

ระบบควบคุมตำแหน่งลูกบอลบนคาน
BALL BEAM CONTROL SYSTEM



เลขานุ.....
เลขทะเบียน.....**62769**
วัน,เดือน,ปี...**22** ส.ค. **2549**

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมตำแหน่งลูกบอลบนลาน
BALL BEAM CONTROL SYSTEM

ผู้จัดทำ นายเป็นเอก เฟื่องวัน 46015314
นายสุริยนต์ โชติะศรี 46015329


.....อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ อธิการบดี บุญจรรยาสุทธิ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมตำแหน่งลูกบอลบนคาน

โดย

นายเป็นเอก เฟื่องวัน

นายสุริยันต์ โชตะศรี

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ถาวร เเบญจนราสุทธิ์

ปีการศึกษา 2548

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอทฤษฎีและการออกแบบระบบควบคุมอัตโนมัติสำหรับระบบควบคุมตำแหน่งลูกบอลบนคาน โดยโครงสร้างของระบบประกอบด้วย คอมพิวเตอร์ กล้องดิจิตอล วิดีโอ และวงจรรีเลย์ทรอนิกส์ที่เกี่ยวข้อง จุดมุ่งหมายของโครงการนี้คือบังคับลูกบอลให้ไปยังตำแหน่งที่ต้องการตามที่กำหนดบนคานและรักษาตำแหน่งของบอลนั้น โดยการควบคุมมอเตอร์ไฟฟ้ากระแสตรงที่ต่อกับคานผ่านทาง पुलเล่ และสายพาน

ขั้นตอนดำเนินการ เริ่มจากการออกแบบและประกอบโครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนคาน ศึกษาและออกแบบวงจรรีเลย์ทรอนิกส์ที่จำเป็นสำหรับระบบควบคุมตำแหน่งลูกบอลบนคาน ประกอบด้วยวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก วงจรซีโรสเปน และวงจรขับมอเตอร์ แล้วจึงเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาวิซวลซีพลัสพลัส เพื่อตรวจจับตำแหน่งของลูกบอลจากข้อมูลภาพจากกล้องดิจิตอลวิดีโอ ซึ่งส่งมายังคอมพิวเตอร์ผ่านทางยูนิเวอร์แซลซีเรียลบัส(USB) ประมวลผลหาตำแหน่งควบคุมที่เหมาะสม โดยอาศัยแนวคิดระบบควบคุมพีไอดีแบบเวลาดีสครีต และส่งสัญญาณควบคุมผ่านทางพอร์ตขนานสำหรับควบคุมมอเตอร์ไฟฟ้ากระแสตรงเพื่อนำไปขับเคลื่อน โดยค่าอัตราขยายของตัวควบคุมพีไอนั้นได้ทำการปรับด้วยวิธีลองผิดลองถูกจนได้ค่าที่เหมาะสม และท้ายสุดทำการทดลอง จากการทดลองพบว่าระบบควบคุมที่ได้ออกแบบนั้น สามารถควบคุมระบบให้ลูกบอลไปอยู่ ณ ตำแหน่งที่ต้องการได้ โดยมีค่าความผิดพลาดอยู่ในระดับที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BALL BEAM CONTROL SYSTEM

By

Mr. Pen-ek Pengwan

Mr. Suriyan Chowtasri

Advisor

Asst. Prof. Taworn Benjanarasuth

Academic Year 2005

ABSTRACT

This thesis presents theory and implementation procedures of ball and beam control system using a digital video camera as a visual feedback sensor. The system composes of the ball and beam structure, computer, digital video camera and interfacing circuits. The goal is to move the ball to any desired position or set point on the beam and maintain the ball that position by controlling the DC motor connected to the beam by pulleys and belt.

The project has been conducted as in the following steps. First, the structure of ball and beam control system is designed and constructed. Second, necessary electronic circuits including D/A converter, signal conditioning circuits and motor drivers are studied and implemented. Then, the computer program written in visual C++ language is composed. Its tasks are to detect the ball position from digital video camera data sending to the computer via universal serial bus (USB), to compute suitable control signal based on the discrete time PID control system and to send such control signal via parallel port for controlling the DC motor to drive the beam. The gains of PID controller are adjusted by trial and error techniques. Lastly, the experiments are conducted. The results show that the ball position can be controlled to the set point with negligible error.

กิตติกรรมประกาศ

การจัดทำปฏิญานិพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก ผศ.ถาวร เบญจนราษฎร์ ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ รศ.ดร.จنگกล งามวิวิทย์ ที่คอยถามถึงความคืบหน้าอยู่ตลอดเวลา และ อาจารย์รัชชชัย คำศรี ที่ให้คำปรึกษาในส่วนของการเขียนโปรแกรม

ขอขอบคุณเพื่อนๆ ทุกคนที่ทำให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงการอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของการจัดทำโครงการ	2
1.2 ขั้นตอนการศึกษาและการจัดทำโครงการ	2
1.3 เนื้อหาในปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	4
2.1 ทฤษฎีการประมวลผลภาพ	4
2.1.1 การกำหนดสี	4
2.1.2 องค์ประกอบของสี	5
2.1.3 ภาพดิจิทัล	6
2.1.3.1 การแทนภาพด้วยข้อมูลดิจิทัล	6
2.1.3.2 ลักษณะการเก็บข้อมูลภาพดิจิทัล	7
2.1.4 การประมวลผลภาพเชิงตัวเลข	7
2.1.5 สัญญาณข้อมูลภาพจากดิจิทัลวิดีโอ	8
2.1.6 ข้อมูลภาพชนิดบิตแมป	8
2.1.7 การสร้างภาพไบนารี	9
2.1.7.1 การหาค่าเทรชโฮลโดยการกำหนดไว้ล่วงหน้า	9
2.1.7.2 การสร้างภาพไบนารีโดยใช้โปรแกรมTRIPOD	10
2.2 มอเตอร์ไฟฟ้ากระแสตรง	14
2.2.1 แรงบิดและงาน	15
2.2.2 แรงบิดที่เกิดขึ้นในอาร์เมเจอร์	15
2.2.3 แรงดันไฟฟ้าต้านกลับ	16
2.2.4 การแบ่งชนิดของมอเตอร์กระแสตรงตามลักษณะการกระตุ้น	17
2.2.5 คุณสมบัติของมอเตอร์	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.1	มอเตอร์แบบขนาน	19
2.2.5.2	มอเตอร์แบบอนุกรม	20
2.2.6	การเปรียบเทียบคุณลักษณะของมอเตอร์	20
2.2.6.1	การเปรียบเทียบของมอเตอร์แบบขนานกับแบบอนุกรม	20
2.2.6.2	คุณลักษณะพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรง	21
2.3	วงจรและภาคขั้วมอเตอร์	22
2.3.1	วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก แบบใช้ความต้านทานสองค่า	23
2.3.2	วงจรปรับซีโรสเปน	26
2.3.3	วงจรขั้วมอเตอร์	26
2.4	พอร์ตขนาน	27
2.4.1	ลักษณะทางกายภาพของพอร์ตขนาน	28
2.4.2	พอร์ตข้อมูล	31
2.4.3	พอร์ตควบคุม	32
2.4.4	พอร์ตสถานะ	33
2.5	ตัวควบคุม	34
2.5.1	ระบบควบคุมแบบพี	34
2.5.2	ระบบควบคุมแบบไอ	37
2.5.3	ระบบควบคุมแบบดี	38
2.5.4	ระบบควบคุมแบบพีไอ	40
2.5.5	ระบบควบคุมแบบพีดี	42
2.5.6	ระบบควบคุมแบบพีไอดี	43
2.5.7	ตัวควบคุมแบบดิจิทัล	44
บทที่ 3 หลักการออกแบบ		45
3.1	กล่องดิจิทัลวีดีโอ	46
3.1.1	สิ่งจำเป็นสำหรับการใช้งานกล่องดิจิทัลวีดีโอ	46
3.1.2	คุณสมบัติของกล่องดิจิทัลวีดีโอ	46
3.2	คอมพิวเตอร์	47
3.3	โครงสร้างทางกายภาพของระบบ	48
3.3.1	การยึดกล่องดิจิทัลวีดีโอ	48
3.3.2	การออกแบบโครงสร้าง	49
3.4	การคำนวณและการออกแบบวงจรและภาคขั้วมอเตอร์	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การออกแบบวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก	51
3.4.2 การออกแบบวงจรปรับซีโรสเปก	52
บทที่ 4 ผลการทดลอง	54
4.1 การทดลองวงจรและภาคขับมอเตอร์	54
4.1.1 วงจรและภาคขับมอเตอร์ที่ไม่มีวงจรตรรก ออเกต	54
4.1.2 วงจรและภาคขับมอเตอร์ที่มีวงจรตรรก ออเกต	56
4.2 การระบุตำแหน่งของลูกบอลโดยการเทียบพิกเซล	59
4.3 การทดสอบการควบคุมตำแหน่งลูกบอลไปยังตำแหน่งต่างๆ ที่กำหนดบนถาด	60
4.3.1 การควบคุมโดยใช้ตัวควบคุมแบบพีดี	60
4.3.2 การเขียนโปรแกรมโดยกำหนดช่วงค่าผิดพลาดที่ยอมรับได้	62
4.3.3 การเขียนโปรแกรมโดยกำหนดช่วงค่าผิดพลาด และการชดเชยตำแหน่งบอลล่วงหน้า	63
บทที่ 5 บทวิจารณ์และสรุป	72
5.1 สรุปผลการทดลอง	72
5.2 ปัญหาที่พบและแนวทางแก้ไข	72
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	73
ภาคผนวก ก โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม	74
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	90
ข-1 เอกสารคู่มือการใช้งาน DAC0832	90
ข-2 เอกสารคู่มือการใช้งาน LM741	94
ข-3 เอกสารคู่มือการใช้งาน TIP31C และ TIP32C	96
ข-4 เอกสารคู่มือการใช้งาน DM74LS32	97
เอกสารอ้างอิง	99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
1.1 โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนคาน	1
2.1 กล้องสี RGB	5
2.2 การเก็บข้อมูลของแต่ละพิกเซล	8
2.3 การสร้างภาพไบนารีของโปรแกรม TRIPOD	10
2.4 โค้ดโปรแกรมการทำภาพไบนารี	11
2.5 การนำเสนอรูปภาพในรูปแบบเมตริกซ์ และเวกเตอร์	13
2.6 การเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกลของมอเตอร์ไฟฟ้าเพื่อใช้งาน	14
2.7 ความเป็นเครื่องกำเนิดในมอเตอร์	16
2.8 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบแยกขั้วกระตุ้น	17
2.9 วงจรมอเตอร์ไฟฟ้าแบบขนาน	18
2.10 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม	18
2.11 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบผสม	18
2.12 คุณลักษณะของมอเตอร์แบบขนาน	19
2.13 คุณลักษณะของมอเตอร์แบบอนุกรม	20
2.14 วงจรเทียบเคียงของมอเตอร์แบบกระตุ้นภายนอก	22
2.15 วงจรและภาคขับมอเตอร์	23
2.16 วงจรเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแบบ R/2R แลคเคอร์	25
2.17 วงจรซีโรสเปน	26
2.18 วงจรขับมอเตอร์	27
2.19 ไคอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์	28
2.20 ระบบบัสภายในของพอร์ตขนาน	31
2.21 วงจรภายในของพอร์ตข้อมูล	32
2.22 วงจรภายในของพอร์ตควบคุม	33
2.23 วงจรภายในของพอร์ตแสดงสถานะ	34
2.24 การกำหนดช่วงจำกัดของเอาต์พุต	35
2.25 ผลการตอบสนองของตัวควบคุมแบบพี	36
2.26 บล็อกไคอะแกรมของระบบควบคุมแบบพี	37
2.27 ผลการตอบสนองของตัวควบคุมแบบ PI	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.28	บล็อกไดอะแกรมของระบบควบคุมแบบโอ	38
2.29	ผลตอบสนองของระบบควบคุมแบบดี	39
2.30	บล็อกไดอะแกรมของระบบควบคุมแบบดี	40
2.31	บล็อกไดอะแกรมของระบบควบคุมแบบพีโอ	40
2.32	การตอบสนองของตัวควบคุมแบบพีโอ	41
2.33	บล็อกไดอะแกรมของระบบควบคุมแบบพีดี	42
2.34	บล็อกไดอะแกรมของระบบควบคุมแบบพีไอดี	43
3.1	แผนผังรวมของระบบควบคุมตำแหน่งลูกบอลบนคาน	45
3.2	กล้องดิจิทัลวิดีโอ	46
3.3	โพลีชาร์ตของโปรแกรมควบคุมระบบ	47
3.4	การเชื่อมกล้องดิจิทัลวิดีโอ	48
3.5	โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนคาน	49
3.6	การเชื่อมต่อ	49
3.7	ระบบควบคุมตำแหน่งลูกบอลบนคาน	50
3.8	การทำงานของวงจรและภาคขับเคลื่อน	51
3.9	วงจรปรับแรงดันมาตรฐานโดยใช้ออปเทค	52
4.1	ผลการทดลองวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกที่ไม่ต่อวงจรตรรก ออเกต	55
4.2	ผลการทดลองของวงจรปรับซีโรสเปกที่ไม่ต่อวงจรตรรก ออเกต	55
4.3	ผลการทดลองของวงจรขับเคลื่อนมอเตอร์ที่ไม่ต่อวงจรตรรก ออเกต	56
4.4	ผลการทดลองวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกที่ต่อวงจรตรรก ออเกต	57
4.5	ผลการทดลองของวงจรปรับซีโรสเปกที่ต่อวงจรตรรก ออเกต	58
4.6	ผลการทดลองของวงจรขับเคลื่อนมอเตอร์ที่ต่อวงจรตรรก ออเกต	58
4.7	การหาตำแหน่งของลูกบอลที่อยู่บนคานเทียบกับตำแหน่งฟิกเชล	60
4.8	การควบคุมลูกบอลจากตำแหน่ง 32 cm ไปยังตำแหน่ง 40 cm	61
4.9	สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 32 cm ไปยังตำแหน่ง 40 cm	61
4.10	การควบคุมลูกบอลจากตำแหน่ง 45 cm ไปยังตำแหน่ง 40 cm	62
4.11	สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 45 cm ไปยังตำแหน่ง 40 cm	63
4.12	การควบคุมลูกบอลจากตำแหน่ง 1 cm ไปยังตำแหน่ง 10 cm	64
4.13	สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 1 cm ไปยังตำแหน่ง 10 cm	64
4.14	การควบคุมลูกบอลจากตำแหน่ง 8 cm ไปยังตำแหน่ง 20 cm	65
4.15	สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 8 cm ไปยังตำแหน่ง 20 cm	66
4.16	การควบคุมลูกบอลจากตำแหน่ง 19 cm ไปยังตำแหน่ง 30 cm	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.17 สัญญาอินพุดสำหรับการควบคุมลูกบอลจากตำแหน่ง 19cm ไปยังตำแหน่ง 30 cm	67
4.18 การควบคุมลูกบอลจากตำแหน่ง 53 cm ไปยังตำแหน่ง 40 cm	68
4.19 สัญญาอินพุดสำหรับการควบคุมลูกบอลจากตำแหน่ง 53 cm ไปยังตำแหน่ง 40 cm	68
4.20 การควบคุมลูกบอลจากตำแหน่ง 64 cm ไปยังตำแหน่ง 50 cm	69
4.21 สัญญาอินพุดสำหรับการควบคุมลูกบอลจากตำแหน่ง 64 cm ไปยังตำแหน่ง 50 cm	70
4.22 การควบคุมลูกบอลจากตำแหน่ง 75 cm ไปยังตำแหน่ง 65 cm	70
4.23 สัญญาอินพุดสำหรับการควบคุมลูกบอลจากตำแหน่ง 75 cm ไปยังตำแหน่ง 65 cm	71
ก.1 หน้าต่างเชื่อมต่อกับผู้ใช้ในโปรแกรมการตรวจสอบตำแหน่งลูกบอลและควบคุม	74



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 สัญญาณสำคัญของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์	29
2.2 สัญญาณทั้งหมดของพอร์ตขนาน	29
4.1 ผลการทดลองวงจรและภาคขับมอเตอร์ที่ไม่มีวงจรตรรก ออเกต	54
4.2 ผลการทดลองวงจรและภาคขับมอเตอร์ที่มีการต่อวงจรตรรก ออเกต	57



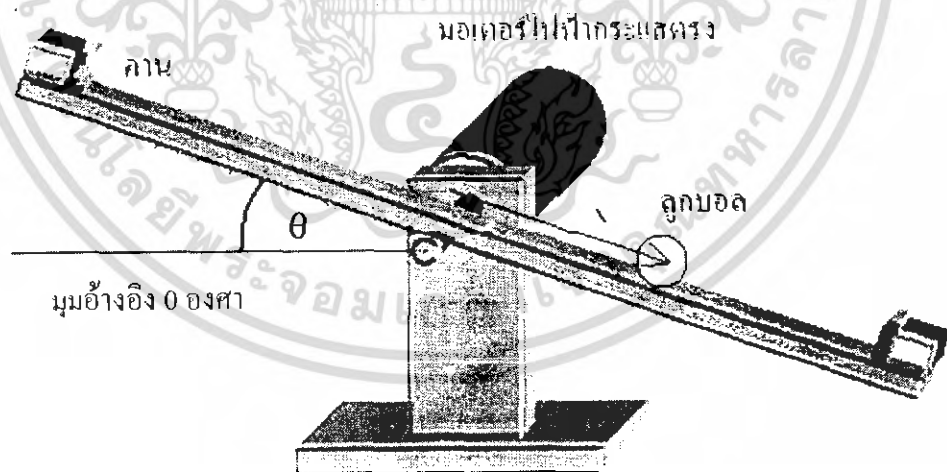
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

การศึกษาในสาขาวิชาวิศวกรรมระบบควบคุม เป็นการศึกษาและประยุกต์ทฤษฎีต่างๆ เพื่อการออกแบบและควบคุมระบบให้มีเสถียรภาพ และมีสมรรถนะตามความต้องการหรือให้เป็นไปตามข้อกำหนด ดังนั้นเพื่อให้เป็นไปตามวัตถุประสงค์ที่ต้องการนี้จึงจำเป็นต้องมีการศึกษาการจำลองระบบควบคุมด้วยสมการทางคณิตศาสตร์ การออกแบบตัวควบคุมหรือตัวชดเชยแบบต่างๆ การศึกษาวงจรรีเลย์ทรอนิกส์และอิเล็กทรอนิกส์กำลังรวมถึงศึกษาการเขียนโปรแกรมคอมพิวเตอร์ การศึกษาและการเลือกอุปกรณ์วัดและแปลงสัญญาณ ตลอดจนการบูรณาการเรื่องที่ศึกษาเหล่านี้ในการประยุกต์ใช้กับระบบควบคุมทางกายภาพจริงซึ่งนับเป็นสิ่งที่จำเป็นในการศึกษาในสาขาวิชานี้

ในโครงการนี้จะเลือกที่จะศึกษาระบบควบคุมตำแหน่งลูกบอลบนคาน ซึ่งเป็นตัวอย่างหนึ่งของระบบควบคุมที่นิยมและน่าสนใจต่อการศึกษา เพราะเป็นระบบวงเปิดที่ไม่มีเสถียรภาพ จึงจำเป็นที่จะต้องมีการควบคุมเพื่อให้เกิดเสถียรภาพ โดยเลือกใช้การควบคุมแบบป้อนกลับอัตโนมัติ ซึ่งมีหลายแบบ เช่น การป้อนกลับสถานะ(State Feedback) การควบคุมแบบพีไอ(PI) พีดี(PD) พีไอดี(PID) และแบบอื่นๆ เป้าหมายของการควบคุมคือการทำให้ลูกบอลหยุดนิ่ง ณ ตำแหน่งใด ตำแหน่งหนึ่งบนคานได้ตามต้องการ



รูปที่ 1.1 โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนคาน

รายละเอียดของบทนี้จะกล่าวถึงวัตถุประสงค์ของการจัดทำโครงการ และรวมถึงขั้นตอน การศึกษาและการจัดทำโครงการโดยภาพรวม ซึ่งรายละเอียดต่างๆ นั้นจะได้กล่าวถึงในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 วัตถุประสงค์ของการจัดทำโครงการ

1. ทำการศึกษาและทดลองการใช้งานอุปกรณ์ต่างๆในระบบควบคุม
2. ทำการศึกษาและสร้างระบบควบคุมตำแหน่งลูกบอลบนคาน ซึ่งประกอบด้วยอุปกรณ์ที่ประกอบเป็นโครงสร้างทางกล อุปกรณ์ทางอิเล็กทรอนิกส์ของระบบควบคุม อุปกรณ์วัดตำแหน่งของลูกบอลซึ่งในโครงการนี้ใช้กล้องเวปแคม(Webcam) รวมถึงส่วนของการควบคุมและประมวลผล โดยโครงการนี้ใช้คอมพิวเตอร์และโปรแกรมวิชวลซีพลัสพลัส (Visual C++)
3. ทำการศึกษาทฤษฎีของการควบคุมเสถียรภาพ ในระบบควบคุมเวลาติสครีต และทดลองการใช้งานตัวชดเชยแบบต่างๆ ในระบบควบคุมจริง

1.2 ขั้นตอนการศึกษาและการจัดทำโครงการ

การศึกษาระบบควบคุมตำแหน่งลูกบอลบนคานนั้น จำเป็นต้องเรียนรู้และเข้าใจในตัวระบบก่อน หลังจากที่ได้ทำการศึกษาแล้วพบว่าระบบควบคุมตำแหน่งลูกบอลบนคานที่เคยพบมานั้นจะมีการตรวจสอบตำแหน่งของลูกบอล โดยใช้ลูกบอลที่เป็นโลหะตัวนำกลิ้งบนคานที่มีเส้นลวดความต้านทานอยู่ โดยการเคลื่อนที่ของลูกบอลจะเปลี่ยนค่าความต้านทานรวมซึ่งแปรผันกับความยาวของลวดความต้านทาน ณ ตำแหน่งนั้นๆ ซึ่งสามารถเปลี่ยนเป็นสัญญาณทางไฟฟ้าที่เหมาะสมต่อการควบคุม จากหลักการนี้ก็จะสามารถระบุตำแหน่งของลูกบอลที่อยู่บนคานได้ แต่ปัญหาที่พบในการใช้ลวดความต้านทานในการระบุตำแหน่งลูกบอลบนคานนั้นก็คือ ค่าความต้านทานหากลูกบอลกลิ้งเร็วเกินไปหรือลูกบอลกระโดด อันเนื่องมาจากการเปลี่ยนแปลงมุมเอียงของคานอาจจะทำให้ค่าความต้านทานที่ได้มีการความผิดพลาดได้ ส่งผลทำให้การควบคุมระบบนั้นมีความผิดพลาดด้วยเช่นกัน

จากที่ได้กล่าวมาแล้วข้างต้นหากจะอาศัยหลักการของการเปลี่ยนแปลงค่าความต้านทานของเส้นลวดความต้านทานนั้นยังมีความผิดพลาดอยู่ นอกจากนั้นเส้นลวดความต้านทานที่มีคุณภาพดีและเหมาะสมในการใช้งานนั้นมีราคาสูง ดังนั้นโครงการนี้จึงเลือกใช้การประมวลผลภาพ(Image Processing) เข้ามาช่วยในการตรวจสอบตำแหน่งลูกบอล โดยต้องคำนึงถึงปัญหาที่อาจเกิดขึ้นคือโปรแกรมที่ใช้ในการประมวลผลภาพนั้นยังมีข้อจำกัดของความเร็วที่ใช้ในการจับภาพ ซึ่งจากที่ผ่านมามีพบว่าเคยมีการใช้หลักการประมวลผลภาพมาช่วยในการตรวจจับตำแหน่งของลูก โดยใช้โปรแกรมวิชวลเบสิก(Visual Basic) แต่ปัญหาที่พบคือความเร็วในการประมวลผลภาพยังมีความล่าช้าทำให้ไม่สามารถควบคุมระบบควบคุมได้ ในโครงการนี้จึงเลือกใช้โปรแกรมวิชวลซีพลัสพลัส ซึ่งใช้เวลาในการประมวลผลภาพได้เร็วกว่าใช้โปรแกรมวิชวลเบสิก ผลที่ได้คือมีความเร็ว

ในการประมวลผลภาพสูงขึ้นมาก ถึงแม้ยังมีการกระตุกของภาพบ้างแต่อยู่ในระดับที่ยอมรับได้และสามารถนำมาใช้งานได้

หลังจากที่สามารถตรวจจับตำแหน่งของลูกบอลบนคานได้แล้ว จึงทำการออกแบบระบบควบคุมตำแหน่งลูกบอลบนคาน โดยศึกษาการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์ภายนอก รวมถึงวงจรที่ใช้ในการควบคุมซึ่งรายละเอียดจะกล่าวในบทต่อไป

1.3 เนื้อหาในปริิณญาณินิพนธ์

เนื้อหาที่จะกล่าวในปริิณญาณินิพนธ์ฉบับนี้ประกอบด้วย บทที่ 2 ซึ่งจะกล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในการประมวลผลภาพ การใช้งานพอร์ตขนาน วงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง และนำเอาความรู้ไปประยุกต์ใช้ในการจัดทำโครงการ บทที่ 3 นำเสนอการประกอบโครงสร้างของระบบ รวมถึงแนวคิดในการออกแบบระบบควบคุม บทที่ 4 เป็นส่วนการทดสอบองค์ประกอบต่างๆ ในระบบ ตลอดจนการทดลองระบบควบคุมตำแหน่งลูกบอลบนคาน หายี่สุดในบทที่ 5 จะสรุปผลการดำเนินงานปัญหาและแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

จากที่ได้กล่าวไว้ในบทที่ 1 แล้วว่า ก่อนที่จะมีการออกแบบและสร้างระบบควบคุมตำแหน่งลูกบอลบนคานนั้น จำเป็นต้องศึกษาองค์ประกอบต่างๆที่จำเป็นของระบบควบคุมที่สนใจให้เข้าใจเสียก่อน พบว่าระบบควบคุมตำแหน่งลูกบอลบนคานนั้นมีส่วนที่สำคัญหลายส่วน ดังนั้นในบทนี้จะศึกษาและอธิบายถึงองค์ประกอบต่างๆ ที่จะนำไปใช้งานจริงในระบบควบคุมตำแหน่งลูกบอลบนคาน ซึ่งประกอบด้วย การประมวลผลภาพ มอเตอร์ไฟฟ้ากระแสตรง วงจรที่ใช้กับระบบควบคุม รวมถึงรายละเอียดของพอร์ตขนาน และตัวควบคุมแบบต่างๆ

2.1 ทฤษฎีการประมวลผลภาพ

ความรู้เกี่ยวกับการประมวลผลภาพ นับเป็นส่วนสำคัญที่จำเป็นจะต้องศึกษาเพื่อให้สามารถระบุตำแหน่งของลูกบอลที่อยู่บนคานได้ โดยต้องทำความเข้าใจเกี่ยวกับหลักการของการประมวลผลภาพเบื้องต้น ซึ่งประกอบไปด้วยหัวข้อต่างๆ ดังนี้

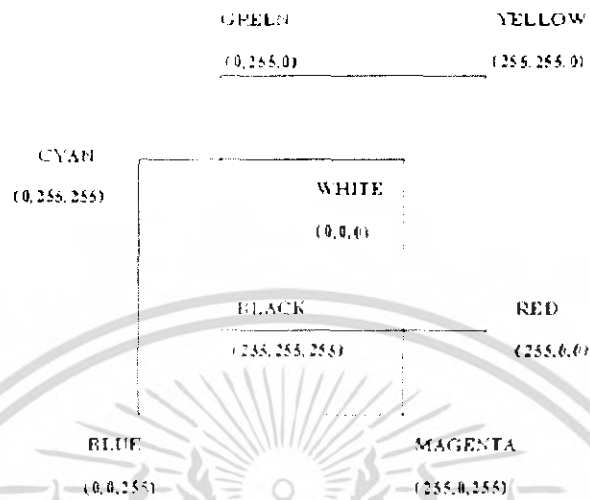
2.1.1 การกำหนดสี

ในคอมพิวเตอร์ สีทุกสีจะประกอบด้วย 3 สีพื้นฐาน คือ สีแดง สีเขียว และสีน้ำเงิน(RGB) ซึ่งสามารถที่จะสังเคราะห์สีอื่นๆ ได้เกือบทุกสี โดยการผสมสีพื้นฐานทั้งสามนี้ในสัดส่วนต่างๆกัน การผสมสีตั้งอยู่บนพื้นฐานของความเข้มข้นขององค์ประกอบของ RGB ซึ่งเรียกว่า RGB model

โดยแต่ละสีถูกนำเสนอ โดยสัดส่วน(Red,Green,Blue) ซึ่งปริมาณของสีพื้นฐานทั้งสาม ในแต่ละสีนิยมนำเสนอด้วยค่าขนาด 1 ไบต์หรือเท่ากับ 8 บิต โดยค่าที่เล็กที่สุดคือค่า 0 สอดคล้องกับการไม่มีสีพื้นฐานสีนั้นเลย และค่าที่ใหญ่ที่สุดคือค่า 255 บ่งบอกถึงความเข้มข้นสูงสุดของสีพื้นฐานสีนั้น อาทิเช่น สัดส่วน (0,0,0) สอดคล้องกับสีดำ เพราะสีพื้นฐานทั้งสามสีไม่มีความเข้มข้นเลย ในขณะที่สัดส่วน (255,255,255) สอดคล้องกับสีขาว ส่วนสีอื่นๆ ซึ่งเกิดจากการรวมกันนั้นก็ได้มากมาย เช่น (255,0,0) จะเป็นสีแดงบริสุทธิ์ (0,255,255) เป็นสีน้ำเงินเข้ม และ (0,128,128) เป็นสีน้ำเงินอ่อนจาง ดังนั้นการรวมกันที่เป็นไปได้ของสีพื้นฐานในกรณีนี้จะมีขนาด 24 บิต หรือมีค่าได้ $256 \times 256 \times 256$ ซึ่งเท่ากับ 16,777,216 ระดับสี จึงนิยมจัดเก็บค่าสีในรูปแบบตัวแปร Long Integer

กระบวนการของการสร้างสีซึ่งมีสีพื้นฐาน 3 สีนี้ ตั้งอยู่บนพื้นฐานของกล่องสี RGB ดังแสดงในรูปที่ 2.1 โดยมุมของกล่องสีนี้จะสอดคล้องกับสีต่างๆดังรูป ปริมาณของสีพื้นฐานทั้งสามสีในสีใดๆจะมีค่าตามแกน x y และ z ตามลำดับ สีที่เป็นส่วนกลับ(Complementary Colors) สามารถคำนวณได้อย่างง่ายดาย โดยการลบค่าของสีจาก 255 ตัวอย่างเช่นสี (0,0,255) เป็นสีน้ำเงินบริสุทธิ์

เป็นส่วนกลับของสีเหลืองบริสุทธิ์ที่สอดคล้องกับค่า (255,255,0) ซึ่งทั้งสองสีดังกล่าวอยู่ที่มุมตรงข้ามกันของกล่องสี ส่วนมุมอื่นๆ ก็เช่นเดียวกัน



รูปที่ 2.1 กล่องสี RGB

องค์ประกอบของสี ที่มุมของกล่องสีจะมีความเข้มข้นสูงที่สุด หรือไม่มีความเข้มข้นเลยเท่านั้น 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 ในตัวแปร Long Integer ดังกล่าวจะประกอบด้วย 4 ไบต์ โดยไบต์แรก(The Most Significant Byte) เก็บ 0 ต่อมาเป็นองค์ประกอบของสีน้ำเงิน เขียว และแดงตามลำดับ ค่า Long Integer ซึ่งสอดคล้องกับสัดส่วน RGB (64,32,192) มีค่าเป็น 12,591,168 ซึ่งค่านี้ดูไม่เหมือนกับจำนวนเต็มเลย ถ้านำเสนอองค์ประกอบของสีด้วยเลขฐานสิบหก(Hexadecimal Format) ซึ่งในตัวอย่างนี้มีค่าเป็น C02040_h จะเป็นเลขฐานสิบหกของ Long Integer 12,591,168 ในเลขฐานสิบ จะเห็นว่าสองหลักสุดท้ายสอดคล้องกับองค์ประกอบสีแดง (40_h = 64) สองหลักถัดมาสอดคล้องกับองค์ประกอบสีเขียว(20_h = 32) และสองหลักที่มีความสำคัญสูงสุดสอดคล้องกับค่าสีน้ำเงิน(C0_h = 192) ด้วยเหตุนี้เลขฐานสิบหกจึงถูกใช้มากในการกำหนดค่าสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ภาพดิจิทัล

ในการที่จะนำภาพหนึ่งมาประมวลผลด้วยเครื่องคอมพิวเตอร์นั้น ภาพดังกล่าวจำเป็นต้องถูกนำเสนอหรือแทนด้วยตัวเลข แต่ภาพที่ได้มาโดยส่วนมากจะเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชัน $f(x,y)$ ที่ต่อเนื่องในระนาบสองมิติ โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพที่ตำแหน่ง (x,y) ซึ่งเรียกว่าระดับสีเทา (Gray Level) สำหรับภาพขาวดำ และจะเป็นสัดส่วนกับระดับค่า RGB สำหรับภาพสี

2.1.3.1 การแทนภาพด้วยข้อมูลดิจิทัล

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

เมื่อนำภาพแบบอนาลอกที่ต้องการประมวลผล มาผ่านส่วนที่เรียกว่าไดจิติไลเซอร์ (Digitizer) ซึ่งมีหน้าที่ในการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิทัล อุปกรณ์ประเภทนี้ได้แก่ กล้องไดจิติไลเซอร์ ซึ่งจะทำการนำเสนอด้วยฟังก์ชันที่ต่อเนื่อง $f(x,y)$ ให้กลายเป็นฟังก์ชันที่ไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเรียกว่าการสุ่ม (Sampling) และผ่านการควอนไทซ์ (Quantization) ก็จะได้ข้อมูลที่เป็นดิจิทัล

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

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

โดยฟังก์ชัน $f(x,y)$ 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)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G = 2^M$$

(2.3)

เมื่อ G คือจำนวนระดับของสีเทา (Gray Scale) ที่ต้องการใช้ในการเก็บภาพ

2.1.3.2 ลักษณะการจัดเก็บข้อมูลภาพดิจิทัล

โดยทั่วไปแล้วข้อมูลภาพสีหรือภาพขาวดำจะมีค่าตั้งแต่สองระดับขึ้นไป โดยเฉพาะในกรณีภาพขาวดำ นิยมใช้ระดับความเข้มของจุดภาพระดับของสีเทาเท่ากับ 256 ระดับ ซึ่งทำให้จุดภาพมีค่าอยู่ในช่วง 0-255 โดยใช้พื้นที่เก็บข้อมูล 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ($M=8$) ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มหรือระดับสีจำนวนมาก อาจจะต้องการจำนวนบิตสำหรับเก็บข้อมูลมากกว่า 8 บิต อาจเป็น 16 หรือ 24 บิต โดยค่าความเข้มของจุดภาพเท่ากับ 65,536 และ 16,777,216 โดยจะแยกให้ชัดเจนดังนี้

1. ภาพ 2 ระดับก็มีเพียงแต่จุดขาวกับจุดดำเท่านั้น 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
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 โดยการประมวลผลภาพในระดับต่ำจะใช้ความเข้มของจุดโดยตรง ส่วนการประมวลผลระดับสูงนั้น ข้อมูลที่นำมาประมวลผลจะถูกแสดงในรูปของสัญลักษณ์ ซึ่งจะแสดงถึงสิ่งต่างๆ ที่อยู่ในภาพ เช่น ขนาด หรือ รูปร่างของวัตถุในภาพ

2.1.5 สัญญาณข้อมูลภาพจากดิจิทัลวิดีโอ

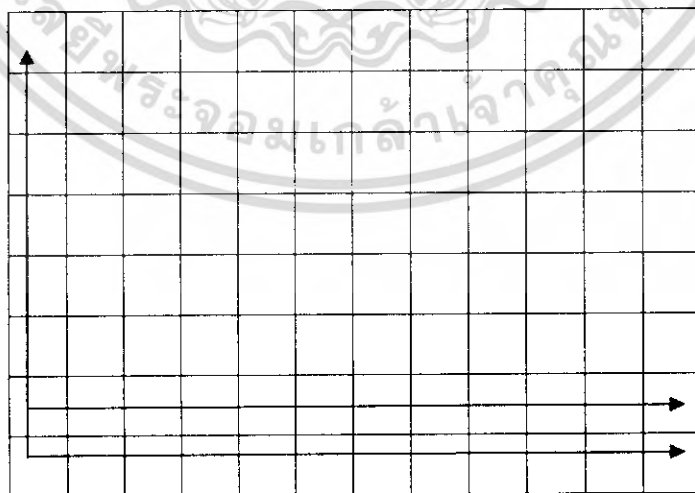
การส่งสัญญาณข้อมูลจากภาพวิดีโอมีลักษณะการส่งที่เป็นลำดับภาพเดี่ยวหรือเฟรม(Frame) ที่ฉายต่อเนื่องกัน เช่นระบบวิดีโอ NTSC จะส่งด้วยอัตราเร็ว 30 เฟรมต่อวินาที โดยดิจิทัลวิดีโอแต่ละเฟรมจะเป็นข้อมูลดิจิทัลในลักษณะเมตริกซ์ซึ่งแต่ละจุดจะเรียกว่าพิกเซล มีค่าระดับของความเข้มสี โดยทั่วไปจะใช้ระดับของสีเทาที่มีค่าตั้งแต่ 0 - 255 โดย 0 จะแทนความมืดมากที่สุด และ 255 จะแทนความสว่างมากที่สุด

2.1.6 ข้อมูลภาพชนิดบิตแมป

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

โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมปจะประกอบด้วย 3 ส่วน คือ

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 โดยข้อมูลแรกจะเป็นค่าสีของพิกเซลที่อยู่แถวล่างสุดที่ตำแหน่งซ้ายสุด ข้อมูลลำดับต่อไปจะเรียงจากทางขวาจากแถวล่างจนถึงแถวบนสุด



รูปที่ 2.2 การเก็บข้อมูลของแต่ละพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7 การสร้างภาพไบนารี

การสร้างภาพไบนารี หมายถึง การแปลงข้อมูลภาพที่มีความเข้มหลายระดับ ให้เป็นภาพที่มีความเข้มเพียง 2 ระดับ คือหนึ่งจุดภาพมีค่าได้สองค่าเท่านั้น โดยเป็นค่า 0 กับค่า 1 ค่าที่เป็น 1 จะหมายถึงจุดภาพที่มีสีดำ และค่าที่เป็น 0 จะหมายถึงจุดภาพที่มีสีขาว

การสร้างภาพไบนารีสามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล(Thresholding Techniques) โดยพิจารณาว่าจุดใดควรเป็นจุดขาวหรือจุดดำ ซึ่งทำได้โดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่ง เรียกว่าค่าเทรชโฮล(Threshold) เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุและพื้นหลัง โดยค่าของจุดภาพใดๆ ที่มีค่าน้อยกว่าค่าเทรชโฮลจะถูกกำหนดค่าเป็น 1 (จุดสีดำ) และถ้าค่าของจุดภาพที่มีค่ามากกว่าค่าเทรชโฮล จะถูกเปลี่ยนให้เป็นค่า 0 (จุดสีขาว) ซึ่งการทำงานสามารถแสดงได้ดังสมการที่ (2.4)

$$\begin{aligned} b(x,y) &= 1 : g(x,y) < Thr \\ b(x,y) &= 0 : g(x,y) > Thr \end{aligned} \quad (2.4)$$

เมื่อ $b(x,y)$ คือ ข้อมูลภาพผลลัพธ์ภาพเป็นไบนารี
 $g(x,y)$ คือ ข้อมูลภาพอินพุตที่มีระดับความเข้ม 0 ถึง L ระดับ
 Th คือ ค่าเทรชโฮลเป็นค่าคงที่ระหว่าง 0 ถึง L ระดับ
 1 คือ จุดภาพที่เป็นสีดำ
 0 คือ จุดภาพที่เป็นสีขาว

โดยที่ L คือระดับความเข้มของจุดภาพสูงสุด

ในการสร้างภาพไบนารีโดยใช้เทคนิคเทรชโฮล เพื่อให้ผลลัพธ์ที่ได้มีความคมชัดและเหมาะสม สิ่งที่สำคัญที่สุด คือ ค่าเทรชโฮล เนื่องจากถ้าเลือกค่าเทรชโฮลที่ไม่เหมาะสม ภาพที่ได้จะไม่คมชัดและรายละเอียดบางส่วนอาจขาดหายไป ดังนั้นปัญหาการสร้างภาพไบนารีด้วยวิธีเทรชโฮล คือ ทำอย่างไรจึงสามารถคำนวณหาค่าเทรชโฮลที่เหมาะสมได้

2.1.7.1 การหาค่าเทรชโฮลโดยการกำหนดไว้ล่วงหน้า

การหาค่าเทรชโฮลโดยวิธีการกำหนดล่วงหน้า(Preassigned Threshold Value) นี้เป็นวิธีที่ง่ายที่สุด เป็นการกำหนดค่าเทรชโฮลเองจากผู้ใช้ ซึ่งจะขึ้นอยู่กับประสบการณ์ของผู้ใช้ โดยการเลือกค่าคงที่ค่าหนึ่งซึ่งจะเป็นค่าที่อยู่ระหว่างค่าต่ำสุดและค่าสูงสุดของระดับความเข้มของข้อมูลอินพุต

วิธีหนึ่งสำหรับการหาค่าเทรชโฮลล่วงหน้าคือ การหาค่าเทรชโฮลจากค่ากลาง(Midrange Threshold Value) การหาวิธีนี้จะพิจารณาจากค่ากลาง โดยอาศัยการคำนวณพื้นฐานทางสถิติในการ

หาค่ากลางหรือค่าเฉลี่ย(Mean) มาประยุกต์ ค่าที่ได้เป็นค่ากึ่งกลางระหว่างค่าระดับความเข้มสูงสุดกับค่าต่ำสุดของข้อมูลภาพอินพุต สำหรับการหาค่ากึ่งกลางจะแสดงได้ดังสมการที่ (2.5)

$$\text{Thr} = \frac{\text{Max}(g(x, y)) + \text{Min}(g(x, y))}{2} \quad (2.5)$$

เมื่อ	Thr	คือ ค่าเทรชโฮล
	$g(x, y)$	คือ ข้อมูลภาพอินพุตที่มีระดับความเข้ม 0 ถึง L ระดับ
	$\text{Max}(g(x, y))$	คือ ค่าสูงสุดเกรย์สเกลของข้อมูลอินพุต
	$\text{Min}(g(x, y))$	คือ ค่าต่ำสุดเกรย์สเกลของข้อมูลอินพุต

การหาค่าเทรชโฮลจากค่าเฉลี่ยเลขคณิต หาได้จากสมการที่ (2.6)

$$\text{Thr} = \frac{\sum g(x, y)}{N \times N} \quad (2.6)$$

2.1.7.2 การสร้างภาพไบนารีโดยใช้โปรแกรม TRIPOD

การสร้างภาพไบนารีโดยใช้โปรแกรม TRIPOD(Template for Real-Time Image Processing Development) ซึ่งเป็นโปรแกรมฟรีแวร์ โดยตัวอย่างการสร้างภาพไบนารีโดยใช้โปรแกรมนี้แสดงโดยรูปที่ 2.3



รูปที่ 2.3 การสร้างภาพไบนารีของโปรแกรม TRIPOD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 รูปด้านบนเป็นรูป ที่จับภาพโดยใช้กล้องเว็บแคมของโลจitech(Logitech USB Camera) ส่วนรูปด้านล่างเป็นรูปที่ถูกแปลงให้เป็นรูปไบนารี 8 บิต สองระดับ ที่เทรชโฮลเท่ากับ 150 ในการใช้งานโปรแกรม TRIPOD จะต้องใช้งานบน MFC ของวิชวลซีพลัสพลัส โดยโค้ดของโปรแกรมที่ใช้ประมวลผลทำภาพไบนารีแสดงในรูปที่ 2.4

สาเหตุที่โปรแกรมนี้ได้สร้างขึ้นเพื่อใช้ร่วมกับกล้องเว็บแคมของโลจitech เพราะผู้คิดโปรแกรมเห็นว่ากล้องเว็บแคมของโลจitechมีใช้อยู่ทั่วไป หาได้ง่าย และมีฟรีแวร์ Quickcam Software Development Kit(QCSDK) ที่ใช้ในการพัฒนาโปรแกรมสำหรับกล้องดังกล่าวให้ดาวน์โหลดได้ฟรีอีกด้วย โดยปกติวินโดวส์จะกำหนดฟอร์แมตรูปภาพเป็น.BMP และ QCSDK จะให้ฟอร์แมตของรูปภาพเป็นภาพทิวทัศน์ 24 บิต เก็บค่าจากซ้ายไปขวาและจากบนลงล่าง แต่ในการใช้งานนิยมใช้งานในรูปแบบระดับของสีเทา 8 บิต

สำหรับภาพขนาด W×H พิกเซล โดย W คือความกว้างของภาพ หมายถึงจำนวนของคอลัมน์ และ H คือค่าความสูง หมายถึงจำนวนของแถว ลำดับขั้นตอนการอ่านในแต่ละพิกเซลมีดังนี้คือ พิกเซลที่มีค่าต่ำกว่าค่าเทรชโฮลจะถูกทำให้พิกเซลนั้นเป็นสีขาวโดยลั้งค่าของพิกเซลนั้นให้เท่ากับ 255 แต่ถ้าพิกเซลมีค่าเป็นอื่นนอกเหนือจากที่กำหนดไว้จะทำพิกเซลนั้นเป็นสีดำ โดยค่าของพิกเซลนั้นจะถูกทำให้เท่ากับ 0 โดยโค้ดการทำเทรชโฮลเป็นดังนี้

```

brinarize - Microsoft Visual C++ - (tripodDlg.cpp)
File Edit View Insert Project Build Tools Window Help
notification
CTripodDlg [All class members] doMyImageProcessing
BMP and all other colors will be black
thresholdValue = 125;
for (row = 0; row < R; row++) {
    for (col = 0; col < V; col++) {
        // Each pixel is composed of 3 bytes
        i = (unsigned long)(row*3*V + 3*col);
        Threshold
        if ( *(_destinationBap + i) > thresholdValue)
            *(_destinationBap + i) =
                *(_destinationBap + i + 1) =
                *(_destinationBap + i + 2) = 0 // Make pixel BLACK
        else
            *(_destinationBap + i) =
                *(_destinationBap + i + 1) =
                *(_destinationBap + i + 2) = 255 // Make pixel WHITE
    }
}
brinarize.exe - 0 error(s) 0 warning(s)
Ln 451, Col 52

```

รูปที่ 2.4 โค้ดโปรแกรมการทำภาพไบนารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(row = 0 ; row < H ; row ++){
    for(col = 0 ; col < W ; col ++){
        if(pixel[row,col] < thresholdValue)
            pixel[row,col] = 0; /* make pixel black*/
        else
            pixel[row,col] = 255; /* make pixel white*/
    }
}

```

โดยเมื่อติดตั้งกล้องเว็บแคมของโลจitechและ QCSDK แล้ว จะสามารถใช้งานโปรแกรม TRIPOD ได้ทันที โดยโค้ดโปรแกรมที่ใช้ทำภาพไบนารีจะพิจารณาเฉพาะส่วนประกอบที่เป็นสีแดงในแต่ละพิกเซลเท่านั้น โค้ดของโปรแกรมแสดงได้ดังนี้

```

Void CTripodDlg::doMyImageProcessing(LPBITMAPINFOHEADER)
lpThisBitmapInfoHeader
{
    // doMyImageProcessing: This is where you'd write your own image processing
code
    // Task: Read a pixel's grayscale value and process accordingly

    unsigned int    W, H;           // Width and Height of current frame [pixels]
    unsigned int    row, col;       // Pixel's row and col positions
    unsigned long   i;              // Dummy variable for row-column vector
    BYTE            thresholdValue; // Value to threshold grayvalue

    W = lpThisBitmapInfoHeader->biWidth; // biWidth: number of columns
    H = lpThisBitmapInfoHeader->biHeight; // biHeight: number of rows

    // In this example, the grayscale image (stored in m_destinationBmp) is
    // thresholded to create a binary image. A threshold value close to 255
    // means that only colors close to white will remain white in binarized
    // BMP and all other colors will be black

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

thresholdValue = 150;

for (row = 0; row < H; row++) {
    for (col = 0; col < W; col++) {

// Recall each pixel is composed of 3 bytes
        i = (unsigned long)(row*3*W + 3*col);

// Add your code to operate on each pixel. For example
// *(m_destinationBmp + i) refers to the ith pixel in the destinationBmp
// Since destinationBmp is a 24-bit grayscale image, you must also apply
// the same operation to *((m_destinationBmp + i + 1) and *((m_destinationBmp + i
+ 2)
// Threshold: if a pixel's grayValue is less than thresholdValue
        if( *(m_destinationBmp + i) <= thresholdValue)
            *(m_destinationBmp + i) =
            *(m_destinationBmp + i + 1) =
            *(m_destinationBmp + i + 2) = 0; // Make pixel BLACK
        else
            *(m_destinationBmp + i) =
            *(m_destinationBmp + i + 1) =
            *(m_destinationBmp + i + 2) = 255; // Make pixel WHITE
    }
}

```

	0	1	2	3
0	A	B	C	D
1	E	F	G	H
2	I	J	K	L

(ก) รูปแบบเมตริกซ์

*(myImage + r*W + c)											
A	B	C	D	E	F	G	H	I	J	K	L
0	1	2	3	4	5	6	7	8	9	10	11

(ข) รูปแบบเวกเตอร์

รูปที่ 2.5 การนำเสนอรูปภาพในรูปแบบเมตริกซ์ และเวกเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.5 (ก) เป็นการนำเสนอรูปภาพด้วยเมตริกซ์(Matrix Form) 2 มิติ ซึ่งสามารถเข้าถึงจุดข้อมูลโดยฟังก์ชัน myImage[r,c] โดยที่ r และ c เป็นค่าแถวและคอลัมน์ของพิกเซลที่สนใจตามลำดับ จากรูปตัวอย่างนี้เมตริกซ์มีขนาด 3 : 4 สมมติว่าพิจารณา myImage[2,1] จะอ้างอิงพิกเซลแถวที่ 2 คอลัมน์ที่ 1 ซึ่งก็คือพิกเซล v ส่วนรูปที่ 2.5 (ข) เป็นการนำเสนอรูปภาพอีกวิธีหนึ่ง โดยทำให้อยู่รูปแบบเวกเตอร์(Vector Form) 1 มิติ โดยการถ่วงตำแหน่งของพิกเซลทำได้โดยสมการ (2.7)

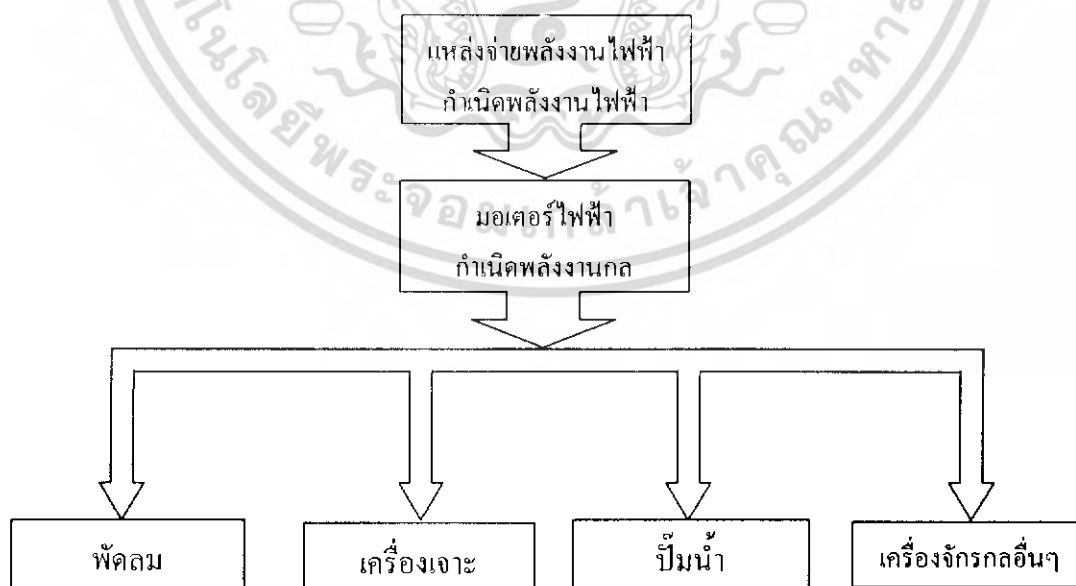
$$(myImage + r*W + C) \quad (2.7)$$

เมื่อ	myImage	คือ	ตำแหน่งเริ่มต้นของพิกเซล
	r	คือ	Pixel's row
	c	คือ	Pixel's column
	W	คือ	จำนวนคอลัมน์ทั้งหมดของภาพ

สำหรับตัวอย่าง r = 2 , c = 1 และ W = 4 แล้ว (myImage + r*W + C) จะให้ค่า (myImage + 9) ในรูปแบบเวกเตอร์ myImage[9] จะเหมือนกับ *(myImage + 9) ซึ่งก็คือค่า "J" นั่นเอง

2.2 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้า คือ อุปกรณ์เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล เพื่อนำพลังงานกลที่ได้ไปขับเคลื่อนสิ่งต่างๆ ตามต้องการ ดังแสดงในรูปที่ 2.6 โครงการนี้ได้อาศัยมอเตอร์ไฟฟ้ากระแสตรงมาใช้งานเพื่อทำการควบคุมความ โดยที่คานจะยึดอยู่กับเพลาของมอเตอร์ ดังนั้นจึงจำเป็นต้องมีการศึกษามอเตอร์ไฟฟ้ากระแสตรง ซึ่งในปัจจุบันนี้มีอยู่หลายแบบหลายขนาดเพื่อสามารถตัดสินใจเลือกใช้มอเตอร์กับระบบควบคุมได้อย่างเหมาะสม



รูปที่ 2.6 การเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกลของมอเตอร์ไฟฟ้าเพื่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 แรงบิดและงาน

แรงบิด(Torque) คือ การหมุนหรือการบิดของโมเมนต์ของแรงๆ หนึ่ง ที่กระทำรอบแกนแกนหนึ่งมีขนาดเท่ากับ ผลคูณของแรงกับรัศมีของวงล้อ จะได้ความสัมพันธ์พื้นฐานดังสมการ (2.8)

$$T = F \cdot r \quad (2.8)$$

เมื่อ T คือ แรงบิด หน่วยเป็น นิวตันเมตร(N-m)

F คือ แรงที่กระทำ หน่วยเป็น นิวตัน(N)

r คือ ระยะทาง หน่วยเป็น เมตร(m)

งาน W ที่ทำได้จากหนึ่งรอบแรงนี้สามารถหาได้จากสมการ (2.9)

$$\begin{aligned} W &= \text{แรง} \times \text{ระยะทาง} \\ W &= F \cdot 2\pi r \end{aligned} \quad (2.9)$$

งานที่ทำต่อวินาทีสามารถหาได้จากสมการ (2.10)

$$\begin{aligned} W &= F \cdot 2\pi r s \\ \text{เมื่อแทนค่า } 2\pi r s &= \theta \\ \text{จะได้สมการเป็น } W &= T\theta \end{aligned} \quad (2.10)$$

เมื่อ s คือ ความเร็ว มีหน่วยเป็น รอบ/วินาที(rpm)

2.2.2 แรงบิดที่เกิดขึ้นในอาร์เมเจอร์

ให้ T_m เป็นแรงบิดที่เกิดขึ้นในอาร์เมเจอร์(Armature Torque of Motor) เมื่อมอเตอร์หมุนด้วยความเร็ว s รอบ/วินาที เมื่อ K_f คือ ค่าคงที่แรงบิดในมอเตอร์ โดย T_m หน่วยเป็นนิวตันเมตร(N-m) ดังสมการ (2.11)

$$P_m = T_m 2\pi s \quad (2.11)$$

เมื่อ P_m คือ กำลังที่เกิดขึ้นที่อาร์มเจอร์ หน่วยเป็นวัตต์ (w)

เมื่อรู้ว่ากำลังไฟฟ้าที่เปลี่ยนไปเป็นกำลังกลในอาร์เมเจอร์ คือ

$$P_m = E_g I_a$$

$$\text{จะได้} \quad T_m 2\pi s = E_g I_a$$

$$\text{จาก} \quad E_g = \frac{\phi Z s P_m}{a}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

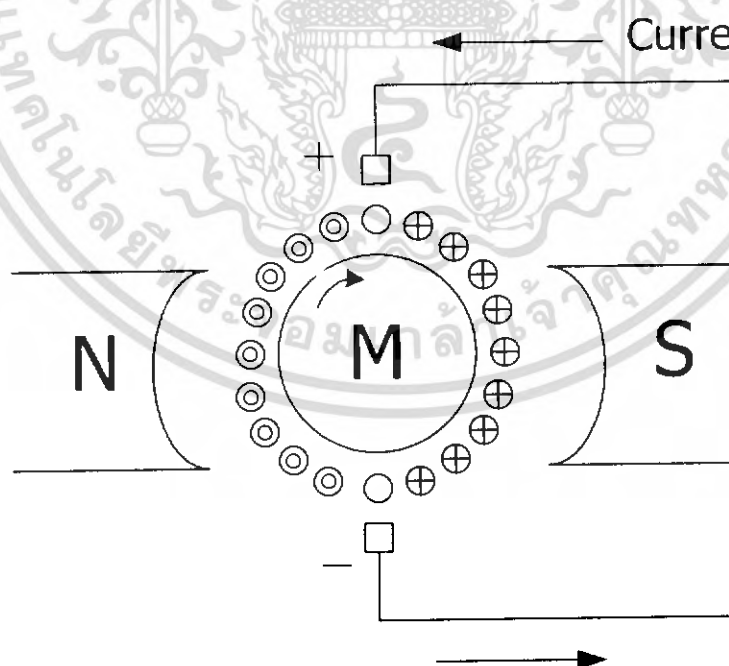
จะได้สมการแรงบิดในมอเตอร์ดังนี้

$$T_m = K_a \phi I_a \quad (2.12)$$

- E_b คือ แรงดันไฟฟ้าต้านกลับ หน่วยเป็น โวลต์ (V)
 ϕ คือ เส้นแรงแม่เหล็กต่อขั้ว หน่วยเป็น เวเบอร์ (Wb)
 P คือ จำนวนขั้วแม่เหล็ก หน่วยเป็น โพล (pole)
 n คือ จำนวนรอบที่มอเตอร์หมุน หน่วยเป็น รอบต่อนาที (rpm)
 Z คือ จำนวนตัวนำบนอาร์เมเจอร์ หน่วยเป็น ตัวนำ
 a คือ จำนวนวงจรไฟฟ้าคู่ขนานบนอาร์เมเจอร์ หน่วยเป็น วงจร
 K คือ ค่าคงที่ของมอเตอร์

2.2.3 แรงดันไฟฟ้าต้านกลับ

พิจารณามอเตอร์ดังรูปที่ 2.7 ขณะที่อาร์เมเจอร์หมุนในสนามแม่เหล็กทำให้มีแรงดันไฟฟ้าเหนี่ยวนำเกิดขึ้นบนตัวนำอาร์เมเจอร์ ตามกฎมือขวาของเฟลมมิ่ง ซึ่งมีทิศทางการไหลของกระแสเหนี่ยวนำตรงกันข้ามในลักษณะที่ต่อต้านการไหลของกระแสอาร์เมเจอร์ ทำให้เกิดแรงดันไฟฟ้าเหนี่ยวนำขึ้นจึงเรียกว่า “แรงดันไฟฟ้าต้านกลับ” (Back หรือ Counter Electromotive Force)



รูปที่ 2.7 ความเป็นเครื่องกำเนิดในมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะได้ความสัมพันธ์ คือ

$$E_g = \frac{\phi Z n P}{60 a}$$

โดย

$$K = \frac{Z P}{60 a}$$

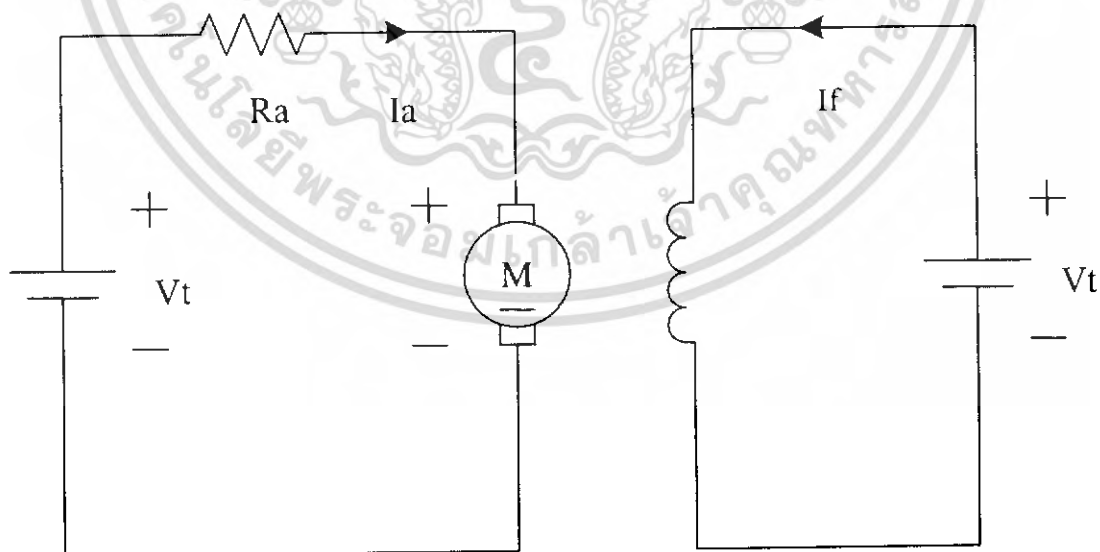
ได้สมการเป็น

$$E_g = K \phi n \tag{2.13}$$

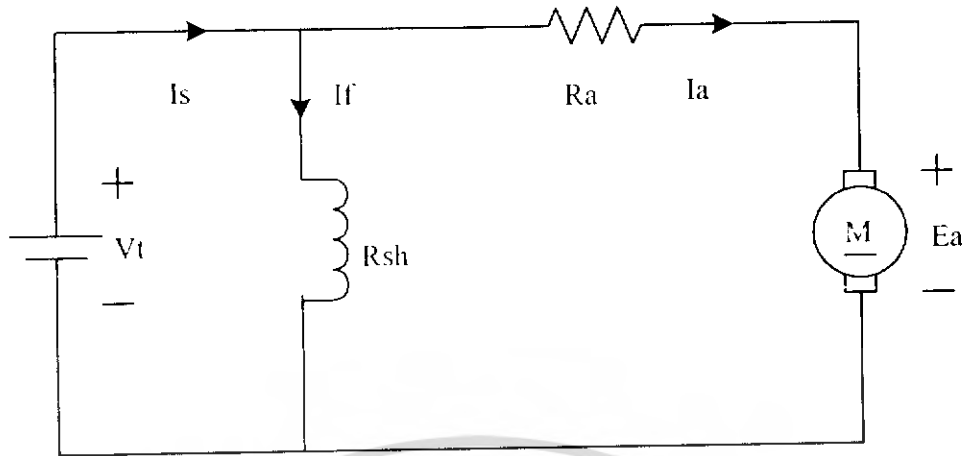
2.2.4 การแบ่งชนิดของมอเตอร์กระแสตรงตามลักษณะการกระตุ้น

การแบ่งชนิดของมอเตอร์กระแสตรงตามลักษณะการกระตุ้น สามารถแบ่งได้เป็นสองแบบ คือ

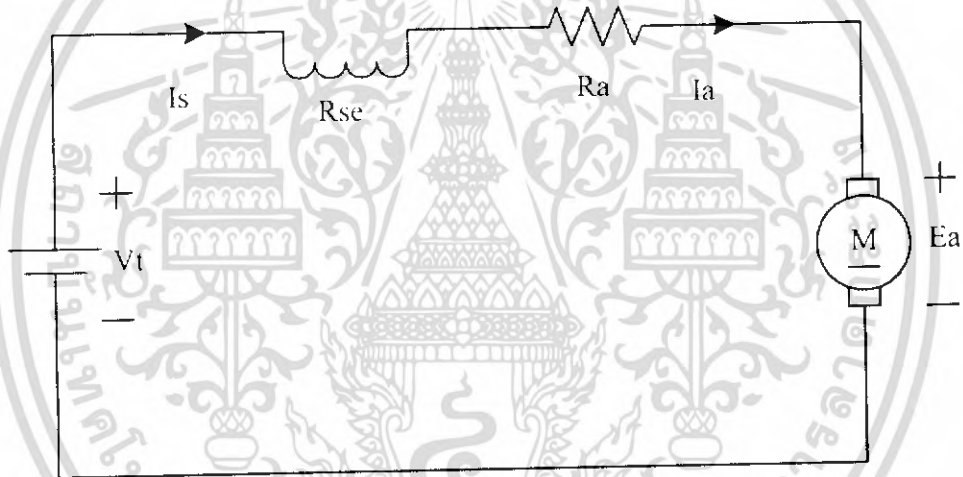
1. มอเตอร์ไฟฟ้ากระแสตรงแบบแยกขดกระตุ้น (Separately Excited Motor) แสดงดังรูปที่ 2.8
2. มอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นด้วยตัวเอง (Self Excited Motor) สามารถแบ่งได้อีกสามแบบคือ
 - มอเตอร์ไฟฟ้ากระแสตรงแบบขนาน (Shunt Motor) แสดงดังรูปที่ 2.9
 - มอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม (Series Motor) แสดงดังรูปที่ 2.10
 - มอเตอร์ไฟฟ้ากระแสตรงแบบผสม แสดงดังรูปที่ 2.11 ซึ่งสามารถแบ่งได้สองแบบคือ
 - แบบขดลวดขนานระยะสั้น (Short Shunt) แสดงดังรูป 2.11 (ก)
 - แบบขดลวดขนานยาว (Long Shunt) แสดงดังรูป 2.11 (ข)



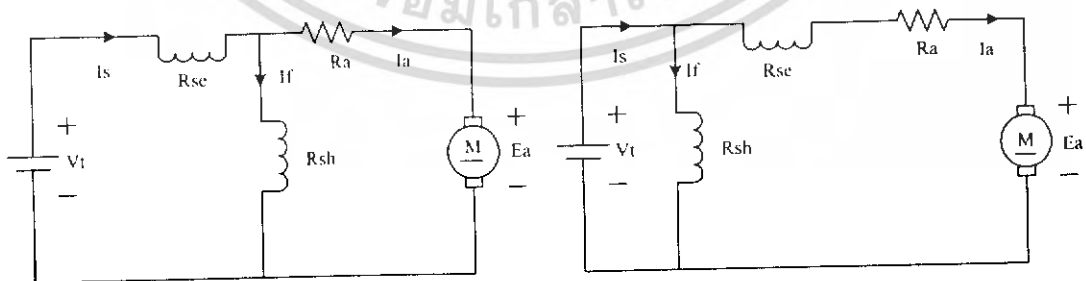
รูปที่ 2.8 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบแยกขดกระตุ้น



รูปที่ 2.9 วงจรมอเตอร์ไฟฟ้าแบบขนาน



รูปที่ 2.10 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบอนุกรม



(ก) แบบขดลวดขนานสั้น (Short Shunt)

(ข) แบบขดลวดขนานยาว (Long Shunt)

รูปที่ 2.11 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

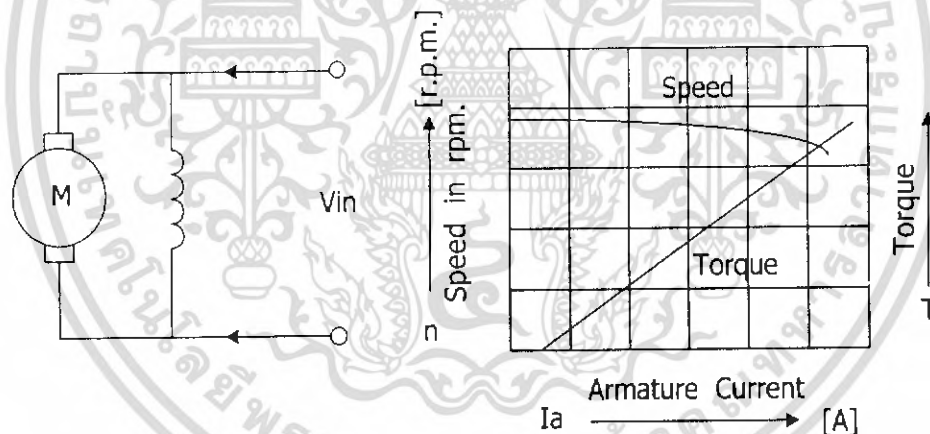
2.2.5 คุณลักษณะของมอเตอร์

Performance Curve คือกราฟหรือเส้นโค้งที่แสดงคุณลักษณะประจำตัวต่างๆ ของมอเตอร์แต่ละแบบ โดยแสดงความสัมพันธ์ต่างๆ เปรียบเทียบกับกำลังม้าที่มอเตอร์จ่ายออกมา อาทิเช่น ความเร็วกับแรงม้า ประสิทธิภาพกับแรงม้า เป็นต้น ในที่นี้จะกล่าวแยกไปตามแบบของมอเตอร์ที่ใช้ ดังนี้

2.2.5.1 มอเตอร์แบบขนาน

มอเตอร์แบบนี้มีคุณลักษณะประจำตัวที่สำคัญๆ อยู่ 4 อย่าง คือ แรงบิด กระแส ความเร็ว และ ประสิทธิภาพ ลักษณะประจำตัวเหล่านี้กับกำลังม้าที่มอเตอร์ส่งออกไปแสดงดังรูปที่ 2.12

จากรูปที่ 2.12 พบว่าในขณะที่มอเตอร์ไม่ได้รับภาระทางกล(Mechanical Load) นั้น จะมีค่าความเร็วที่ค่าๆ หนึ่ง และความเร็วของมอเตอร์จะไม่วิ่งเร็วกว่าความเร็วปกติ(Run Away) เมื่อทำการตัดภาระทางกลออกไปทันที อย่างไรก็ตามความเร็วของมอเตอร์ขณะไม่มีภาระและขณะมีภาระเต็มที่นั้น ไม่แตกต่างกันไปมากนัก ดังนั้นจึงถือว่ามอเตอร์มีความเร็วคงที่ ความเร็วในช่วงที่นำไปใช้งานนี้ สามารถปรับให้เร็วได้ โดยการต่อความต้านทานอนุกรมเข้ากับขดลวดขนาน(Shunt Field)



รูปที่ 2.12 คุณลักษณะของมอเตอร์แบบขนาน

จากรูปที่ 2.12 พบว่ามอเตอร์จะกินกระแสไฟอยู่ค่าหนึ่งในขณะที่มอเตอร์ยังไม่จ่ายโหลดออกไป ทั้งนี้เพราะต้องนำกำลังไฟฟ้าค่านั้นไปจ่ายให้ส่วนสูญเสียกำลังคงที่นั่นเอง

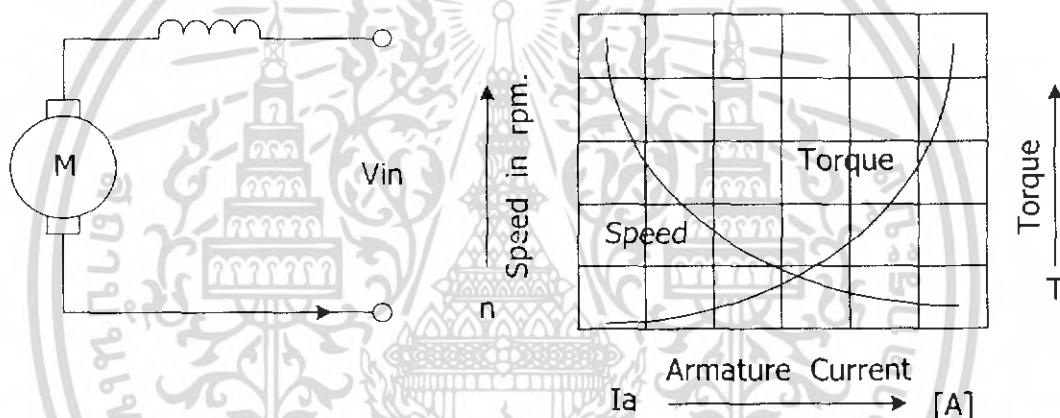
การทำให้มอเตอร์ขนานเริ่มหมุนโหลด จะพบว่ามอเตอร์แบบขนานนี้มีแรงบิดหมุนต่ำกว่าแบบอนุกรมหรือแบบผสม นั่นคือมีแรงเริ่มหมุนต่ำ อย่างไรก็ตามไม่ได้หมายความว่ามอเตอร์แบบขนานนี้จะไม่สามารถเริ่มหมุนที่ภาระทางกลหนักๆ ได้ แต่มอเตอร์แบบอนุกรมหรือแบบผสมนั้นสามารถที่จะทำให้ภาระทางกลที่หนักๆ หมุนได้ โดยที่กินกระแสไฟขณะที่ทำให้เริ่มหมุนน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.2 มอเตอร์แบบอนุกรม

คุณลักษณะของมอเตอร์แบบนี้แสดงดังรูปที่ 2.13 ซึ่งประกอบด้วยเส้นกราฟต่างๆ ที่แสดงคุณลักษณะประจำตัวของมอเตอร์ คิดในทำนองเดียวกับคุณลักษณะของมอเตอร์แบบขนานนั่นเอง แต่สำหรับมอเตอร์แบบอนุกรมนี้ เส้นกราฟแสดงให้เห็นชัดว่าความเร็วจะลดลงอย่างมากเมื่อต้องจ่ายกำลังออกไปมาก ส่วนในขณะที่ไม่ได้จ่ายกำลังนั้น ความเร็วของมอเตอร์จะเพิ่มสูงขึ้นอย่างรวดเร็ว เมื่อเปรียบเทียบกับขณะที่ต้องจ่ายกำลังไฟเต็มที่ นั่นคือ ความเร็วจะไม่คงที่และจะเปลี่ยนแปลงไปมาก เมื่อต้องจ่ายกำลังกลมาก ฉะนั้นจึงไม่นิยมใช้กับโหลดที่ต้องการให้มีค่าความเร็วคงที่หรือค่อนข้างสม่ำเสมอ

สำหรับกระแสค่าหนึ่งที่มีมอเตอร์แบบนี้ใช้ในการทำให้เริ่มหมุนนั้นจะปรากฏว่าให้แรงบิดเริ่มแรกมากกว่ามอเตอร์แบบขนานที่ค่ากระแสเดียวกัน ดังนั้นจึงนิยมใช้ในกรณีที่ต้องการจุดโหลดเริ่มแรกที่หนักๆ ซึ่งเป็นเอกลักษณ์หรือข้อเด่นของมอเตอร์แบบอนุกรม



รูปที่ 2.13 คุณลักษณะของมอเตอร์แบบอนุกรม

2.2.6 การเปรียบเทียบคุณลักษณะของมอเตอร์

เครื่องกลไฟฟ้าที่มีโครงสร้างโครงเดียวกันนั้นย่อมมี คุณลักษณะเหมือนกัน แม้ว่าเครื่องกลไฟฟ้านั้นจะเป็นมอเตอร์แบบอนุกรม แบบขนาน หรือแบบผสมก็ตาม

2.2.6.1 การเปรียบเทียบของมอเตอร์แบบขนานกับแบบอนุกรม

มอเตอร์แบบขนาน คุณลักษณะประจำตัวของมอเตอร์แบบขนานสรุปได้ดังนี้

- มอเตอร์แบบขนานจะมีความเร็วคงที่ในการใช้งานที่โหลดต่างกัน
- แรงบิดเริ่มแรกของมอเตอร์แบบขนานไม่สูงเหมือนกับมอเตอร์แบบอนุกรมเมื่อเปรียบเทียบในกรณีที่ต้องกินกระแสไฟเท่าๆกัน

การนำไปใช้งาน

- ใช้กับงานที่ต้องการให้มีความเร็วคงที่โดยประมาณ จากช่วงที่ไม่ได้รับการชะงักจนถึงช่วงที่ได้รับการชะงักเต็มที่(Full Load)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้หมุนหรือใช้ขับโหลด ณ ที่ความเร็วต่างๆ กัน โดยจะมีค่าคงที่ในแต่ละความเร็วตามที่ตั้งไว้ ขึ้นอยู่กับงานนั้นๆ ว่าต้องการใช้ความเร็วค่านี้นานเท่าไร

มอเตอร์แบบอนุกรม คุณลักษณะประจำตัวของมอเตอร์แบบนี้สรุปได้ดังนี้

- มีแรงบิดเริ่มแรกสูงมาก
- มีความเร็วต่ำเมื่อต้องการหมุนโหลดมากๆ และมีความเร็วสูงมากจนอาจเป็นอันตรายกับตัวมอเตอร์เองเมื่อมีโหลดต่ำมากๆ

การนำไปใช้งาน

- ใช้กับงานที่ต้องการแรงบิดสูงๆ เช่น ใช้ในการขับรถราง เป็นต้น
- ใช้กับงานที่มอเตอร์สามารถต่อกับโหลดได้โดยตรง เช่น พัดลม ซึ่งความเร็วจะเพิ่มขึ้นได้ก็ต่อเมื่อมีแรงบิดเพิ่มขึ้น
- ใช้ในกรณีที่ไม่จำเป็นต้องให้ความเร็วคงที่เมื่อได้รับ โหลดมากขึ้น ซึ่งจะมีข้อดีอยู่ด้วยคือมอเตอร์จะไม่กินไฟเพิ่มมากขึ้นตามแรงบิดที่เพิ่มขึ้นมา
- มอเตอร์แบบอนุกรมนี้ ไม่ควรนำไปใช้กับโหลดที่มีโอกาสจะเกิดค่าโหลดน้อยๆ เช่น นำไปใช้กับปั๊มหนีศูนย์กลาง(Centrifugal Pump) หรือขับโหลดชนิดที่ใช้สายพานเป็นตัวนำ เป็นต้น

2.2.6.2 คุณลักษณะพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรง

วงจรเทียบเคียงของมอเตอร์แบบกระตุ้นภายนอกแสดงได้โดยรูปที่ 2.13 ซึ่งสามารถหาค่ากระแสที่ฟลักซ์ชั่วขณะได้จากสมการ (2.14)

$$I_f = R_f I_f + L_f \frac{dI_f}{dt} \quad (2.14)$$

และหาค่ากระแสที่อาร์เมเจอร์ชั่วขณะได้จากสมการ (2.15)

$$I_a = R_a I_a + L_a \frac{dI_a}{dt} + E_g \quad (2.15)$$

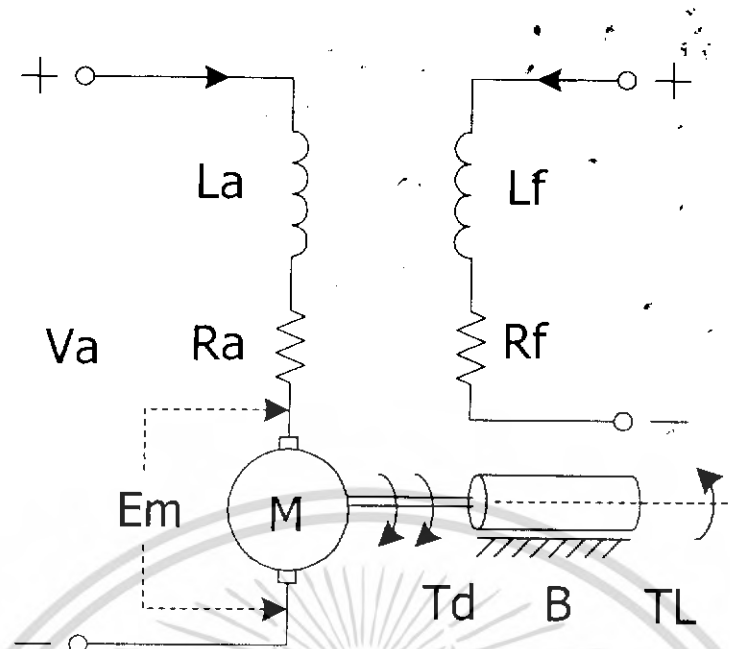
แรงดันไฟฟ้าต้านกลับ(E_m) หรือที่เรียกว่าแรงดันความเร็ว(Speed Voltage) มีดังสมการ (2.16)

$$E_m = K_v \omega I_f \quad (2.16)$$

และแรงบิดที่เพลลาของมอเตอร์(T) คำนวณได้จากสมการ (2.17)

$$T = K_t \omega I_f \quad (2.17)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 วงจรเทียบเคียงของมอเตอร์แบบกระตุ้นภายนอก

โดยแรงบิดที่เพลารวมของมอเตอร์จะต้องเท่ากับ โหลดของแรงบิด อธิบาย ได้ดังสมการ (2.18)

$$T = J \frac{d\omega}{dt} + B\omega + T_l \quad (2.18)$$

เมื่อ ω คือ ความเร็วมอเตอร์ (rad/s)

B คือ Viscous friction constant ($N-m/(rad \cdot s)$)

2.3 วงจรและภาคขับมอเตอร์

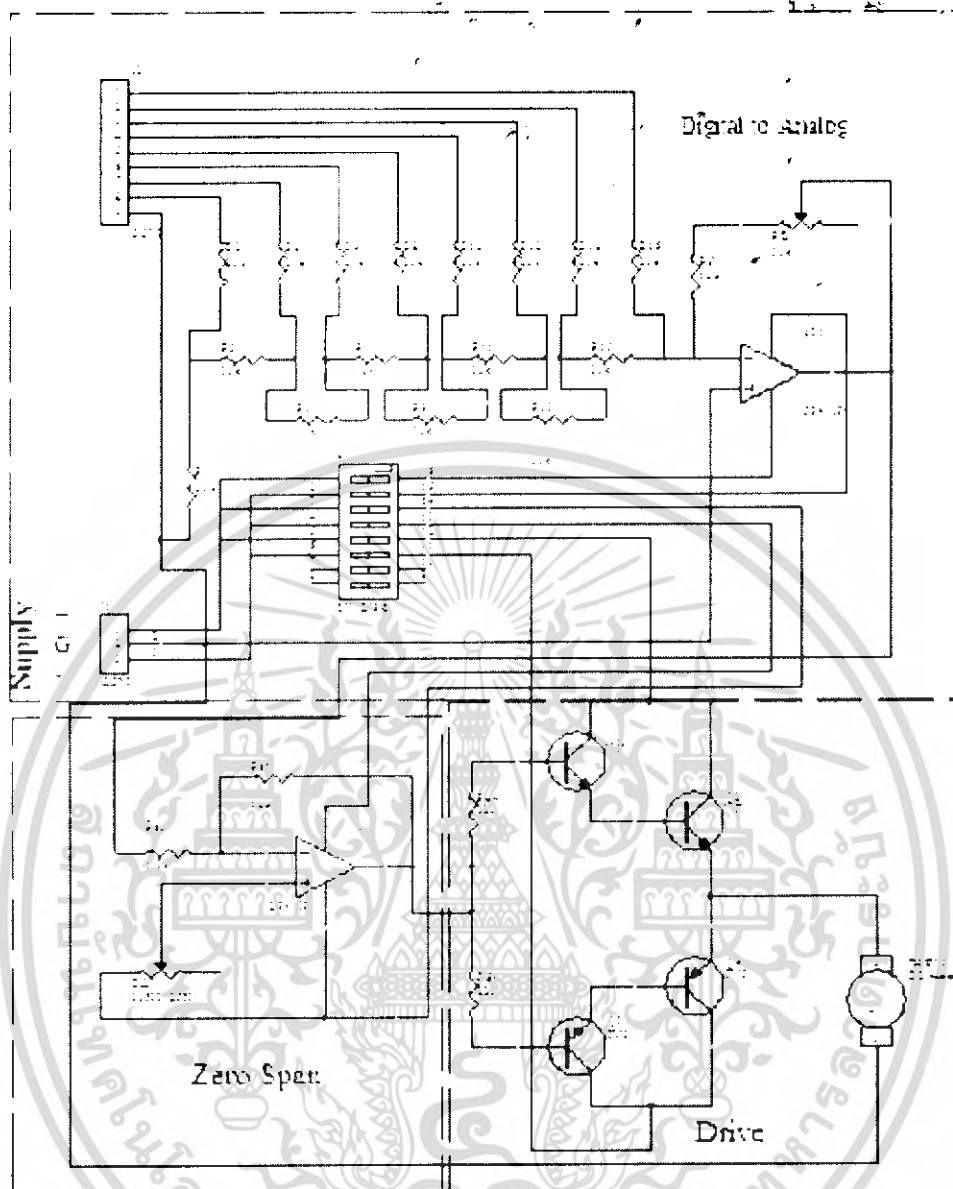
ส่วนของวงจรและภาคขับมอเตอร์นั้น ถูกออกแบบมาเพื่อให้สามารถควบคุมมอเตอร์เพื่อนำไปควบคุมแกนเล็งลูกบอลที่ยึดติดอยู่กับเพลารวมของมอเตอร์ โดยมอเตอร์จะต้องหมุนได้ทั้งสองทิศทางคือหมุนในทิศทางตามเข็มนาฬิกาและทิศทางทวนเข็มนาฬิกา ดังนั้นจึงจำเป็นที่จะต้องออกแบบวงจรเพื่อควบคุมมอเตอร์ให้ทำงานตามต้องการได้ โดยเฉพาะจะต้องสามารถทำงานร่วมกับคอมพิวเตอรืที่จะใช้ในการควบคุม โดยคอมพิวเตอรืจะส่งค่าต่างๆ ออกมาทางพอร์ตขนาน วงจรที่ใช้สำหรับโครงงานนี้ประกอบด้วยส่วนต่างๆ แสดงดังรูปที่ 2.15 ดังนี้คือ

2.3.1 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบใช้ความต้านทานสองค่า

2.3.2 วงจรปรับซีโรสแปน (Zero Span)

2.3.3 วงจรขับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 วงจรและภาคขับมอเตอร์

2.3.1 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแบบใช้ความต้านทานสองค่า

วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกที่ใช้กันทั่วไปมีอยู่ 2 แบบคือ วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแบบใช้ตัวต้านทานหลายค่า และวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแบบใช้ความต้านทานสองค่า วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแบบใช้ตัวต้านทานหลายค่า นั้นนิยมใช้กับสัญญาณดิจิทัลขนาด 4 บิต แต่ถ้าต้องการใช้แปลงสัญญาณสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลที่มีขนาดมากกว่า 4 บิต จะเกิดปัญหาเนื่องจากต้องการค่าความต้านทานที่มีช่วงกว้างมาก ส่วนวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

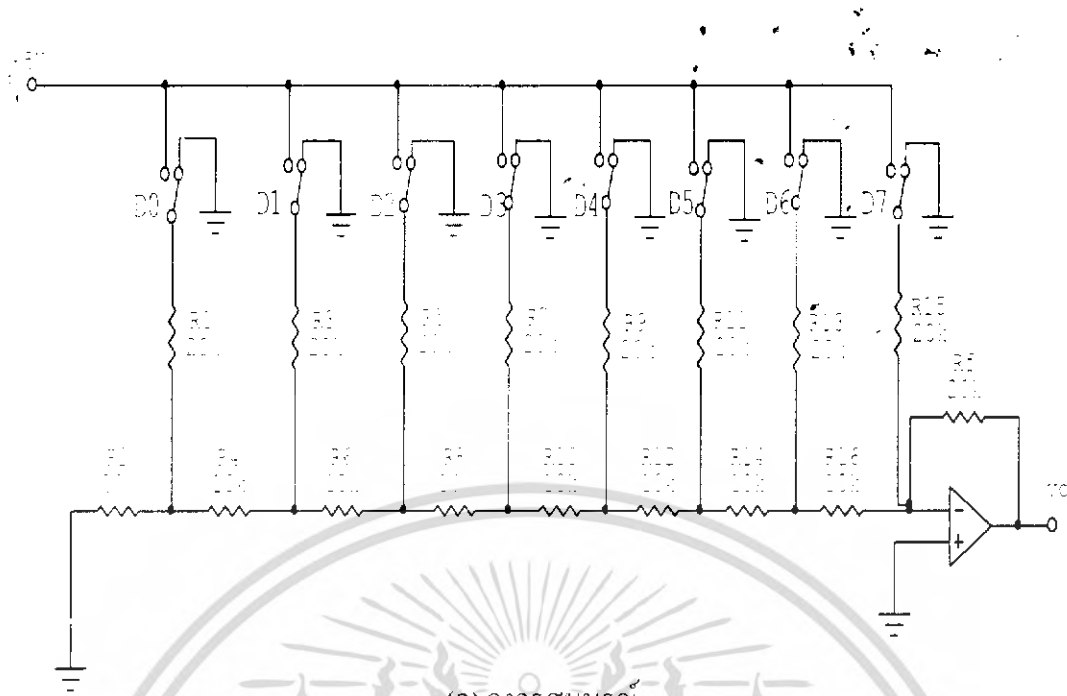
แบบใช้ค่าความต้านทานสองค่า นั้นจะใช้ความต้านทานเพียง 2 ค่า ซึ่งกระแสจะถูกเปลี่ยนเป็นค่าแรงดันโดยออปแอมป์และตัวต้านทานป้อนกลับ Rf หลักการสวามด้านทาน 2 ค่า โน้มนั้นแรกสมมติว่าสวิทช์ D7 ซึ่งเป็นสวิทช์ในบิตที่มีนัยสำคัญสูงสุดนั้น ต่อกับแรงดันอ้างอิง 5 โวลต์ ในขณะที่สวิทช์ตัวอื่นปิดลงกราวด์ ดังนั้น R1 และ R2 จึงต่อขนานกันลงกราวด์ จะเห็นว่าตัวต้านทาน 2R ต่อขนานกับ 2R อีกตัวหนึ่ง จึงมีค่าเท่ากับ R ค่า R นี้จะถูกบวกกับ R4 กลายเป็นค่า 2R แล้วขนานกันกับ R3 ลงกราวด์

การรวมของ R3 และตัวต้านทานก่อนหน้าจึงทำให้เหลือเพียงค่า R ต่ออนุกรมกับ R6 พิจารณาเช่นเดียวกับวงจรส่วนที่เหลือก็จะได้เป็นวงจรง่ายขึ้น ดังรูปที่ 2.16 (ข) โดยเหตุที่กราวด์เทียบของออปแอมป์มีแรงดัน 0 โวลต์ ทำให้ไม่มีกระแสไหลผ่านค่าความต้านทานเหล่านี้ลงกราวด์ จึงไม่ต้องสนใจส่วนนี้ ดังนั้นแรงดัน 5 โวลต์ ที่ปลายข้างหนึ่งของ R15 ที่มีค่า 20 กิโลโอห์ม ทำให้มีกระแส 0.25 มิลลิแอมป์ ผ่านที่จุดต่อ Rf ซึ่งมีค่า 20 กิโลโอห์ม แรงดันเอาต์พุตที่ได้จากบิตที่มีนัยสำคัญสูงสุดจึงมีค่า -5 โวลต์

แรงดันที่ได้จากบิตที่มีนัยสำคัญรองลงมาหาได้โดยปิดสวิทช์ D6 ไปยัง -5 โวลต์ และ D5 ลงกราวด์ การวิเคราะห์วงจรสามารถอาศัยทฤษฎีของเทวินิน โดยมีการแบ่งแรงดันระหว่าง R15 และ 2R ที่ต่อลงกราวด์ ดังรูปที่ 2.16 (ค) แรงดันของเทวินิน คือแรงดันที่รอยต่อซึ่งมีค่า 2.5 โวลต์ ตัวต้านทานเทวินินมีค่าเท่ากับตัวต้านทาน 2 ตัวต่อขนานกัน(หรือ R) ความต้านทานรวมของอุปกรณ์ที่อยู่ทางซ้าย R14 สามารถลดรูปได้เหลือค่า R ต่อกับแรงดัน 2.5 โวลต์ โดยสามารถละทิ้ง R15 ได้ เพราะจุดปลายทั้งสองค่าต่อลงกราวด์ ค่าความต้านทานรวมที่จุดรวมคือ 2R หรือ 20 กิโลโอห์ม และแรงดันเทวินินคือ 2.5 โวลต์ กระแสที่ผ่าน Rf 20 กิโลโอห์ม ทำให้เกิดแรงดันเอาต์พุต -2.5 โวลต์ (สำหรับบิตที่มีนัยสำคัญถัดมา)

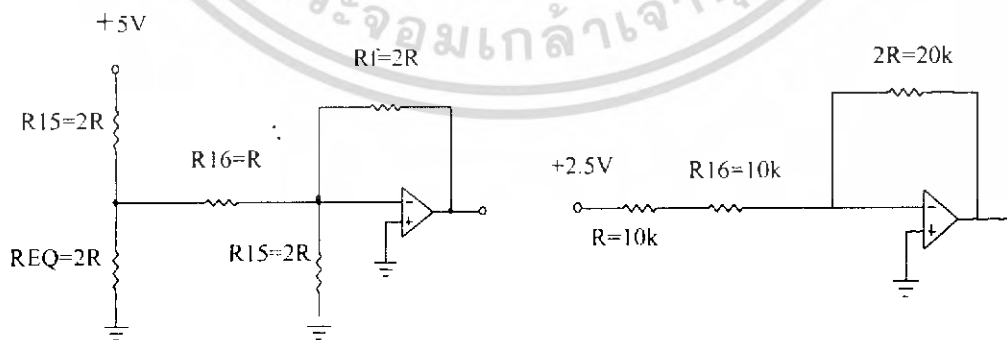
ด้วยการวิเคราะห์เช่นเดียวกันนี้ สามารถหาแรงดันเอาต์พุตที่บิตต่ำลงมาได้อีก 1.25 โวลต์ และที่ค่าดิจิตอลต่ำสุดได้ 0.625 โวลต์ ในขณะที่ค่าดิจิตอลสูงสุดซึ่งสอดคล้องกับสวิทช์ทุกตัวต่อไปที่ +5 โวลต์ ได้เอาต์พุตเต็มสเกลคือ 9.375 โวลต์ แม้ว่าวงจรเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกแบบ R/2R แลคเคอร์จะวิเคราะห์ยากกว่าแบบใช้ความต้านทานหลายค่า แต่ง่ายสำหรับการต่อวงจรให้ถูกต้อง เพราะใช้ค่าความต้านทานเพียง 2 ค่าเท่านั้น จำนวนบิตก็เพิ่มได้โดยเพิ่มส่วนของ R/2R ลงไป วงจรนับ 4 บิต ที่เป็น TTL หรือ CMOS อาจนำมาต่อแทนตำแหน่งของสวิทช์ในวงจรรูปที่ 2.16 (ค) เพื่อให้แรงดันเอาต์พุตเป็นขั้นบันไดได้

สำหรับการทดลองจริงได้ใช้ไอซีเบอร์ DAC0832 ทำหน้าที่แปลงสัญญาณดิจิตอลขนาด 8 บิต เป็นสัญญาณอนาล็อก เนื่องจากมีเสถียรภาพดีกว่าใช้วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกแบบใช้ความต้านทานสองค่า และยังสามารถนำไปใช้กับคอมพิวเตอร์เครื่องอื่นๆ ได้ โดยไม่ต้องมีการปรับตั้งอีก การใช้งานไอซีเบอร์ DAC0832 กล่าวไว้ในภาคผนวก ข-1



(ก) วงจรสมบูรณ์

(ข) วงจรเสมือนในขณะที่สวิตช์ของบิตที่มีนัยสำคัญสูงสุดปิด



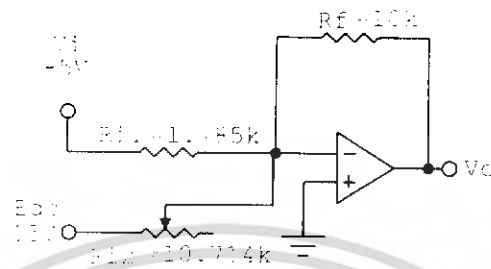
(ค) วงจรเสมือนในขณะที่สวิตช์ของบิตที่มีนัยสำคัญสูงสุดรองลงมาปิด

รูปที่ 2.16 วงจรเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณอนาลอกแบบ R/2R แลตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 วงจรปรับซีโรสเปน

วงจรซีโรสเปน เป็นวงจรปรับสภาพสัญญาณ(Signal Conditioning) ชนิดหนึ่ง ซึ่งมีหน้าที่ปรับระดับสัญญาณให้อยู่ในย่านที่ต้องการ โดยในโครงงานนี้จำเป็นต้องใช้วงจรซีโรสเปนร่วมกับวงจรแปลงสัญญาณดิจิทัลเป็นอนาลอกและวงจรขั้วมอเตอร์ วงจรซีโรสเปนแสดงดังรูปที่ 2.17



รูปที่ 2.17 วงจรซีโรสเปน

เนื่องจากแรงดันเอาต์พุตที่ออกมาจากวงจรแปลงสัญญาณดิจิทัลเป็นอนาลอก ให้ค่าแรงดันจาก 0 ถึง -5 โวลต์ แต่แรงดันที่วงจรขั้วมอเตอร์สามารถทำงานได้มีค่าจาก -12 โวลต์ ถึง +12 โวลต์ จึงจำเป็นต้องใช้วงจรปรับซีโรสเปนปรับระดับแรงดันให้มีทั้งค่าบวกและลบก่อนที่จะเข้าวงจรขั้วมอเตอร์ เพื่อขยายแรงดันและกระแสให้มากเพียงพอที่จะสามารถขั้วมอเตอร์ได้ โดยแรงดันเอาต์พุตของวงจรปรับซีโรสเปนนั้น อธิบายได้ดังสมการที่ (2.19)

$$V_o = (-V_i) \left(-R_{f1} / R_{i1} \right) - (R_{f2} / R_{i2})(E_{dc}) \quad (2.19)$$

2.3.3 วงจรขั้วมอเตอร์

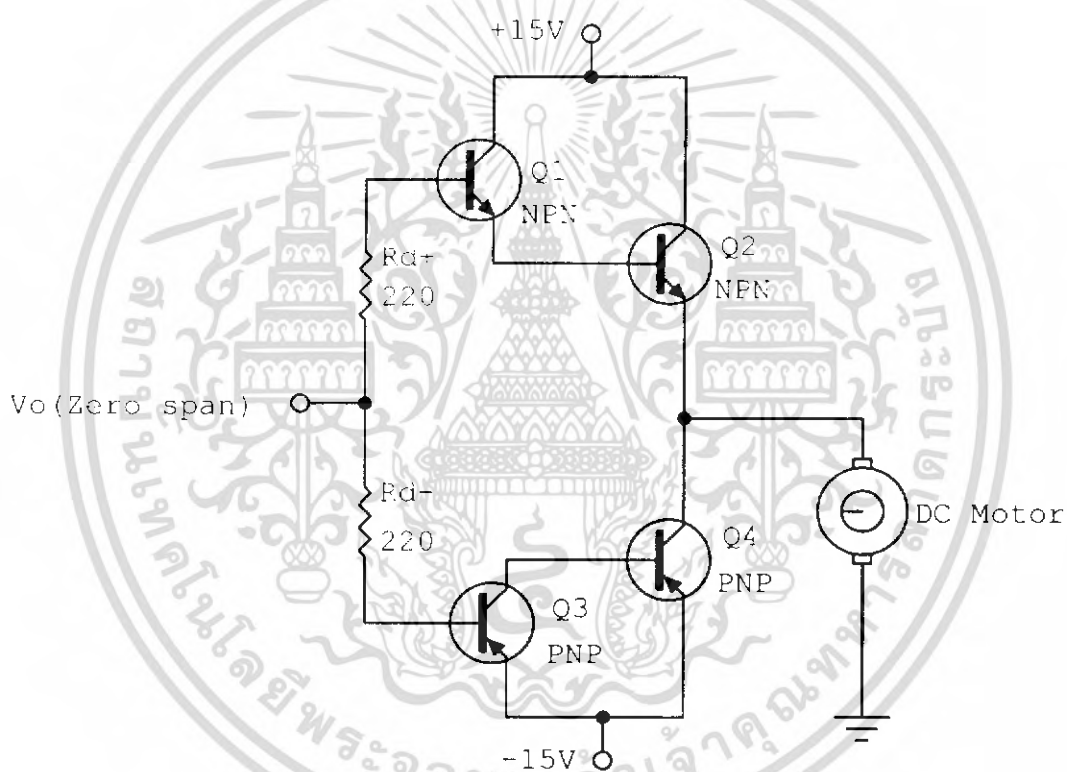
จากหัวข้อที่ 2.3.2 พบว่ากระแสเอาต์พุตที่ออกมาจากวงจรซีโรสเปนจะมีค่าน้อย ไม่เพียงพอที่จะขั้วมอเตอร์โดยตรงได้ เนื่องจากมอเตอร์ต้องการกระแสมาก จึงจำเป็นต้องมีวงจรที่ใช้ขยายแรงดันและกระแสเพื่อให้สามารถขั้วมอเตอร์ได้ โดยโครงงานนี้ใช้วงจรขยายแรงดันและกระแส โดยมีพื้นฐานมาจากวงจรดาแล็งคัตตัน(Darlington) แบบใช้ทรานซิสเตอร์(Transistor) ซึ่งมีวงจรดังแสดงในรูปที่ 2.18

เมื่อ Q1 และ Q2 เป็นทรานซิสเตอร์ชนิดเอ็นพีเอ็น(NPN) เบอร์ TIP31C ส่วน Q3 และ Q4 เป็นทรานซิสเตอร์ชนิดพีเอ็นพี(PNP) เบอร์ TIP32C มีรายละเอียดในภาคผนวก ข-3

หลักการของวงจรดาแล็งคัตตัน คือการนำทรานซิสเตอร์ ทั้งชนิดพีเอ็นพี และ เอ็นพีเอ็นมาต่อร่วมกันเพื่อช่วยกันทำงานในกรณีที่สัญญาณอินพุตมีทั้งซีกบวกและซีกลบดังนี้

ถ้าอินพุตเป็นแรงดันไฟบวกทรานซิสเตอร์ Q1 จะทำงาน ทำให้ Q2 ทำงานโดยได้รับไบแอสตรงจากกระแสมีตเตอร์ของทรานซิสเตอร์ Q1 ซึ่งจะทำให้มีแรงดันไฟบวกออกทางเอาต์พุต โดยมีค่ามากที่สุดไม่เกินแหล่งจ่ายไฟบวก ส่วนทรานซิสเตอร์ Q3 และ Q4 เป็นทรานซิสเตอร์ชนิดพีเอ็นทีจะไม่ทำงานเพราะไม่ได้รับการไบแอส

ถ้าอินพุตเป็นลบ ทรานซิสเตอร์ Q3 จะทำงาน ทำให้ Q4 ทำงานโดยได้รับไบแอสตรงจากกระแสมีตเตอร์ของ Q3 ซึ่งจะทำให้มีแรงดันไฟลบออกทางเอาต์พุต โดยมีค่ามากที่สุดไม่เกินแหล่งจ่ายไฟลบ คุณสมบัติของวงจรถือค่าความต้านทานอินพุตสูงมากและค่าความต้านทานเอาต์พุตต่ำมาก ทำให้วงจรนี้ไม่ดึงกระแสทางด้านอินพุต และเอาต์พุตสามารถจ่ายกระแสได้สูงด้วย โดยวงจรนี้มีอัตราขยายแรงดัน $\frac{V_{out}}{V_{in}} \cong 1$



รูปที่ 2.18 วงจรขับมอเตอร์

2.4 พอร์ตขนาน

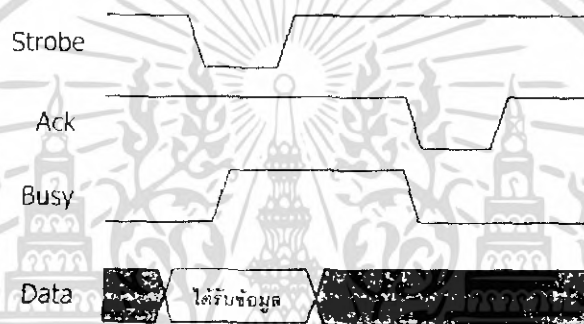
การควบคุมระบบควบคุมตำแหน่งลูกบอลบนคานนั้นเป็นระบบควบคุมแบบอัตโนมัติ โดยจะมีการส่งสัญญาณควบคุมออกมาจากคอมพิวเตอร์ โครงการนี้ใช้วิธีการส่งข้อมูลโดยใช้พอร์ตขนานเนื่องจากง่ายต่อการใช้งาน โดยสามารถที่จะส่งข้อมูลขนาด 8 บิต ออกมาจากคอมพิวเตอร์ได้พร้อมกัน ทำให้สามารถนำสัญญาณ ไปใช้งานกับวงจรควบคุมและภาคขับมอเตอร์ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 ลักษณะทางกายภาพของพอร์ตขนาน

เพื่อให้เข้าใจถึงการนำเอาพอร์ตขนานไปใช้งาน ในเบื้องต้นควรทำความเข้าใจก่อนว่าโดยปกติการส่งพื้งานจากคอมพิวเตอร์ไปยังพอร์ตขนานนั้นมีรูปแบบการทำงานภายในอย่างไร

ในรูปที่ 2.19 แสดงไคอะแกรมเวลาของการติดต่อระหว่างพอร์ตขนานกับเครื่องพิมพ์ จะเห็นได้ว่าสัญญาณที่นำไปใช้งานจริงๆ มีไม่มาก เริ่มจากสัญญาณพอร์ตข้อมูล(Data Port) ถูกส่งออกไปยังเครื่องพิมพ์พร้อมทั้งส่งสัญญาณสโตรบ(Strobe) ออกไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่าการส่งข้อมูลใหม่มาที่ขาข้อมูล จากนั้นคอมพิวเตอร์จะต้องรอการตอบกลับจากเครื่องพิมพ์ นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ Busy เพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่องพิมพ์พร้อม เครื่องพิมพ์จะสร้างสัญญาณ ACK(Acknowledge) ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว



รูปที่ 2.19 ไคอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์

สัญญาณข้อมูลขนาด 8 บิต สัญญาณสโตรบ และสัญญาณ ACK เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้ว การติดต่อกับเครื่องพิมพ์โดยมากยังต้องมีสัญญาณอื่นๆ ร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ถึงสามอย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์ พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนรูปแบบตัวพิมพ์(Font) เป็นต้น บางครั้งอาจเกิดเหตุการณ์ที่ไม่ปกติ เช่น บัฟเฟอร์สำหรับรับข้อมูลเต็ม เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกลย่อมทำงานได้ช้ากว่าการส่งข้อมูลของคอมพิวเตอร์ ในกรณีนี้เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลได้มากกว่านี้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์คือสัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่น กระดาษติด เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์ในกรณีนี้เรียกว่าสัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซ็ตเครื่องพิมพ์ คอมพิวเตอร์จะต้องส่งสัญญาณไปยังเครื่องพิมพ์เพื่อรีเซ็ตเครื่องพิมพ์ด้วย สามารถสรุปหาสัญญาณที่จำเป็นสำหรับการติดต่อดังในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 สัญญาณสำคัญของพอร์ตนานที่ใช้ติดต่อกับเครื่องพิมพ์

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซ็ตเครื่องพิมพ์	คอมพิวเตอร์

ตารางที่ 2.2 สัญญาณทั้งหมดของพอร์ตนาน

DR-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ
1	Control	Out	\overline{CO}	\overline{STROBE}
2-9	Data	Out	D0-D7	DATA1 – DATA5
10	Status	In	S6	ACK
11	Status	In	$\overline{S7}$	\overline{BUSY}
12	Status	In	S5	PE
13	Status	In	S4	SELECT
14	Control	Out	$\overline{C1}$	$\overline{AUTO FEED}$
15	Status	In	S3	\overline{ERROR}
16	Control	Out	C2	\overline{INIT}
17	Control	Out	$\overline{C3}$	$\overline{SELECT - IN}$
18-25			GND	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2.1 จะเห็นว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีกสามพอร์ตได้แก่

1. พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์
2. พอร์ตเอาต์พุตสำหรับสัญญาณสโตรบและรีเซต และ
3. พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ ACK, Busy และ Error จากเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น โดยสายสัญญาณเหล่านั้นจะมีรีจิสเตอร์ควบคุมการทำงานดังนี้

1. พอร์ตเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ข้อมูล(Data Register) ควบคุม
2. พอร์ตอินพุตสำหรับการอ่านค่าสถานะต่างๆ จากภายนอกมี 5 เส้น ใช้รีจิสเตอร์สถานะ (Status Register) ในการควบคุม
3. พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอกมี 4 เส้น ใช้รีจิสเตอร์ควบคุม(Control Register) ในการควบคุม

บล็อกไดอะแกรมในรูปที่ 2.20 แสดงระบบบัสของคอมพิวเตอร์สำหรับการติดต่อกับพอร์ตขนาน สัญญาณเอาต์พุตจากพอร์ตขนานจะถูกส่งไปยังคอนเน็คเตอร์แบบ DB-25 สำหรับคอมพิวเตอร์ส่วนใหญ่ในปัจจุบัน พอร์ตขนานจะมีมาพร้อมกับเมนบอร์ดไม่จำเป็นต้องใช้การ์ดเสียบเพิ่มเติมเหมือนในอดีต พร้อมทั้งมีฟังก์ชันการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ตขนานในรูปแบบมาตรฐาน(SPP) อยู่

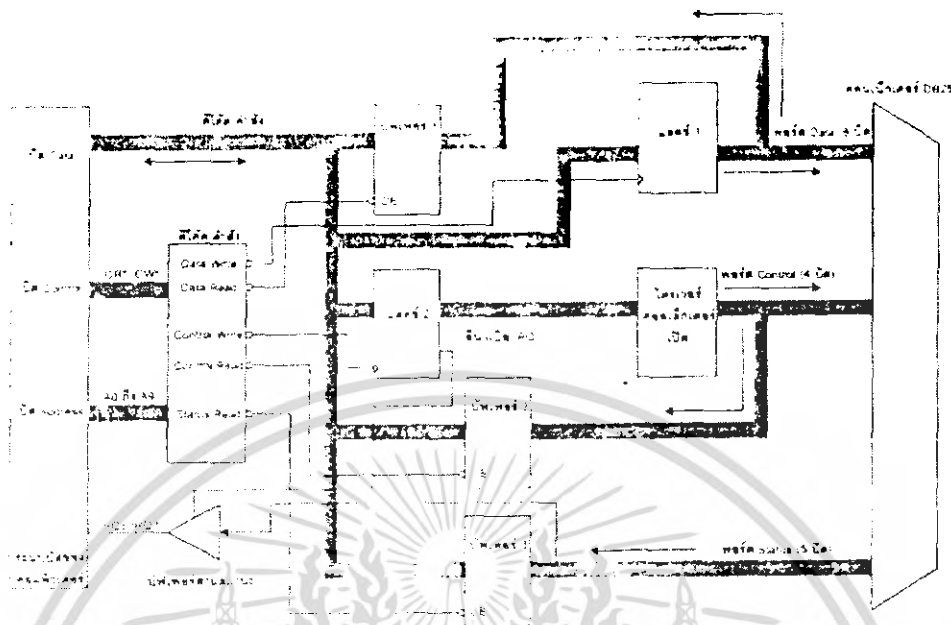
การติดต่อกับพอร์ตขนานจะต้องมีการอ้างแอดเดรสตำแหน่ง โดยแอดเดรสที่ใช้อ้างถึงจะเป็นตำแหน่ง A0-A9 และใช้ขา IOR และ IOW สำหรับเป็นตัวเลือก ว่าต้องการอ่าน หรือเขียนรีจิสเตอร์ตัวใด จากการถอดรหัสแอดเดรส A0-A9 ทำให้ได้สัญญาณออกมาเพื่อไปควบคุมหรืออินทิเกรตวงจรเฟิร์มแวร์ต่างๆ ดังนี้

Data Write คือสัญญาณอินทิเกรตสำหรับนำข้อมูลที่อยู่ในบัสข้อมูล(Data Bus) ไปออกที่ขาข้อมูล ของพอร์ตขนานส่วน Data Read คือสัญญาณอินทิเกรตสำหรับอ่านข้อมูลจากขาข้อมูล ของพอร์ตขนานมาเก็บไว้ในบัสข้อมูล

Control Write คือสัญญาณอินทิเกรตสำหรับนำข้อมูลที่อยู่ในบัสข้อมูลส่งไปออกที่ขาควบคุมของพอร์ตขนาน สำหรับพอร์ตนี้ นอกจากจะส่งข้อมูลไปยังพอร์ตขนานแล้ว ยังทำหน้าที่อินทิเกรตการอินเทอร์รัพต์ของการเปลี่ยนแปลงสัญญาณที่พอร์ตสถานะ(Status Port) อีกด้วย

Control Read คือสัญญาณอินทิเกรตสำหรับอ่านค่าข้อมูลจากขาควบคุมมาเก็บไว้ในบัสข้อมูล

Status Read คือสัญญาณอินทิเกรตสำหรับอ่านค่าข้อมูลจากขาพอร์ตสถานะ มาเก็บไว้ในบัสข้อมูลแอดเดรสของพอร์ตขนานถูกกำหนดไว้ 3 ตำแหน่ง คือ LPT1, LPT2 และ LPT3



รูปที่ 2.20 ระบบบัสภายในของพอร์ตขนาน

2.4.2 พอร์ตข้อมูล

จากรูปที่ 2.21 แสดงให้เห็นว่าพอร์ตข้อมูล ประกอบไปด้วยบัฟเฟอร์ 1 ตัวและไอซีแลตซ์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตซ์ตัวที่หนึ่งทั้ง 8 บิต เอาต์พุตของไอซีแลตซ์ตัวที่หนึ่งคือ D0-D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึงขา 9 และที่ขาเอาต์พุตนี้สัญญาณข้อมูล จะส่งกลับไปเป็นอินพุตของบัฟเฟอร์ตัวที่หนึ่งด้วย ทำให้คอมพิวเตอร์สามารถอ่านค่าสถานะปัจจุบันที่เกิดขึ้นกับพอร์ตข้อมูลได้

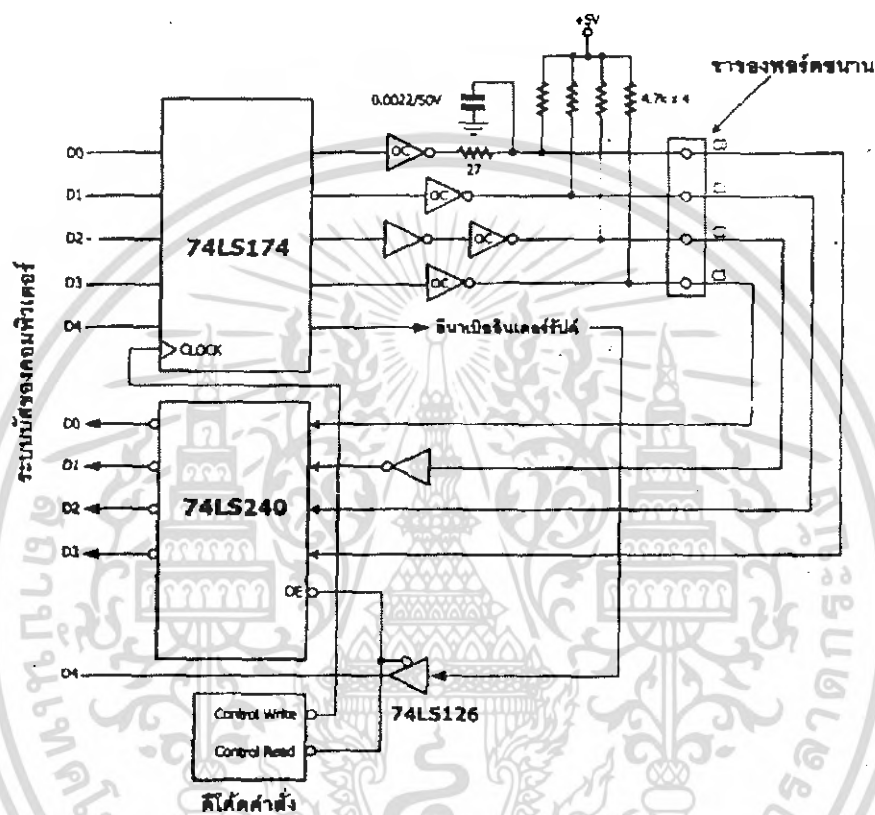
เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมาจากบัสข้อมูลของคอมพิวเตอร์ผ่าน ไปให้กับไอซี 74LS374 ซึ่งเป็นไอซีแลตซ์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุตคอมพิวเตอร์จะส่งสัญญาณ Data Write ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกกรองด้วยวงจร RC ซึ่งประกอบด้วยตัวต้านทานค่า 27 โอห์ม และตัวเก็บประจุ 0.0022 ไมโครฟารัด เพื่อให้ช่วงเวลาเปลี่ยนแปลงจาก ลอจิก "0" เป็นลอจิก "1" หรือจากลอจิก "1" เป็นลอจิก "0" เป็นไปอย่างช้าๆ เนื่องจากการเปลี่ยนแปลงแรงดันที่รวดเร็วอาจจะทำให้เกิดสัญญาณรบกวนเหนี่ยวนำข้ามไปยังข้อมูลบิตอื่นๆ ได้ ทำให้ข้อมูลที่ส่งออกไปเกิดการผิดพลาด โดยค่าตัวต้านทานและตัวเก็บประจุในวงจรทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที

จากวงจรในรูปที่ 2.21 จะได้เอาต์พุตของพอร์ตข้อมูล ซึ่งมีคุณสมบัติดังนี้

- กระแสซิงค์สูงสุด 24 mA

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเอาต์พุตของ 74LS240 มีอินเวอร์เตอร์อยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่านและเขียนข้อมูลกับพอร์ตควบคุม คอมพิวเตอร์จะส่งข้อมูลมาที่ขา Control Write และ Control Read เนื่องจากเอาต์พุตของพอร์ตควบคุม เป็นแบบคอลเล็กเตอร์เปิด ดังนั้นผู้ใช้งานสามารถใช้พอร์ตนี้ในการอ่านค่าสัญญาณอินพุตจากภายนอกได้ โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาของพอร์ตที่ต้องการอ่านค่าเป็นลอจิก “1” เสียก่อน

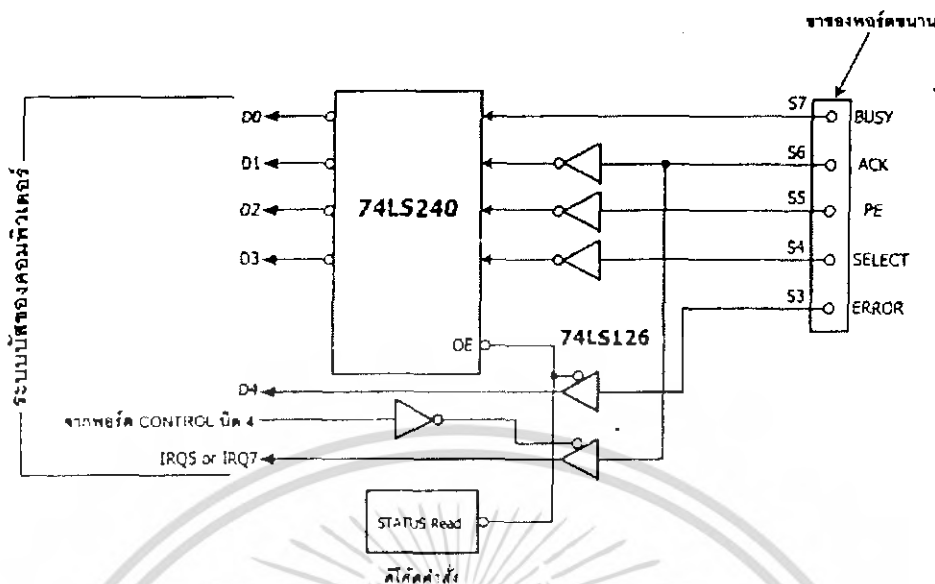


รูปที่ 2.22 วงจรภายในของพอร์ตควบคุม

2.4.4 พอร์ตสถานะ

พอร์ตสถานะเป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะจากเครื่องพิมพ์ รูปที่ 2.23 แสดงรายละเอียดภายในของพอร์ตสถานะ จะสังเกตเห็นว่ามีขาสัญญาณอยู่ทั้งหมดห้าสัญญาณด้วยกันและจะเรียกชื่อเป็น S3, S4, S5, S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจิสเตอร์สถานะนั่นเอง สำหรับ S7 จะมีชื่อแตกต่างจากบิตอื่นๆ ที่เมื่อสัญญาณจากภายนอกส่งเข้ามาแล้วจะไม่ผ่านอินเวอร์เตอร์ ในขณะที่ขาอื่นๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้นเมื่อข้อมูลจากขาอินพุตไปยัง 74LS240 ซึ่งเอาต์พุตมีการกลับสถานะทำให้บิต S⁻ เป็นบิตเดียวที่มีการกลับสถานะ นอกจากนี้ในการใช้งานถ้าต้องการให้มีการสร้างสัญญาณอินเวอร์รต์จากขาขาขึ้นของขา S6 สามารถกำหนดค่าได้จากพอร์ตควบคุม บิตที่ 4

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้กับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 วงจรภายในของพอร์ตแสดงสถานะ

2.5 ตัวควบคุม

ตัวควบคุม(Controller)เป็นส่วนประกอบหนึ่งในระบบควบคุมแบบป้อนกลับ ซึ่งมีสัญญาณความผิดพลาดเป็นอินพุตและมีเอาต์พุตเป็นอินพุตของอุปกรณ์ขับเคลื่อนสำหรับพลานท์(Plant) ที่ต้องการควบคุม ส่วนนี้จะกล่าวถึงวิธีการเลือกใช้ตัวควบคุมที่เหมาะสมกับระบบควบคุมแบบป้อนกลับและวิธีการหาพารามิเตอร์ที่เหมาะสมสำหรับตัวควบคุม โดยที่ความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของตัวควบคุม นิยมเรียก กฎการควบคุม(Control Law) โดยที่นิยมมี 3 รูปแบบ คือ

1. ระบบควบคุมแบบพี(P: Proportional Control)
2. ระบบควบคุมแบบไอ(I: Integral Control)
3. ระบบควบคุมแบบดี(D: Derivative Control)

ในระบบบางประเภท มีความจำเป็นที่ต้องปรับปรุงสมรรถนะของระบบควบคุม ซึ่งสามารถทำได้โดยการใช้ส่วนประกอบอื่นเพิ่มเติมเข้ากับระบบควบคุม การที่ปรับเปลี่ยนสมรรถนะของระบบควบคุมป้อนกลับแบบนี้เรียกว่า การชดเชย(Compensation)

2.5.1 ระบบควบคุมแบบพี

ในระบบควบคุมแบบพี(P: Proportional Control) เอาต์พุตของตัวควบคุม $u(t)$ จะเป็นสัดส่วนกับอินพุตของตัวควบคุม และถ้ากำหนดสัญญาณอินพุตให้กับตัวควบคุมเป็นค่าความผิดพลาด $e(t)$ ซึ่งเป็นฟังก์ชันของเวลา จะได้

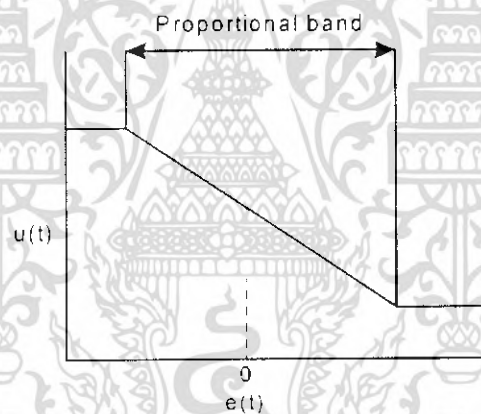
$$u(t) = K_p e(t) \tag{2.20}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ใด ๆ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ K_p เป็นค่าคงที่เรียกว่า อัตราขยายแบบสัดส่วน(Proportional Gain) จะพบว่าเอาต์พุตที่ออกจากตัวควบคุมแบบพี เป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดในขณะที่กำลังพิจารณา ทำให้ฟังก์ชันถ่ายโอนของตัวควบคุม $G_c(s)$ มีค่าเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \quad (2.21)$$

ดังนั้นการควบคุมด้วยตัวควบคุมแบบนี้จะเป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้น การที่ได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง ทำให้เกิดเอาต์พุตที่มีขนาดใหญ่จากตัวควบคุมในเวลานั้น อย่างไรก็ตามการที่ให้อัตราขยายคงที่นั้นในทางปฏิบัติอาจกำหนดไว้ในบางช่วงของสัญญาณความผิดพลาดเท่านั้น หรืออาจกำหนดให้ตัวควบคุมมีค่าเอาต์พุตไม่น้อยกว่าค่าค่าหนึ่งและไม่มากเกินไปกว่าค่าค่าหนึ่งก็ได้ ซึ่งการกำหนดช่วงจำกัดของเอาต์พุตจะมีลักษณะดังรูปที่ 2.24 และการกำหนดเอาต์พุตของตัวควบคุมแบบพี ช่วงที่มีการกำหนดสัดส่วนนี้ จะเรียกว่า Proportional Band



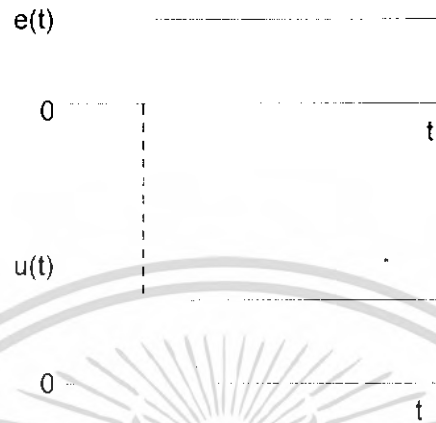
รูปที่ 2.24 การกำหนดช่วงจำกัดของเอาต์พุต

การกำหนด Proportional Band นี้ จะช่วยให้สัญญาณเอาต์พุตมีค่าจำกัดไม่ไปสู่ออนันต์ ทั้งทางด้านบวกและทางด้านลบ และเมื่อตัวควบคุมมีเอาต์พุตสูงที่สุดที่เป็นไปได้ค่าหนึ่ง ก็นิยมที่จะกำหนดเอาต์พุตค่าใดๆ เป็นร้อยละของค่าสูงสุดที่เป็นไปได้ ดังนั้นการเปลี่ยนแปลงค่าเอาต์พุตของตัวควบคุม 100% ก็หมายถึงว่าเอาต์พุตจะเปลี่ยนจากค่าต่ำสุดที่เป็นไปได้ ไปเป็นค่าสูงสุดที่เป็นไปได้ ซึ่งจะได้ว่า

$$K_p = \frac{100}{\text{Proportional Band}} \quad (2.22)$$

เนื่องจากเอาต์พุตของตัวควบคุมจะเป็นสัดส่วนกับอินพุต ดังนั้น ถ้าหากอินพุตมีลักษณะเป็นสัญญาณระดับ(Step) เอาต์พุตที่ได้ก็จะมีลักษณะเป็นสัญญาณระดับเช่นกัน โดยลักษณะของกราฟค่าไม่วากรณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงอินพุตและเอาต์พุตจะมีสัดส่วนที่แน่นอนค่าหนึ่ง แสดงดังรูปที่ 2.25 โดยรูปนี้แสดงถึงการตอบสนองของตัวควบคุม เมื่ออินพุตอยู่ในช่วง Proportional Band



รูปที่ 2.25 ผลการตอบสนองของตัวควบคุมแบบพี

ในทางปฏิบัติตัวควบคุมแบบพีนี้มีลักษณะเหมือนกับเครื่องขยายสัญญาณรูปแบบหนึ่ง ซึ่งอาจจะเป็นในลักษณะของอุปกรณ์ไฟฟ้า หรืออาจจะเป็นเครื่องขยายสัญญาณเชิงกล เช่น กาน ก็ได้ ลักษณะของระบบที่มีตัวควบคุมแบบพี ได้ตั้งบล็อกโคแอสเทรรมแสดงในรูปที่ 2.26 และได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) = K_p G_p(s) \quad (2.23)$$

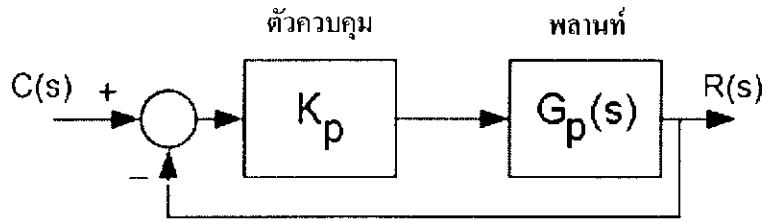
เมื่อ $G_p(s)$ เป็นฟังก์ชันถ่ายโอนของพลานท์ที่ต้องการควบคุม

ข้อเสียของระบบที่มีตัวควบคุมแบบพี คือไม่มีการเพิ่มอินทิเกรตหรือ $1/s$ ในส่วน Forward-Path ซึ่งหมายความว่า ถ้าพลานท์เป็นชนิด (Type) 0 ตัวควบคุมจะไม่เปลี่ยนแปลงชนิด ทำให้ระบบเป็นชนิด 0 เหมือนเดิม ซึ่งจะทำให้เกิดความผิดพลาดที่สถานะคงตัว (Steady State Error) จากรูปที่ 2.26 ระบบควบคุมแบบพี ซึ่งมีการป้อนกลับหนึ่งหน่วย มีฟังก์ชันถ่ายโอนของระบบเป็น

$$G(s) = \frac{K_p G_p(s)}{1 + K_p G_p(s)} \quad (2.24)$$

สมการคุณลักษณะเป็น $(1 + K_p G_p(s))$ และมีรากเปลี่ยนไปตามค่าของ K_p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 บล็อกไคอะแกรมของระบบควบคุมแบบพี

2.5.2 ระบบควบคุมแบบไอ

ในระบบควบคุมแบบไอ (I: Integral Control) เอาท์พุทของตัวควบคุมจะเป็นสัดส่วนกับการอินทิเกรตสัญญาณของค่าผิดพลาดเทียบกับเวลา หรือ

$$u(t) = K_i \int_0^t e(t) dt \quad (2.25)$$

เมื่อ K_i เป็นค่าคงที่ที่เรียกว่า อัตราขยายแบบอินทิเกรต (Integral Gain) ซึ่งจะมีหน่วยเป็น sec^{-1} รูปที่ 2.27 แสดงลักษณะการตอบสนองของตัวควบคุมแบบไอ เมื่อได้รับสัญญาณอินพุทแบบระดับ ค่า $u(t)$ ณ เวลา t ใดๆ หมายถึงพื้นที่ใต้กราฟของสัญญาณอินพุทจากเวลา 0 ถึง t คูณด้วยอัตราขยายอินทิเกรต ดังนั้นเนื่องจากระบบเริ่มการมีสัญญาณอินพุทแบบระดับ เอาท์พุทที่ออกจากตัวควบคุมจึงมีค่ามากขึ้นเรื่อยๆ ด้วยอัตราที่คงที่ในลักษณะรูปสัญญาณความชัน (Ramp)

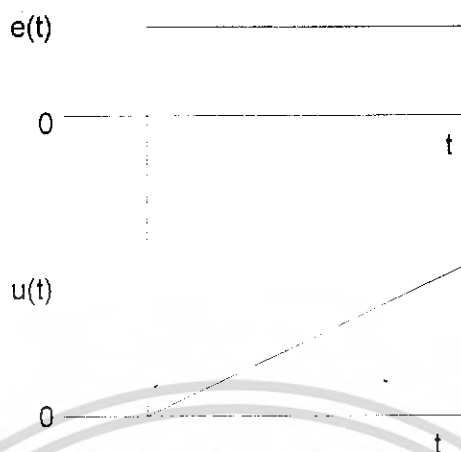
จากสมการที่ (2.25) ได้ฟังก์ชันถ่ายโอนของตัวควบคุมแบบไอเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.26)$$

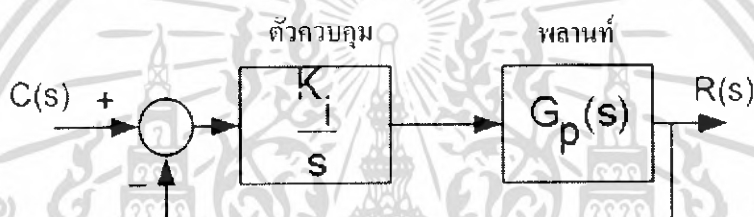
ทำให้ระบบควบคุมแบบไอที่แสดงในรูปที่ 2.28 มีฟังก์ชันถ่ายโอน Forward-Path เป็น $\frac{K_i}{s} G_p(s)$ และมีฟังก์ชันถ่ายโอนระบบปิดเป็น

$$G(s) = \frac{\frac{K_i}{s} G_p(s)}{1 + \frac{K_i}{s} G_p(s)} \quad (2.27)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 ผลการตอบสนองของตัวควบคุมแบบไอ



รูปที่ 2.28 บล็อกไดอะแกรมของระบบควบคุมแบบไอ

ดังนั้นสามารถพิจารณาถึงข้อได้เปรียบของการควบคุมแบบไอ ได้จากฟังก์ชันถ่ายโอน Forward-Path ซึ่งจะเห็นว่าระบบควบคุมแบบไอ จะเพิ่มชนิดของระบบขึ้นมา 1 ระดับ เช่น หากพลานท์เดิมเป็นชนิด 0 ระบบควบคุมแบบไอจะเป็น ชนิด 1 ซึ่งทำให้ระบบมีความผิดพลาดที่สถานะคงตัวเป็นศูนย์เทียบกับสัญญาณอินพุตแบบระดับ อย่างไรก็ตามการเพิ่มโพลที่ $s=0$ ในระบบวงเปิดและไม่มีการเพิ่มซีโรให้กับระบบควบคุม ทำให้ความแตกต่างระหว่างจำนวนโพล (n) และจำนวนซีโร (m) เพิ่มขึ้นอีก 1 ซึ่งมีผลให้มุมของเส้นอะซิมโทต (Asymptote Angles) ของเส้นทางรากลดลง และจุดตัดเคลื่อนไปทางครึ่งขวาของระนาบ s (s -Plane) มากขึ้น มีผลทำให้ความเสถียรสัมพัทธ์ของระบบลดลง

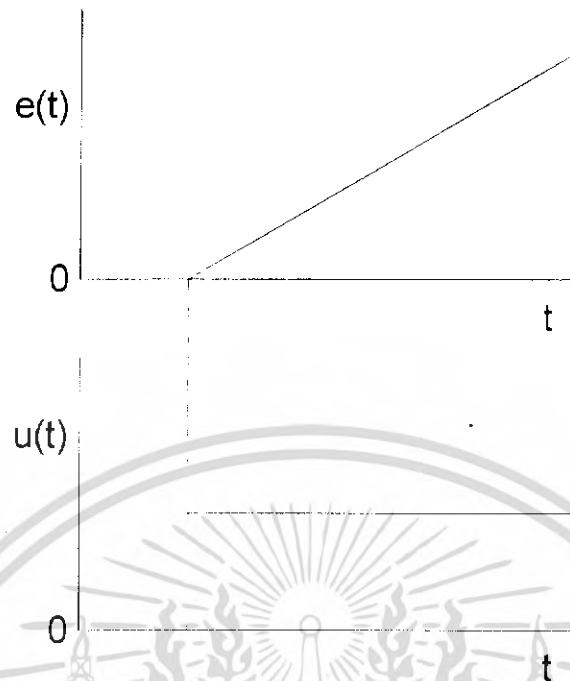
2.5.3 ระบบควบคุมแบบดี

ระบบควบคุมแบบดี (D: Derivative Control) นี้ เอาท์พุทของตัวควบคุมจะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงอินพุตความผิดพลาดเทียบกับเวลา นั่นคือ

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.28)$$

เมื่อ K_d คือ อัตราขยายอนุพันธ์ (Derivative Gain) และมีหน่วยเป็นวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้ว่าห้ามเผยแพร่ในวงกว้างเพื่อวัตถุประสงค์อื่นใดโดยไม่ขออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 ผลตอบสนองของระบบควบคุมแบบค้ำ

รูปที่ 2.29 แสดงผลตอบสนองของระบบควบคุมแบบค้ำ เมื่อสัญญาณอินพุตเป็นสัญญาณแบบความชัน จากสมการที่ (2.28) จะเห็นว่าตัวควบคุมแบบค้ำจะให้สัญญาณส่งออกจกตัวควบคุมมีค่ามากเมื่อค่าอินพุตความผิดพลาดมีแนวโน้มเพิ่มในอัตราที่สูงก่อนที่จะเกิดความผิดพลาดขึ้นมากจริงๆ

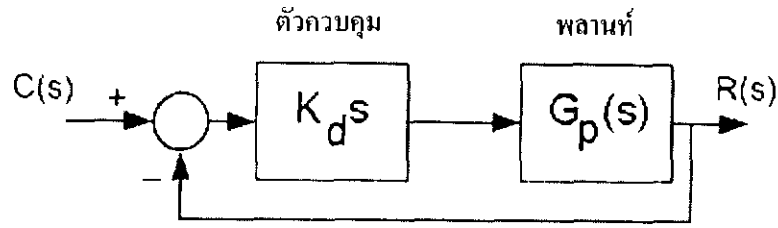
อย่างไรก็ตามหากความผิดพลาดมีค่าคงที่ก็จะไม่มีการสะสมค่าความผิดพลาดแม้ว่าค่าความผิดพลาดจะมีมากก็ตาม ทำให้การควบคุมแบบค้ำนี้ ไม่เปลี่ยนแปลงต่อค่าความผิดพลาดที่คงที่หรือเปลี่ยนแปลงอย่างช้าๆ ดังนั้นการควบคุมแบบค้ำจึงไม่นิยมใช้เพียงลำพัง แต่มักจะใช้ควบคู่ร่วมกับการควบคุมแบบอื่น ฟังก์ชันถ่ายโอนของตัวควบคุมแบบค้ำ จะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_d s \quad (2.29)$$

ดังนั้น สำหรับระบบควบคุมแบบค้ำจะได้ฟังก์ชันถ่ายโอนระบบปิดเป็น

$$G(s) = \frac{K_d s G_p(s)}{1 + K_d s G_p(s)} \quad (2.30)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 บล็อกไดอะแกรมของระบบควบคุมแบบดี

ถ้าหากว่าระบบเป็นแบบชนิด 1 หรือสูงกว่า การควบคุมแบบดีจะลดเทอม s ในพหุนามส่วนของฟังก์ชันถ่ายโอนของ Forward-Path ลง ซึ่งจะไปลดชนิดของระบบลง 1 อย่างไรก็ตามดังที่ได้กล่าวก่อนหน้านี้นี้แล้วว่า การควบคุมแบบดีนี้จะไม่ใช่เพียงลำพังแต่จะใช้ร่วมกับการตัวควบคุมแบบอื่น เพราะเมื่อใช้การควบคุมแบบดี ทำให้เพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้น

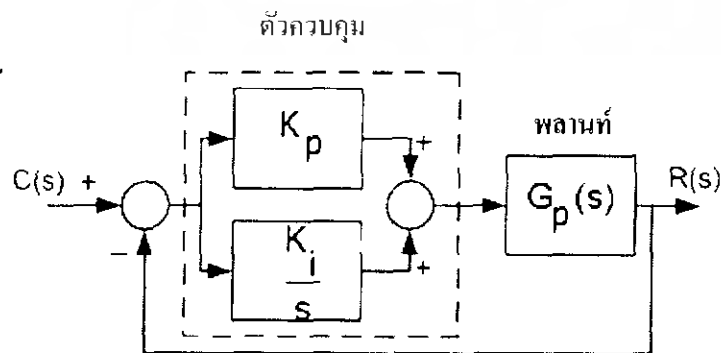
ในทางปฏิบัติการนำการควบคุมดีไปใช้นั้นค่อนข้างจะลำบาก ดังนั้น ในทางปฏิบัติโดยทั่วไปจะเป็นการประมาณการควบคุมแบบดีโดยใช้ตัวชดเชยแบบม้วนนำ(Lead Compensator)

2.5.4 ระบบควบคุมแบบพีไอ

การที่ระบบควบคุมมีความเสถียรสัมพัทธ์ลดลง เมื่อใช้การควบคุมแบบไอ สามารถที่จะแก้ไขได้ในระดับหนึ่งโดยการใช้การควบคุมแบบพีร่วมกับแบบไอ ซึ่งลักษณะของระบบควบคุมเป็นตามรูปที่ 2.31

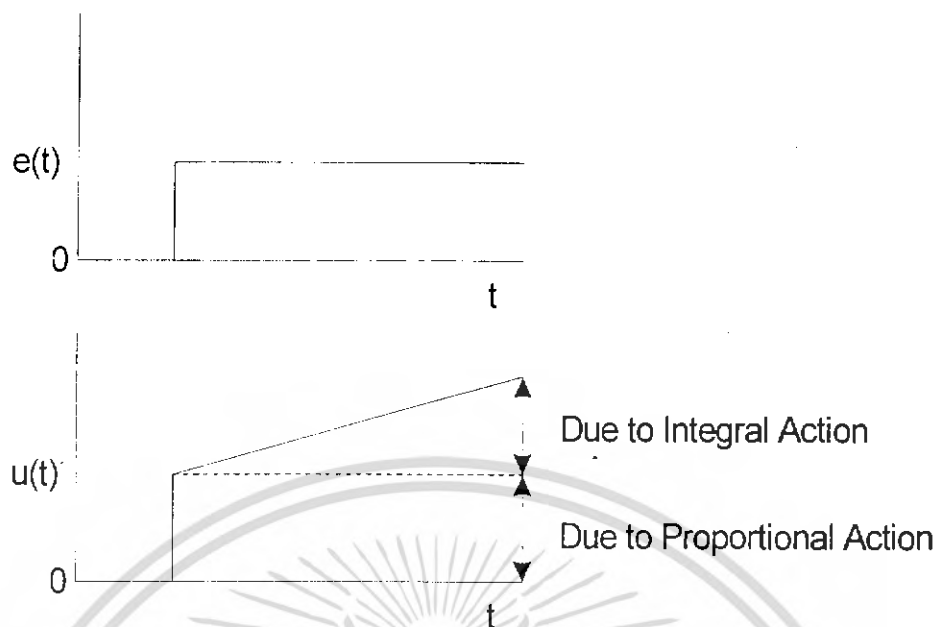
สำหรับระบบดังกล่าวจะมีเอาต์พุตของตัวควบคุมเป็น

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \tag{2.31}$$



รูปที่ 2.31 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการใช้งานเฉพาะเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 การตอบสนองของตัวควบคุมแบบพีไอ

รูปที่ 2.32 แสดงเอาต์พุตของตัวควบคุมที่ได้รับเมื่อมีอินพุตเป็นแบบระดับ จากสมการที่ (2.31) ฟังก์ชันถ่ายโอนของตัวควบคุมแบบพีไอ(PI: Proportional plus Integral Control) เป็น

$$\begin{aligned}
 G_c(s) &= \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \\
 &= \frac{sK_p + K_i}{s} \\
 &= K_p \frac{(s + K_i/K_p)}{s}
 \end{aligned}
 \tag{2.32}$$

นิยามค่าเวลากงตัวอินทิกรัล(Integral Time Constant) เป็น

$$\tau_i = \frac{K_p}{K_i}
 \tag{2.33}$$

ดังนั้นจะได้

$$G_c(s) = \frac{K_p [s + (1/\tau_i)]}{s}
 \tag{2.34}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะทำให้ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) = G_c(s)G_p(s) \quad (2.35)$$

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

จะเห็นว่ามิซีโรที่ $s = -1/\tau_i$ และโพลที่ $s = 0$ เพิ่มให้กับฟังก์ชันถ่ายโอนของระบบเมื่อใช้การควบคุมแบบพีไอ การที่เพิ่มตัวประกอบ s เข้ากับพหุนามส่วนของฟังก์ชันถ่ายโอนวงเปิดเป็นการเพิ่มชนิดของระบบขึ้นไป 1 จึงทำให้ระบบควบคุมแบบพีไอนี้ไม่มีความผิดพลาดที่สภาพคงตัวสำหรับอินพุตแบบระดับ นอกจากนั้นการที่เพิ่มซีโรให้กับระบบไปพร้อมๆ กัน ก็ทำให้ความแตกต่างระหว่างจำนวนโพล(n) และจำนวนซีโร(m) มีค่าคงที่ ดังนั้นมุมของเส้นอะซิมโทตสำหรับเส้นทางรากมีค่าคงเดิม แต่จุดตัดของเส้นอะซิมโทตบนแกนจริง(Centroid) จะเคลื่อนที่เข้าหาจุดกำเนิดมากขึ้น เนื่องจาก

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

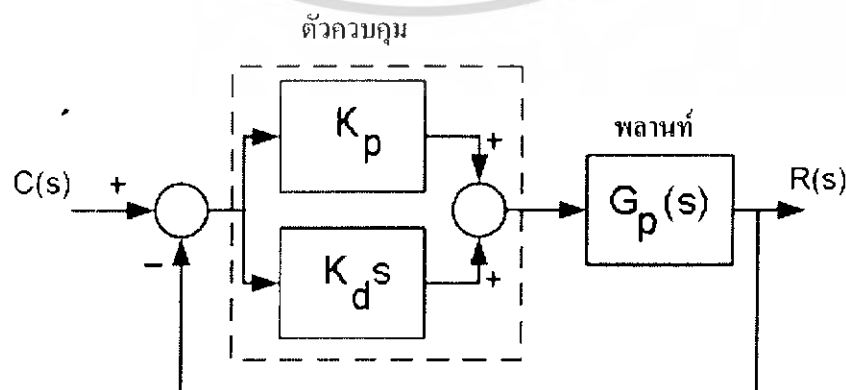
ดังนั้นจุดตัดของเส้นอะซิมโทตจะเปลี่ยนไปเท่ากับ $-(1/\tau_i)/(n-m)$ ส่งผลให้ประสิทธิภาพของระบบลดลง อย่างไรก็ตามการลดลงของความเสถียรสัมพัทธ์นี้จะน้อยกว่าการที่ใช้การควบคุมแบบไอเพียงอย่างเดียว

2.5.5 ระบบควบคุมแบบพีดี

ถ้าการควบคุมแบบดีใช้ร่วมกับการควบคุมแบบพี(PD: Proportional plus Derivative Control) ดังที่แสดงในรูปที่ 2.33 ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) = (K_p + K_d s)G_p(s) \quad (2.36)$$

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



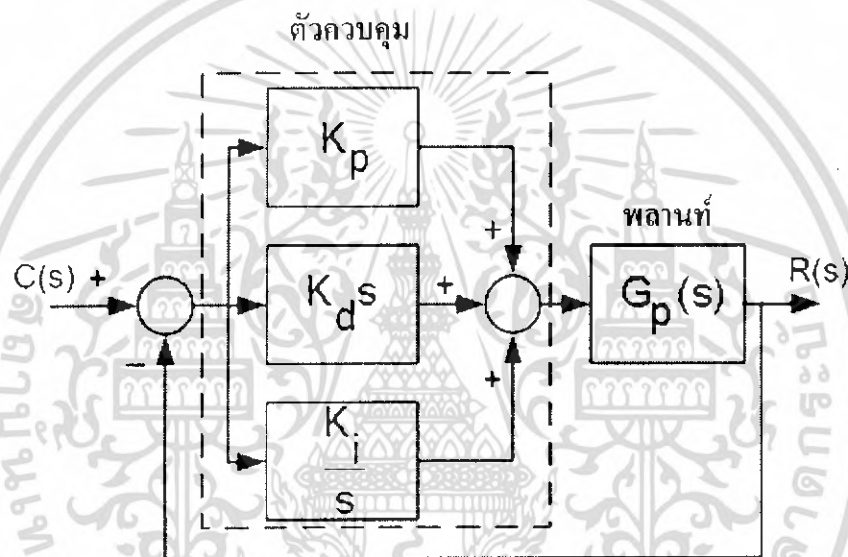
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.33 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2.5.6 ระบบควบคุมแบบพีไอดี

การควบคุมโดยใช้แบบพีรวมกับแบบไอและรวมกับแบบดี (PID: Proportional plus Integral plus Derivative Control) หรือที่เรียก การควบคุมแบบ 3 เทอม (Three-term Control) ระบบจะมีลักษณะตามรูปที่ 2.34 ทำให้เอาท์พุทของตัวควบคุมเมื่อรับอินพุตเป็นความผิดพลาด $e(t)$ ดังนี้



รูปที่ 2.34 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอดี

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.37)$$

ฟังก์ชันถ่ายโอนของตัวควบคุม จะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2.38)$$

ซึ่งสามารถจัดรูปได้เป็น

$$G_c(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2.39)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันถ่ายโอนระบบแบบเปิดของระบบควบคุมแบบพีไอดี ที่แสดงในรูปที่ 2.34 จะเป็น

$$G_o(s) = G_c(s)G_p(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s\right) G_p(s) \quad (2.40)$$

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

ดังนั้น การควบคุมแบบพีไอดี จะเพิ่มจำนวนซีโรให้กับระบบเท่ากับ 2 ตัวและเพิ่มจำนวนโพล 1 ตัว และทำให้ชนิดระบบเพิ่มขึ้น 1

2.5.7 ตัวควบคุมแบบดิจิตอล

ในการใช้งานจริง เนื่องจากระบบของเราเป็นระบบเวลาติดสกริต ดังนั้นตัวควบคุมที่ใช้จึงต้องเป็นแบบดิจิตอล(Digital Control) แสดงได้ดังสมการดังต่อไปนี้

ตัวควบคุมแบบพี

$$U_p[n] = K_p e[n] \quad (2.41)$$

ตัวควบคุมแบบไอ

$$U_i[n] = U_i[n-1] + K_i \frac{T_s}{2} (e[n] + e[n-1]) \quad (2.42)$$

ตัวควบคุมแบบดี

$$U_d[n] = K_d \frac{e[n] - e[n-1]}{T_s} \quad (2.43)$$

ตัวควบคุมแบบพีดี

$$U_{PD}[n] = U_p[n] + U_d[n] \quad (2.44)$$

ตัวควบคุมแบบพีไอ

$$U_{PI}[n] = U_p[n] + U_i[n] \quad (2.45)$$

ตัวควบคุมแบบพีไอดี

$$U_{PID}[n] = U_p[n] + U_i[n] + U_d[n] \quad (2.46)$$

เมื่อ T_s คือคาบเวลาการสุ่มสัญญาณ(Sampling Time)

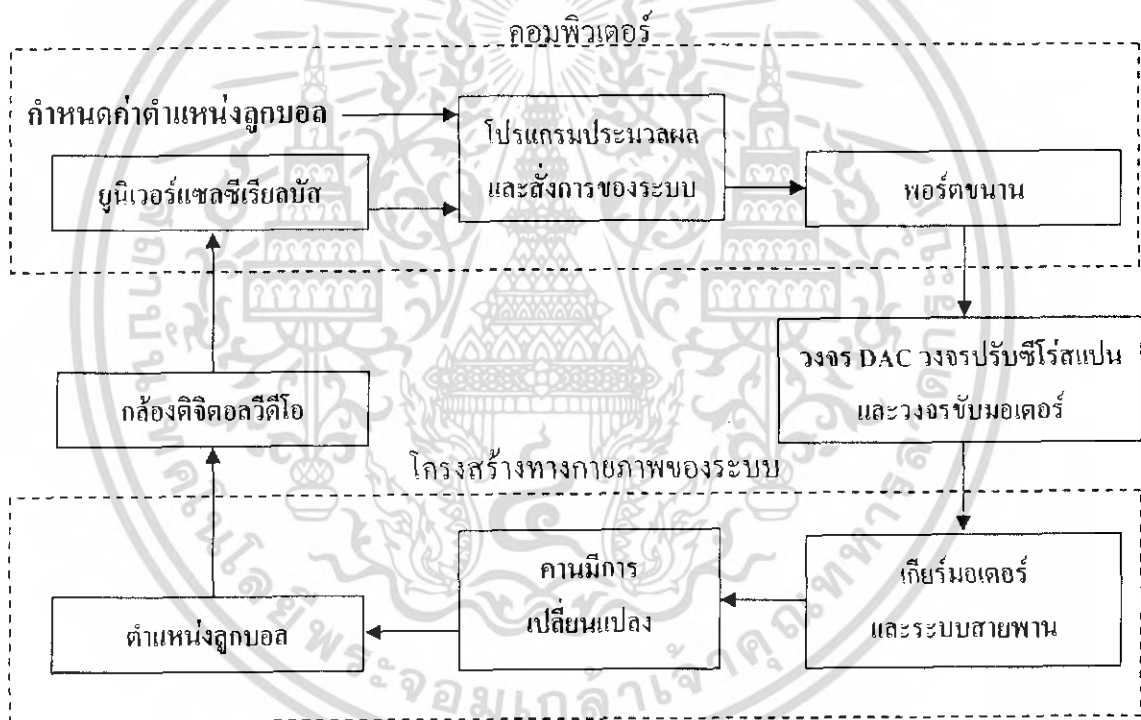
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการออกแบบ

การออกแบบระบบควบคุมตำแหน่งลูกบอลบนคาน ในเบื้องต้นนั้นต้องเข้าใจหลักการทำงานในภาพรวมของระบบควบคุมก่อนว่ามีหลักการทำงานอย่างไร แล้วจึงพิจารณาส่วนย่อยต่างๆ อันประกอบไปด้วย ส่วนที่เป็นโครงสร้างของระบบควบคุม ส่วนที่เป็นวงจรรีเลย์ทรอนิกส์ และ ส่วนของโปรแกรมที่ใช้ในการประมวลผลภาพและควบคุมระบบ

หลักการทำงานในภาพรวมของระบบควบคุมตำแหน่งลูกบอลบนคาน สามารถแสดงดังแผนผังในรูปที่ 3.1



รูปที่ 3.1 แผนผังรวมของระบบควบคุมตำแหน่งลูกบอลบนคาน

จากรูปที่ 3.1 เห็นว่าระบบควบคุมตำแหน่งลูกบอลบนคานประกอบด้วยส่วนหลักๆ ดังนี้คือ

- กล้องดิจิทัลวีดีโอ
- คอมพิวเตอร์
- โครงสร้างทางกายภาพของระบบ
- วงจรรีเลย์ทรอนิกส์ที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 กล้องดิจิทัลวิดีโอ

กล้องดิจิทัลวิดีโอเป็นอุปกรณ์ที่สำคัญสำหรับโครงการนี้ เนื่องจากเป็นอุปกรณ์ที่ใช้สำหรับตรวจสอบตำแหน่งของลูกบอลที่อยู่บนคาน เพื่อนำมาประมวลผลหาค่าสัญญาณควบคุมที่เหมาะสม กล้องดิจิทัลวิดีโอที่เลือกใช้ในโครงการนี้เป็นของโลจitech แสดงได้ดังรูปที่ 3.2

การเลือกใช้กล้องดิจิทัลวิดีโอ เพื่อที่จะนำมาใช้กับระบบควบคุมนั้น ต้องพิจารณาในเรื่องต่างๆ ดังนี้

- สิ่งจำเป็นสำหรับการใช้งานกล้องดิจิทัลวิดีโอ
- คุณสมบัติของกล้องดิจิทัลวิดีโอ



รูปที่ 3.2 กล้องดิจิทัลวิดีโอ

3.1.1 สิ่งจำเป็นสำหรับการใช้งานกล้องดิจิทัลวิดีโอ

- เครื่องไมโครคอมพิวเตอร์ที่ทำงานตั้งแต่สัญญาณนาฬิกา 400 เมกะเฮิร์ตขึ้นไป
- โปรแกรมระบบปฏิบัติการ ไมโครซอฟท์วินโดวส์มิลินเนียม (Window Millinuum)
- หน่วยความจำ 128 เมกะไบต์ หรือมากกว่า
- โปรแกรม TRIPOD และ QCSDK
- พอร์ตยูนิเวอร์แซลซีเรียลบัส
- หน่วยความจำบนฮาร์ดดิสก์อย่างน้อย 500 เมกะไบต์ สำหรับโปรแกรมที่เกี่ยวข้อง

3.1.2 คุณสมบัติของกล้องดิจิทัลวิดีโอ

- ให้ภาพที่มีความละเอียดสูงสุดถึง 640x 480 พิกเซล ในภาพสี 24 บิต
 - อัตราการส่งภาพด้วยความเร็วสูงสุดถึง 30 ภาพ ต่อวินาที
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ต้องการไฟเลี้ยงจากคอมพิวเตอร์ 2 วัตต์
- ปรับโฟกัสได้ตั้งแต่ 1 นิ้ว ถึงอนันต์

3.2 คอมพิวเตอร์

คอมพิวเตอร์เป็นอุปกรณ์ที่ทำหน้าที่รับข้อมูลจากกล้องดิจิทัลวิดีโอ ผ่านทางยูนิเวอร์แซลซีเรียลบัส โดยมีโปรแกรมซึ่งทำหน้าที่ประมวลผลภาพเพื่อหาตำแหน่งของลูกบอล แล้วจึงประมวลผลหาสัญญาณควบคุมที่เหมาะสม จากนั้นจึงส่งคำสั่งสัญญาณควบคุมที่คำนวณได้ผ่านทางพอร์ตขนานไปยังวงจรถอนิเล็กทรอนิกส์เพื่อควบคุมคันต่อไป

เนื่องจากการควบคุมในโครงการนี้ใช้คอมพิวเตอร์เป็นอุปกรณ์ประมวลผลหลัก ดังนั้นการเขียนโปรแกรมจึงเป็นการดำเนินการบนระบบเวลาคิสิกส์รีต แทนการควบคุมแบบระบบเวลาต่อเนื่อง ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องมีความรัดกุม และทำงานได้รวดเร็ว ซึ่งคอมพิวเตอร์ที่ใช้จะต้องมีความเร็วพอที่จะทำให้การสุ่มสัญญาณมีค่าน้อยเพียงพอ จนไม่มีผลต่อการควบคุมระบบทั้งหมดด้วย สำหรับโครงการนี้เลือกใช้การเขียนโปรแกรมโดยภาษาวิชวลซีพลัสพลัส



รูปที่ 3.3 โฟลว์ชาร์ตของ โปรแกรมควบคุมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

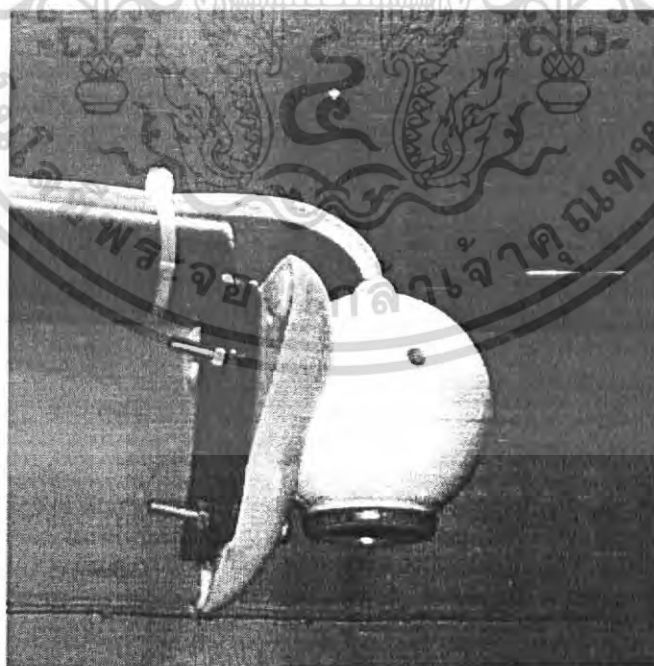
ในส่วนของโปรแกรมควบคุมระบบ เริ่มจากการเขียนโปรแกรมรับภาพจากกล้องดิจิทัลวีดีโอ และหาตำแหน่งของลูกบอลที่อยู่บนคาน จากนั้นเปรียบเทียบกับตำแหน่งของคานที่ต้องการให้ลูกบอลเคลื่อนที่ไป โปรแกรมส่วนควบคุมจะประมวลผลเพื่อหาสัญญาณควบคุมที่เหมาะสมซึ่งอาศัยตัวควบคุมแบบพีไอดี ดังที่กล่าวไว้ในบทที่ 2 แล้วจึงสั่งให้คอมพิวเตอร์ส่งค่าคำสั่งสัญญาณควบคุมออกทางพอร์ตขนาน เพื่อควบคุมคานให้เลี้ยงลูกบอลไปยังตำแหน่งที่ต้องการให้เร็วและแม่นยำตามต้องการ สามารถเขียนโฟลว์ชาร์ต(Flowchart) แสดงการทำงานของโปรแกรมได้ดังรูปที่ 3.3

3.3 โครงสร้างทางกายภาพของระบบ

โครงสร้างทางกายภาพของระบบควบคุมตำแหน่งลูกบอลบนคานประกอบด้วยส่วนต่างๆ หลายส่วน โดยหัวข้อนี้จะอธิบายถึงการออกแบบในแต่ละส่วน ดังนี้

3.3.1 การยึดกล้องดิจิทัลวีดีโอ

การติดตั้งกล้องดิจิทัลวีดีโอต้องมีการพิจารณาถึงความเหมาะสมในการออกแบบตัวยึดกล้อง ให้มีความมั่นคง สามารถรับน้ำหนักของกล้องดิจิทัลวีดีโอได้ ฐานยึดกล้องสำหรับโครงการนี้เลือกใช้ท่อพีวีซี เพราะสามารถตัดให้โค้งงอได้ตามความต้องการและมีน้ำหนักเบา ทำให้สะดวกต่อการเคลื่อนย้ายและปรับตั้ง การยึดกล้องดิจิทัลวีดีโอแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 การยึดกล้องดิจิทัลวีดีโอ

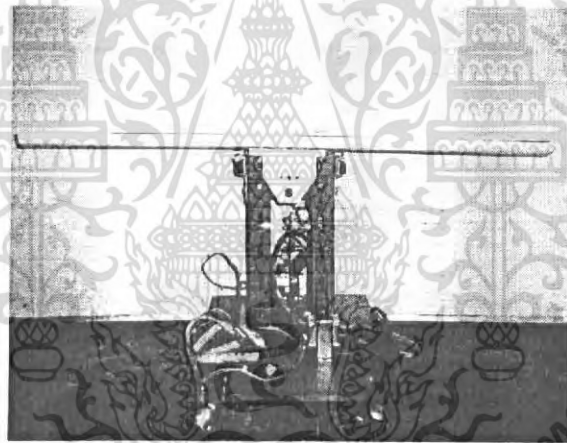
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.4 จะเห็นว่ากล่องถูกติดตั้งในลักษณะที่มีการตรวจจับด้านบนของคาน เนื่องจากจะสะดวกในการเขียนโปรแกรมมากกว่าติดตั้งกล่องในด้านหน้า ที่จำเป็นต้องมีการคำนวณเอียงเพื่อหาตำแหน่งของลูกบอล

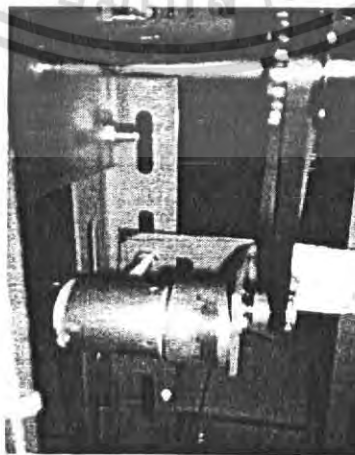
3.3.2 การออกแบบโครงสร้าง

จากหัวข้อที่ 3.3.1 เป็นการสร้างฐานยึดกล่องและตำแหน่งที่เหมาะสมในการติดตั้งกล่องดิจิตอลวีดีโอ ในหัวข้อนี้จะอธิบายถึงส่วนต่างๆ ของโครงสร้างระบบควบคุมตำแหน่งลูกบอลบนคาน

จากรูปที่ 3.5 จะเห็นว่าในส่วน โครงของโครงสร้างทำมาจากเหล็กฉาก ทำให้ง่ายต่อการขึ้นรูปตามแบบ โดยส่วนของโครงสร้างนี้จะยึดอยู่บนฐานซึ่งทำมาจากอาคริลิก และมีล้อเพื่อง่ายต่อการเลื่อนในขณะที่ทำการปรับตั้งเพื่อใช้งานร่วมกับกล่องดิจิตอลวีดีโอ ส่วนของโครงสร้างต้องมีความแน่นหนา มั่นคง เพื่อป้องกันไม่ให้ชิ้นงานเลื่อนไปจากเดิม โดยเฉพาะเมื่อมีแรงบิดจากมอเตอร์

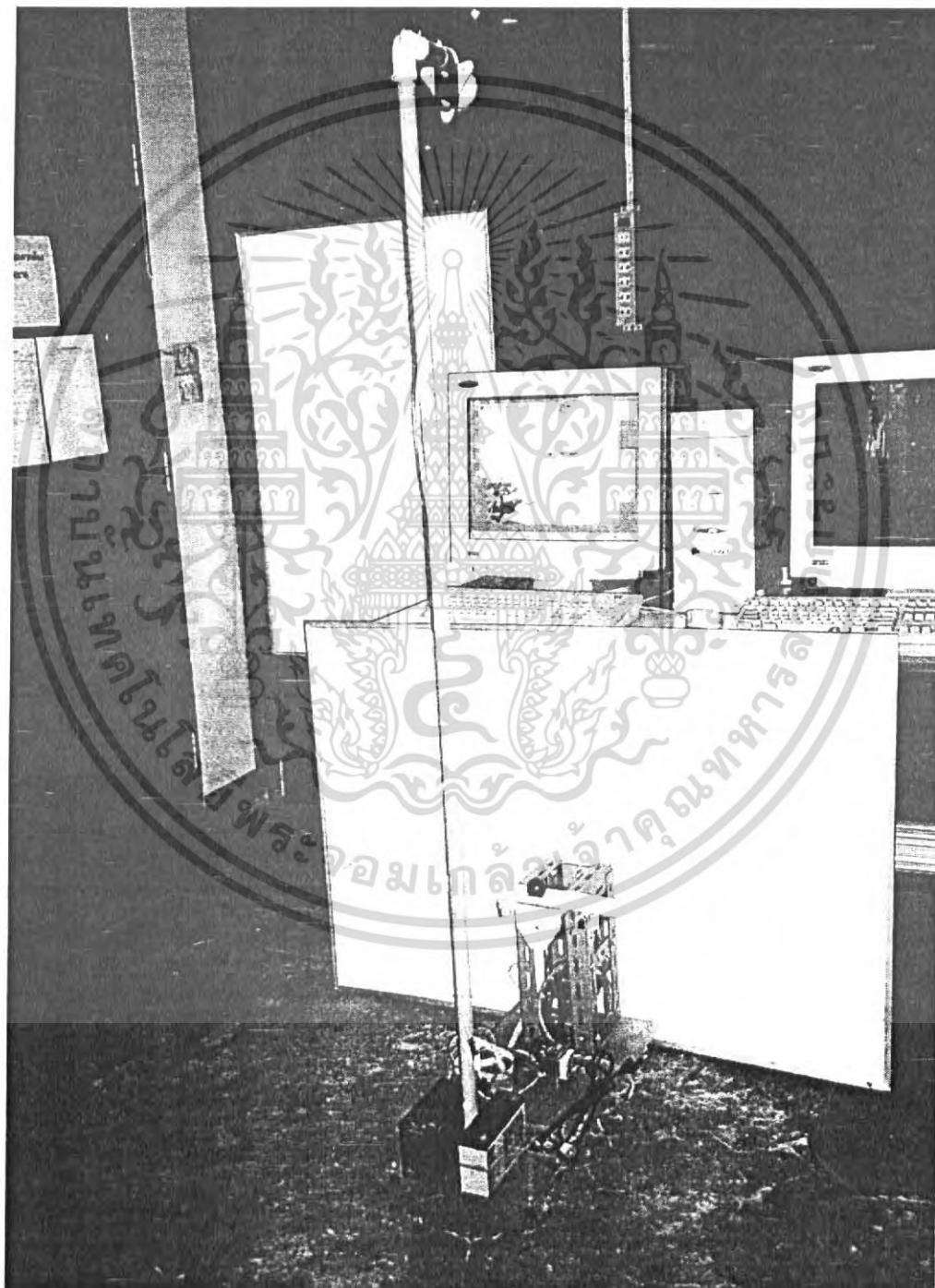


รูปที่ 3.5 โครงสร้างของระบบควบคุมตำแหน่งลูกบอลบนคาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 3.6 การยึดมอเตอร์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

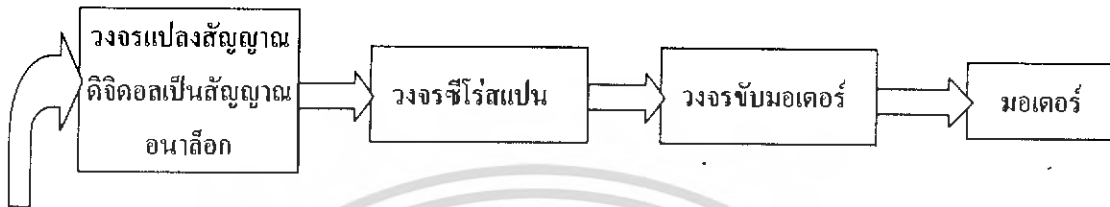
หลังจากประกอบโครงสร้างของระบบควบคุมเรียบร้อยแล้ว ได้ติดตั้งมอเตอร์ไฟฟ้า กระแสตรง เข้ากับตัวโครงสร้างดังรูปที่ 3.6 การติดตั้งมอเตอร์เข้ากับโครงสร้างต้องทำอย่างรอบคอบเพื่อป้องกันมอเตอร์เคลื่อนที่ เนื่องจากต้องใช้แรงบิดมากขณะหมุนบังคับคัน และการติดตั้งพูล์เลย์ของมอเตอร์กับพูล์เลย์ที่อยู่บนแกนเพลลาบังคับคันต้องได้ศูนย์ โดยสายพานต้องไม่หย่อนหรือตึงจนเกินไป เมื่อนำมาประกอกรวมกันจะได้เป็นโครงสร้างรวมของระบบควบคุมตำแหน่งลูกบอลบนกานดังรูปที่ 3.7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.7 ระบบที่ควบคุมตำแหน่งลูกบอลบนกานหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การคำนวณและการออกแบบวงจรและภาคขับมอเตอร์

จากที่ได้กล่าวมาแล้วในบทที่ 2 ว่า วงจรและภาคขับมอเตอร์ประกอบด้วยส่วนสำคัญ 3 ส่วน คือ วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก วงจรปรับซีโรสเปน และวงจรขับมอเตอร์ ในบทนี้จะอธิบายถึงหลักการออกแบบวงจรที่ได้กล่าวมาแล้ว ดังนี้



สัญญาณขนาด 8 บิต จากพอร์ตขนาน

รูปที่ 3.8 การทำงานของวงจรและภาคขับมอเตอร์

3.4.1 การออกแบบวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก

วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกเป็นวงจรส่วนแรกที่จะทำงาน โดยจะรับสัญญาณดิจิทัลขนาด 8 บิต จากพอร์ตขนานที่มีข้อมูล 256 ค่า คือ จากค่า 0 ถึง 255 และทำการเปลี่ยนเป็นสัญญาณอนาล็อกที่มีขนาด 0 ถึง 5 โวลต์ จากความสัมพันธ์นี้สามารถแสดงได้ดังสมการ (3.1)

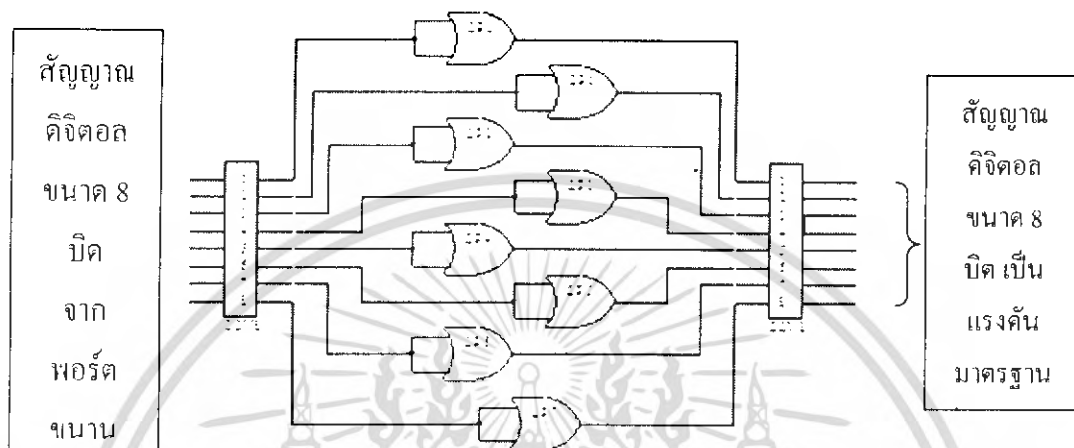
$$V_o = \frac{BCD \times 5}{255} \quad (3.1)$$

เมื่อ V_o คือแรงดันเอาต์พุตของวงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกมีค่าจาก 0 ถึง 5 โวลต์ BCD คือค่าของเลข ไบนารี 16 บิต สามารถแปลงให้อยู่ในรูปของเลขฐานสิบคือ 0 ถึง 255

การออกแบบวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแบบใช้ความต้านทานสองค่า ต้องทำการวัดขนาดแรงดันไฟฟ้าที่ออกมาจากพอร์ตขนาน เพราะแรงดันไฟฟ้าที่ออกมาจากพอร์ตขนานของคอมพิวเตอร์แต่ละเครื่องจะมีขนาดไม่เท่ากัน โดยคอมพิวเตอร์ที่ใช้ทดลองในโครงงานนี้ วัดค่าแรงดันไฟฟ้าในแต่ละบิตของพอร์ตขนานได้ 4.5 โวลต์ ปัญหาที่ตามมาก็คือวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกเป็นวงจรที่ประกอบด้วย ออปแอมป์ และ คิวต้านทาน ขนาดของแรงดันไฟฟ้าที่เข้าวงจรจะมีผลต่อแรงดันเอาต์พุตทำให้วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกที่ออกแบบนั้นมีข้อจำกัดคือ ไม่สามารถใช้กับคอมพิวเตอร์ที่มีขนาดของแรงดันไฟฟ้าในแต่ละบิตของพอร์ตขนาน ไม่เท่ากับ 4.5 โวลต์ ได้ จึงแก้ปัญหาโดยนำสัญญาณที่

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมาจากพอร์ตนานของคอมพิวเตอร์ในแต่ละบิตมาผ่านวงจรรรอก ออกเกต(OR Gate) ดังรูปที่ 3.9 แต่ก็ยังเกิดปัญหาแรงดันที่ออกจากเกตแต่ละตัวมีค่าไม่เท่ากัน ทำให้อาจเลือกใช้ไอซีแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกเพราะมีเสถียรภาพดีกว่า โดยไอซีที่เลือกใช้คือเบอร์ DAC0832 ซึ่งการใช้งานสามารถศึกษาได้จากภาคผนวก ข-1



รูปที่ 3.9 วงจรปรับแรงดันมาตรฐานโดยใช้ออเกต

3.4.2 การออกแบบวงจรปรับซีโรสเปน

จากรูปที่ 3.8 พบว่าหลังจากรับสัญญาณที่ออกมาจากพอร์ตนาน ผ่านวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก ทำให้ได้แรงดันเอาต์พุต 0 ถึง 5 โวลต์ แต่เราต้องการแรงดันทั้งในช่วงบวกและลบ เพื่อให้มอเตอร์สามารถหมุนได้ทั้งสองทิศทาง ดังนั้นจึงใช้วงจรซีโรสเปนดังรูปที่ 2.17 ปรับระดับแรงดันให้ได้แรงดันไฟทั้งช่วงบวกและลบตามต้องการ ในหัวข้อนี้จะแสดงวิธีการออกแบบและการคำนวณค่าตัวแปรต่างๆ ของวงจรปรับซีโรสเปน โดยการคำนวณหาค่าของอุปกรณ์ต่างๆ ของวงจรสามารถคำนวณได้จากสมการ (3.2)

$$V_o = (-V_f)(-R_f/R_{11}) - (R_f/R_{12})(E_{dc}) \quad (3.2)$$

เมื่อ V_o คือ แรงดันเอาต์พุตของวงจรปรับซีโรสเปน

V_f คือ แรงดันอินพุตของวงจร (เอาต์พุตของวงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก)

E_{dc} คือ แรงดันไฟฟ้าอ้างอิง

R_f คือ ความต้านทานป้อนกลับของวงจรปรับซีโรสเปน

R_{11} คือ ความต้านทานของแรงดันอินพุตของวงจร

R_{12} คือ ความต้านทานของแรงดันไฟฟ้าอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณจะเริ่มจากการเลือกกำหนดค่าความต้านทานป้อนกลับของวงจรปรับซีโรสแปนก่อน โครงการนี้เลือกค่าความต้านทานป้อนกลับของวงจรปรับซีโรสแปน $R_f = 10$ กิโลโห์ม แรงดันไฟฟ้าอ้างอิง E_{dc} มีค่าเท่ากับ 15 โวลต์ จากนั้นกำหนดเอาต์พุตของวงจรปรับซีโรสแปนให้เป็น 2 กรณี คือ

- เมื่อแรงดันอินพุตของวงจรปรับซีโรสแปน $V_i = 0$ โวลต์ ต้องการแรงดันเอาต์พุตของวงจรปรับซีโรสแปนเป็น -14 โวลต์
- เมื่อแรงดันอินพุตของวงจรปรับซีโรสแปน $V_i = 5$ โวลต์ ต้องการแรงดันเอาต์พุตของวงจรปรับซีโรสแปนเป็น $+14$ โวลต์

จากเงื่อนไข 2 เงื่อนไขนี้ เมื่อแทนลงในสมการที่ (3.2) จะได้สมการ (3.3) และ (3.4) ตามลำดับ

$$-14 = (0)(R_f/R_{f1}) - (R_f/R_{f2})(15) \quad (3.3)$$

$$14 = (5)(R_f/R_{f1}) - (R_f/R_{f2})(15) \quad (3.4)$$

จากสมการที่ (3.3) และ (3.4) ได้อัตราส่วนของ $R_f/R_{f1} = 5.6$ และได้อัตราส่วนของ $R_f/R_{f2} = 0.9333$

ดังนั้นคำนวณ R_{f1} ได้จากสมการ (3.5) และ R_{f2} ได้จากสมการ (3.6)

$$R_{f1} = \frac{10 \times 10^3}{5.6} \quad (3.5)$$

$$R_{f2} = \frac{10 \times 10^3}{0.9333} \quad (3.6)$$

จะได้ค่า $R_{f1} = 1.785$ กิโลโห์ม

$R_{f2} = 10.714$ กิโลโห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้ จะกล่าวถึงการทดลองและผลการทดลอง วงจรและภาคขับมอเตอร์ และการทำงานของระบบควบคุมตำแหน่งลูกบอลบนคานแบบระบบวงปิด โดยมีรายละเอียดของการทดลองดังนี้

4.1 การทดลองวงจรและภาคขับมอเตอร์

ในส่วนนี้ เป็นการศึกษาเฉพาะส่วนวงจรและภาคขับมอเตอร์ ซึ่งนับแต่การสั่งค่า BCD โดยโปรแกรมคอมพิวเตอร์ผ่านออกพอร์ตขนาน ไปสู่วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแบบใช้ความต้านทานสองค่า วงจรซีโรสแปน เข้าสู่ภาคขับเพื่อขับมอเตอร์ ซึ่งผลการทดลองมีดังนี้

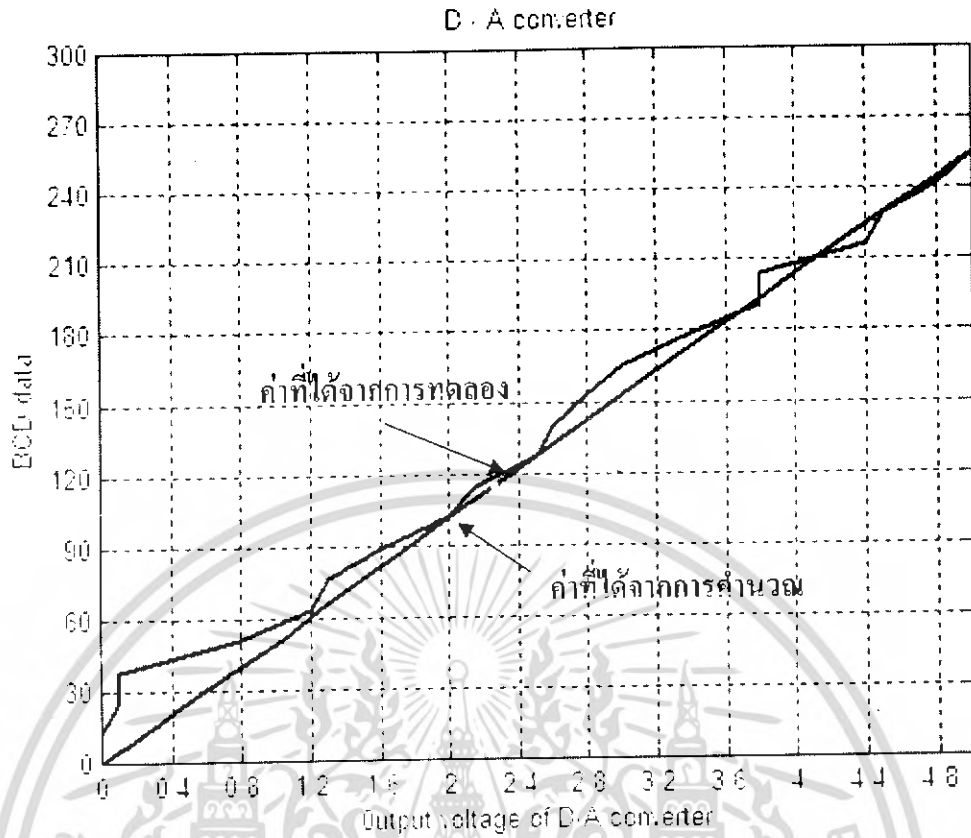
4.1.1 วงจรและภาคขับมอเตอร์ที่ไม่มีวงจรถรก ออกกต

ผลการทดลองในส่วนนี้แปรค่า BCD ที่สั่งจากคอมพิวเตอร์ที่เป็นการทดลองต่างๆ กัน แล้วบันทึกการทำงาน ณ จุดต่างๆ ของวงจร บันทึกผลได้ดังตารางที่ 4.1 และนำไปวาดกราฟได้ดังรูปที่ 4.1 ถึง 4.3

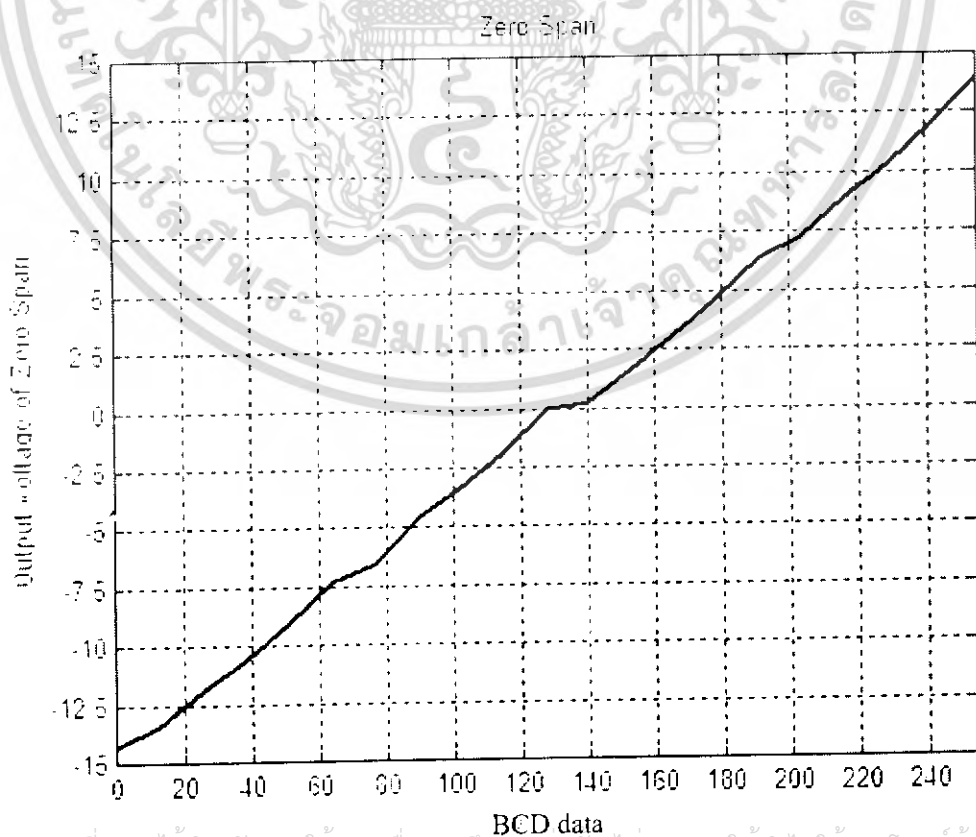
ตารางที่ 4.1 ผลการทดลองวงจรและภาคขับมอเตอร์ที่ไม่มีวงจรถรก ออกกต

BCD Code	D/A output voltage(V) จากการวัดค่า	D/A output voltage(V) จากการคำนวณ	Output voltage Zero Span(V)	Driver output voltage(V)	Motor action
0	0	0	-14.32	-13	หมุนซ้าย
38.25	0.7	1	-10.57	-10	หมุนซ้าย
63.75	1.2	1.25	-7.38	-6.8	หมุนซ้าย
102	2	2	-3.37	-2.8	หยุด
127.5	2.5	2.5	0	0	หยุด
153	2.8	3	1.7	1.2	หยุด
191.25	2.8	3.75	6.39	5.6	หมุนขวา
229	4.4	4.49	10.37	9.6	หมุนขวา
255	5	5	13.94	13	หมุนขวา

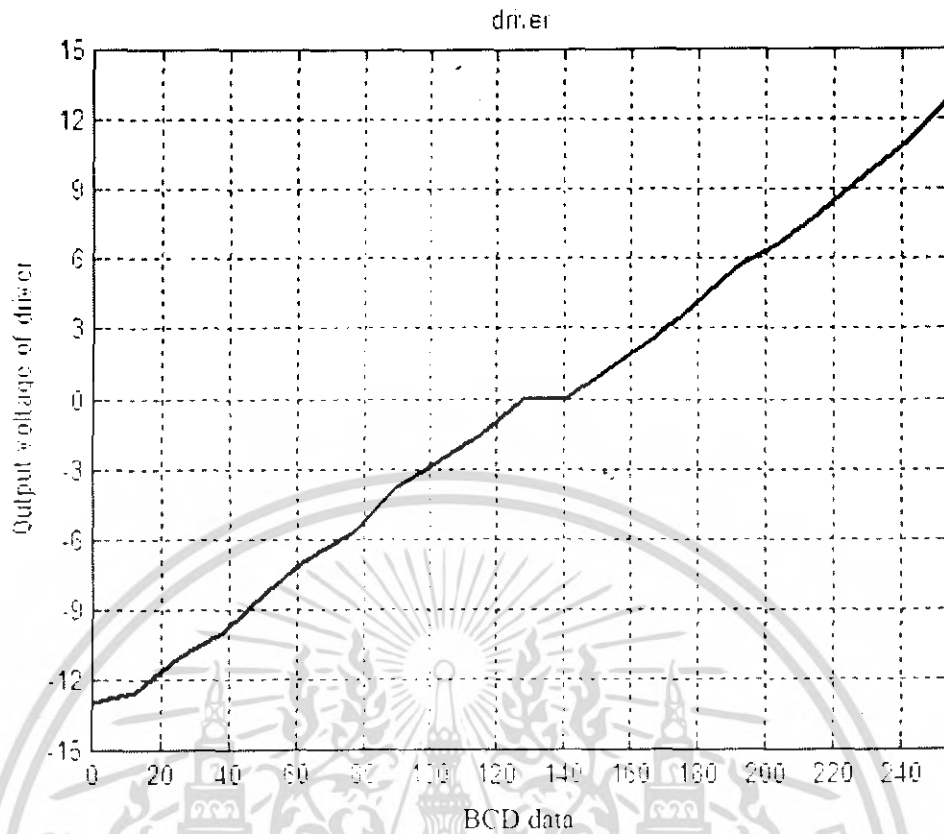
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 ผลการทดลองวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกที่ไม่ต้องวงจรทรานซิสเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 4.2 ผลการทดลองของวงจรปรับซีโรสเปนนที่ไม่ต้องวงจรทรานซิสเตอร์



รูปที่ 4.3 ผลการทดลองของวงจรขับมอเตอร์ที่ไม่ใช่วงจรตรรก ออกเกต

จากผลการทดลองวงจรและภาคขับมอเตอร์ จะเห็นว่าสัญญาณเอาต์พุตที่ได้จากวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกมีความผิดพลาดจากการคำนวณพอสมควร เนื่องจากค่าความต้านทานที่ใช้งานนั้นมีค่าความผิดพลาดสูง ปรกติตามพบว่ายังสามารถนำมาใช้งานกับวงจร ซีโรสแปนและวงจรขับมอเตอร์ได้

วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกแบบใช้ความต้านทานสองค่าที่ใช้ในการทดลองนี้มีข้อเสียอยู่ประการหนึ่ง กล่าวคือเอาต์พุตของวงจรจะขึ้นกับระดับอินพุตที่ได้รับจากพอร์ตขนานของคอมพิวเตอร์ซึ่งแต่ละเครื่องมักมีค่าไม่เท่ากัน จึงทำให้อาจไม่สามารถนำไปใช้กับคอมพิวเตอร์เครื่องอื่นได้ ดังนั้นจึงป้องกันปัญหาดังกล่าวโดยการเพิ่มวงจรตรรก ออกเกต ระหว่างจากพอร์ตขนานกับวงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก เพื่อให้เอาต์พุต 0 และ 1 ของสัญญาณดิจิทัลที่จะเข้าสู่วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกมีค่าสม่ำเสมอ คือ ลอจิก 0 เป็นสัญญาณ 0 โวลต์ และลอจิก 1 เป็นสัญญาณ 5 โวลต์ แม้ว่าระดับสัญญาณเอาต์พุตจากพอร์ตขนานของคอมพิวเตอร์อาจมีค่าต่างกัน ดังการทดลองในหัวข้อ 4.1.2

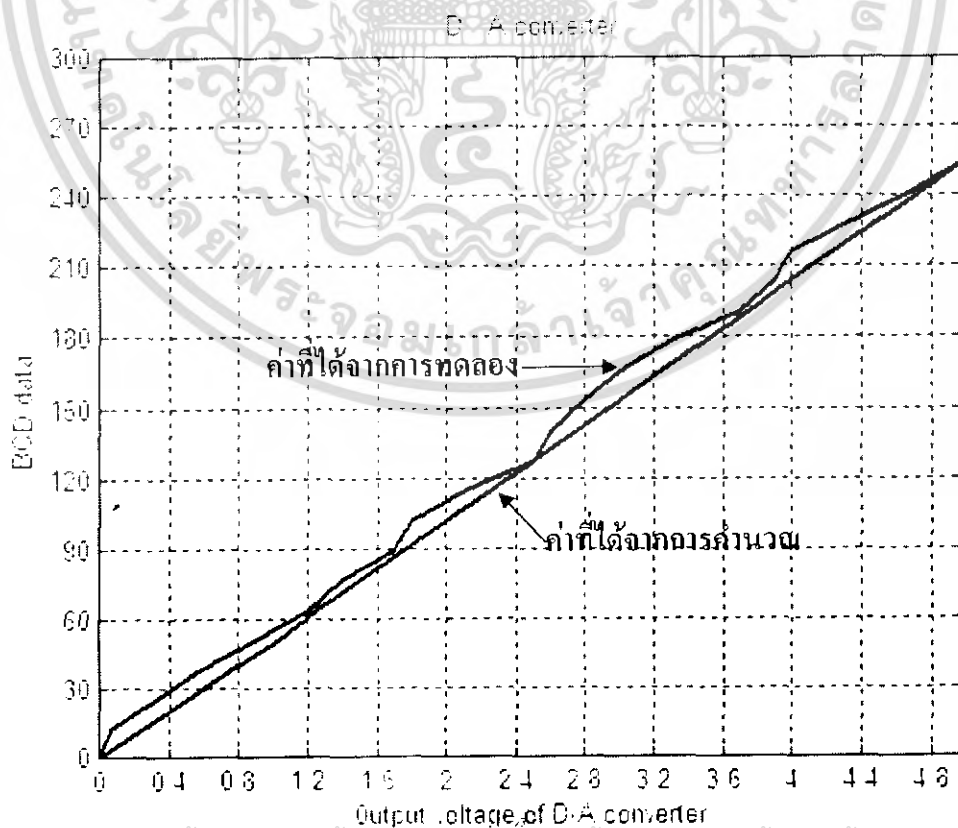
4.1.2 วงจรและภาคขับมอเตอร์ที่มีวงจรตรรก ออกเกต

วงจรและภาคขับมอเตอร์แบบนี้เป็นการนำเอาวงจรตรรก ออกเกต เพื่อแก้ไขข้อจำกัดในเรื่องระดับสัญญาณจากพอร์ตขนานซึ่งอาจให้แรงดันไฟฟ้าไม่ได้มาตรฐาน เพื่อให้สามารถนำวงจรไปใช้

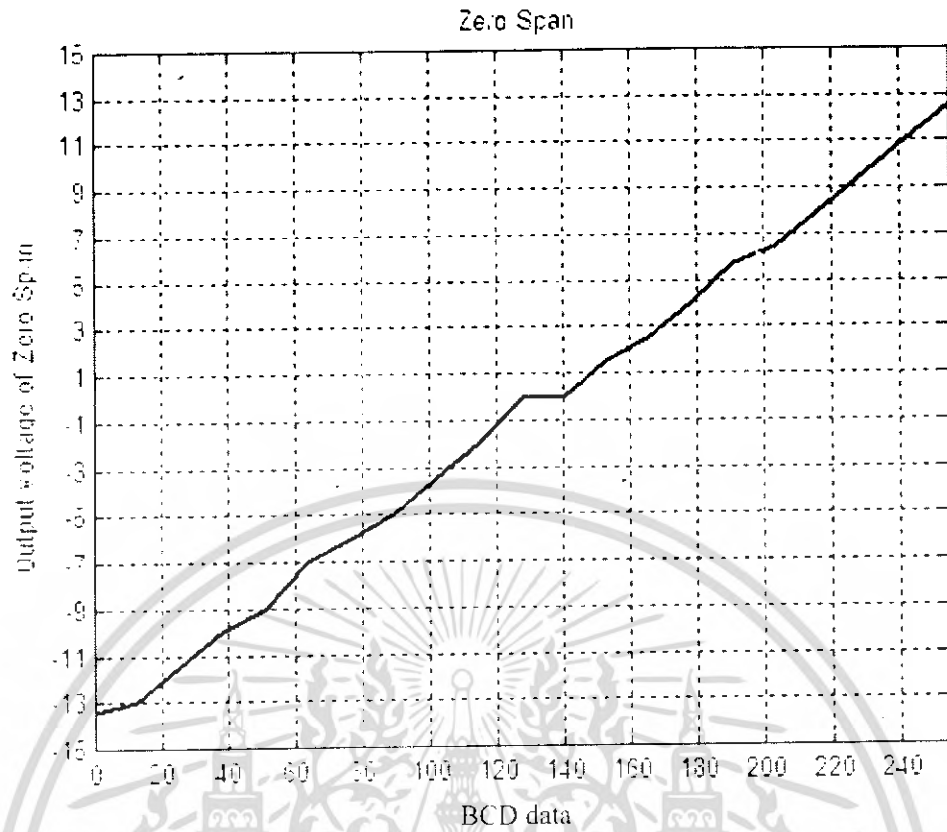
กับคอมพิวเตอร์เครื่องอื่นได้ การทำงานของวงจรนี้จะทำงานคล้ายกับวงจรแบบแรก โดยผลการทดลองแสดงดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองวงจรและภาคขับมอเตอร์ที่มีการต่อวงจรครบ ออกเกิด

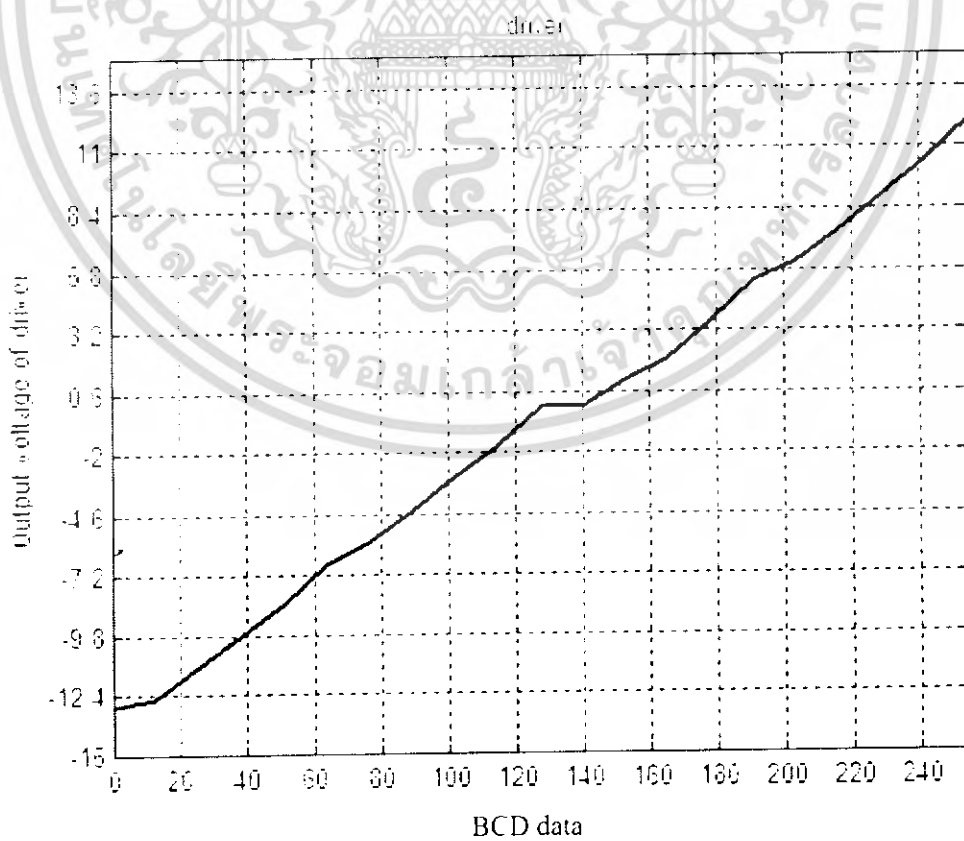
BCD Code	D/A output voltage(V) จากการวัดค่า	D/A output voltage(V) จากการคำนวณ	Output voltage Zero Span(V)	Drive output voltage(V)	Motor action
0	0	0	-13	-12.9	หมุนซ้าย
38.25	0.58	1	-10	-9.8	หมุนซ้าย
63.75	1.2	1.25	-7	-6.7	หมุนซ้าย
102	1.8	2	-3.5	-3	หยุด
127.5	2.5	2.5	0	0	หยุด
153	2.8	2.8	1.5	1.1	หยุด
191.25	3.7	3.8	5.8	5.3	หมุนขวา
216	4	4	8	7.4	หมุนขวา
255	5	5	12.5	12.2	หมุนขวา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.4 ผลการทดลองวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกที่ต่อวงจรครบ ออกเกิด



รูปที่ 4.5 ผลการทดลองของวงจรปรับซีโรสแปนที่ต่อวงจรตรรก ออเกต



รูปที่ 4.6 ผลการทดลองของวงจรขมอมอเตอร์ที่ต่อวงจรตรรก ออเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.2 แรงดันเอาต์พุตของวงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกจะเท่ากับ แรงดันเอาต์พุตของวงจรแบบแรกโดยจะมีค่า 0 ถึง 5 โวลต์ เทียบกับข้อมูลของพอร์ตขนานซึ่งจะมี ค่าจาก 0 ถึง 255 เอาต์พุตของวงจรซีโรสแปนมีค่า -13 โวลต์ ถึง 12.5 โวลต์ ทำให้เอาต์พุตของ วงจรขับมอเตอร์มีค่า -12.9 โวลต์ ถึง 12.2 โวลต์ โดยมอเตอร์จะเริ่มหมุนที่แรงดันไฟฟ้าประมาณ 5.6 โวลต์ กรณีหมุนไปทางขวา และ -5.3 โวลต์ กรณีหมุนไปทางซ้าย(ไม่มีลูกบอลบนคาน)

จากการทดลองทั้งสองกรณีข้างต้น พบว่าการใช้วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณ อนาล็อกแบบความต้านทานสองค่านี้มีความผิดพลาด เนื่องจากปัญหาที่ค่าความต้านทานไม่เป็นไป ตามอุดมคติ ในการใช้จริงต่อไปจึงแก้ไขโดยใช้วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก สำเร็จรูปด้วยไอซี DAC0832 ดังแสดงในภาคผนวก ข-1 ซึ่งเมื่อทดลองจะพบว่าสามารถใช้งานได้ดี โดยระดับสัญญาณเอาต์พุตจากวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกมีค่าใกล้เคียงกับ ค่าคำนวณอย่างมาก

4.2 การระบุตำแหน่งของลูกบอลโดยการเทียบพิกเซล

โครงการนี้ได้ใช้กล้องดิจิทัลวีดีโอสำหรับตรวจจับตำแหน่งของลูกบอลที่อยู่บนคาน ดังนั้น ตำแหน่งของลูกบอลที่อ่านได้จากหน้าจอกอมพิวเตอร์จะอยู่ในรูปของพิกเซล จึงต้องทำการแปลง โดยเลือกใช้ความกว้างของหน้าจอที่รับค่าจากกล้องคือ 320 พิกเซล เท่ากับความยาวของคาน 80 เซนติเมตร ความสัมพันธ์ดังกล่าวสามารถเขียนได้ดังสมการที่ (4.1)

$$A = \frac{B \times 80}{320} \quad (4.1)$$

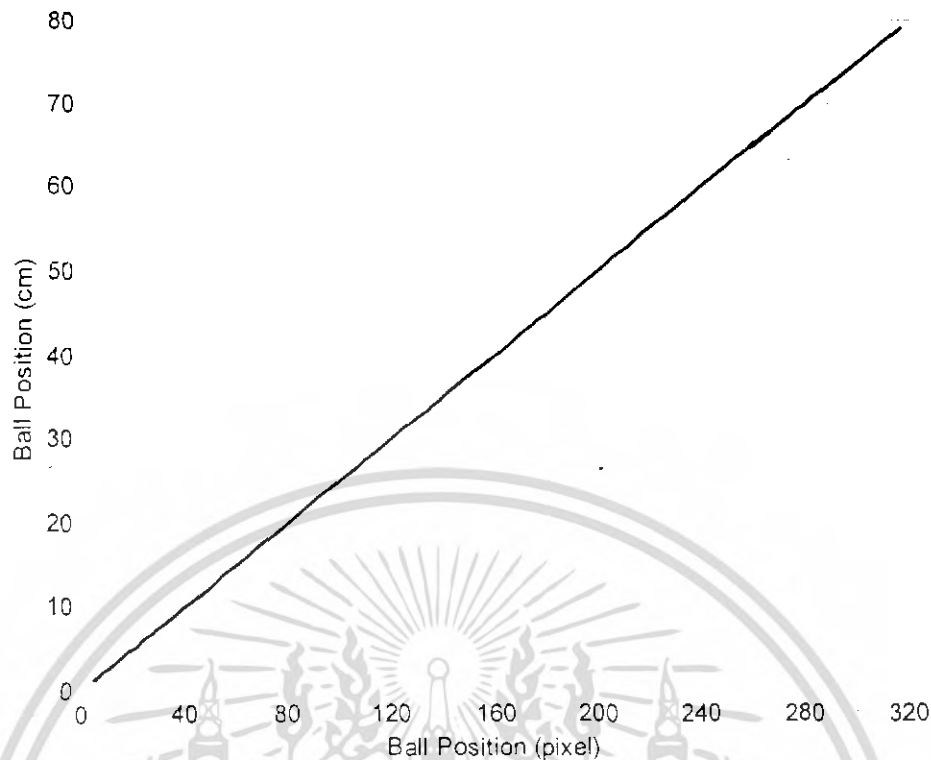
เมื่อ A คือ ตำแหน่งจริงของลูกบอลบนคานมีหน่วยเป็น เซนติเมตร(cm)

B คือ ตำแหน่งของลูกบอลที่อ่านได้จากหน้าจอกอมพิวเตอร์ มีหน่วยเป็น พิกเซล(pixel)

จากสมการที่ (4.1) เมื่อนำมาเขียน โปรแกรมเพื่อหาตำแหน่งของลูกบอลที่อ่านได้จากหน้า จอกอมพิวเตอร์ โดยให้แสดงตำแหน่งของลูกบอลอยู่ในหน่วยเซนติเมตร เนื่องจากเวลานำไปใช้ งาน จะระบุจุดเป้าหมายที่ต้องการให้ระบบควบคุมลูกบอลไปยังตำแหน่งนั้นๆ อยู่ในหน่วยของ เซนติเมตร โดยผลการทดลองการตรวจวัดตำแหน่งของลูกบอลที่อยู่บนคานเทียบกับตำแหน่ง พิกเซลแสดงได้ดังรูปที่ 4.7

จากรูปที่ 4.7 พบว่าไม่สามารถตรวจจับลูกบอลที่ใกล้ตำแหน่ง 0 เซนติเมตร และ 80 เซนติเมตรได้ เนื่องจากทำการตรวจจับลูกบอลโดยคิดจากค่ากลางระหว่างตำแหน่งพิกเซลเริ่มต้น และตำแหน่งพิกเซลสุดท้ายที่ตรวจพบลูกบอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 การหาตำแหน่งของลูกบอลที่อยู่บนคานเทียบกับตำแหน่งพิกเซล

4.3 การทดสอบการควบคุมตำแหน่งลูกบอลไปยังตำแหน่งต่างๆ ที่กำหนดบนคาน

การทดลองนี้เป็นการทดสอบการควบคุมตำแหน่งลูกบอลไปยังตำแหน่งเป้าหมายต่างๆ บนคานแบบระบบวงปิดโดยใช้ตัวควบคุมแบบพีดีแต่ละหน่วยพบว่าผลตอบสนองไม่ดีนัก จึงมีการปรับปรุงโดยการเพิ่มเงื่อนไขในการควบคุมลงโปรแกรม เพื่อให้ระบบสามารถควบคุมได้ดีขึ้นในระดับที่ยอมรับได้ ดังแสดงในหัวข้อย่อต่อไปนี้ โดยในการทดลองมีค่าคาบเวลาการสุ่มสัญญาณ T_s เป็น 0.185 วินาที โปรแกรมที่ใช้ในการระบุตำแหน่งลูกบอลและควบคุมแสดงในภาคผนวก ก

4.3.1 การควบคุมโดยใช้ตัวควบคุมแบบพีดี

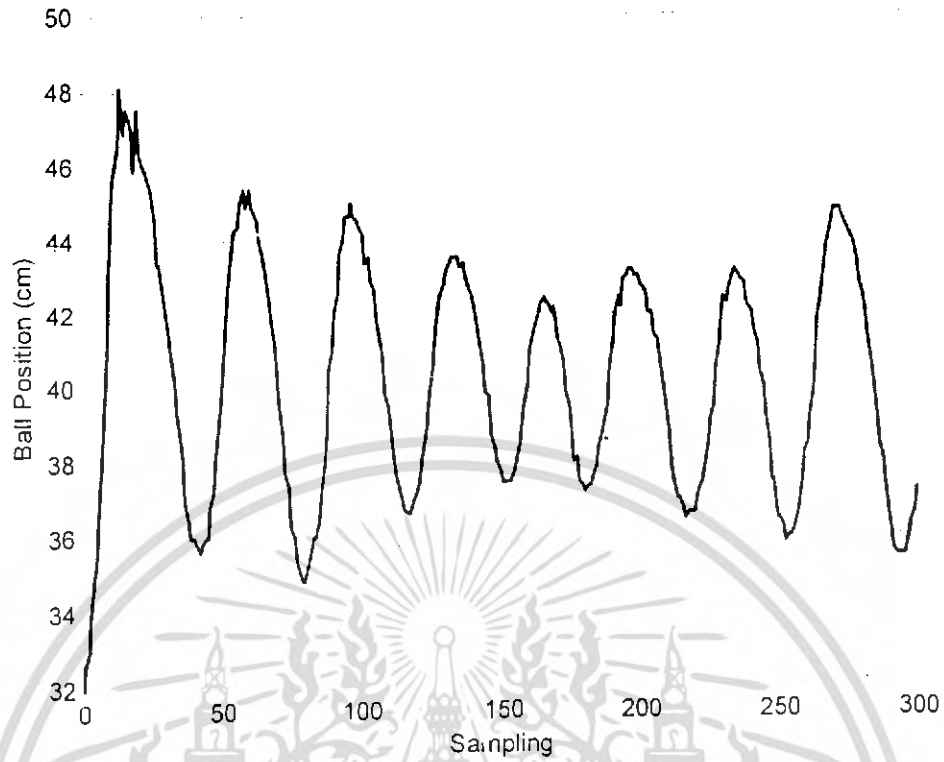
การทดลองนี้ อาศัยโปรแกรมซึ่งใช้ตัวควบคุมแบบพีดีตามปกติ โดยทดลองควบคุมลูกบอลจากตำแหน่ง 32 เซนติเมตร ไปยังตำแหน่ง 40 เซนติเมตร ได้ผลการทดลอง ดังรูปที่ 4.8 และ 4.9 ซึ่งแสดงเอาท์พุทตำแหน่งลูกบอล และสัญญาณอินพุทของระบบ ตามลำดับ

จากผลการทดลอง พบว่าไม่สามารถควบคุมตำแหน่งของลูกบอลให้เข้าสู่จุดเป้าหมายที่กำหนด เนื่องจากขณะที่ลูกบอลเข้าสู่จุดเป้าหมาย โปรแกรมยังช้ากว่าระบบจริงอยู่เล็กน้อย และคานยังเอียงอยู่ไม่สามารถเอียงกลับได้ทันที ระบบจะควบคุมคานให้เอียงกลับได้ก็ต่อเมื่อลูกบอลเลขจุด

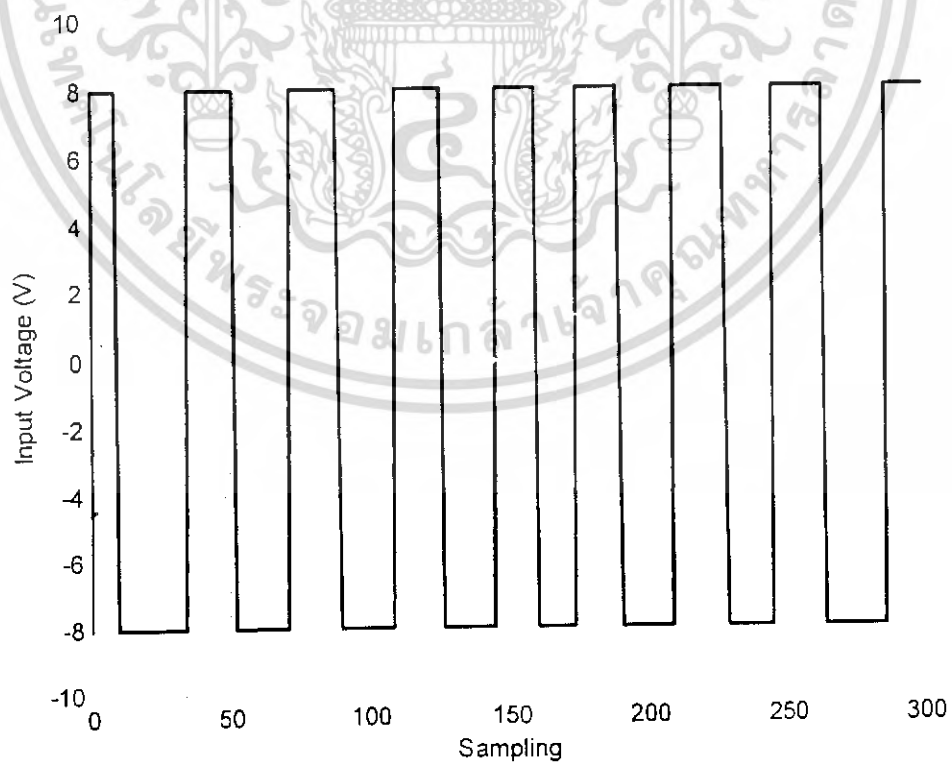
กำหนดแล้ว ดังนั้นลูกบอลจึงไม่สามารถเข้าหาจุดกำหนดได้พอดี จึงเกิดการแกว่งรอบจุดเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 4.8

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 การควบคุมลูกบอลจากตำแหน่ง 32 cm ไปยังตำแหน่ง 40 cm



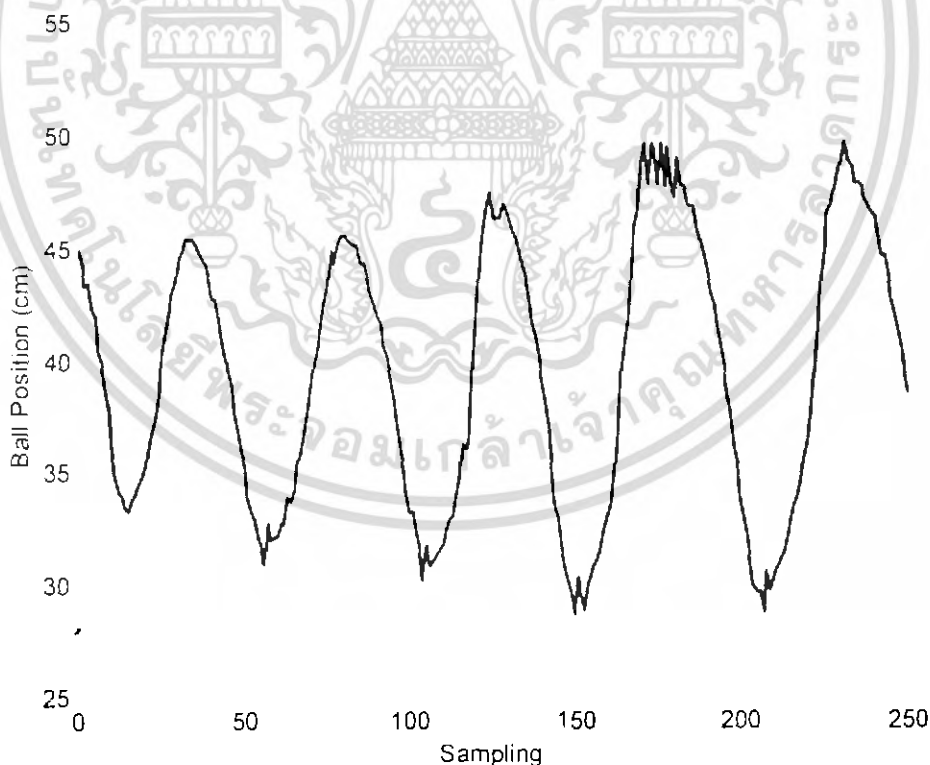
รูปที่ 4.9 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 32 cm ไปยังตำแหน่ง 40 cm

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการเรียนเพื่อการศึกษาเท่านั้น เมื่อผู้ญาติเห็นใบโฆษณาบนการค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองกำหนดให้ $K_p = 3$, $K_d = 4.625 \times 10^{-3}$ ค่าแรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางขวา คือ 8 โวลต์ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางขวา คือ 10 โวลต์ แรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางซ้าย คือ -8 โวลต์ และ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางซ้าย คือ -10 โวลต์

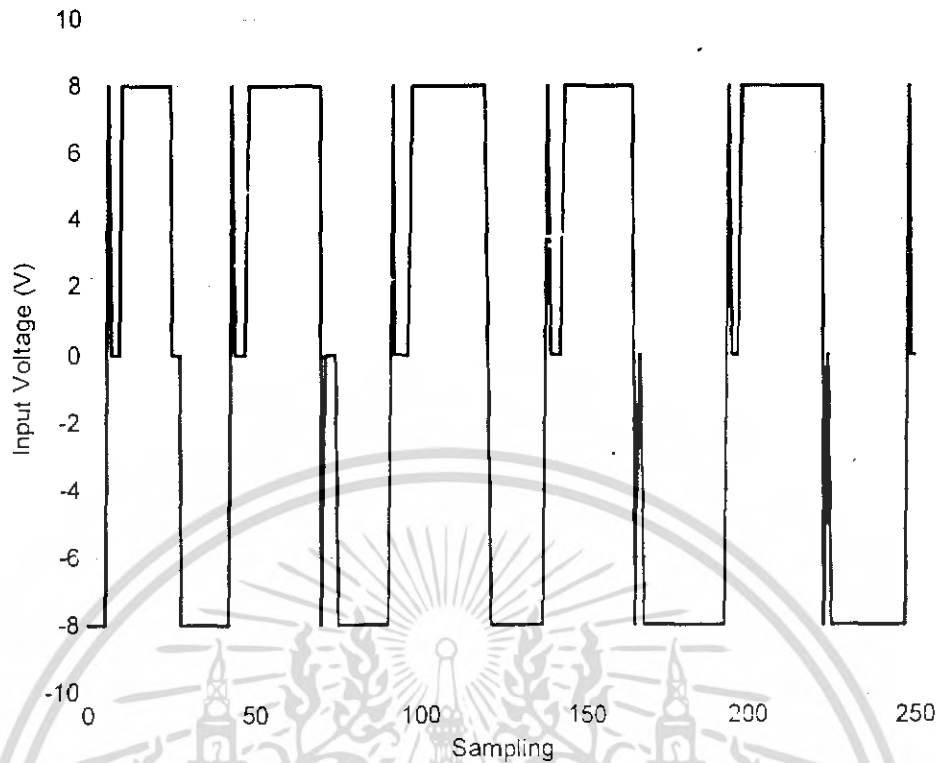
4.3.2 การเขียนโปรแกรมโดยการกำหนดช่วงค่าผิดพลาดที่ยอมรับได้

จากการทดลองที่ 4.3.1 เห็นว่าระบบไม่สามารถควบคุมตำแหน่งของลูกบอลให้เข้าสู่จุดกำหนดได้ จึงได้เขียน โปรแกรมขึ้นมาใหม่ โดยมีเงื่อนไขว่าขณะที่ลูกบอลอยู่ในช่วงของค่าผิดพลาดที่กำหนดไว้ ให้มอเตอร์หยุดหมุน จากการทดลองกำหนดค่าตัวแปรต่างๆ เหมือนกับการทดลองที่ 4.3.1 กำหนดให้ค่าความผิดพลาดคือ ± 1.5 เซนติเมตร จากการทดลองพบว่า เมื่อลูกบอลอยู่ในช่วงของค่าผิดพลาดที่กำหนดในขณะที่คานเอียงอยู่แต่มอเตอร์หยุดหมุน ลูกบอลจะลิ่งต่อไปเรื่อยๆ จนกว่าจะเลยช่วงของค่าผิดพลาดที่กำหนดระบบจึงสั่งให้มอเตอร์หมุนกลับทาง ทำให้เกิดค่าผิดพลาดระหว่างลูกบอลและจุดกำหนดมากกว่าการทดลองที่ 4.3.1 โดยมีผลการทดลองดังรูปที่ 4.10 สัญญาณอินพุตของระบบแสดงได้ดังรูปที่ 4.11



รูปที่ 4.10 การควบคุมลูกบอลจากตำแหน่ง 45 cm ไปยังตำแหน่ง 40 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 45 cm ไปยังตำแหน่ง 40 cm

4.3.3 การเขียนโปรแกรมโดยกำหนดช่วงค่าผิดพลาดและการชดเชยตำแหน่งบอลล่วงหน้า

จากการทดลองที่ 4.3.1 และ 4.3.2 พบว่าคานจะเอียงกลับก็คือเมื่อลูกบอลเลยจุดกำหนดไปแล้ว และคานจะหยุดนิ่งเมื่อลูกบอลอยู่ในช่วงของค่าผิดพลาดที่กำหนดไว้ จึงเขียน โปรแกรมขึ้นมาใหม่ โดยนำข้อเสียของการทดลองที่ 4.3.1 และ 4.3.2 มาปรับปรุงให้ดีขึ้น โดยโปรแกรมมีเงื่อนไขว่า ให้ระบบสั่งมอเตอร์ควบคุมคานให้เอียงกลับก่อนที่ลูกบอลจะถึงจุดกำหนด และเมื่อลูกบอลอยู่ในช่วงของของค่าผิดพลาดที่กำหนดไว้ระบบจะสั่งให้มอเตอร์หยุดหมุน จากการทดลองพบว่าระบบสามารถควบคุมลูกบอลให้เข้าสู่จุดกำหนด โดยมีค่าผิดพลาดอยู่ในช่วงที่กำหนดได้ ช่วงของค่าผิดพลาดที่กำหนดคือ ± 1.5 เซนติเมตร และกำหนดว่าถ้ามีค่าผิดพลาด 10 เซนติเมตร ให้มีการชดเชยตำแหน่งล่วงหน้า 6.5 เซนติเมตร โดยที่ค่าความผิดพลาดค่าอื่นๆ สามารถหาค่าชดเชยตำแหน่งล่วงหน้าได้ดังสมการที่ (4.2)

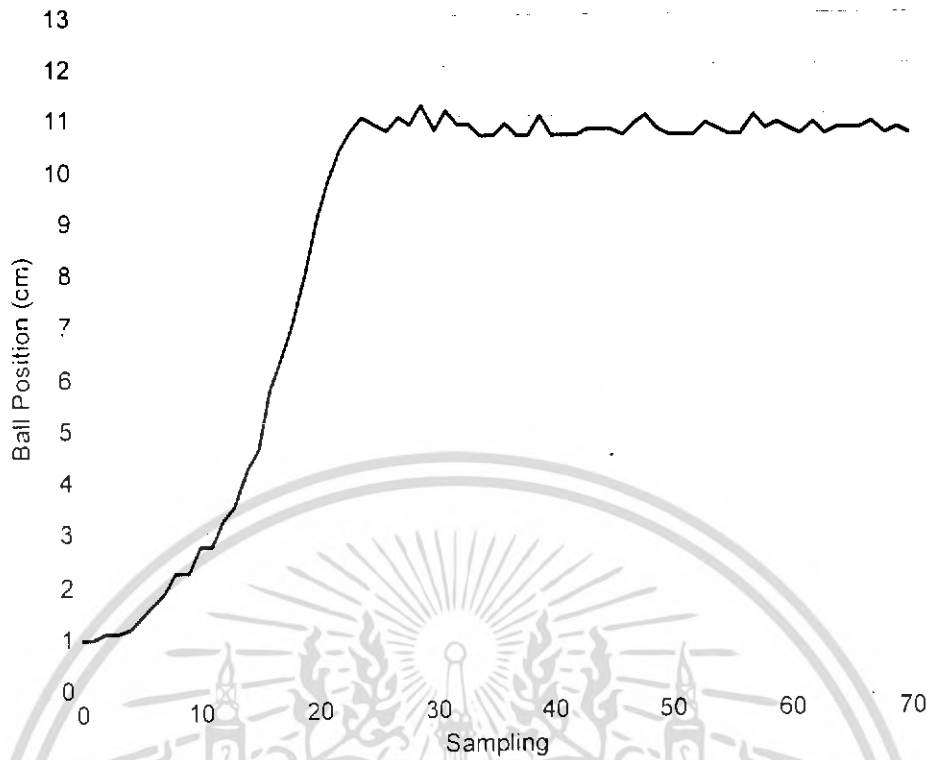
$$0.65 \times \text{ค่าผิดพลาดเริ่มต้น} \quad (4.2)$$

จากการทดลองโดยใช้สมการที่ (4.1) หาค่าชดเชยตำแหน่งล่วงหน้าพบว่าสามารถควบคุมลูกบอลให้ไปยังตำแหน่งที่ต้องการได้ภายในระยะ 20 เซนติเมตรเท่านั้น เนื่องจากระยะทางที่เพิ่มขึ้น มี

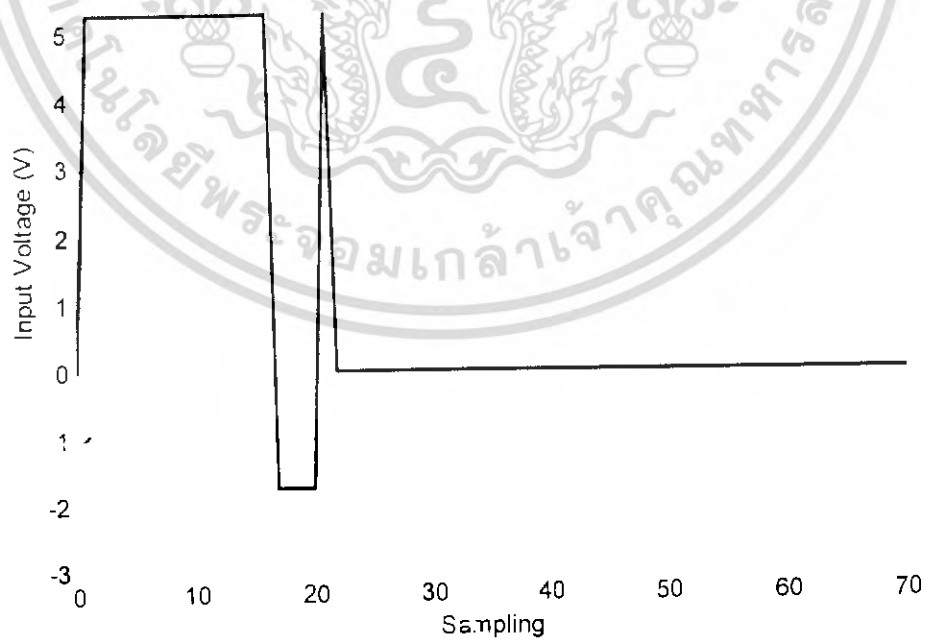
ผลทำให้ความเร็ว และความเร่งของลูกบอลไม่เท่ากัน ผลการทดลองของการควบคุมลูกบอลไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปชนดานการค้า
ตำแหน่งต่างๆ โดยใช้ตัวควบคุมแบบพีดี มีค่า $K_p = 3$ และ $K_d = 4.625 \times 10^{-3}$ มีดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



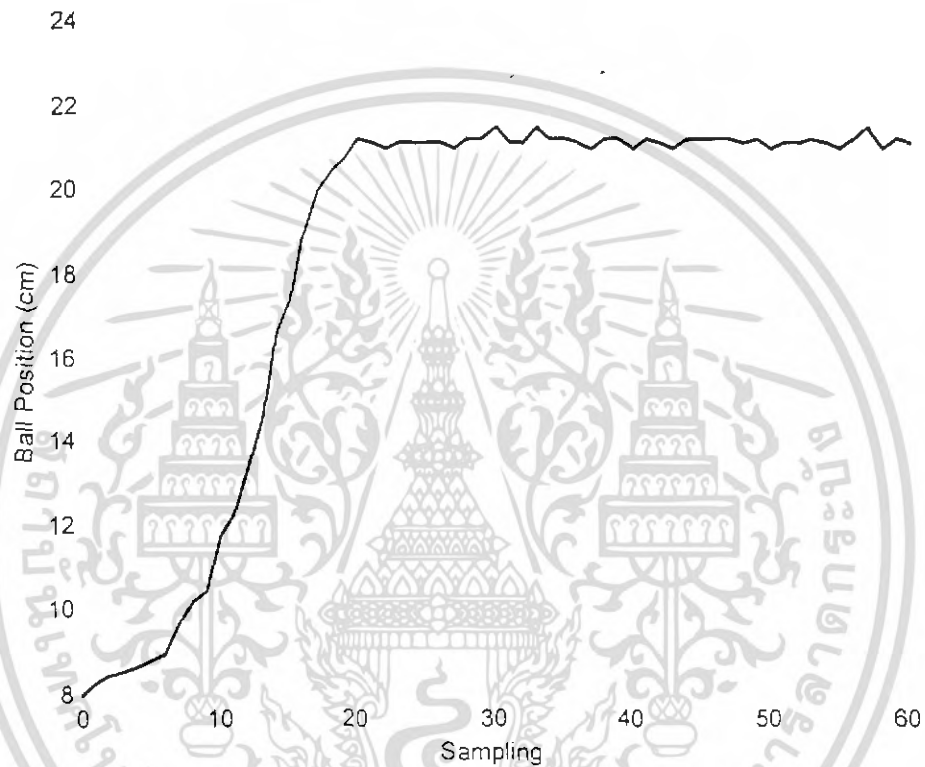
รูปที่ 4.12 การควบคุมลูกบอลจากตำแหน่ง 1 cm ไปยังตำแหน่ง 10 cm



รูปที่ 4.13 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 1 cm ไปยังตำแหน่ง 10 cm

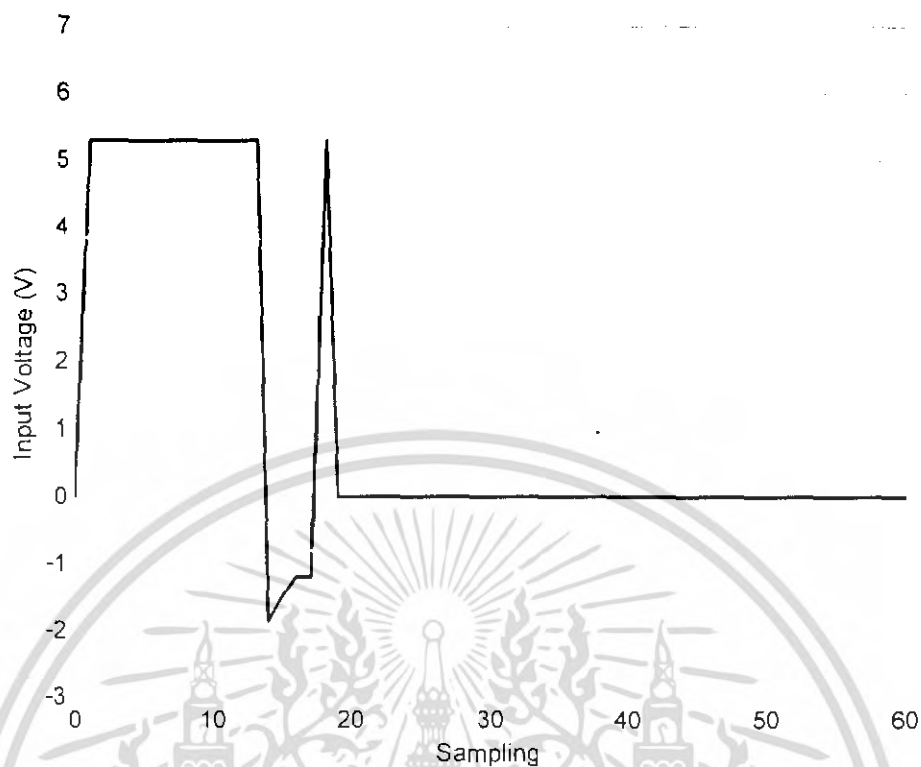
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.12 เป็นการควบคุมลูกบอลจากตำแหน่ง 1 เซนติเมตร ไปยังตำแหน่ง 10 เซนติเมตรของคาน ลูกบอลจะเข้าสู่จุดกำหนดที่คาบการสุมประมาณ 24 คาบ หรือเท่ากับ 4.44 วินาที กำหนดค่าแรงดันไฟฟ้าค่าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 5.3 โวลต์ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 7.5 โวลต์ แรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -1.75 โวลต์ และ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -4 โวลต์ แสดงสัญญาณอินพุตของการควบคุม ได้ดังรูปที่ 4.13

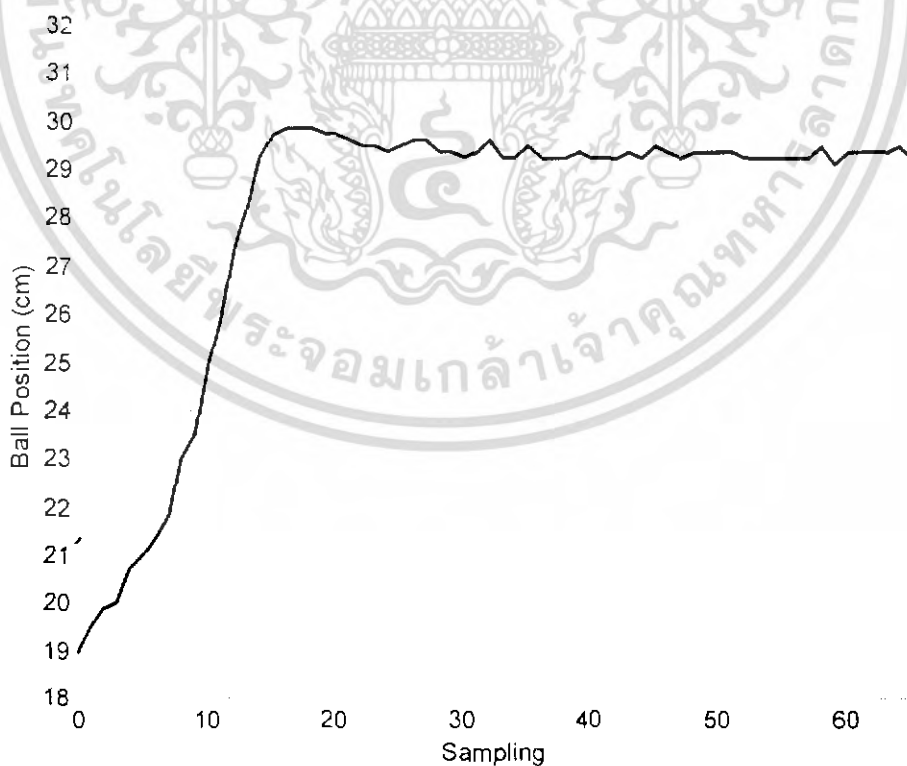


รูปที่ 4.14 การควบคุมลูกบอลจากตำแหน่ง 8 cm ไปยังตำแหน่ง 20 cm

จากรูปที่ 4.14 เป็นการควบคุมลูกบอลจากตำแหน่ง 8 เซนติเมตร ไปยังตำแหน่ง 20 เซนติเมตรของคาน ลูกบอลจะเข้าสู่จุดกำหนดที่คาบการสุมประมาณ 20 คาบ หรือเท่ากับ 3.7 วินาที กำหนดค่าขอบเขตแรงดันไฟฟ้าค่าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 5.3 โวลต์ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 7.5 โวลต์ แรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -1.8 โวลต์ และ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -4 โวลต์ แสดงสัญญาณอินพุตของการควบคุม ได้ดังรูปที่ 4.15



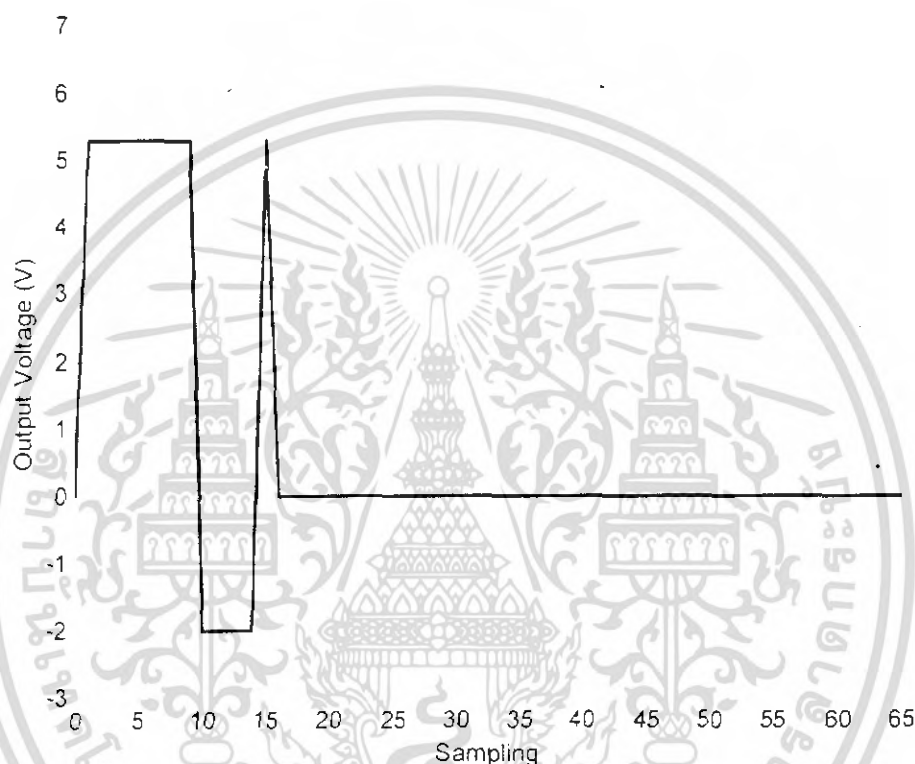
รูปที่ 4.15 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 8 cm ไปยังตำแหน่ง 20 cm



รูปที่ 4.16 การควบคุมลูกบอลจากตำแหน่ง 19 cm ไปยังตำแหน่ง 30 cm

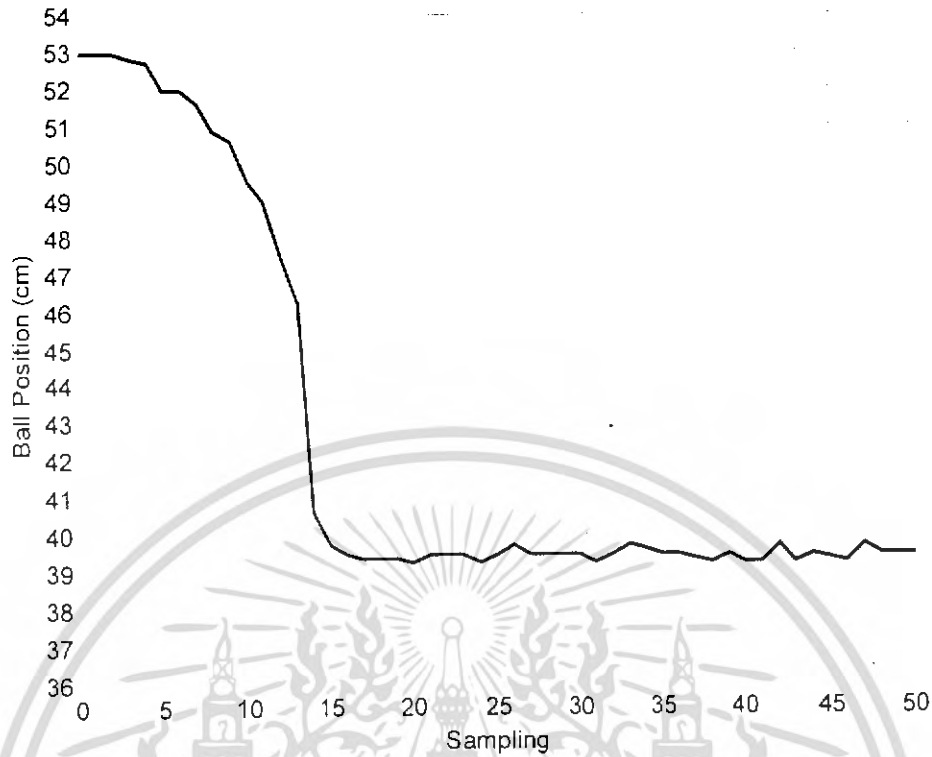
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.16 เป็นการควบคุมลูกบอลจากตำแหน่ง 19 เซนติเมตร ไปยังตำแหน่ง 30 เซนติเมตรของกาน ลูกบอลจะเข้าสู่จุดกำหนดที่คาบการสุมประมาณ 24 คาบ หรือเท่ากับ 4.44 วินาที กำหนดค่าแรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางขวาคือ 5.3 โวลต์ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางขวาคือ 7.5 โวลต์ แรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางซ้ายคือ -2 โวลต์ และ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางซ้ายคือ -4 โวลต์ แสดงสัญญาณอินพุตของการควบคุม ได้ดังรูปที่ 4.17

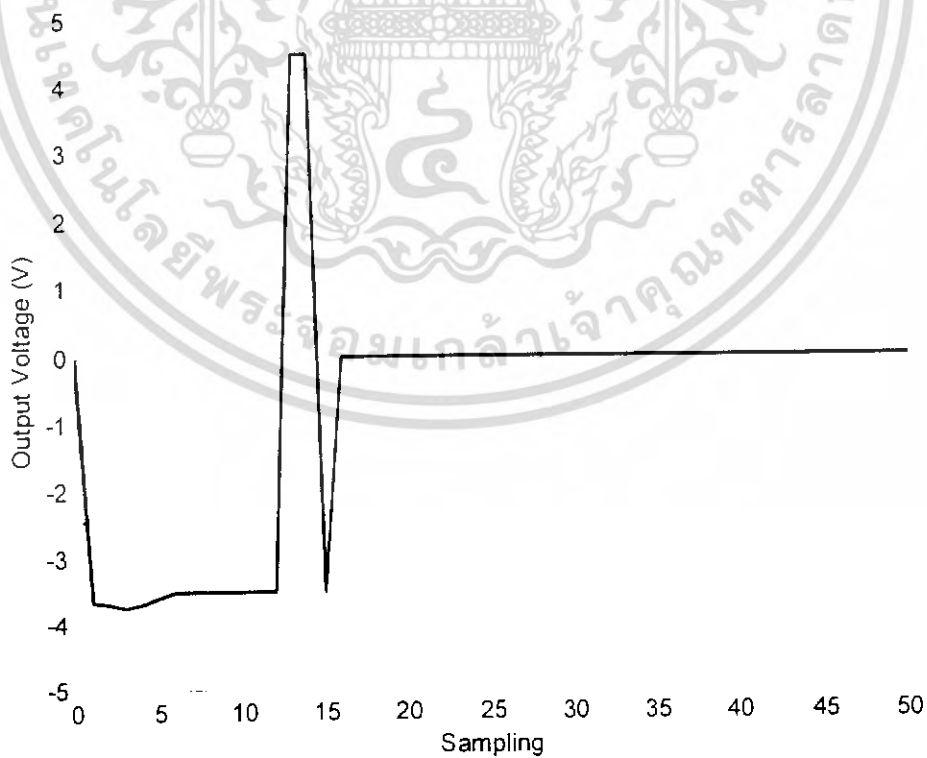


รูปที่ 4.17 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 19 cm ไปยังตำแหน่ง 30 cm

จากรูปที่ 4.18 เป็นการควบคุมลูกบอลจากตำแหน่ง 53 เซนติเมตร ไปยังตำแหน่ง 40 เซนติเมตรของกาน ลูกบอลจะเข้าสู่จุดกำหนดที่คาบการสุมประมาณ 16 คาบ หรือเท่ากับ 2.96 วินาที กำหนดค่าแรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางขวาคือ 4.5 โวลต์ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางขวาคือ 7 โวลต์ แรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางซ้ายคือ -3.2 โวลต์ และ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขั้วคานเอียงไปทางซ้ายคือ -5 โวลต์ แสดงสัญญาณอินพุตของการควบคุม ได้ดังรูปที่ 4.19



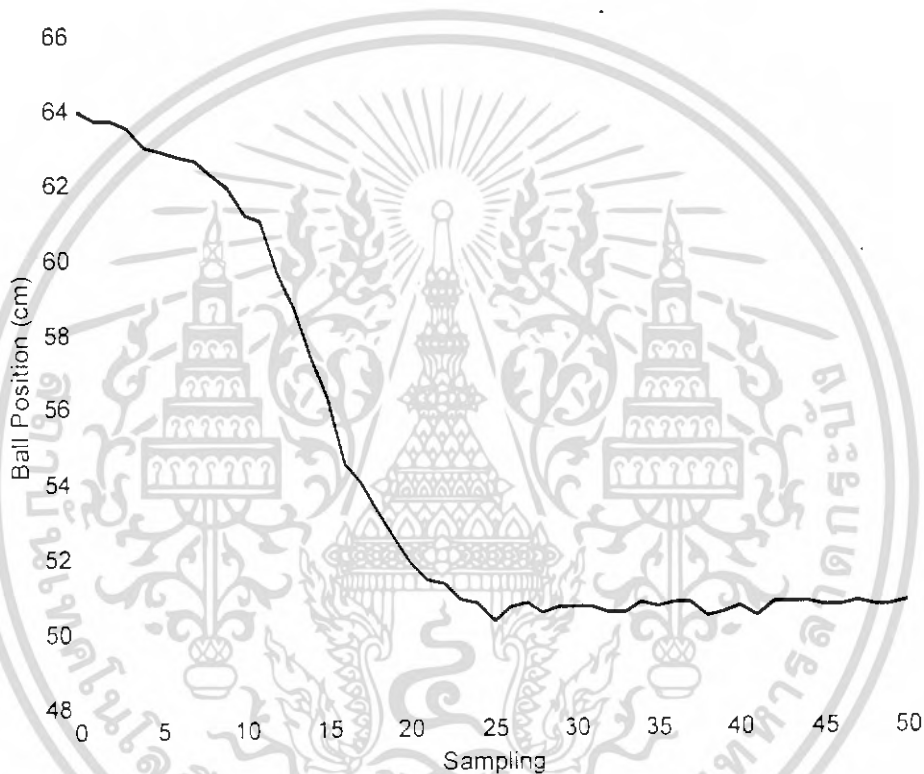
รูปที่ 4.18 การควบคุมลูกบอลจากตำแหน่ง 53 cm ไปยังตำแหน่ง 40 cm



รูปที่ 4.19 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 53 cm ไปยังตำแหน่ง 40 cm

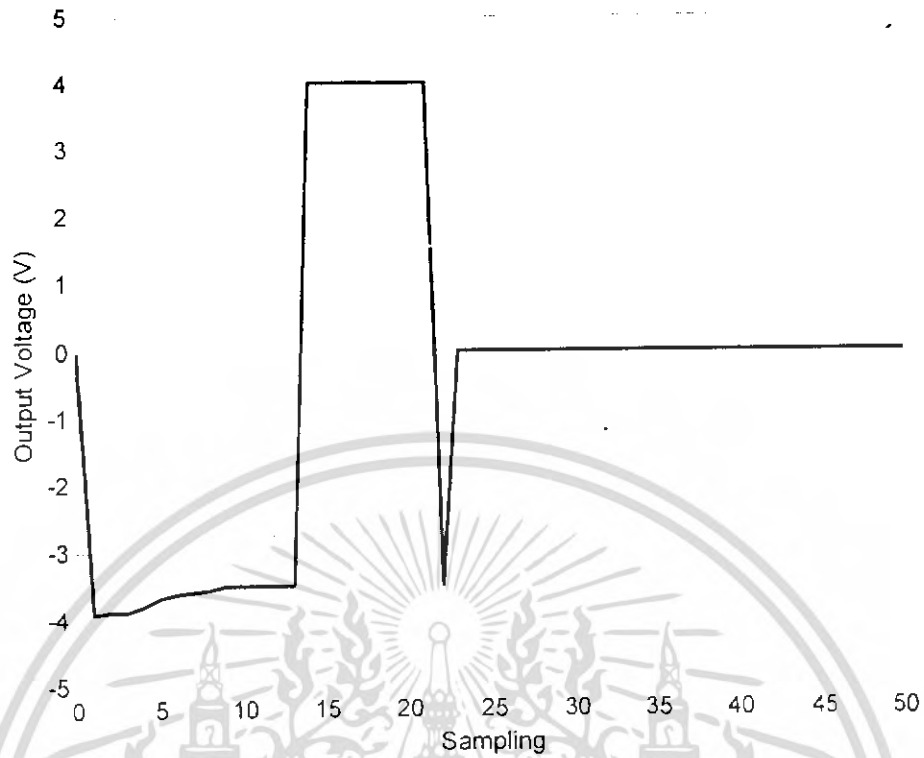
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.20 เป็นการควบคุมลูกบอลจากตำแหน่ง 64 เซนติเมตร ไปยังตำแหน่ง 50 เซนติเมตรของกาน ลูกบอลจะเข้าสู่จุดกำหนดที่คาบการสุมประมาณ 25 คาบ หรือเท่ากับ 4.625 วินาที กำหนดค่าแรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 4 โวลต์ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 5 โวลต์ แรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -3.6 โวลต์ และ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -5 โวลต์ แสดงสัญญาณอินพุตของการควบคุม ได้ดังรูปที่ 4.21

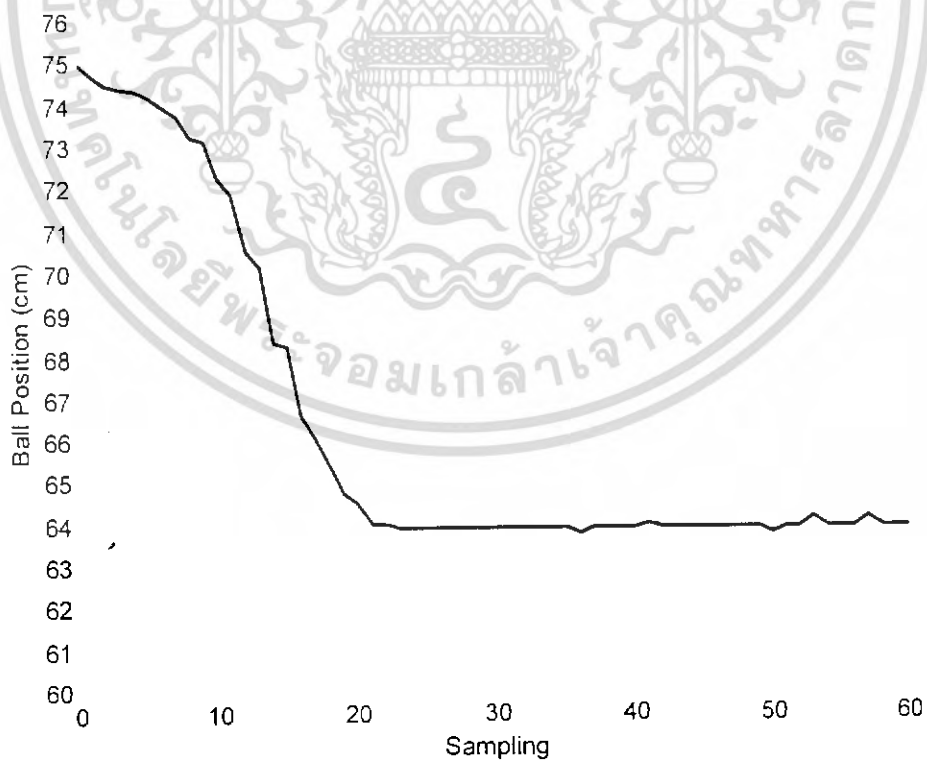


รูปที่ 4.20 การควบคุมลูกบอลจากตำแหน่ง 64 cm ไปยังตำแหน่ง 50 cm

จากรูปที่ 4.22 เป็นการควบคุมลูกบอลจากตำแหน่ง 75 เซนติเมตรไปยังตำแหน่ง 65 เซนติเมตรของกาน ลูกบอลจะเข้าสู่จุดกำหนดที่คาบการสุมประมาณ 21 คาบ หรือเท่ากับ 3.885 วินาที กำหนดค่าแรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 3.7 โวลต์ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางขวาคือ 5 โวลต์ แรงดันไฟฟ้าต่ำสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -3.6 โวลต์ และ แรงดันไฟฟ้าสูงสุดที่ทำให้มอเตอร์หมุนขับเคลื่อนไปทางซ้ายคือ -5 โวลต์ แสดงสัญญาณอินพุตของการควบคุม ได้ดังรูปที่ 4.23

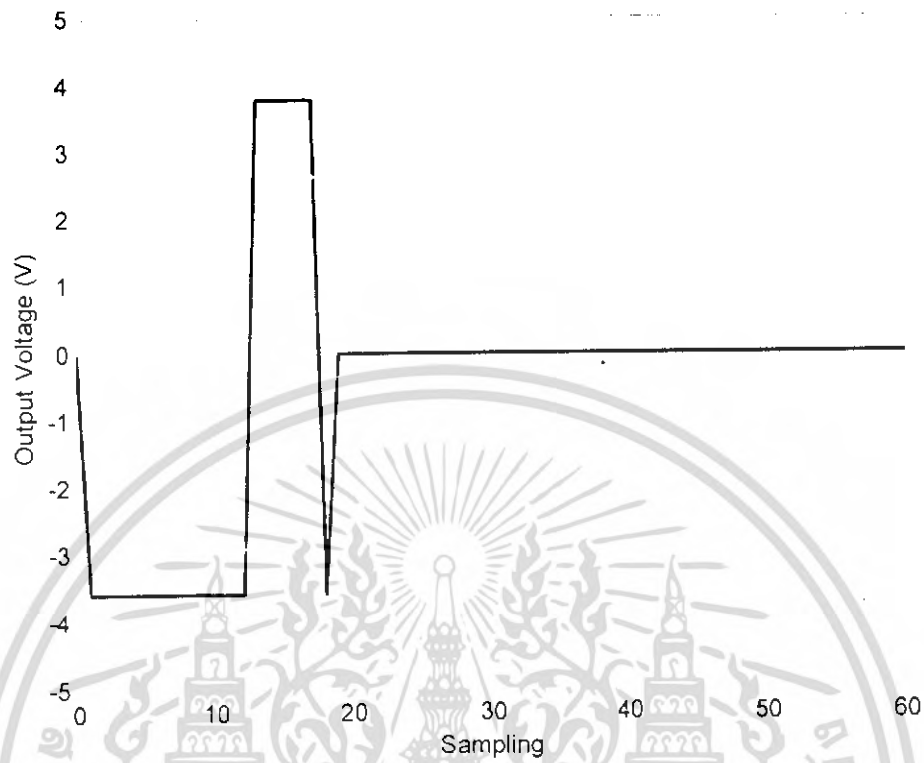


รูปที่ 4.21 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 64 cm ไปยังตำแหน่ง 50 cm



รูปที่ 4.22 การควบคุมลูกบอลจากตำแหน่ง 75 cm ไปยังตำแหน่ง 65 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 สัญญาณอินพุตสำหรับการควบคุมลูกบอลจากตำแหน่ง 75 cm ไปยังตำแหน่ง 65 cm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากการทดลองระบบควบคุมลูกบอลบนคาน เมื่อให้ลูกบอลเคลื่อนที่ไปยังตำแหน่งเป้าหมาย ระบบสามารถควบคุมลูกบอลให้เคลื่อนที่ไปยังบริเวณตำแหน่งที่ต้องการนั้นได้จริง แต่ยังคงมีความผิดพลาดของตำแหน่งลูกบอลที่อยู่บนคานกับตำแหน่งเป้าหมายอยู่เล็กน้อย เนื่องจากระบบได้ถูกออกแบบให้มีการวัดตำแหน่งของลูกบอลเพื่อใช้ในการควบคุมเพียงอย่างเดียว โดยไม่สามารถวัดมุมเอียงของคานได้ ดังนั้นในการควบคุมแบบพีดีพื้นฐาน เมื่อลูกบอลหยุดใกล้ตำแหน่งเป้าหมายแต่ยังมีค่าความผิดพลาดอยู่ ระบบจะยังสั่งให้มีสัญญาณควบคุมแก้มอเตอร์ไปขับเคลื่อนเพื่อเลี้ยงลูกบอลไปยังตำแหน่งเป้าหมาย แต่เมื่อมีสัญญาณแรงดันเข้าไปที่มอเตอร์เพียงเล็กน้อยมอเตอร์ก็อาจจะขับเคลื่อนให้เอียงมากเกินไป จึงทำให้ลูกบอลถ่วงเร็ว มีผลให้ไม่สามารถควบคุมลูกบอลให้เข้าสู่ตำแหน่งเป้าหมายได้และเกิดการแกว่งรอบตำแหน่งเป้าหมาย หรือถึงแม้จะได้แก้ไขโปรแกรมให้ยอมรับช่วงค่าความผิดพลาด แต่ในกรณีนี้เมื่อลูกบอลเข้าสู่บริเวณที่ยอมรับได้แล้วระบบก็จะสั่งให้มอเตอร์หยุดหมุน โดยไม่ได้พิจารณาว่าคานยังคงเอียงอยู่หรือไม่ ซึ่งอาจทำให้ลูกบอลยังถ่วงต่อไปได้ในกรณีที่ยังคงมีความเอียงของคานอยู่ จึงเกิดการแกว่งรอบตำแหน่งเป้าหมายเช่นกัน ซึ่งปัญหาทั้งสองนี้เกิดจากที่ระบบไม่สามารถใช้มุมเอียงของคานในการควบคุมได้ ดังนั้นต้องมีการเขียนโปรแกรมเพื่อแก้ไขข้อผิดพลาดเหล่านี้ตามเนื้อหาที่ได้กล่าวมาแล้วในหัวข้อ 4.3.3

5.2 ปัญหาที่พบและแนวทางแก้ไข

จากการศึกษาและทำโครงงานนี้ในช่วงแรกเกิดปัญหาคือ ใช้เวลาในการศึกษาโปรแกรมรับค่าจากกล้องดิจิตอลวีดีโอเพื่อระบุตำแหน่งของลูกบอลนาน เนื่องจากไม่เคยศึกษาโปรแกรมภาษาวิซวลซีพลัสพลัสมาก่อน นอกจากนั้น เมื่อได้ทำการทดลองพบว่ามอเตอร์ไฟฟ้ากระแสตรงที่ใช้มีความเร็วรอบต่ออนาทีมากเกินไป จึงเปลี่ยนมอเตอร์ให้มีความเร็วรอบต่ออนาทีน้อยลง แต่ก็ยังเร็วเกินไป จึงควรถอดรอบพูลส์ของเพลามอเตอร์กับพูลส์ของเพลารับคานให้มีอัตราทดเพิ่มขึ้น หรือเปลี่ยนมอเตอร์ใหม่ให้มีความเร็วรอบต่ออนาทีลดลง

ในส่วนปัญหาที่พบและยังไม่ได้ปรับปรุงแก้ไข ได้แก่ ปัญหาที่ระบบไม่มีการวัดและควบคุมมุมเอียงของคาน ทำให้การควบคุมทำได้ยากและต้องใช้เงื่อนไขอื่นๆ เพื่อปรับปรุงผลตอบสนองของระบบดังหัวข้อ 4.3 ซึ่งพบถึงแม้จะทำให้ระบบสามารถควบคุมตำแหน่งลูกบอลบนคานได้ แต่ยังคงยอมรับค่าความผิดพลาดที่เกิดขึ้นพอสมควร ดังนั้นวิธีที่ดีที่สุดควรมีการติดตั้งโพเทนทิโอเมทริกเป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการศึกษาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มิเตอร์เพื่อวัดตำแหน่งมุมเอียงของคาน เพื่อให้ระบบสามารถนำความเอียงของคานมาใช้ในการพิจารณาในการควบคุมตำแหน่งลูกบอลเพื่อให้มีค่าความผิดพลาดลดลง

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

ตัวควบคุมที่ใช้ในปัจจุบันนี้มีอยู่อีกหลายแบบ ควรศึกษาเพิ่มเติมว่ามีตัวควบคุมแบบใดบ้างที่เหมาะสมกับระบบควบคุมตำแหน่งลูกบอลบนคาน และควรวางแบบจำลองทางคณิตศาสตร์ของระบบซึ่งจะทำให้สามารถเลือกใช้และออกแบบตัวควบคุมได้ง่ายขึ้น อีกทั้งยังสามารถนำแบบจำลองทางคณิตศาสตร์ที่ได้ศึกษาไปเลียนแบบระบบจริง(Simulate) ในโปรแกรมคอมพิวเตอร์เพื่อทดสอบระบบที่ได้ออกแบบไว้ได้ก่อนการทำการทดลองจริงอีกด้วย

นอกจากนั้นโปรแกรมรับค่าจากกล้องดิจิตอลวีดีโอของ QCSDK ไม่สามารถใช้งานร่วมกับกล้องวีดีโออื่น ๆ หรือรุ่นอื่นๆ ของลอจitech ได้ จึงควรศึกษาหาโปรแกรมตัวใหม่เพื่อให้ใช้งานได้กับกล้องดิจิตอลวีดีโอที่หลากหลายขึ้น



เอกสารอ้างอิง

- [1] สิทธิศักดิ์ มณีแสง. “การควบคุมตำแหน่งลูกบอลบนคาน.” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2545.
- [2] สัจवाल บกสุวรรณ, สุเมธ จันทร์ทำจัน. “ระบบติดตามวัตถุ.” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2546.
- [3] ปิยะภูมิ สิทธิสูงเนิน, วราวุธ พรินทรากุล. “การควบคุมอากาศยานจำลอง.” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2548.
- [4] ประจัน พลังสันติกุล. เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : อินโนเวทีฟ เอ็กเพอริเมนต์. 2547.
- [5] นิรุช อำนวยศิลป์. คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0 ฉบับเพื่อการใช้งานจริง. กรุงเทพมหานคร : ชัคเชส นีเดีย.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม

โปรแกรมวิซวลซีพลัสพลัส ซึ่งเขียนเพื่อใช้ในการตรวจสอบตำแหน่งลูกบอลและควบคุม ประกอบด้วย หน้าต่างเชื่อมต่อกับผู้ใช้(Graphical User Interface) ดังรูปที่ ก.1 และโค้ดโปรแกรม ดังต่อไปนี้



รูปที่ ก.1 หน้าต่างเชื่อมต่อกับผู้ใช้ใน โปรแกรมการตรวจสอบตำแหน่งลูกบอลและควบคุม

```
// tripodDlg.cpp : implementation file
```

```
#include <conio.h>
```

```
#include "stdafx.h"
```

```
#include "tripod.h"
```

```
#include "tripodDlg.h"
```

```
#include "LVServerDefs.h"
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <math.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

int count=0 , count1=0 , col , inc=0;

float data.error=0, err_n=0, err_o=0, Ts=0, Ki_o=0, Kp=0, Ki=0, Kd=0, PID=0, out_p=0, V_input=0, m, m1, m2, m3;

float induce[50], err[2000], Set_point, Kp_in, Ki_in, Kd_in, CompensatorL, CompensatorH;

float BCDH_Left, BCDH_Right, BCDL_Left, BCDL_Right, VLeftH, VLeftL, VRightH, VRightL;

struct _jobuf *textf, *textf1, *textf2;

// ////////////////////////////////////////

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

    Dialog Data
    //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //{{AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CAboutDlg)

protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    //{{AFX_VIRTUAL

    Implementation

protected:

    //{{AFX_MSG(CAboutDlg)
    //{{AFX_MSG
    DECLARE_MESSAGE_MAP()

};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //{{AFX_DATA_INIT

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    ///AFX_DATA_MAP(CAboutDlg)
    ///AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    ///AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    ///AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// C tripod dialog
CTripodDlg::CTripodDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CTripodDlg::IDD, pParent)
{
    ///AFX_DATA_INIT(CTripodDlg)
    m_Ball_P = 0.0f;
    m_Ts = 0.0f;
    m_Kd = 0.0f;
    m_Ki = 0.0f;
    m_Kp = 0.0f;
    m_CompensatorH = 0.0f;
    m_CompensatorL = 0.0f;
    m_VLeftH = 0.0f;
    m_VRightH = 0.0f;
    m_VLeftL = 0.0f;
    m_VRightL = 0.0f;
    ///AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win5x
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);

    ////////////////////////////////////// Set destination BMP to NULL first
    m_destinationBitmapInfoHeader = NULL;
}

//////////////////////////////////// Additional generic functions

static unsigned PixelBytes(int w, int bpp)
{
    return (w * bpp + 7) / 8;
}

static unsigned DIBRowSize(int w, int bpp)
{
    return (w * bpp + 31) / 32 * 4;
}

```

```

static unsigned DibRowSize(LPBITMAPINFOHEADER pbi)
    {return DibRowSize(pbi->biWidth, pbi->biBitCount);}

static unsigned DibRowPadding(int w, int bpp)
    {return DibRowSize(w, bpp) - PixelBytes(w, bpp);}

static unsigned DibRowPadding(LPBITMAPINFOHEADER pbi)
    {return DibRowPadding(pbi->biWidth, pbi->biBitCount);}

static unsigned DibImageSize(int w, int h, int bpp)
    {return h * DibRowSize(w, bpp);}

static size_t DibSize(int w, int h, int bpp)
    {return sizeof (BITMAPINFOHEADER) + DibImageSize(w, h, bpp);}

##### end of generic functions

void CTripodDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    ::AFX_DATA_MAP(CTripodDlg)
    DDX_Control(pDX, IDC_PROCESSEDDVIEW, m_eVideoProcessedView);
    DDX_Control(pDX, IDC_UNPROCESSEDDVIEW, m_eVideoUnprocessedView);
    DDX_Text(pDX, IDC_Ball_P2, m_Ball_P2);
    DDX_Text(pDX, IDC_Ts2, m_Ts);
    DDX_Text(pDX, IDC_Kd2, m_Kd);
    DDX_Text(pDX, IDC_Ki2, m_Ki);
    DDX_Text(pDX, IDC_Kp2, m_Kp);
    DDX_Text(pDX, IDC_V_HN, m_VLeftH);
    DDX_Text(pDX, IDC_V_HP, m_VRightH);
    DDX_Text(pDX, IDC_V_LN, m_VLeftL);
    DDX_Text(pDX, IDC_V_LP, m_VRightL);
    ::AFX_DATA_MAP

}

BEGIN_MESSAGE_MAP(CTripodDlg, CDialog)
    ::AFX_MSG_MAP(CTripodDlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDEXIT, OnExit)
    ON_BN_CLICKED(IDStart, OnStart)
    ON_BN_CLICKED(IDStop, OnStop)
    ::AFX_MSG_MAP
END_MESSAGE_MAP()

```

```

////////////////////////////////////
// CTripodDlg message handlers
BOOL CTripodDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
                strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here

    return TRUE; // return TRUE unless you set the focus to a control
}

void CTripodDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else {
        CDialog::OnSysCommand(nID, lParam);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

// If you add a minimize button to your dialog, you will need the code below
 // to draw the icon. For MFC applications using the document/view model,
 // this is automatically done for you by the framework.

```
void CTripodDlg::OnPaint()
{
    if (!IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon - 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }

    // The system calls this to obtain the cursor to display while the user drags
    // the minimized window.
    HCURSOR CTripodDlg::OnQueryDragIcon()
    {
        return (HCURSOR) m_hIcon;
    }

    //ปุ่ม Start
    void CTripodDlg::OnStart()
    {
        // For Unprocessed view: videoportal (top one)
        char sRegUnprocessedView[] = "HKEY_LOCAL_MACHINE\\Software\\UnprocessedView";
        m_cVideoUnprocessedView.PrepareControl("UnprocessedView", sRegUnprocessedView, 0);
        m_cVideoUnprocessedView.EnableUIElements(UIELEMENT_STATUSBAR, 0, TRUE);
        m_cVideoUnprocessedView.ConnectCamera2();
        m_cVideoUnprocessedView.SetEnablePreview(TRUE);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// For binary view videoportal (bottom one)
char sRegProcessedView[] = "HKEY_LOCAL_MACHINE\\Software\\ProcessedView";
m_cVideoProcessedView.PrepareControl("ProcessedView", sRegProcessedView, 0);
m_cVideoProcessedView.EnableUIElements(UIELEMENT_STATUSBAR,0,TRUE);
m_cVideoProcessedView.ConnectCamera2();
m_cVideoProcessedView.SetEnablePreview(TRUE);

Initialize the size of binary videoportal
m_cVideoProcessedView.SetPreviewMaxHeight(240);
m_cVideoProcessedView.SetPreviewMaxWidth(320);

Uncomment if you wish to fix the live videoportal's size
m_cVideoUnprocessedView.SetPreviewMaxHeight(240);
m_cVideoUnprocessedView.SetPreviewMaxWidth(320);

Find the screen coordinates of the binary videoportal
m_cVideoProcessedView.GetWindowRect(m_rectForProcessedView);
ScreenToClient(m_rectForProcessedView);
allocateDib(CSize(320, 240));

// Start grabbing frame data for Processed videoportal (bottom one)
m_cVideoProcessedView.StartVideoHook(0);
text1=fopen("ballp.txt", "w");
text2=fopen("sampl.txt", "w");
text3=fopen("commd.txt", "w");
;

//ปุ่ม Stop
void CTripodDlg::OnStop()
;

// TODO: Add your control notification handler code here

// Kill live view videoportal (top one)
m_cVideoUnprocessedView.StopVideoHook(0);
m_cVideoUnprocessedView.DisconnectCamera();

// Kill binary view videoportal (bottom one)
m_cVideoProcessedView.StopVideoHook(0);
m_cVideoProcessedView.DisconnectCamera();

_outp(0x378,127,5);
inc = 0;
Ki_o = 0;
sampi = 0;

fclose(text1);
fclose(text2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        fclose(textf2);
    }

    //ปุ่ม Exit
void CTripodDlg::OnExit()
{ // Kill program
    fclose(textf);
    fclose(textf1);
    fclose(textf2);
    DestroyWindow();
    _outp(0x378,127.5);
}

BEGIN_EVENTSINK_MAP(CTripodDlg, CDialog)
    {{{AFX_EVENTSINK_MAP(CTripodDlg)
        ON_EVENT(CTripodDlg, IDC_PROCESSEDVIEW, 1 /* PortalNotification */, OnPortalNotificationProcessedview,
VTS_I4 VTS_I4 VTS_I4 VTS_I4)
        {{{AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()
void CTripodDlg::OnPortalNotificationProcessedview(long lMsg, long lParam1, long lParam2, long lParam3)
{
    // TODO: Add your control notification handler code here
    // This function is called at the camera's frame rate

#define NOTIFICATIONMSG_VIDEOHOOK 10

    // Declare some useful variables
    // QCSDKMFC.pdf (Quickcam MFC documentation) p. 103 explains the variables lParam1, lParam2, lParam3 too

    LPBITMAPINFOHEADER lpBitmapInfoHeader; // Frame's info header contains info like width and height
    LPBYTE lpBitmapPixelData; // This pointer-to-long will point to the start of the frame's pixel data
    unsigned long lTimeStamp; // Time when frame was grabbed

    switch(lMsg) {
        case NOTIFICATIONMSG_VIDEOHOOK:

            lpBitmapInfoHeader = (LPBITMAPINFOHEADER) lParam1;
            lpBitmapPixelData = (LPBYTE) lParam2;
            lTimeStamp = (unsigned long) lParam3;
            grayScaleTheFrameData(lpBitmapInfoHeader, lpBitmapPixelData);
            doMyImageProcessing(lpBitmapInfoHeader);
            // Place where you'd add your image processing code
            displayMyResults(lpBitmapInfoHeader);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    default:
        break;
    }
}

void CTripodDlg::allocateDib(CSize sz)
{
    // Purpose: allocate information for a device independent bitmap (DIB)
    // Called from OnInitVideo
    if(m_destinationBitmapInfoHeader) {
        free(m_destinationBitmapInfoHeader);
        m_destinationBitmapInfoHeader = NULL;
    }

    if(sz.cx | sz.cy) {
        m_destinationBitmapInfoHeader = (LPBITMAPINFOHEADER)malloc(sizeof(BITMAPINFOHEADER));
        ASSERT(m_destinationBitmapInfoHeader);
        m_destinationBitmapInfoHeader->biSize = sizeof(BITMAPINFOHEADER);
        m_destinationBitmapInfoHeader->biWidth = sz.cx;
        m_destinationBitmapInfoHeader->biHeight = sz.cy;
        m_destinationBitmapInfoHeader->biPlanes = 1;
        m_destinationBitmapInfoHeader->biBitCount = 24;
        m_destinationBitmapInfoHeader->biCompression = 0;
        m_destinationBitmapInfoHeader->biSizeImage = DIBImageSize(sz.cx, sz.cy, 24);
        m_destinationBitmapInfoHeader->biXPelsPerMeter = 0;
        m_destinationBitmapInfoHeader->biYPelsPerMeter = 0;
        m_destinationBitmapInfoHeader->biClrImportant = 0;
        m_destinationBitmapInfoHeader->biClrUsed = 0;
    }
}

void CTripodDlg::displayMyResults(LPBITMAPINFOHEADER lpThisBitmapInfoHeader)
{
    // displayMyResults: Displays results of doMyImageProcessing() in the videoport
    // Notes: StretchDIBits stretches a device-independent bitmap to the appropriate size

    CDC *pDC;          // Device context to display bitmap data

    pDC = GetDC();
    int nOldMode = SetStretchBltMode(pDC->GetSafeHdc(), COLORONCOLOR);

    StretchDIBits(
        pDC->GetSafeHdc(),
        m_rectForProcessedView, left, top, right, bottom,
        lpThisBitmapInfoHeader->biWidth, lpThisBitmapInfoHeader->biHeight,
        0, 0, DIB_RGB_COLORS, NULL);
}

```

```

        m_rectForProcessedView.top,           // videoportal top-most coordinate
        m_rectForProcessedView.Width(),      // videoportal width
        m_rectForProcessedView.Height(),     // videoportal height
        0,                                   // Row position to display bitmap in videoportal
        0,                                   // Col position to display bitmap in videoportal
        lpThisBitmapInfoHeader->biWidth,     // m_destinationBmp's number of columns
        lpThisBitmapInfoHeader->biHeight,    // m_destinationBmp's number of rows
        m_destinationBmp,
        // The bitmap to display; use the one resulting from doMyImageProcessing
        (BITMAPINFO*)m_destinationBitmapInfoHeader,
        // The bitmap's header info e.g. width, height, number of bits etc
        DIB_RGB_COLORS,                      // Use default 24-bit color table
        SRCCOPY                               // Just display
    );

    SetStretchBltMode(pDC->GetSafeHdc(),nOldMode);

    ReleaseDC(pDC);
}

//แปลงภาพที่รับเข้ามาเป็นภาพของระดับสีเทา
void CTripodDlg::grayScaleTheFrameData(LPBITMAPINFOHEADER lpThisBitmapInfoHeader, LPBYTE
lpThisBitmapPixelData)
{
    // grayScaleTheFrameData: Called by CTripodDlg::OnPortalNotificationBinaryview
    // Task: Read current frame pixel data and computes a gray scale version

    unsigned int    W, H;                    // Width and Height of current frame [pixels]
    BYTE            *sourceBmp;              // Pointer to current frame of data
    unsigned int    row, col;
    unsigned long   i;
    BYTE            grayValue;
    BYTE            redValue;
    BYTE            greenValue;
    BYTE            blueValue;

    W = lpThisBitmapInfoHeader->biWidth; // biWidth: number of columns
    H = lpThisBitmapInfoHeader->biHeight; // biHeight: number of rows

    // Store pixel data in row-column vector format
    // Recall that each pixel requires 3 bytes (red, blue and green bytes)
    // m_destinationBmp is a protected member and declared in binarizeDlg.h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

m_destinationBmp = (BYTE*)malloc(H*3*W*sizeof(BYTE));

// Point to the current frame's pixel data
sourceBmp = lpThisBitmapPixelData;

for (row = 110; row < 131; row++) {
    for (col = 0; col < W; col++) {

        // Recall each pixel is composed of 3 bytes
        i = (unsigned long)(row*3*W + 3*col);

        // The source pixel has a blue, green and red value:
        blueValue = *(sourceBmp + i);
        greenValue = *(sourceBmp + i + 1);
        redValue = *(sourceBmp + i + 2);

        // A standard equation for computing a grayscale value based on RGB values
        grayValue = (BYTE)(0.299*redValue + 0.587*greenValue + 0.114*blueValue);

        // The destination BMP will be a grayscale version of the source BMP
        *(m_destinationBmp + i) = grayValue;
        *(m_destinationBmp + i + 1) = grayValue;
        *(m_destinationBmp + i + 2) = grayValue;
    }
}

void CTripodDlg::doMyImageProcessing(LPBITMAPINFOHEADER lpThisBitmapInfoHeader)
{
    // doMyImageProcessing: This is where you'd write your own image processing code
    // Task: Read a pixel's grayscale value and process accordingly
    unsigned int W, H; // Width and Height of current frame [pixels]
    unsigned int row; // Pixel's row and col positions
    unsigned long i; // Dummy variable for row-column vector
    BYTE thresholdValue; // Value to threshold grayvalue

    char str[80]; // To print message
    char str1[80];
    char str2[80];
    char str3[80];
    char str4[80];
    char str5[80];
    char str6[80];
    char str7[80];
    char str8[80];
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char          str9[80];
char          str10[80];
char          str11[80];
char          str12[80];
char          str13[80];
char          str14[80];

CDC           *pDC;                               Device context need to print message

```

```

W = lpThisBitmapInfoHeader->biWidth; // biWidth: number of columns

```

```

H = lpThisBitmapInfoHeader->biHeight; // biHeight: number of rows

```

```

// In this example, the grayscale image (stored in m_destinationBmp) is
// thresholded to create a binary image. A threshold value close to 255
// means that only colors close to white will remain white in binarized
// BMP and all other colors will be black

```

```

thresholdValue = 50; // ค่าเทรชโวลที่ใช้งาน

```

```

// Map message from Text Box.

```

```

UpdateData(TRUE);

```

```

Set_point = m_Ball_P;

```

```

Ts      = m_Ts;

```

```

Kp_in  = m_Kp;

```

```

Ki_in  = m_Ki;

```

```

Kd_in  = m_Kd;

```

```

VLeftH = m_VLeftH;

```

```

VRightH = m_VRightH;

```

```

VLeftL = m_VLeftL;

```

```

VRightL = m_VRightL;

```

```

CompensatorL = 1.5;

```

```

// ตรวจสอบตำแหน่งของลูกบอลที่อยู่บนคาน

```

```

// Scan ball position on the beam.

```

```

for (row = 120; row < 121; row++) {

```

```

    for (col = 0; col < W; col++) {

```

```

        // Recall each pixel is composed of 3 bytes

```

```

        i = (unsigned long)(row*3*W + 3*col);

```

```

        // Add your code to operate on each pixel. For example

```

```

        // *(m_destinationBmp + i) refers to the ith pixel in the destinationBmp

```

```

        // Since destinationBmp is a 24-bit grayscale image, you must also apply

```

```

        // the same operation to *(m_destinationBmp + i + 1) and *(m_destinationBmp + i + 2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในวิชาเท่านั้น การคัดลอกหรือการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Threshold: if a pixel's grayValue is less than thresholdValue
iff(*(m_destinationBmp + i) <= thresholdValue)
{
    induce[count] = col;
    count++;
}
else {
    if (count < 5 )
        count = 0;
    if (count >= 5)
        {
            m = ((induce[count-1] - induce[0])/2.0) +
                induce[0];
            m1 = induce[0];
            m2 = induce[count-1];
            m3 = (80*m)/W;
            count = 0;
            //Keep ball position to text file
            fprintf(textf, "%f\n", m3);
            fprintf(textf1, "%f\n", samp1);
            fprintf(textf2, "%f\n", V_input);
            error = Set_point - m3;
            err[inc] = error;
            inc++;
            pDC = GetDC();
            sprintf(str, "Ball Position is %f pixel = %f cm. on
            beam ".m,m3);
            sprintf(str1, "First point is %f pixel", m1);
            sprintf(str2, "Last point is %f pixel", m2);
            sprintf(str3, "Set point = %f ", Set_point);
            sprintf(str4, "Ts = %f ", Ts);
            sprintf(str5, "Kp = %f ", Kp_in);
            sprintf(str6, "Ki = %f ", Ki_in);
            sprintf(str7, "Kd = %f ", Kd_in);
            sprintf(str8, "Error = %f ", error);
            sprintf(str9, "Gain Kp = %f ", Kp);
            sprintf(str10, "Gain Ki = %f ", Ki);
            sprintf(str11, "Gain Kd = %f ", Kd);
            sprintf(str12, "Gain PID = %f ", PID);
            sprintf(str13, "Out Port = %f ", out_p);
            sprintf(str14, "Command Signal = %f ", V_input);
            pDC->TextOut(20, 285, str);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นองานใดไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pDC->TextOut(20, 305, str1);
pDC->TextOut(20, 325, str2);
pDC->TextOut(20, 345, str9);
pDC->TextOut(20, 365, str10);
pDC->TextOut(20, 385, str11);
pDC->TextOut(20, 405, str12);
pDC->TextOut(20, 425, str13);
pDC->TextOut(450, 285, str3);
pDC->TextOut(450, 305, str4);
pDC->TextOut(450, 325, str5);
pDC->TextOut(450, 345, str6);
pDC->TextOut(450, 365, str7);
pDC->TextOut(450, 385, str8);
pDC->TextOut(450, 405, str9);
ReleaseDC(pDC);

// จุดเซตค่าหนึ่งล่างหน้า
if (err[0] >= 0)
    CompensatorH = err[0] * 0.55;
else if (err[0] < 0)
    CompensatorH = err[0] * -0.55;

if (error < -CompensatorH)
    err_n = Set_point - m3;
if (error > CompensatorH)
    err_n = Set_point - m3;

if (error >= -CompensatorH && error < -CompensatorL)
    err_n = m3 - Set_point;

if (error > CompensatorL && error <= CompensatorH)
    err_n = m3 - Set_point;

if (error >= -CompensatorL && error <= CompensatorL)
    err_n = 0;
}

// ตัวควบคุม
Kp = Kp_in * err_n;
Ki = Ki_o + (Ki_in * ((Ts/2) * (err_o - err_n)));
Kd = ((err_n - err_o)/Ts) * Kd_in;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ขอสงวนสิทธิ์ในสิ่งตีพิมพ์และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PID = Kp + Ki + Kd;
data = _inp(0x379);

// Convert Voltage to BCD
BCDL_Right = 127.5 + ((127.5 * VRightL) / 12); // BCD low for motor turn right
BCDH_Right = 127.5 + ((127.5 * VRightH) / 12); // BCD high for motor turn right
BCDL_Left = 127.5 - ((127.5 * VLeftL) / 12); // BCD low for motor turn left
BCDH_Left = 127.5 - ((127.5 * VLeftH) / 12); // BCD high for motor turn right
;

out_p = PID + 127.5;
if (PID > 0)
{
    if (out_p <= BCDL_Right)
        out_p = BCDL_Right;
    if (out_p >= BCDH_Right)
        out_p = BCDH_Right;
}
if (PID < 0)
{
    if (out_p >= BCDL_Left)
        out_p = BCDL_Left;
    if (out_p <= BCDH_Left)
        out_p = BCDH_Left;
}
else if (PID == 0)
{
    out_p = 127.5;
}

// ตรวจสอบลิมิตสวิตช์

if (data == 119)
{
    if (err_n > 0)
    {
        _outp(0x378, out_p);
    }
    if (err_n <= 0)
    {
        _outp(0x378, 127.5);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข-1 เอกสารคู่มือการใช้งาน DAC0832

ไอซี DAC0832 เป็นอุปกรณ์แปลงสัญญาณดิจิทัลเป็นอนาล็อก 8 บิต มีรายละเอียดต่างๆ ดังนี้



March 2002

DAC0830/DAC0832

8-Bit μ P Compatible, Double-Buffered D to A Converters

General Description

The DAC0830 is an advanced CMOS/SiC-MOS 8-bit multiplying DAC designed to interface directly with the 6800, 6840, 6865, Z80[™], and other popular microprocessors. A deposited silicon-chromium R-2R resistor ladder network provides the reference current and provides the circuit with excellent temperature tracking characteristics (0.05% of Full Scale Range maximum linearity error over temperature). The circuit uses CMOS current switches and control logic to achieve low power consumption and low output leakage current errors. Special circuitry provides TTL logic input voltage level compatibility.

Double buffering allows these DACs to output a voltage corresponding to one digital word while holding the next digital word. This permits the simultaneous updating of any number of DACs.

The DAC0830 series are the 8-bit members of a family of microprocessor-compatible DACs (MICRO-DAC[™]).

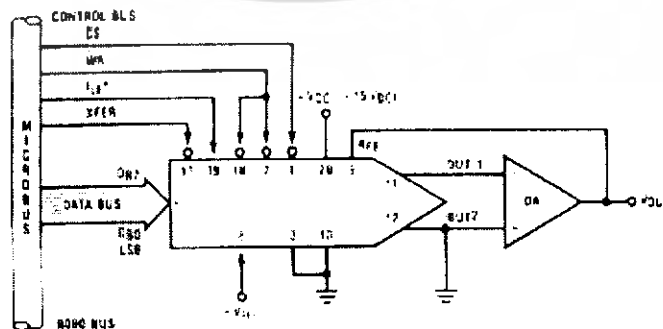
Features

- Double-buffered, single-buffered or flow-through digital data inputs
- Easy interchange and pin-compatible with 12-bit DAC1250 series
- Direct interface to all popular microprocessors
- Linearity specified with zero and full scale adjust only — NOT BEST STRAIGHT LINE FIT
- Works with ± 10 V reference full 4-quadrant multiplication
- Can be used in the voltage switching mode
- Logic inputs which meet TTL voltage level specification (4V logic threshold)
- Operates STAND ALONE without μ P (if desired)
- Available in 20-pin small-outline or molded chip carrier package

Key Specifications

- Current settling time: 1 μ s
- Resolution: 3 bits
- Linearity: 8, 9, or 10 bits (guaranteed over temp)
- Gain Tempco: 0.0001% FS/°C
- Low power dissipation: 20 mW
- Single power supply: 5 to 15 Vcc

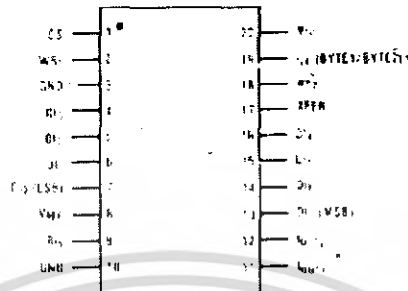
Typical Application



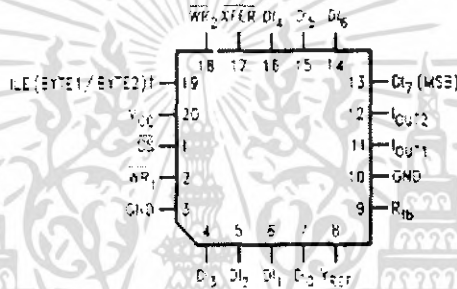
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagrams (Top Views)

Dual-In-Line and Small-Outline Packages



Molded Chip Carrier Package



Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V _{CC})	-1 to +5.5
Voltage of Any Digital Input	V _{CC} to GND
Voltage of Logic Input	-0.5 to +2.5
Storage Temperature Range	-65°C to +150°C
Package Dissipation	500 mW
at T _a = 25°C (Note 3)	
DIC Voltage Applied to	-100 mV to +50
Output (Note 4)	500V
ESD Susceptibility (Note 4)	500V
Lead Temperature (Soldering, 10 sec)	

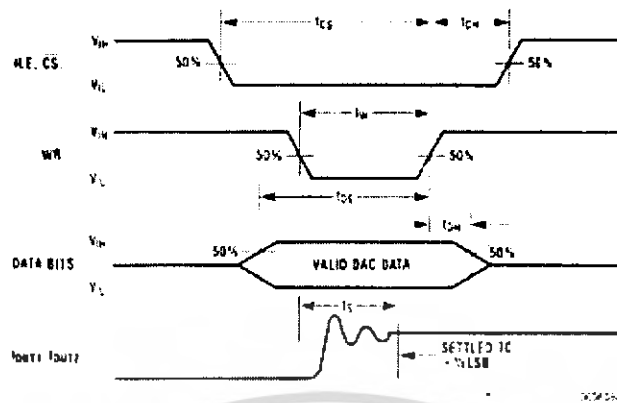
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Weld Phase (60 sec)	215°C
Infrared (15 sec)	220°C

Operating Conditions

Temperature Range	T _{min} to T _{max}
Part numbers with "LCM" suffix	0°C to +70°C
Part numbers with "LCWM" suffix	0°C to +70°C
Part numbers with "LC" suffix	0°C to +70°C
Part numbers with "LCU" suffix	-40°C to +65°C
Part numbers with "L" suffix	-55°C to +125°C
Voltage of Any Digital Input	V _{CC} to GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switching Waveform



Definition of Package Pinouts

Control Signals

(All control signals level actuated)

CS: Chip Select (active low). The CS in combination with ILE will enable WR.

ILE: Input Latch Enable (active high). The ILE in combination with CS enables WR.

WR₁: Write 1. The active low WR₁ is used to load the digital input data bits (D_n) into the input latch. The data in the input latch is latched when WR₁ is high. To update the input latch CS and WR₁ must be low while ILE is high.

WR₂: Write 2 (active low). This signal in combination with VREF causes the 8-bit data which is available in the input latch to transfer to the DAC register.

XFER: Transfer control signal (active low). The XFER will erase WR₂.

VREF:

Reference Voltage Input. This input connects an external precision voltage source to the internal R-2R ladder. It can be selected over the range of +10 to -10. This is also the analog voltage input for a 4-quadrant multiplying DAC application.

VDD:

Digital Supply Voltage. This is the power supply pin for the device. It can be from +5 to +15Vdc. Operation is optimum for +5Vdc.

GND:

The on-chip voltage must be at the same ground potential as VDD and IOUT2 for current switching applications. Any difference of potential (VDD pin I/O) will result in a linearity change of

Other Pin Functions

D₀-D₇: Digital Inputs. D₇ is the least significant bit (LSB) and D₀ is the most significant bit (MSB).

IOUT1: DAC Current Output 1. IOUT1 is a maximum for a digital code of all 1's in the DAC register, and is zero for all 0's in DAC register.

IOUT2: DAC Current Output 2. IOUT2 is a constant minus IOUT1, or IOUT2 + IOUT1 = constant (full scale for a fixed reference voltage).

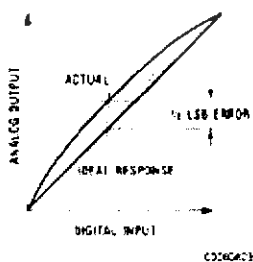
R_{FB}: Feedback Resistor. The feedback resistor is provided on the IC chip for use as the shunt feedback

$$\frac{V_{CS} \text{ pin } 10}{3V_{REF}}$$

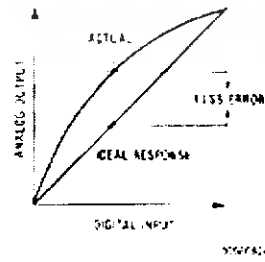
For example, if VREF = 10, and pin 10 is 5mV offset from IOUT1 and IOUT2 the linearity change will be 0.03%.

Pin 3 can be offset = 100mV with no linearity change, but the logic input threshold will shift.

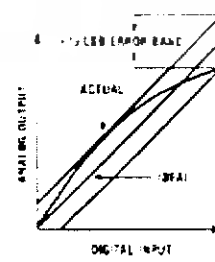
Linearity Error



a) End point test after zero and fs adj.



b) Best straight line



c) Shifting fs adj. to pass best straight line test

Definition of Terms

Resolution: Resolution is directly related to the number of switches or bits within the DAC. For example, the DAC0820 has 2^8 or 256 steps and therefore has 8-bit resolution.

Linearity Error: Linearity Error is the maximum deviation from a straight line passing through the endpoints of the DAC transfer characteristic. It is measured after adjusting for zero and full-scale. Linearity error is a parameter intrinsic to the device and cannot be externally adjusted.

National's linearity "end point test" (a) and the "best straight line" test (b) used by other suppliers are illustrated above. The "end point test" greatly simplifies the adjustment procedure by eliminating the need for multiple iterations of checking the linearity and then adjusting full scale until the linearity is met. The "end point test" guarantees that linearity is met

after a single full scale adjust. (One adjustment vs. multiple iterations of the adjustment.) The "end point test" uses a standard zero and F.S. adjustment procedure and is a much more stringent test for DAC linearity.

Power Supply Sensitivity: Power supply sensitivity is a measure of the effect of power supply changes on the DAC full-scale output.

Settling Time: Settling time is the time required from a code transition until the DAC output reaches within $\pm 1/2$ LSB of the final output value. Full-scale settling time requires a zero to full-scale or full-scale to zero output change.

Full Scale Error: Full scale error is a measure of the output error between an ideal DAC and the actual device output.

Definition of Terms (Continued)

Linearity: for the DAC0830 series, full-scale error $\pm 1/2$ LSB. For $V_{REF} = 10V$ and unipolar operation, $\pm 1/2$ LSB = $\pm 0.00074296mV$ (9.981%). Full-scale error is adjustable to zero.

Differential Nonlinearity: The difference between any two consecutive codes in the transfer curve from the theoretical $\pm 1/2$ LSB to differential nonlinearity.

Monotonic: If the output of a DAC increases for increasing digital input code, then the DAC is monotonic. An 8-bit DAC which is monotonic to 8 bits simply means that increasing digital input codes will produce an increasing analog output.

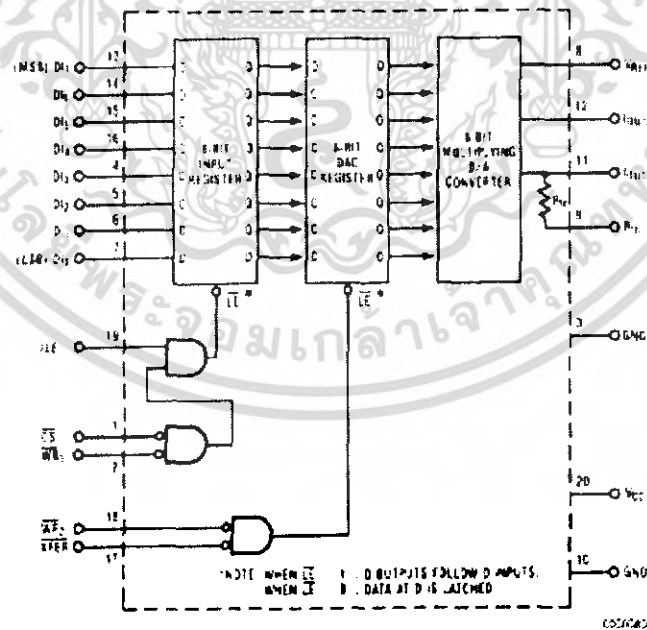


FIGURE 1. DAC0830 Functional Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข-2 เอกสารคู่มือการใช้งาน LM741

ไอซี LM741 เป็นไอซีออปแอมป์ ที่ใช้ในวงจรซีโรสเปนในรูปแบบที่ 2.17 โดยมีรายละเอียดต่างๆดังนี้

FAIRCHILD
SEMICONDUCTOR

www.fairchildsemi.com

LM741

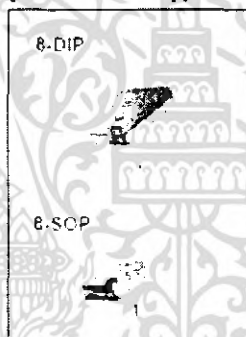
Single Operational Amplifier

Features

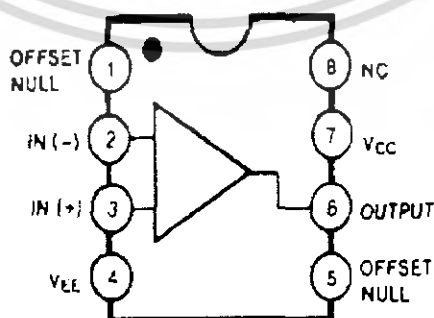
- Short circuit protection
- Excellent temperature stability
- Internal frequency compensation
- High input voltage range
- Null of offset

Description

The LM741 series are general purpose operational amplifiers. It is intended for a wide range of analog applications. The high gain and wide range of operating voltage provide superior performance in integrator, summing amplifier, and general feedback applications.



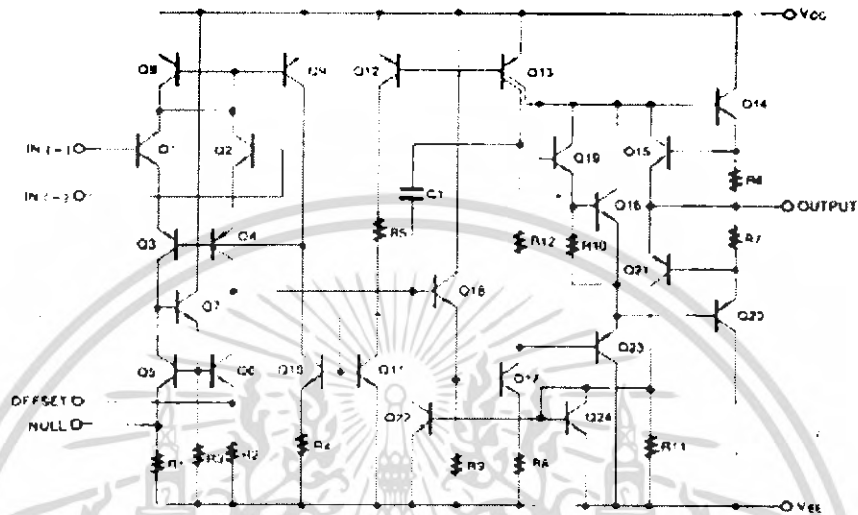
Internal Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM741

Schematic Diagram

Absolute Maximum Ratings ($T_A = 25^\circ\text{C}$)

Parameter	Symbol	Value	Unit
Supply Voltage	V_{CC}	± 18	V
Differential Input Voltage	V_{DIFF}	30	V
Input Voltage	V_I	± 15	V
Output Short Circuit Duration	-	Indefinite	-
Power Dissipation	P_D	500	mW
Operating Temperature Range	T_{OPR}	0 ~ +70 -40 ~ +25	$^\circ\text{C}$
Storage Temperature Range	T_{STG}	-65 ~ +150	$^\circ\text{C}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข-3 เอกสารคู่มือการใช้งาน TIP31C และ TIP32C

ไอซี TIP31C และ TIP32C เป็นไอซีทรานซิสเตอร์แบบเอ็นพีเอ็น และพีเอ็นพี ตามลำดับ ซึ่งใช้ในภาคขับเคลื่อนมอเตอร์ ในรูปที่ 2.18 โดยมีรายละเอียดต่างๆดังนี้

TIP31, TIP31A, TIP31B, TIP31C, (NPN), TIP32, TIP32A, TIP32B, TIP32C, (PNP)

Complementary Silicon Plastic Power Transistors

Designed for use in general purpose amplifier and switching applications.

- Collector-Emitter Saturation Voltage -
 $V_{CE(sat)} = 1.2 V_{dc} (Max) @ I_C = 3.0 A_{dc}$
- Collector-Emitter Sustaining Voltage -
 $V_{CE(sus)} = 40 V_{dc} (Min) - TIP31, TIP32$
 $= 60 V_{dc} (Min) - TIP31A, TIP32A$
 $= 80 V_{dc} (Min) - TIP31B, TIP32B$
 $= 100 V_{dc} (Min) - TIP31C, TIP32C$
- High Current Gain - Bandwidth Product
 $f_T = 3.0 MHz (Min) @ I_C = 500 mA_{dc}$
- Compact TO-220 AB Package

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector-Emitter Voltage TIP31, TIP32 TIP31A, TIP32A TIP31B, TIP32B TIP31C, TIP32C	V_{CE}	40 60 80 100	Vdc
Collector-Base Voltage TIP31, TIP32 TIP31A, TIP32A TIP31B, TIP32B TIP31C, TIP32C	V_{CB}	40 60 80 100	Vdc
Emitter-Base Voltage	V_{EB}	50	Vdc
Collector Current - Continuous Peak	I_C	3.0 5.0	A _{dc}
Base Current	I_B	1.0	A _{dc}
Total Power Dissipation $@ T_c = 25^\circ C$ Derate above 25 $^\circ C$	P_D	40 0.32	Watts W/ $^\circ C$
Total Power Dissipation $@ T_A = 25^\circ C$ Derate above 25 $^\circ C$	P_D	20 0.216	Watts W/ $^\circ C$
Unclamped Inductive Load Energy (Note 1)	E	32	mJ
Operating and Storage Junction Temperature Range	T_j, T_{stg}	-55 to +150	$^\circ C$

1. $I_B = 1.0 A_{dc}, t = 20 \mu s, P_{RF} = 10 Hz, V_{CE} = 10 V, R_{EE} = 100 \Omega$

ON Semiconductor

<http://onsemi.com>

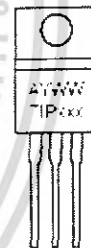
3 AMPERE
POWER TRANSISTORS
COMPLEMENTARY
SILICON
40-60-80-100 VOLTS
40 WATTS

MARKING
DIAGRAM



TO-220 AB
CASE 221A-09
STYLE 1

xxx = Specific Device Code
21 31A 31B 31C 32 32A 32B 32C
A = Assembly Location
y = Year
WW = Work Week



ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 6 of this data sheet.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข-4 เอกสารคู่มือการใช้งาน DM74LS32

ไอซี DM74LS32 เป็นไอซีตรรกะ แบบออกเกต ซึ่งใช้วงจรปรับแรงดันไฟฟ้ามาตรฐาน ในกรณีใช้วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกแบบใช้ความต้านทานสองค่า ในรูปที่ 3.9 โดยมีรายละเอียดต่างๆดังนี้



June 1968
Revised March 2000

DM74LS32 Quad 2-Input OR Gate

General Description

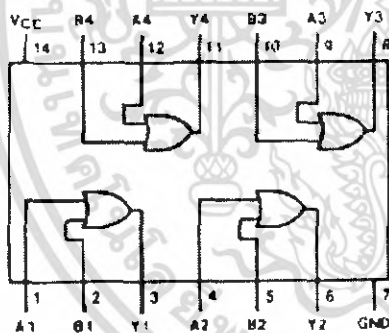
This device contains four independent gates each of which performs the logic OR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS32M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS32G	M14D	14-Lead Small Outline Package (SOP), BIAJ TYPE I, 5.5mm Wide
DM74LS32H	N14A	14-Lead Plastic Dual-In-Line Package (DIP), JEDEC MS-001, 0.500 Wide

Devices also available in Tape and Reel formats, by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = A + B$$

Inputs		Output
A	B	Y
L	L	L
L	-	-
-	L	-
-	-	-

H = HIGH Logic Level
L = LOW Logic Level

Absolute Maximum Ratings (Note 1):

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Note 1: The Absolute Maximum Ratings are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The Recommended Operating Conditions table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V _{CC}	Supply Voltage	4.75	5	5.25	V
V _H	HIGH Level Input Voltage	2			V
V _L	LOW Level Input Voltage			0.8	V
I _{OH}	HIGH Level Output Current			-0.4	mA
I _{OL}	LOW Level Output Current			2	mA
T _a	Free Air Operating Temperature	0		70	°C

เอกสารนี้เป็นเอกสารของบริษัทที่สงวนลิขสิทธิ์ไว้ ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

over recommended operating free air temperature range unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V_{OL}	Input Clamp Voltage	$V_{CC} = V_{IH} I_{ICL} = 18 \text{ mA}$			-1.5	V
V_{OH}	HIGH Level Output Voltage	$V_{CC} = V_{IH} I_{OH} = \text{Max}$ $V_{OL} = \text{Min}$	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	$V_{CC} = V_{IH} I_{OL} = \text{Max}$ $V_{OH} = \text{Max}$ $I_{OL} = 8 \text{ mA}$ $V_{CC} = \text{Min}$		0.35	0.6	V
I_{I}	Input Current @ Maximum Output Voltage	$V_{CC} = \text{Max}$ $V_{I} = 7 \text{ V}$			0.1	mA
I_{IH}	HIGH Level Input Current	$V_{CC} = \text{Max}$ $V_{I} = 2.7 \text{ V}$			20	μA
I_{IL}	LOW Level Input Current	$V_{CC} = \text{Max}$ $V_{I} = 0 \text{ V}$			-0.35	mA
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}$ (Note 3)	-20		-60	mA
I_{CC1}	Supply Current with Outputs HIGH	$V_{CC} = \text{Max}$		3.1	6.2	mA
I_{CC2}	Supply Current with Outputs LOW	$V_{CC} = \text{Max}$		4.9	9.8	mA

Note 2: All speeds are at $V_{CC} = 5 \text{ V}$ $T_A = 25^\circ\text{C}$

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

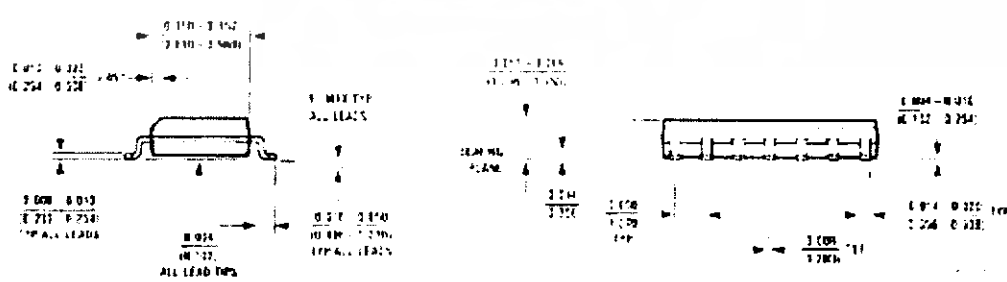
Switching Characteristics

at $V_{CC} = 5 \text{ V}$ and $T_A = 25^\circ\text{C}$

Symbol	Parameter	$R_L = 2 \text{ k}\Omega$				Units
		$C_L = 15 \text{ pF}$		$C_L = 50 \text{ pF}$		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	3	11	4	15	ns
t_{PLL}	Propagation Delay Time H-to-LOW Level Output	3	11	4	15	ns

Physical Dimensions

inches in millimeters unless otherwise noted



14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow Package Number M14A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้