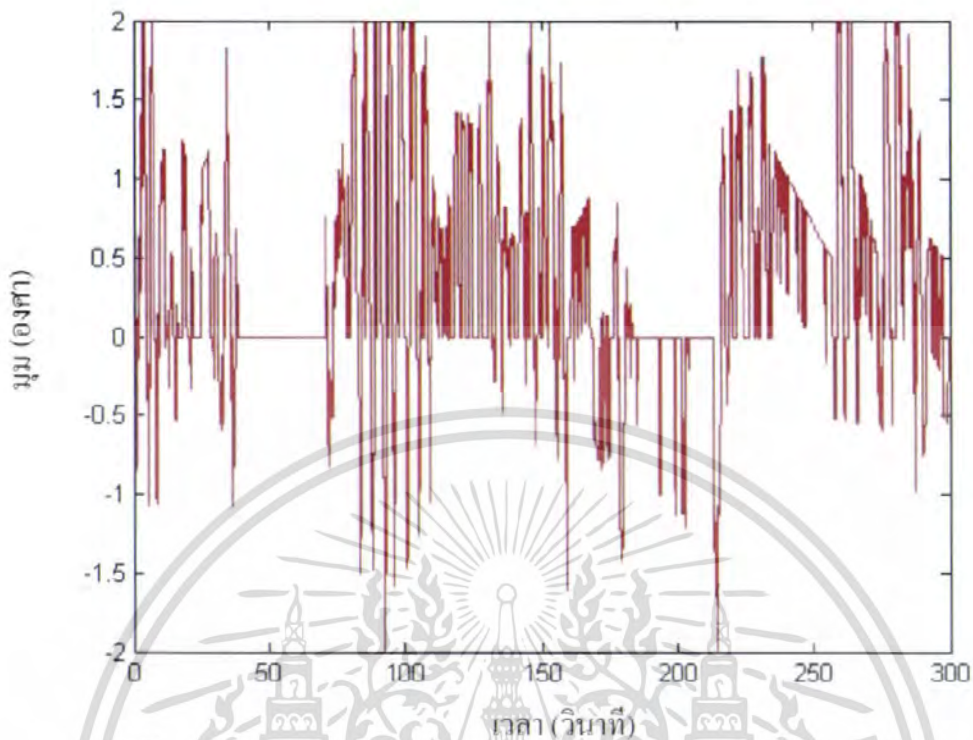


รูปที่ 4.32 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายต่างๆแบบต่อเนื่องในแนวแกน X โดยตัวควบคุมแบบป้อนกลับสเตท

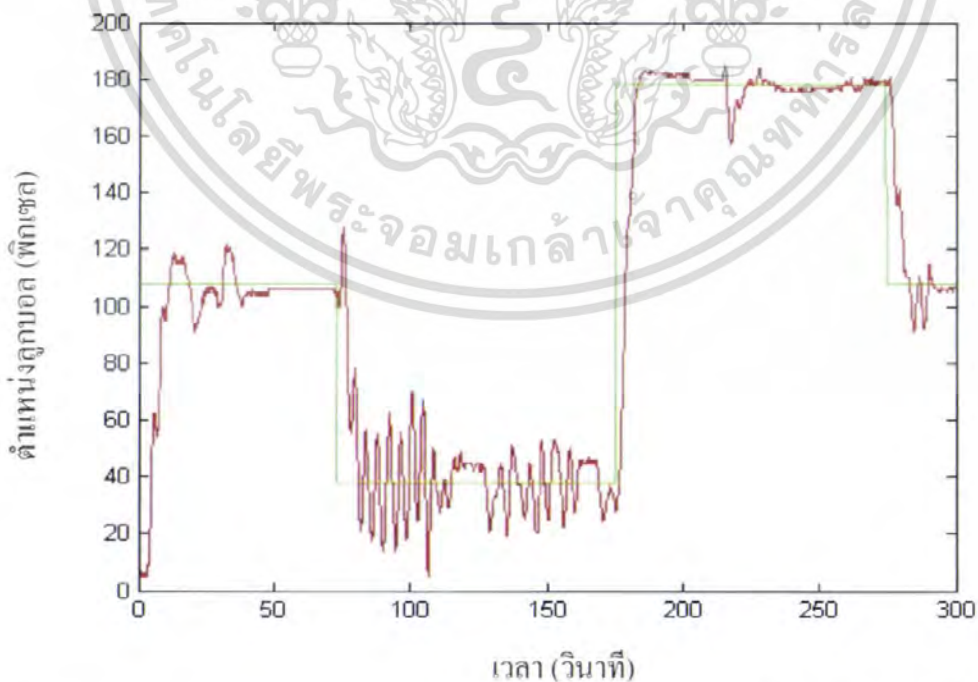


รูปที่ 4.33 สัญญาณควบคุมตัวแปรสเตทความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน X โดยตัวควบคุมแบบป้อนกลับสเตท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

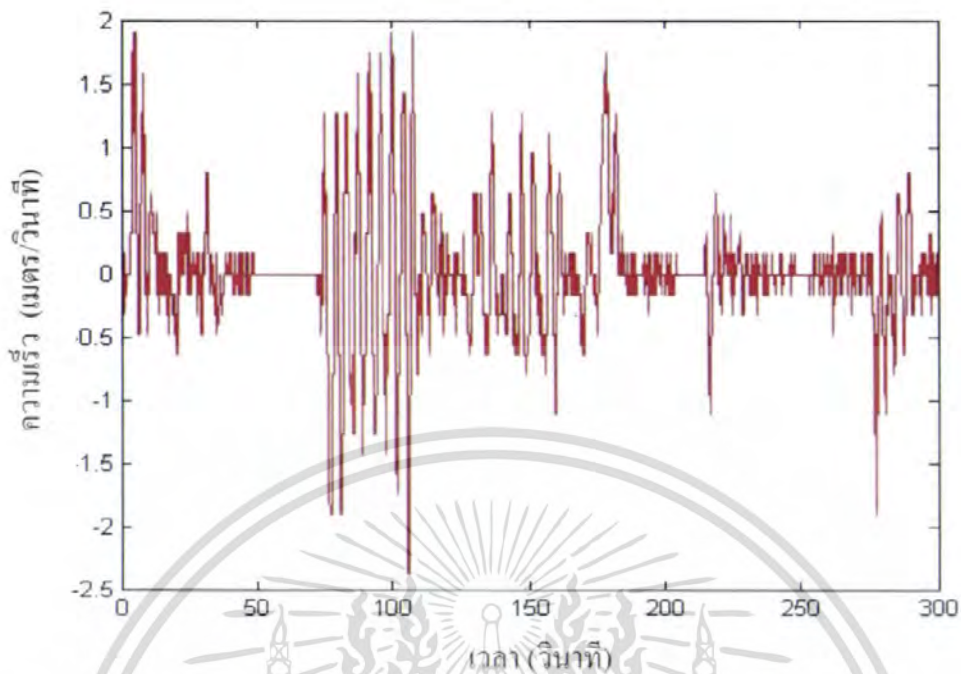


รูปที่ 4.34 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน X โดยอาศัยตัวควบคุมแบบป้อนกลับสเลท

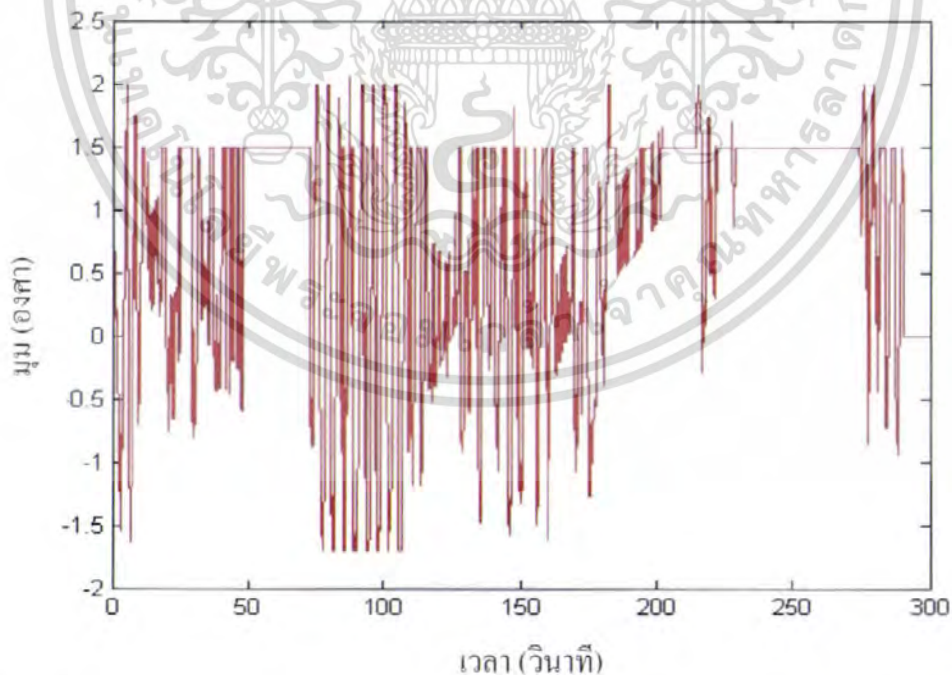


รูปที่ 4.35 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยัง

ตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน Y โดยอาศัยตัวควบคุมแบบป้อนกลับสเลท การคำนวณการคำนวณค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

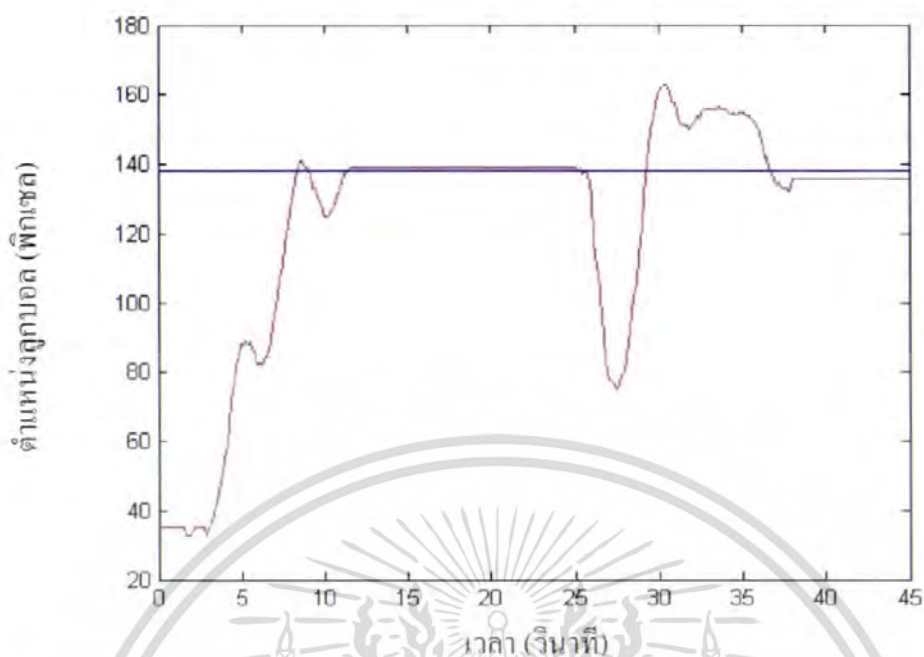


รูปที่ 4.36 สัญญาณควบคุมตัวแปรสตกความเร็วของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน Y

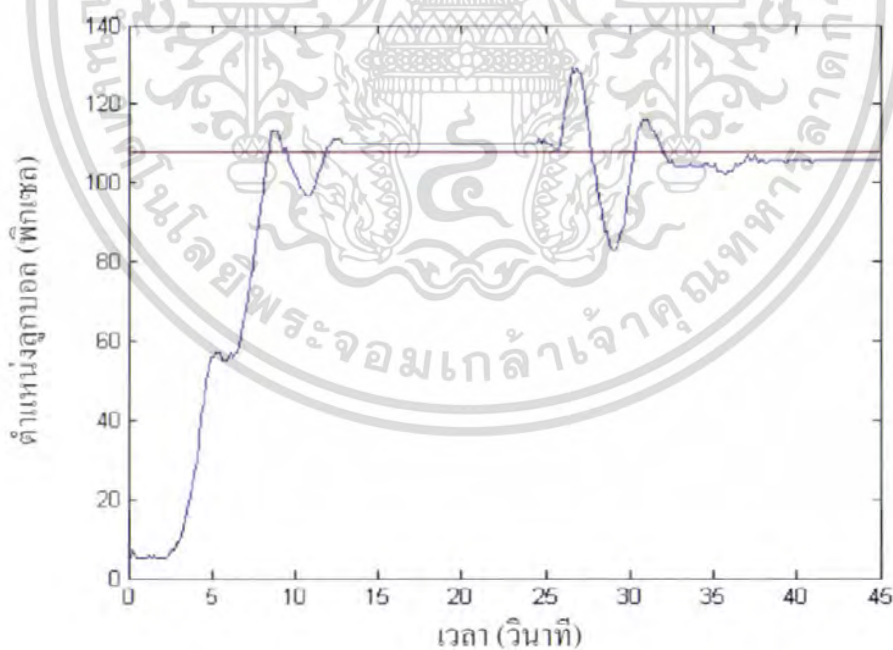


รูปที่ 4.37 สัญญาณควบคุมของระบบควบคุมตำแหน่งลูกบอลบนระนาบ เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายต่างๆ แบบต่อเนื่องในแนวแกน Y โดยอาศัยตัวควบคุมแบบป้อนกลับสเตรต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.38 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบในแนวแกน X จากตำแหน่ง 38 พิกเซล ไปยังตำแหน่ง 138 พิกเซล และมีการรบกวนระบบจากภายนอกโดยตัวควบคุมแบบป้อนกลับสเตท



รูปที่ 4.39 ผลตอบสนองของระบบควบคุมตำแหน่งลูกบอลบนระนาบในแนวแกน Y จากตำแหน่ง 8 พิกเซล ไปยังตำแหน่ง 108 พิกเซล และมีการรบกวนระบบจากภายนอกโดยตัวควบคุมแบบป้อนกลับสเตท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุปผล

5.1 สรุปผลการทดลอง

จากการทดลองระบบควบคุมตำแหน่งลูกบอลบนระนาบ สามารถแบ่งออกเป็น 2 ส่วนหลักๆ ด้วยกัน คือ ส่วนควบคุมมุมระนาบ และควบคุมตำแหน่งบอล

ในส่วนควบคุมตำแหน่งมุมระนาบ ทำการทดลองโดยใช้ตัวควบคุมแบบพีไอดี ในการควบคุมมอเตอร์ ซึ่งมีไมโครคอนโทรลเลอร์เป็นหน่วยประมวลผล และใช้โพเทนชิโอมอเตอร์ขนาด 10 กิโลโอห์ม เป็นอุปกรณ์วัดมุมระนาบ พบว่าสามารถควบคุมตำแหน่งมุมระนาบ ได้ตามที่กำหนด

ในส่วนควบคุมตำแหน่งลูกบอลบนระนาบ เพื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายโดยอาศัยตัวควบคุมแบบตัวชดเชยชนิดมูมนำ ระบบสามารถควบคุมลูกบอลให้เคลื่อนที่ไปยังบริเวณเป้าหมายที่ต้องการ ได้จริง แต่ในช่วงเริ่มต้นมีการแกว่งรอบบริเวณเป้าหมาย และยังมีคามผิดพลาดของตำแหน่งลูกบอลอยู่ จึงได้ทดลองใช้ตัวควบคุมแบบป้อนกลับสเตรท พบว่าระบบสามารถควบคุมลูกบอลให้เคลื่อนที่ไปยังบริเวณเป้าหมาย ได้จริงแต่ยังมีการแกว่งรอบเป้าหมาย และเกิดคามผิดพลาดของตำแหน่งลูกบอล จึงได้ทดลองเพิ่มมูมในช่วงเริ่มต้นขณะลูกบอลหยุดนิ่งและยอมรับคามผิดพลาด ซึ่งทำให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมายที่ต้องการได้

5.2 ปัญหาที่พบและแนวทางการแก้ไข

จากการทดลองพบว่ามอเตอร์ไฟฟ้ากระแสตรงที่ใช้ในเบื้องต้นมีความเร็วรอบมากเกินไป จึงทำให้การควบคุมมุมระนาบ เมื่อมูมมีการเปลี่ยนแปลงเล็กน้อยทำได้ยาก จึงแก้ไขด้วยการเลือกใช้มอเตอร์ที่มีความเร็วรอบน้อยลง

ในการควบคุมมุมของระนาบนั้น ในช่วงแรกมีคามพุ่งเกินเกิดขึ้น จึงแก้ไขโดยการใส่แอนไท-ไวต์อัพ ที่ตัวควบคุมแบบไอ เพื่อเป็นการลดช่วงเวลาอิมพัลส์ของสัญญาณควบคุม

ในการตรวจจับตำแหน่งลูกบอลโดยใช้กล้องดิจิตอลวีดีโอ นั้น พบว่าแสงที่มีการเปลี่ยนแปลงมีผลทำให้การตรวจจับตำแหน่งลูกบอลผิดพลาด แนวทางการแก้ไข คือการทำการทดลองในห้องปิดที่ไม่มีการเปลี่ยนแปลงของแสง

ปัญหาอีกประการหนึ่ง คือผลกระทบเนื่องจากแรงเสียดทานสถิตย์ ซึ่งมีผลทำให้ลูกบอลไม่เคลื่อนที่เมื่อมูมเอียงมีค่าน้อยๆ แก้ไขได้โดยการ โปรแกรมชดเชยมุมเริ่มต้นในกรณีที่ลูกบอลหยุดนิ่ง เพื่อชดเชยแรงเสียดทานสถิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่พบและยังไม่สามารถทำการปรับปรุงแก้ไข ได้แก่ ปัญหาการวัดค่ามุมของระนาบ ที่ไม่สามารถทำได้อย่างเที่ยงตรงและแม่นยำ เนื่องจากขาดอุปกรณ์วัดเทียบที่เที่ยงตรงและแม่นยำเพื่อใช้ปรับแต่งอุปกรณ์วัด

5.3 ข้อเสนอแนะ

ตัวควบคุมที่ใช้ในปัจจุบันมีหลายแบบ จึงควรศึกษาเพิ่มเติมว่าการใช้ตัวควบคุมแบบใดที่จะเหมาะสมกับการควบคุมตำแหน่งลูกบอลบนระนาบ

ในส่วนการพัฒนาโปรแกรมควบคุม ควรเปลี่ยนมาใช้โปรแกรมวิชวลซีพลัสพลัส แทนวิชวลเบสิก ซึ่งจะช่วยให้การประมวลผลทำได้เร็วมากขึ้น นอกจากนั้นควรเลือกใช้กล้องดิจิตอลวีดีโอที่มีความไวและความละเอียดสูง เพื่อช่วยให้การตรวจจับตำแหน่งลูกบอลทำได้ดียิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] เป็นเอก เพ็งวัน, สุริยันต์ โชตะศรี. “การควบคุมตำแหน่งลูกบอลบนคาน.” ปรินญาณิพนธ์ วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2548.
- [2] สัจจะ จรัสรุ่งรวีวร. คู่มือการเขียนโปรแกรม Visual Basic 6.0 ฉบับผู้เริ่มต้น. นนทบุรี : ไอดี ซีๆ. 2548.
- [3] ประภาพร ช่างไม้. คู่มือการเขียนโปรแกรมภาษา C ฉบับผู้เริ่มต้น. นนทบุรี : อินโฟเพรส. 2545.
- [4] ประจัน พลังสันติกุล. เรียนรู้และการใช้งาน CCS C คอมไพเลอร์. กรุงเทพฯ : อินโนเวตีฟ เอ็ดจิวเออร์เมนท์. 2548.

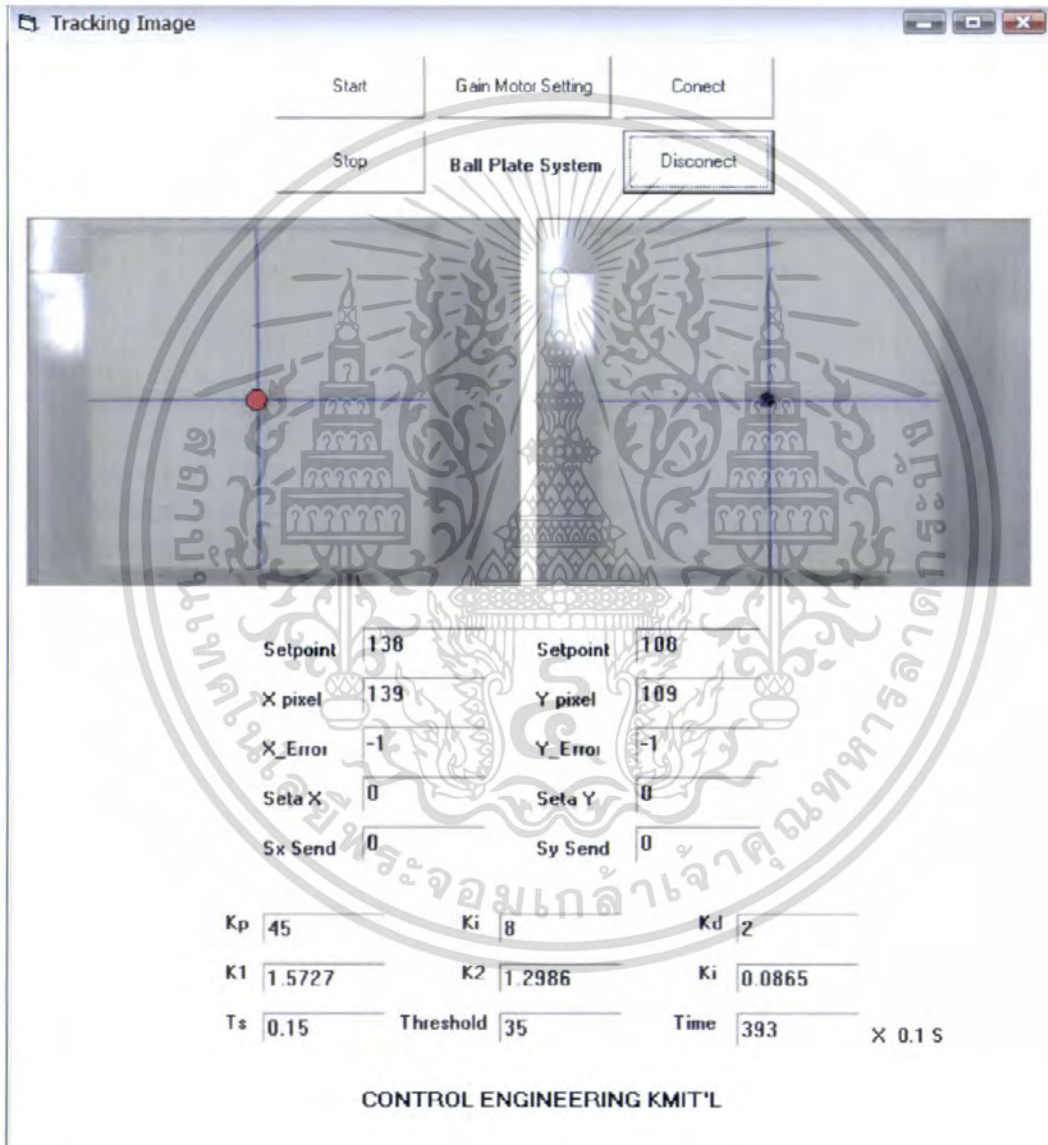


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม

โปรแกรมวิหวลเบสิก ซึ่งเขียนเพื่อใช้ในการตรวจสอบตำแหน่งลูกบอลบนคาน และควบคุม ประกอบด้วย หน้าต่างเชื่อมต่อกับผู้ใช้ (Graphical User Interface) ดังรูปที่ ก.1 และ โค้ดโปรแกรมหาดังต่อไปนี้



รูปที่ ก.1 หน้าต่างเชื่อมต่อกับผู้ใช้ใน โปรแกรมการตรวจสอบตำแหน่งลูกบอลบนระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมมอเตอร์ในไมโครคอนโทรลเลอร์

```

#include <16F877.h>
#define adc=10
#define FUSES NOWDT           //No Watch Dog Timer
#define FUSES HS              //High speed Osc (> 4mhz)
#define FUSES NOPUT          //No Power Up Timer
#define FUSES NOPROTECT      //Code not protected from reading
#define FUSES NOBROWNOUT     //No brownout reset
#define FUSES NOLVP          //No low voltage prgming. B3(PIC16) or B5(PIC18) used for I/O
#define FUSES NOCPD           //No EE protection
#define FUSES NOWRT          //Program memory not write protected
#define FUSES NODEBUG        //No Debug mode for ICD
#define use delay(clock=2000000)
#define use rs232(baud=57600,parity=N,xmit=PIN_C6,rev=PIN_C7,bits=8)
int8 tmp,count,TimeCount,data[10],start;
int16 Uxi,Uyi;
float Kp,Kd,Ki,Ts,ex,Upx,Uix,Udx,Ux,Uix0,ex0,ADC1,X,value_data_x,sign_x;
float      ey,Upy,Uiy,Udy,Uy,Uiy0,ey0,ADC2,Y,value_data_y,sign_y;
#define byte rcreg = 0x1a
void convert_show(void)
{
float joke;
if(count == 10)
{
data[0] = 0x30;
data[1] = 0x30;
data[2] = 0x30;
data[3] = 0x30;
data[4] = 0x30;
data[5] = 0x30;
data[6] = 0x30;
data[7] = 0x30;
data[8] = 0x30;
data[9] = 0x30;
count=0;
start = data[0];
if(start==1)
{
value_data_x = data[1];
joke      = data[6];
value_data_x += joke/10;
joke      = data[7];
value_data_x += joke/100;
sign_x    = data[2];
value_data_y = data[3];
joke      = data[8];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

value_data y += joke/10;
joke = data[9];
value_data y += joke/100;
sign_y = data[4];
//Kp=35;
//Ki=8.5;
//Kd=1;
if(sign_x == 0)
{
value_data x = value_data x;
X = (value_data x+10.1016244)/0.0203252;
}
else if (sign_x == 1)
{
value_data x = -value_data x;
X = (value_data x+10.1016244)/0.0203252;
}
if(sign_y == 0)
{
value_data y = value_data y;
Y = (value_data y+10.61181615)/0.02109705;
}
else if (sign_y == 1)
{
value_data y = -value_data y;
Y = (value_data y+10.61181615)/0.02109705;
}
}
else if(start == 2)
{
Kp = (data[1]*10)+data[2];
Ki = (data[3]*10)+data[4];
Kd = data[5];
X = 10.1016244/0.0203252;
Y = 10.61181615/0.02109705;
}
}
}

```

```
void ADC(void)
```

```

{
set_adc_channel(0);
delay_us(10);
ADC1 = read_adc();
delay_us(10);
set_adc_channel(1);
delay_us(10);
ADC2 = read_adc();
delay_us(10);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Gain_PID(void)
{
  ex = X-ADC1;
  ey = Y-ADC2;
  Upx = Kp*ex;
  Upy = Kp*ey;
  Udx = Kd*(ex-ex0)/Ts;
  Udy = Kd*(ey-ey0)/Ts;
  if((Ux>=350)&(Ux<=350)) { Uix=Uix0+Ki*(Ts/2)*(ex+ex0);}
  Uix0 = Uix;
  ex0 = ex;
  Ux = Upx+Uix+Udx+150;
  if((Uy>=150)&(Uy<=150)) { Uiy=Uiy0+Ki*(Ts/2)*(ey+ey0);}
  Uiy0 = Uiy;
  ey0 = ey;
  Uy = Upy+Uiy+Udy+600;
}

void DriveMotor_X(void)
{
  if(Ux>1023.0)
  {
    Uxi=1023;
    output_high(pin_b0);
    output_low(PIN_b1);
  }
  if((Ux>0.0)&(Ux<1023.0))
  {
    Uxi=(int16)Ux;
    output_high(pin_b0);
    output_low(PIN_b1);
  }
  if(Ux==0.0)
  {
    Uxi=0;
    output_high(pin_b0);
    output_high(PIN_b1);
  }
  if(Ux<0.0)
  {
    Ux=(-1)*(Ux);
    Uxi=(int16)Ux;
    if(Uxi>1023)
    {
      Uxi=1023;
      output_high(pin_b1);
      output_low(PIN_b0);
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else
    {
        Uxi=Uxi;
        output_high(pin_b1);
        output_low(PIN_b0);
    }
    set_pwm1_duty(Uxi);
}
void DriveMotor_Y(void)
{
    if(Uy>1023.0)
    {
        Uyi=1023;
        output_high(pin_b2);
        output_low(PIN_b3);
    }
    if((Uy>0.0)&(Uy<1023.0))
    {
        Uyi=(int16)Uy;
        output_high(pin_b2);
        output_low(PIN_b3);
    }
    if(Uy==0.0)
    {
        Uyi=0;
        output_high(pin_b2);
        output_high(PIN_b3);
    }
    if(Uy<0.0)
    {
        Uy=(-1)*Uy;
        Uyi=(int16)Uy;
        if(Uyi>1023)
        {
            Uyi=1023;
            output_high(pin_b3);
            output_low(PIN_b2);
        }
        else
        {
            Uyi=Uyi;
            output_high(pin_b3);
            output_low(PIN_b2);
        }
    }
    set_pwm2_duty(Uyi);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
#endif
void RDA_isr()
{
  if(count < 10)
  {
    data[count] = rcreg;
    count++;
  }
  else
  {
    tmp = rcreg;
  }
}
#endif
void TIMER0_isr()
{
  //restart_wdt();
  TimeCount++;
}
void main()
{
  port_b_pullups(TRUE);
  setup_adc_ports(RA0_RA1_RA3_ANALOG);
  setup_adc(ADC_CLOCK_INTERNAL);
  setup_psp(PSP_DISABLED);
  setup_spi(FALSE);
  setup_timer_0(RTCC_INTERNAL/RTCC_DIV_32); // Overflow very 1.6msec
  set_timer0(6);
  setup_timer_1(T1_DISABLED);
  setup_timer_2 ( T2_DIV_BY_1,255.1);
  enable_interrupts(INT_TIMER0);
  enable_interrupts(INT_RDA);
  enable_interrupts(GLOBAL);
  setup_ccp1(CCP_PWM);
  setup_ccp2(CCP_PWM);
  set_tris_c(0x81);
  Ts=0.0016;
  Ux=0;
  ex0=0;
  Uix0=0;
  Uy=0;
  ey0=0;
  Uiy0=0;
  count = 0;
  TimeCount = 0;
  while(true)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสำหรับที่ใช้ในการควบคุมตำแหน่งลูกบอลบนระนาบ

Option Explicit

Dim RoddersMotion As New ClassMotion

Dim Xi, X1, X1N_1, X2, Ex As Single

Dim Ux As Single

Dim Yi, Y1, Y1N_1, Y2, Ey As Single

Dim Uy As Single

Dim co, Th As Integer

Dim time, Nub, loops As Long

Public OutY, OutX, comX, comY, IX, IY, Xdot, Ydot

Public Fso

Public m As Integer

Public N As Integer

Public O As Integer

Public p As Integer

Public Q As Integer

Public r As Integer

Public S As Integer

Public T As Integer

Public U As Integer

Public V As Integer

Public minx As Integer

Public miny As Integer

Public maxx As Integer

Public maxy As Integer

Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" _

(ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Long,

lParam As Any) As Long

Private Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias _

"capCreateCaptureWindowA" (ByVal lpszWindowName As String, ByVal

dwStyle As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long,

ByVal nHeight As Long, ByVal hwndParent As Long, ByVal nID As Long) As Long

Private mCapHwnd As Long

Private Const CONNECT As Long = 1034

Private Const DISCONNECT As Long = 1035

Private Const GET_FRAME As Long = 1084

Private Const COPY As Long = 1054

Private Const VIDEOFORMAT As Long = 1065

Private Const VIDEOSOURCE As Long = 1066

Private Const VIDEODISPLAY As Long = 1067

Private Const VIDEOCOMPRESSION As Long = 1070

Private Const PREVIEW As Long = 1074

Private Declare Function GetObject Lib "gdi32" Alias "GetObjectA" _

(ByVal hObject As Long, ByVal nCount As Long, lpObject As Any) As Long

Private Declare Function GetDIBits Lib "gdi32" (ByVal aHDC As Long, _

ByVal hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans As

เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
Private Declare Function SetDIBits Lib "gdi32" (ByVal hdc As Long, _
ByVal hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans _
As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
Private Type BITMAP '14 bytes
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type
Private Type BITMAPINFOHEADER '40 bytes
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As Long
    biSizeImage As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type
Private Type RGBQUAD
    rgbBlue As Byte
    rgbGreen As Byte
    rgbRed As Byte
    rgbReserved As Byte
End Type
Private Type BITMAPINFO
    bmiHeader As BITMAPINFOHEADER
    bmiColors As RGBQUAD
End Type
Private Const DIB_RGB_COLORS = 0&
Private Const BI_RGB = 0&
Private Const pixR As Integer = 3
Private Const pixG As Integer = 2
Private Const pixB As Integer = 1
Private Sub Find_Object(ByVal picColor As PictureBox)
Dim bitmap_info As BITMAPINFO
Dim pixels() As Byte
Dim bytes_per_scanLine As Integer
Dim pad_per_scanLine As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim x As Integer
Dim y As Integer
Dim ave_color As Byte
minx = picColor.Width
miny = picColor.Height
maxx = 0
maxy = 0
' Prepare the bitmap description.
With bitmap_info.bmiHeader
.biSize = 40
.biWidth = picColor.ScaleWidth
' Use negative height to scan top-down.
.biHeight = -picColor.ScaleHeight
.biPlanes = 1
.biBitCount = 32
.biCompression = BI_RGB
bytes_per_scanLine = (((biWidth * biBitCount) + 31) \ 32) * 4
pad_per_scanLine = bytes_per_scanLine - (((biWidth * biBitCount) + 7) \ 8)
.biSizeImage = bytes_per_scanLine * Abs(biHeight)
End With
' Load the bitmap's data.
ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
GetDIBits picColor.hdc, picColor.Image,
0, picColor.ScaleHeight, pixels(1, 1, 1),
bitmap_info, DIB_RGB_COLORS
' Modify the pixels.
Th = Text15.Text
For y = 1 To picColor.ScaleHeight
For x = 1 To picColor.ScaleWidth
If (pixels(pixB, x, y) < Th) And (pixels(pixG, x, y) < Th) And (pixels(pixR, x, y) < Th) Then
pixels(pixR, x, y) = 0
pixels(pixG, x, y) = 0
pixels(pixB, x, y) = 0
If minx > x Then minx = x
If maxx < x Then maxx = x
If miny > y Then miny = y
If maxy < y Then maxy = y
Else
'pixels(pixR, X, Y) = 255
'pixels(pixG, X, Y) = 255
'pixels(pixB, X, Y) = 255
End If
Next x
Next y
N = (minx + maxx) / 2
m = (miny + maxy) / 2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'picColor.Line (N, miny)-(M, maxy), vbGreen
O = (miny + maxy) / 2
p = (miny + maxy) / 2
'picColor.Line (minx, O)-(maxx, P), vbGreen
picColor.Line (minx, miny)-(maxx, miny), vbRed
picColor.Line (maxx, miny)-(maxx, maxy), vbRed
picColor.Line (minx, maxy)-(maxx, maxy), vbRed
picColor.Line (minx, miny)-(minx, maxy), vbRed
picColor.Line (160, 0)-(160, 240), vbGreen
picColor.Line (0, 120)-(320, 120), vbGreen
shpLED0(0).Visible = True
shpLED0(0).BackColor = vbRed
Q = (N + m) / 2
r = (O + p) / 2
shpLED0(0).Width = 14
shpLED0(0).Height = 14
shpLED0(0).Top = O - 7
shpLED0(0).Left = N - 7
Text5.Text = shpLED0(0).Top
Text2.Text = shpLED0(0).Left
Label1.Caption = "ตำแหน่งหลอด (X,Y) = (" & (N - 8) & ", " & (O - 8) & ")
U = N - 7
V = O - 7
End Sub
Private Sub cmdcon_Click()
mCapHwnd = capCreateCaptureWindow("WebcamCapture", 0, 0, 0, 320, 240, Me, hWnd, 0)
DoEvents
SendMessage mCapHwnd, CONNECT, 0, 0
Picture3.Width = 320 * Screen.TwipsPerPixelX
Picture3.Height = 240 * Screen.TwipsPerPixelY
picColor.AutoSize = True
picColor.AutoRedraw = True
shpLED0(0).Visible = False
Timer1.Enabled = True
End Sub
Private Sub cmdDiscon_Click()
DoEvents
SendMessage mCapHwnd, DISCONNECT, 0, 0
Timer1.Enabled = False
End Sub
Private Sub Command1_Click()
Dim cc As Long
cc = 0
If MSComm1.PortOpen = False Then
MSComm1.PortOpen = True
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For cc = 1 To 1000
cc = cc + 1
Next cc
If cc > 999 Then
Timer2.Enabled = True
Timer3.Enabled = True
End If
cc = 0
End Sub

Private Sub Command2_Click()
Timer2.Enabled = False
Timer3.Enabled = False
'EÁØ'¸ÒÃ-Ó§Ò'εÍ§ÁÌµÍÃi
For co = 1 To 10
MSComm1.Output = Chr(48)
Next co
'MSComm1.PortOpen = False
End Sub

Private Sub Command3_Click()
Dim Gp, Gi, g(1 To 10), Ng As Integer
Timer2.Enabled = False
Gp = Text18
g(2) = Int(Gp / 10) + 48
Gp = Gp Mod 10
g(3) = Int(Gp) + 48
Gi = Text17
g(4) = Int(Gi / 10) + 48
Gi = Gi Mod 10
g(5) = Int(Gi) + 48
g(1) = 50
g(6) = Text22.Text + 48 * Kd
g(7) = 48
g(8) = 48
g(9) = 48
g(10) = 48
If MSComm1.PortOpen = False Then
MSComm1.PortOpen = True
End If
For Ng = 1 To 10
MSComm1.Output = Chr(g(Ng))
Next Ng
'MSComm1.PortOpen = False
End Sub

Private Sub Form_Load()
Set Fso = CreateObject("Scripting.FileSystemObject")
Set OutX = Fso.CreateTextFile("c:\ที่ตั้งหน่วยบอลแกนX.dat", True)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set OutY = Fso.CreateTextFile("c:\ตำแหน่งบอกลบแกนY.dat", True)
Set comX = Fso.CreateTextFile("c:\มุมแกน X.dat", True)
Set comY = Fso.CreateTextFile("c:\มุมแกน Y.dat", True)
Set IX = Fso.CreateTextFile("c:\InputX.dat", True)
Set IY = Fso.CreateTextFile("c:\InputY.dat", True)
Set Xdot = Fso.CreateTextFile("c:\Xdot.dat", True)
Set Ydot = Fso.CreateTextFile("c:\Ydot.dat", True)
MSComm1.Settings = "57600,N,8,1"
MSComm1.CommPort = 4
MSComm1.RThreshold = 1
MSComm1.PortOpen = True
Xi = 0
X1N_1 = 0
Yi = 0
Y1N_1 = 0
Ux = 0
Uy = 0
Ex = 0
Ey = 0
time = 0
End Sub
Private Sub Timer1_Timer()
    SendMessage mCapHwnd, GET_FRAME, 0, 0
    SendMessage mCapHwnd, COPY, 0, 0
    Picture3.Picture = Clipboard.GetData
    Clipboard.Clear
    picColor.Picture = Picture3.Image
    picColor.Visible = True
    Screen.MousePointer = vbHourglass
    DoEvents
    Find_Object picColor
    Screen.MousePointer = vbDefault
End Sub
Private Sub Timer2_Timer()
Dim data(1 To 10) As Integer
Dim setpoint_X, setpoint_Y, K1, K2, Ki, Ts As Single
K1 = Text7.Text
K2 = Text9.Text
Ki = Text12.Text
Ts = Text13.Text
If loops <= 2 Then
setpoint_X = Text1.Text * 0.0238
    X1 = Text2.Text * 0.0238
    Ex = setpoint_X - X1
    X2 = (X1 - X1N_1) / Ts
    Xi = Ex + Xi

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setpoint_Y = Text4.Text * 0.0238
  Y1 = Text5.Text * 0.0238
  Ey = setpoint_Y - Y1
  Y2 = (Y1 - Y1N_1) / Ts
  Yi = Ey + Yi
End If
X1N_1 = X1
Y1N_1 = Y1
'-----
'setpoint_X = Text1.Text
'setpoint_Y = Text4.Text
If loops > 2 Then
Text3.Text = Text1.Text - Text2.Text 'Show error X pixel
  setpoint_X = Text1.Text * 0.0238
  X1 = Text2.Text * 0.0238
  Ex = setpoint_X - X1
  X2 = (X1 - X1N_1) / Ts
  Xi = Ex + Xi
  Ux = -(K1 * X1) - (K2 * X2) + (K1 * Xi)
  Text8.Text = Ux
If (Abs(Ex)) > (3 * 0.0238) Then
  If (Ux > 0) And (X2 = 0) Then Ux = Ux + 0.5
  If (Ux < 0) And (X2 = 0) Then Ux = Ux - 0.5
  If (Ux > 2) Then '-2.01 down
    Ux = 2
    data(2) = 48 + Ux
    data(3) = 48 '(+)
    data(7) = 48
    data(8) = 48
    Text10.Text = Ux
  Elseif (Ux <= -2) Then '-2.01 down
    Ux = -2
    data(2) = 48 + Abs(Ux)
    data(3) = 49 '(-)
    data(7) = 48
    data(8) = 48
    Text10.Text = Ux
  Elseif ((Ux < 2) And (Ux > -2)) Then '-1.99 To 1.99
    If (Ux >= 0) Then
      data(3) = 48 '(+)
    Elseif (Ux < 0) Then
      data(3) = 49 '(-)
    End If

    Ux = Abs(Ux) * 100
    Ux = Int(Ux)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'Show = Ux
  If (data(3) = 48) Then
    Text10.Text = Ux / 100
  ElseIf (data(3) = 49) Then
    Text10.Text = -(Ux / 100)
  End If

  data(2) = Int(Ux / 100) + 48
  Ux = Ux Mod 100
  data(7) = Int(Ux / 10) + 48
  data(8) = Int(Ux Mod 10) + 48
End If

XIN_1 = X1
ElseIf (Abs(Ex)) <= (3 * 0.0238) Then
  data(2) = 48
  data(3) = 48 ' +,-
  data(7) = 48
  data(8) = 48
  Ux = 0
  Text10.Text = Ux
End If
Text8.Text = Ux
Text6.Text = Text4.Text - Text5.Text
setpoint_Y = Text4.Text * 0.0238
Y1 = Text5.Text * 0.0238
Ey = setpoint_Y - Y1
Y2 = (Y1 - YIN_1) / Ts
Yi = Ey + Yi
Uy = -(K1 * Y1) - (K2 * Y2) + (K3 * Yi)
Uy = -Uy
Text11.Text = Uy
If (Abs(Ey) > (3 * 0.0238)) Then
  If (Uy > 0) And (Y2 = 0) Then Uy = Uy + 1
  If (Uy < 0) And (Y2 = 0) Then Uy = Uy - 0.5
  If (Uy >= 1.5) Then ' 3.00 up
    Uy = 1.5
    data(4) = 48 + Int(Uy)
    data(5) = 48
    data(9) = 48 + 5
    data(10) = 48
    Text14.Text = Uy
  ElseIf (Uy <= -3.2) Then ' -3.00 down
    Uy = -3.2
    data(4) = 48 + Int(Abs(Uy))
    data(5) = 49
    data(9) = 48 + 2
    data(10) = 48

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text14.Text = Uy
Elseif (Uy < 1.5) And (Uy > -3.2) Then '-2.99 To 2.99
  If (Uy >= 0) Then
    data(5) = 48
  Elseif (Uy < 0) Then
    data(5) = 49
  End If
  Uy = Abs(Uy) * 100
  Uy = Int(Uy)
  'Show = Uy
  If (data(5) = 48) Then
    Text14.Text = Uy / 100
  Elseif (data(5) = 49) Then
    Text14.Text = -(Uy / 100)
  End If

  data(4) = Int(Uy / 100) + 48
  Uy = Uy Mod 100
  data(9) = Int(Uy / 10) + 48
  data(10) = Int(Uy Mod 10) + 48
End If
Y IN_1 = Y1
Elseif (Abs(Ey)) <= (3 * 0.0238) Then
  data(4) = 48
  data(5) = 49
  data(9) = 48 + 7
  data(10) = 48 + 5
  Uy = 0
  Text14.Text = Uy
End If
Text11.Text = Uy
data(1) = 49
data(6) = 48 + Text22.Text * Kd

For co = 1 To 10
  MSComm1.Output = Chr(data(co))
Next co
End If

```

```

loops = loops + 1
End Sub
'Private Sub picColor_MouseMove(Button As Integer, Shif As Integer, Xpix As Single, Ypix As Single)
'Label23.Caption = " ตำแหน่งลูกบอล(X,Y) = (" & Str(Xpix) & ", " & Str(Ypix) & ")"
'Text1.Text = Xpix
'Text4.Text = Ypix
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Timer3_Timer()
Dim datapix_X, datapix_Y, Seta_X, Seta_Y As Single
time = time + 1
Text23.Text = time
datapix_X = Text2.Text
datapix_Y = Text5.Text
Seta_X = Text10.Text
Seta_Y = Text14.Text
IX.writeline (Text1.Text)
IY.writeline (Text4.Text)
OutX.writeline (datapix_X)
OutY.writeline (datapix_Y)
comX.writeline (Seta_X)
comY.writeline (Seta_Y)
Xdot.writeline (X2)
Ydot.writeline (Y2)
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Base API Code

Option Explicit

Public Type RGBTRIPLE

 rgbRed As Byte

 rgbGreen As Byte

 rgbBlue As Byte

End Type

Public Type RGBthingy

 Value As Long

End Type

Public Type RGBpoint

 Red As Byte

 Green As Byte

 Blue As Byte

End Type

Public Const ws_child As Long = &H4000000

Public Const ws_visible As Long = &H10000000

Public Const SWP_NOSIZE As Long = &H11&

Public Const SWP_NOMOVE As Long = &H2&

Public Const SWP_NOZORDER As Long = &H4&

Public Const SWP_NOSENDCHANGING As Long = &H400& ' /* Don't send WM_WINDOWPOSCHANGING */

Public Const HWND_BOTTOM As Long = 1&

Public Const SM_CYCAPTION As Long = 4

Public Const SM_CXBORDER As Long = 5

Public Const SM_CYBORDER As Long = 6

Public Const SM_CYMENU As Long = 15

Public Const SM_CXEDGE As Long = 45

Public Const SM_CYEDGE As Long = 46

Declare Function ShellAbout Lib "shell32" Alias "ShellAboutA" _

 (ByVal hWnd As Long, _

 ByVal szApp As String, _

 ByVal szOtherStuff As String, _

 ByVal hIcon As Long) As Long

Declare Function SetWindowTextAsLong Lib "user32" Alias "SetWindowTextA" (ByVal hWnd As Long, ByVal LPCSTR As Long) As

Long 'C BOOL

Declare Function GetSystemMetrics Lib "user32" (ByVal nIndex As Long) As Long

Declare Function SetParent Lib "user32" (ByVal hWndChild As Long, ByVal hWndNewParent As Long) As Long

Declare Function SetWindowPos Lib "user32" (ByVal hWnd As Long, ByVal hWndInsertAfter As Long, ByVal x As Long, ByVal y As

Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long

Declare Function DestroyWindow Lib "user32" (ByVal hWnd As Long) As Long 'C BOOL

Declare Function GetDiskFreeSpace Lib "kernel32" Alias "GetDiskFreeSpaceA" _

 (ByVal lpRootPathName As String, _

 lpSectorsPerCluster As Long, _

 lpBytesPerSector As Long, _

 lpNumberOfFreeClusters As Long, _

 lpTotalNumberOfClusters As Long) As Long 'C BOOL

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 Global Memory Functions

```

Declare Function GlobalAlloc Lib "kernel32" (ByVal wFlags As Long, ByVal dwBytes As Long) As Long
Declare Function GlobalFree Lib "kernel32" (ByVal hMem As Long) As Long
Declare Function GlobalLock Lib "kernel32" (ByVal hMem As Long) As Long
Declare Function GlobalUnlock Lib "kernel32" (ByVal hMem As Long) As Long
Public Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (lPvDest As Any, lPvSource As Any, ByVal cbCopy As Long)
Declare Function lstrcpy Lib "kernel32" Alias "lstrcpyA" (ByVal lPStringDest As Long, ByVal lPStringSrc As Long) As Long
Declare Sub CopyPTRtoANY Lib "kernel32.dll" Alias "RtlMoveMemory" (ByRef Dest As Any, ByVal PtrSrc As Long, ByVal length As
Long)
Declare Sub CopyPTRtoLONG Lib "kernel32.dll" Alias "RtlMoveMemory" (ByRef LONGDest As Long, ByVal PtrSrc As Long, ByVal
length As Long)
Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal
nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long
Declare Function CreateBitmap Lib "gdi32" (ByVal nWidth As Long, ByVal nHeight As Long, ByVal nPlanes As Long, ByVal nBitCount
As Long, lPBits As Any) As Long
Public Declare Function SetBkColor Lib "gdi32" (ByVal hdc As Long, ByVal crColor As Long) As Long
Public Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject As Long) As Long
Public Declare Function CreateCompatibleBitmap Lib "gdi32" (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) As
Long
Public Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long
Public Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
Public Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Public Const GMEM_MOVEABLE = &H2&
Public Const GMEM_SHARE = &H2000&
Public Const GMEM_ZEROINIT = &H40&
'VFW stuff
Global Const WM_USER = 1024
Global Const WM_CAP_EDIT_COPY = WM_USER + 30
Global Const wm_cap_driver_connect = WM_USER + 10
Global Const wm_cap_set_preview = WM_USER + 50
Global Const wm_cap_set_overlay = WM_USER + 51
Global Const WM_CAP_SET_PREVIEWRATE = WM_USER + 52
Global Const WM_CAP_SEQUENCE = WM_USER + 62
Global Const WM_CAP_SINGLE_FRAME_OPEN = WM_USER + 70
Global Const WM_CAP_SINGLE_FRAME_CLOSE = WM_USER + 71
Global Const WM_CAP_SINGLE_FRAME = WM_USER + 72
Public Const WM_CAP_DLG_VIDEOFORMAT As Long = WM_USER + 41
Global Const DRV_USER = &H4000
Global Const DVM_DIALOG = DRV_USER + 100
Global Const WM_CAP_DRIVER_DISCONNECT As Long = WM_USER + 11
Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As
Long, ByVal lParam As Long) As Long
Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA" (ByVal a As String, ByVal b As Long,
ByVal c As Integer, ByVal d As Integer, ByVal e As Integer, ByVal f As Integer, ByVal g As Long, ByVal h As Integer) As Long
Declare Function SendMessageAsLong Lib "user32" Alias "SendMessageA" _
  (ByVal hWnd As Long, _

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ByVal wParam As Long, _
        ByVal lParam As Long) As Long
Global Const WM_CAP_GRAB_FRAME As Long = WM_USER + 60
Private Type PALETTEENTRY
    peRed As Byte
    peGreen As Byte
    peBlue As Byte
    peFlags As Byte
End Type
Public Type BITMAP
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type
Public Declare Function GetBitmapBits Lib "gdi32" (ByVal hBitmap As Long, _
    ByVal dwCount As Long, lpBits As Any) As Long
Public Declare Function SetBitmapBits Lib "gdi32" (ByVal hBitmap As Long, _
    ByVal dwCount As Long, lpBits As Any) As Long
Public Declare Function GetObject Lib "gdi32" Alias "GetObjectA" (ByVal hObject As Long, _
    ByVal nCount As Long, lpObject As Any) As Long
Private Declare Function GetNearestPaletteIndex Lib "gdi32" (ByVal hPalette As Long, _
    ByVal crColor As Long) As Long
Private Declare Function GetPaletteEntries Lib "gdi32" (ByVal hPalette As Long, ByVal _
    wStartIndex As Long, ByVal wNumEntries As Long, lpPaletteEntries As PALETTEENTRY) As Long
Private Declare Function RealizePalette Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function GetSystemPaletteEntries Lib "gdi32" (ByVal hdc As Long, _
    ByVal wStartIndex As Long, ByVal wNumEntries As Long, _
    lpPaletteEntries As PALETTEENTRY) As Long
Private Declare Function ResizePalette Lib "gdi32" (ByVal hPalette As Long, _
    ByVal nNumEntries As Long) As Long
Private Declare Function SetPaletteEntries Lib "gdi32" (ByVal hPalette As Long, _
    ByVal wStartIndex As Long, ByVal wNumEntries As Long, _
    lpPaletteEntries As PALETTEENTRY) As Long
Private Const Size_of_palate = 2 ^ 8
Private Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As Long, _
    ByVal nIndex As Long) As Long
Public Sub GetImagePixels(ByVal Image As PictureBox, ByRef pixel() _
    As RGBTRIPLE, ByVal Color_dept As Integer)
    Dim Temp_Picture As Long
    Dim Picture_Detail As BITMAP
    Dim Pixels_data() As Byte

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim Palette_entry(0 To Size_of_palette - 1) As PALETTEENTRY
Dim Palette_index As Integer
Dim Object_Width As Integer
Dim Object_Height As Integer
Dim x As Integer
Dim y As Integer
Dim Pixels_data_16bits As Long
Temp_Picture = Image.Image
GetObject Temp_Picture, Len(Picture_Detail), Picture_Detail
Color_dept = Picture_Detail.bmBitsPixel
' Debug.Print "BITMAP Data "
' Debug.Print "bmType "; bm.bmType
' Debug.Print "bmWidth "; bm.bmWidth
' Debug.Print "bmHeight "; bm.bmHeight
' Debug.Print "bmWidthBytes "; bm.bmWidthBytes
' Debug.Print "bmPlanes "; bm.bmPlanes
' Debug.Print "bmBitsPixel "; bm.bmBitsPixel
If (Color_dept = 8) Or
(Color_dept = 16) Or
(Color_dept = 24) Or
(Color_dept = 32)
Then
ReDim Pixels_data(0 To Picture_Detail.bmWidthBytes - 1, 0 To Picture_Detail.bmHeight - 1)
GetBitmapBits Temp_Picture, Picture_Detail.bmWidthBytes, Picture_Detail.bmHeight, Pixels_data(0, 0)
Else
MsgBox ("Error # " & CStr(Err.Number) & " " & Err.Description)
Err.Clear
End If
Object_Width = Picture_Detail.bmWidth
Object_Height = Picture_Detail.bmHeight
ReDim pixel(0 To Object_Width - 1, 0 To Object_Height - 1)
Select Case Color_dept
Case 8
For y = 0 To Object_Height - 1
For x = 0 To Object_Width - 1
Palette_index = Pixels_data(x, y)
pixel(x, y).rgbtRed = Palette_entry(Palette_index).peRed
pixel(x, y).rgbtGreen = Palette_entry(Palette_index).peGreen
pixel(x, y).rgbtBlue = Palette_entry(Palette_index).peBlue
Next x
Next y
Case 16
For y = 0 To Object_Height - 1
For x = 0 To Object_Width - 1
Pixels_data_16bits = Pixels_data(x * 2, y) + Pixels_data(x * 2 + 1, y) * 256&
pixel(x, y).rgbtBlue = Pixels_data_16bits Mod 32

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Pixels_data_16bits = Pixels_data_16bits \ 32
        pixel(x, y).rgbGreen = Pixels_data_16bits Mod 64
        Pixels_data_16bits = Pixels_data_16bits \ 64
        pixel(x, y).rgbRed = Pixels_data_16bits
    Next x
Next y
Case 24
For y = 0 To Object_Height - 1
    CopyMemory pixel(0, y), Pixels_data(0, y), Object_Width * 3
Next y
Case 32
For y = 0 To Object_Height - 1
    For x = 0 To Object_Width - 1
        pixel(x, y).rgbBlue = Pixels_data(x * 4, y)
        pixel(x, y).rgbGreen = Pixels_data(x * 4 + 1, y)
        pixel(x, y).rgbRed = Pixels_data(x * 4 + 2, y)
    Next x
Next y
End Select
End Sub
Public Sub SetImagePixels(ByVal Image As PictureBox, ByVal Color_dept As Integer,
pixel() As RGBTRIPLE)
Dim wid_bytes As Long
Dim Object_Width As Integer
Dim Object_Height As Integer
Dim x As Integer
Dim y As Integer
Dim Pixels_data() As Byte
Dim hpal As Long
Dim Pixels_data_16bits As Long
Object_Width = UBound(pixels, 1) + 1
Object_Height = UBound(pixels, 2) + 1
Select Case Color_dept
    Case 8
        wid_bytes = Object_Width
    Case 16
        wid_bytes = Object_Width * 2
    Case 24
        wid_bytes = Object_Width * 3
    Case 32
        wid_bytes = Object_Width * 4
    Case Else
        MsgBox ("Error # " & CStr(Err.Number) & " " & Err.Description)
        Err.Clear
    End Select

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If wid_bytes Mod 2 = 1 Then wid_bytes = wid_bytes + 1
ReDim Pixels_data(0 To wid_bytes - 1, 0 To Object_Height - 1)
Select Case Color_dept
Case 8
    hpal = Image.Picture.hpal
    For y = 0 To Object_Height - 1
        For x = 0 To Object_Width - 1
            Pixels_data(x, y) = (&HFF And _
                GetNearestPaletteIndex(hpal, RGB(pixel(x, y).rgbRed, _
                    pixel(x, y).rgbGreen, pixel(x, y).rgbBlue) + &H2000000))
        Next x
    Next y
Case 16
    For y = 0 To Object_Height - 1
        For x = 0 To Object_Width - 1
            If pixel(x, y).rgbRed > &H1F Then pixel(x, y).rgbRed = &H1F
            If pixel(x, y).rgbGreen > &H3F Then pixel(x, y).rgbGreen = &H3F
            If pixel(x, y).rgbBlue > &H1F Then pixel(x, y).rgbBlue = &H1F
            Pixels_data_16bits = pixel(x, y).rgbBlue + 32 * (pixel(x, y).
                .rgbGreen + CLng(pixel(x, y).rgbRed) * 64)
            Pixels_data(x * 2, y) = (Pixels_data_16bits Mod 256) And &HFF
            Pixels_data(x * 2 + 1, y) = (Pixels_data_16bits \ 256) And &HFF
        Next x
    Next y
Case 24
    For y = 0 To Object_Height - 1
        CopyMemory Pixels_data(0, y), pixel(0, y), Object_Width * 3
    Next y
Case 32
    For y = 0 To Object_Height - 1
        For x = 0 To Object_Width - 1
            Pixels_data(x * 4, y) = pixel(x, y).rgbBlue
            Pixels_data(x * 4 + 1, y) = pixel(x, y).rgbGreen
            Pixels_data(x * 4 + 2, y) = pixel(x, y).rgbRed
        Next x
    Next y
End Select
SetBitmapBits Image.Image, wid_bytes * Object_Height, _
    Pixels_data(0, 0)
Image.Refresh
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข-1 เอกสารคู่มือการใช้งาน PIC 16F877A



MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873
- PIC16F875
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/75/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT), with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram

PDIP



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

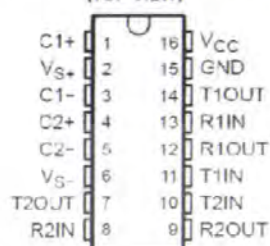
ข-2 เอกสารคู่มือการใช้งาน IC MAX 232

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLL6047L – FEBRUARY 1985 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202
- Applications
 - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC library.

ORDERING INFORMATION

TA	PACKAGE	ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N
	SOIC (D)	Tube of 40	MAX232D
		Reel of 2500	MAX232DR
	SOIC (DW)	Tube of 40	MAX232DW
Reel of 2000		MAX232DWR	
-40°C to 85°C	SOP (NS)	Reel of 2000	MAX232NSR
	PDIP (N)	Tube of 25	MAX232IN
		Tube of 40	MAX232ID
		Reel of 2500	MAX232IDR
SOIC (DW)	Tube of 40	MAX232IDW	
	Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2004, Texas Instruments Incorporated

1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข-3 เอกสารคู่มือการใช้งาน L293D

SGS-THOMSON
MICROELECTRONICS

L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

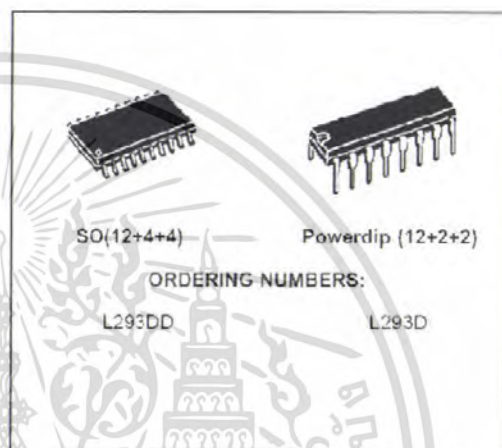
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

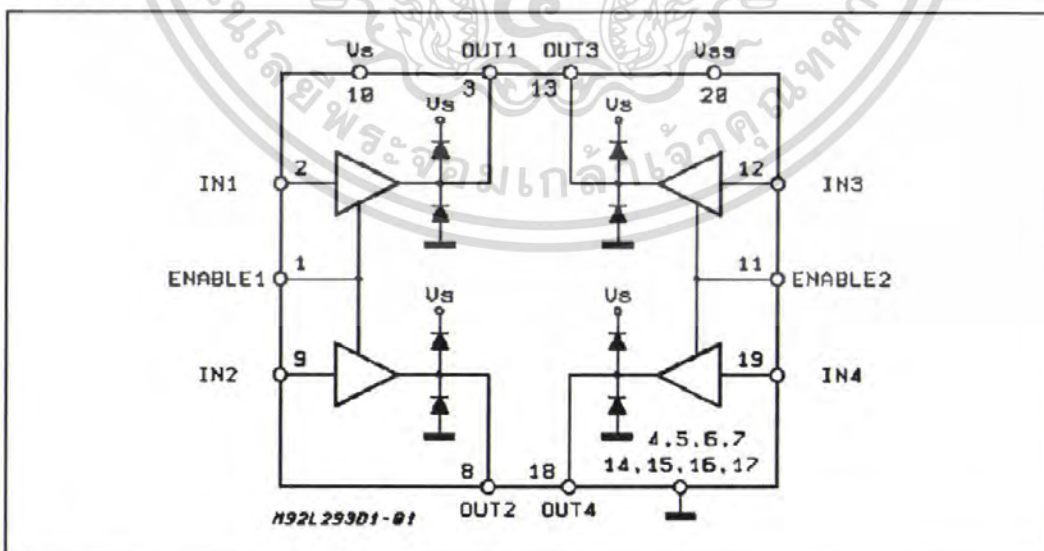
This device is suitable for use in switching applications at frequencies up to 5 kHz.



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking.

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM



June 1993

1/7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้