

# โครงสร้างการควบคุมแบบนิร-ตามู-พีซี

นางสาวสุภารัตน์ ขวัญอ่อน

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2547

ISBN 974-533-382-4

# **NEURO-TABU-FUZZY CONTROL STRUCTURE**

**Miss Sudarat Khwanon**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the**

**Degree of Master of Engineering in Electrical Engineering**

**Suranaree University of Technology**

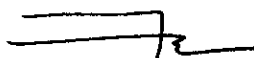
**Academic Year 2004**

**ISBN 974-533-382-4**

## โครงสร้างตัวควบคุมแบบนิร-ตาบ-พีซี

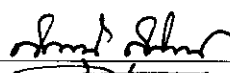
มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์



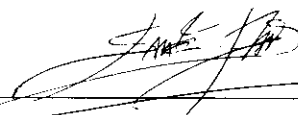
(ผศ. ดร.อาทิตย์ ศรีแก้ว)

ประธานกรรมการ



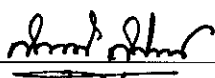
(รศ. น.ท. ดร.สรารุฒิ สุจิตจร)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(อ. ดร.ชนันชัย กุลสุวรรณพงษ์)

กรรมการ



(รศ. น.ท. ดร.สรารุฒิ สุจิตจร)

รองอธิการบดีฝ่ายวิชาการ



(รศ. น.อ. ดร.วรพจน์ จำพิศ)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

สุชาติร์นต์ ขวัญอ่อน : โครงสร้างการควบคุมแบบนิวโร-ตาบูลู-ฟัซซี่ (NEURO-TABU-FUZZY CONTROL STRUCTURE) อาจารย์ที่ปรึกษา : รศ. น.ท. ดร. สราวุฒิ สุจิตจร, 283 หน้า. ISBN 974-533-382-4

งานวิจัยวิทยานิพนธ์นี้ได้นำเสนอโครงสร้างการควบคุมแบบใหม่ เพื่อสร้างเสถียรภาพให้ระบบที่มีความไม่เป็นเชิงเส้น โครงสร้างการควบคุมดังกล่าวเป็นการผสมผสานระหว่างโครงข่ายประสาทเทียม ฟัซซี่ลอจิก และการค้นหาแบบตาบูลูเชิงปรับตัว เรียกว่า ตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ ตัวควบคุมที่นำเสนอนี้ประกอบด้วยมอดูลฟัซซี่แบบกฎอินพุตเดียวและค่าระดับความสำคัญพลวัต โดยในส่วนเงื่อนไขของแต่ละกฎฟัซซี่จะมีตัวแปรอินพุตเพียงตัวเดียวเท่านั้น ทำให้ช่วยลดจำนวนกฎฟัซซี่และง่ายต่อการกำหนดกฎฟัซซี่ให้เหมาะสม โดยมีค่าระดับความสำคัญพลวัตเป็นตัวบ่งชี้ความสำคัญของตัวแปรอินพุตแต่ละตัว ซึ่งค่าระดับความสำคัญพลวัตเหล่านี้ได้จากการเรียนรู้ของโครงข่ายประสาทเทียม และเพื่อให้กระบวนการเรียนรู้มีสมรรถนะสูงขึ้นได้นำวิธีการค้นหาแบบตาบูลูเชิงปรับตัวเข้าช่วยค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมที่สุดให้กับโครงข่ายประสาทเทียม ในการทดสอบสมรรถนะของตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ได้ดำเนินการจำลองสถานการณ์เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ผลการจำลองสถานการณ์แสดงให้เห็นว่า ตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวที่มีขนาดของส่วนประกอบหลากหลายได้อย่างดี รวมทั้งสามารถฟื้นฟูสภาพคืนจากการรบกวนภายนอกได้อย่างรวดเร็ว นอกจากนี้เมื่อเปรียบเทียบเวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมที่นำเสนอกับตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวจากผลงานวิจัยดั้งเดิม พบว่า ตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่สามารถสร้างเสถียรภาพให้ระบบได้รวดเร็วกว่าตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว รวมทั้งมีขอบเขตการสร้างเสถียรภาพให้ระบบที่กว้างกว่า

SUDARAT KHWANON : NEURO-TABU-FUZZY CONTROL STRUCTURE

THESIS ADVISOR : ASSOC. PROF. SARAWUT SUJITJORN, Ph.D. 283 PP.

ISBN 974-533-382-4

## ARTIFICIAL NEURAL NETWORK/FUZZY LOGIC/ADAPTIVE TABU SEARCH

This thesis presents a new control structure to stabilize a nonlinear system. The proposed control structure is the combination of artificial neural network, fuzzy logic and adaptive tabu search, which is called “neuro-tabu-fuzzy” controller. This proposed controller consists of the single input rule modules (SIRMs) and the dynamic importance degrees (DIDs). Each fuzzy rule has only one input variable which helps to decrease the number of rules and ease the rule development. Each DID indicates the importance of each input variable. The values of DIDs are obtained from the learning of neural network. To improve the performance of the learning process, adaptive tabu search is adopted to find optimum initial parameters for the neural network. Simulations have been conducted to evaluate the neuro-tabu-fuzzy controller. The proposed controller can effectively stabilize an inverted pendulum system of various dimensions. It can rapidly recover the pendulum from external disturbances. Beside, the comparison of stabilization times between the proposed and the previous controllers indicates that the neuro-tabu-fuzzy controller stabilizes the inverted pendulum of various dimensions more rapidly than the previous controller does.

School of Electrical Engineering

Academic Year 2004

Student's Signature \_\_\_\_\_

Advisor's Signature \_\_\_\_\_

## กิตติกรรมประกาศ

วิทยานิพนธ์นี้ดำเนินการสำเร็จลุล่วงด้วยดี ผู้วิจัยขอขอบพระคุณบุคคลและกลุ่มบุคคลต่างๆ ที่ได้กรุณาให้คำปรึกษา แนะนำ รวมทั้งได้ให้ความช่วยเหลืออย่างดียิ่ง ทั้งด้านวิชาการและด้านการดำเนินงานวิจัย ซึ่งได้แก่

รองศาสตราจารย์ นาวาอากาศโท ดร.สราวุฒิ สุจิตจร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้คำปรึกษา แนะนำ และแนะแนวทางอันเป็นประโยชน์ยิ่งต่องานวิจัย รวมถึงได้ช่วยตรวจทาน และแก้ไขรายงานวิทยานิพนธ์เล่มนี้จนทำให้มีความสมบูรณ์ยิ่งขึ้น รวมทั้งเป็นกำลังใจ และเป็นแบบอย่างที่ดีในการดำเนินชีวิตหลายๆ ด้านให้กับผู้วิจัยเสมอมา

ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ศรีแก้ว ดร.ชนัดชัช กุลวรวานิชพงษ์ และอาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารีทุกท่าน รวมถึงผู้ช่วยศาสตราจารย์เฉชา พวงดาวเรือง ที่ได้กรุณาให้คำปรึกษา แนะนำ และความรู้ทางด้านวิชาการอย่างดียิ่งมาโดยตลอด

ขอขอบคุณคุณวิชัย ศรีสุรภัย สำหรับคำแนะนำดีๆ อันเป็นประโยชน์ต่องานวิจัย รวมทั้งขอขอบคุณบุคลากรศูนย์เครื่องมือวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีทุกท่าน ที่อำนวยความสะดวกในการทำงานของข้าพเจ้า

ขอบคุณ พี่ๆ เพื่อนๆ น้องๆ บัณฑิตศึกษาทุกท่าน รวมถึงมิตรสหายทั้งในอดีตและปัจจุบันที่คอยถามไถ่และให้กำลังใจในการทำวิจัยมาโดยตลอด

ขอขอบคุณ รณเดช จันทรมัส ที่ได้ให้ความช่วยเหลือ ดูแล ห่วงใย และเป็นกำลังใจแก่ผู้วิจัยเสมอมา รวมถึงขอบคุณความเสียสละของรณเดช จนทำให้งานวิจัยนี้สำเร็จลุล่วงด้วยดี

สุดท้ายนี้ ผู้วิจัยขอขอบคุณอาจารย์ผู้สอนทุกท่านที่ประสิทธิ์ประสาทความรู้ทางด้านต่างๆ ทั้งในอดีตและปัจจุบัน และขอกราบขอบพระคุณ บิดา มารดา รวมถึงญาติพี่น้องของผู้วิจัยทุกท่าน ที่ได้ให้ความรัก ความอบอุ่น ความห่วงใย การอบรมเลี้ยงดู และให้การสนับสนุนทางการศึกษาอย่างดียิ่งมาโดยตลอด รวมทั้งเป็นกำลังใจที่ยิ่งใหญ่ในยามที่ผู้วิจัยท้อและท้อแท้ใจ ช่วยให้มีพลังเข้มแข็งพร้อมเผชิญกับปัญหาอุปสรรคต่างๆ จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา

สุภารัตน์ ขวัญอ่อน

# สารบัญ

## หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง .....	ช
สารบัญรูป .....	ญ
<b>บทที่</b>	
<b>1 บทนำ.....</b>	<b>1</b>
1.1 ความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์การวิจัย .....	3
1.3 ข้อยกเว้นเบื้องต้น .....	3
1.4 ขอบเขตของการวิจัย.....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	4
1.6 การจัดรูปเล่มวิทยานิพนธ์.....	5
<b>2 ระบบควบคุมแบบฟัซซี่ .....</b>	<b>7</b>
2.1 บทนำ.....	7
2.2 ทฤษฎีฟัซซี่เซต.....	8
2.3 องค์ประกอบของระบบควบคุมแบบฟัซซี่ .....	10
2.4 การออกแบบระบบควบคุมแบบฟัซซี่.....	12
2.5 กฎฟัซซี่.....	13
2.6 การอนุมานฟัซซี่.....	17
2.7 การกำหนดฟังก์ชันความเป็นสมาชิก.....	17
2.8 เทคนิคการดีฟัซซี่ฟิเคชัน .....	19
2.9 สรุป .....	22

## สารบัญ (ต่อ)

หน้า

<b>3</b>	<b>ตัวควบคุมฟuzzyแบบกฎอินพุตเดียว.....</b>	<b>23</b>
3.1	บทนำ.....	23
3.2	ระบบลูกค้่มผกผันแบบข้อต่อเดียว .....	24
3.3	หลักการของตัวควบคุมฟuzzyแบบกฎอินพุตเดียว .....	25
3.4	การออกแบบตัวควบคุมฟuzzyแบบกฎอินพุตเดียว.....	29
3.4.1	การกำหนดกฎฟuzzyแบบอินพุตเดียว .....	29
3.4.2	การกำหนดค่าระดับความสำคัญพลวัต .....	32
3.5	การอนุมานกฎฟuzzyด้วยวิธี simplified fuzzy reasoning method.....	35
3.6	การทำงานของตัวควบคุมฟuzzyแบบกฎอินพุตเดียว .....	37
3.7	การสร้างเสถียรภาพให้ระบบลูกค้่มผกผันแบบข้อต่อเดียว.....	39
3.8	การปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิก.....	50
3.9	การเพิ่มการแบ่งส่วนตัวแปรฟuzzy.....	53
3.10	สรุป .....	57
<b>4</b>	<b>ตัวควบคุมนิวโร-ฟuzzy .....</b>	<b>59</b>
4.1	บทนำ.....	59
4.2	โครงข่ายประสาทเทียม.....	60
4.2.1	องค์ประกอบของโครงข่ายประสาทเทียม .....	61
4.2.2	การเรียนรู้แบบแพร่กลับ .....	63
4.3	หลักการของตัวควบคุมนิวโร-ฟuzzy.....	65
4.4	การออกแบบตัวควบคุมนิวโร-ฟuzzy .....	67
4.4.1	การกำหนดค่า SIRM.....	68
4.4.2	การปรับค่าระดับความสำคัญพลวัตโดยใช้โครงข่ายประสาทเทียม.....	68
4.5	การทำงานของตัวควบคุมนิวโร-ฟuzzy .....	71
4.6	การทดสอบเพื่อเลือกค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม .....	73
4.7	การสร้างเสถียรภาพให้ระบบลูกค้่มผกผันแบบข้อต่อเดียว.....	77



## สารบัญ (ต่อ)

### หน้า

4.8	การทดสอบสุ่มค่าพารามิเตอร์เริ่มต้นให้โครงข่ายประสาทเทียม.....	88
4.9	สรุป .....	90
5	ตัวควบคุมนิวโร-ตาบู-พีซซี่.....	91
5.1	บทนำ.....	91
5.2	การค้นหาแบบตาบูเชิงปรับตัว.....	92
5.3	หลักการของตัวควบคุมนิวโร-ตาบู-พีซซี่ .....	97
5.4	การทำงานของตัวควบคุมนิวโร-ตาบู-พีซซี่ .....	98
5.5	การพิจารณาเลือกค่าพารามิเตอร์ของการค้นหาแบบตาบูเชิงปรับตัว.....	100
5.6	การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว.....	103
5.7	การทดสอบตัวควบคุมนิวโร-ตาบู-พีซซี่.....	114
5.8	สรุป .....	116
6	อภิปรายสมรรถนะของตัวควบคุม .....	118
6.1	บทนำ.....	118
6.2	การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันด้วยตัวควบคุมแบบต่างๆ.....	119
6.3	เวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันเมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบ .....	124
6.4	ขอบเขตการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันของตัวควบคุมแบบต่างๆ .....	128
6.5	การทดสอบสมรรถนะของตัวควบคุมเมื่อมีการรบกวนจากภายนอก .....	130
6.5.1	การรบกวนระบบที่เวลา 2 วินาที .....	131
6.5.2	การรบกวนระบบที่เวลา 10 วินาที .....	135
6.6	ผลทดสอบค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสมของโครงข่ายประสาทเทียม .....	138
6.7	สรุป .....	146
7	บทสรุปและข้อเสนอแนะ.....	148
7.1	สรุป .....	148
7.2	ข้อเสนอแนะ .....	150
	รายการอ้างอิง .....	152

## สารบัญ (ต่อ)

หน้า

### ภาคผนวก

ภาคผนวก ก.	แบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบข้อต่อเดียว.....	155
ภาคผนวก ข.	แบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบสองข้อต่อ .....	163
ภาคผนวก ค.	การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อด้วย ตัวควบคุมนิวโร-ตาบู-ฟิชชี .....	174
ภาคผนวก ง.	โครงข่ายประสาทเทียมและการเรียนรู้แบบแพร่กลับ .....	189
ภาคผนวก จ.	เวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบ โดยใช้ตัวควบคุมแบบต่างๆ.....	204
ภาคผนวก ฉ.	โปรแกรมสำหรับตัวควบคุมฟิชชีแบบกฎอินพุตเดียว.....	214
ภาคผนวก ช.	โปรแกรมสำหรับตัวควบคุมนิวโร-ฟิชชี .....	224
ภาคผนวก ซ.	โปรแกรมสำหรับตัวควบคุมนิวโร-ตาบู-ฟิชชี .....	239
ภาคผนวก ฌ.	โปรแกรมสำหรับการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน แบบสองข้อต่อด้วยตัวควบคุมนิวโร-ตาบู-ฟิชชี .....	256
ภาคผนวก ฎ.	บทความที่ได้รับการตีพิมพ์เผยแพร่ .....	281
ประวัติผู้เขียน .....		283

## สารบัญตาราง

ตารางที่	หน้า
2.1	การกำหนดกฎฟิชชีสำหรับตัวแปรฟิชชีที่กำหนดขึ้น.....14
2.2	ขอบเขตฟิชชีเชิงทฤษฎีของฟังก์ชันความเป็นสมาชิก.....18
2.3	ขอบเขตฟิชชีเชิงทฤษฎีจากการแบ่งส่วนตัวแปรฟิชชี.....19
3.1	การกำหนดกฎฟิชชีใน SIRM.....31
3.2	การกำหนดกฎฟิชชีสำหรับตัวแปรพลวัตของมุมและความเร็วเชิงมุม.....33
3.3	การกำหนดกฎฟิชชีสำหรับตัวแปรพลวัตของตำแหน่งและความเร็วรูด.....34
3.4	ค่าฐานและค่าขยายสำหรับตัวแปรอินพุตแต่ละตัว.....35
3.5	สถานการณ์ที่กำหนดขึ้นเพื่อสร้างเสถียรภาพให้ระบบลูกคัมผกผันแบบข้อต่อเดียว.....40
3.6	ผลการเปรียบเทียบเวลาเข้าสู่เสถียรภาพของระบบลูกคัมผกผันแบบข้อต่อเดียว จากโปรแกรมที่พัฒนาขึ้นและผลงานวิจัยอ้างอิง.....44
3.7	เวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมฟิชชีแบบกฎอินพุตเดียว.....48
3.8	การกำหนดกฎฟิชชีใน SIRM เมื่อเพิ่มการแบ่งส่วนตัวแปรฟิชชี.....54
3.9	การกำหนดกฎฟิชชีสำหรับตัวแปรพลวัตของมุมและความเร็วเชิงมุม.....56
3.10	การกำหนดกฎฟิชชีสำหรับตัวแปรพลวัตของตำแหน่งและความเร็วรูด.....56
4.1	องค์ประกอบของสมองและโครงข่ายประสาทเทียม.....62
4.2	พารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการสุ่มค่า.....78
4.3	ค่าระดับความสำคัญพลวัตของตัวควบคุมนิวโร-ฟิชชีในสถานการณ์ต่างๆ.....85
4.4	เวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมนิวโร-ฟิชชี.....86
4.5	การทดสอบสุ่มค่าพารามิเตอร์เริ่มต้นให้โครงข่ายประสาทเทียม.....89
5.1	การทดสอบเลือกค่าพารามิเตอร์ของการค้นหาแบบตามูเชิงปรับตัว.....100
5.2	พารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการค้นหาแบบ ตามูเชิงปรับตัว.....103
5.3	ค่าระดับความสำคัญพลวัตของตัวควบคุมนิวโร-ตามู-ฟิชชีในสถานการณ์ต่างๆ.....110
5.4	เวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมนิวโร-ตามู-ฟิชชี.....111

## สารบัญตาราง (ต่อ)

ตารางที่	หน้า
5.5 การทดสอบเมื่อใช้ค่าพารามิเตอร์เริ่มต้นที่ได้จากการค้นหาแบบตามูเชิงปรับตัว.....	115
6.1 สถานการณ์ที่จัดตั้งให้มีการสร้างเสถียรภาพแก่ระบบลูกคุ่มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมแบบต่างๆ .....	119
6.2 ค่าระดับความสำคัญพลวัตของตัวควบคุมนิวโร-ฟิชชี และตัวควบคุม นิวโร-คาบู-ฟิชชี .....	123
6.3 เวลาในการสร้างเสถียรภาพให้ระบบลูกคุ่มผกผันด้วยตัวควบคุมแบบต่างๆ.....	123
6.4 เวลาเข้าสู่เสถียรภาพของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 2 วินาที.....	133
6.5 เวลาเข้าสู่เสถียรภาพของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 10 วินาที.....	137
6.6 ค่าระดับความสำคัญพลวัตจากการเรียนรู้ของโครงข่ายประสาทเทียม.....	141
6.7 ปริมาณการเข้าสู่เสถียรภาพของตัวควบคุมนิวโร-ฟิชชี และตัวควบคุม นิวโร-คาบู-ฟิชชี .....	145

## สารบัญรูป

รูปที่	หน้า
2.1	ความแตกต่างของความเป็นสมาชิกระหว่างเซตปกติธรรมดา กับ ฟัซซี่เซต.....8
2.2	ฟังก์ชันลักษณะเฉพาะของเซตปกติธรรมดา.....9
2.3	ฟังก์ชันความเป็นสมาชิกของฟัซซี่เซต ..... 10
2.4	โครงสร้างพื้นฐานของระบบควบคุมแบบฟัซซี่ ..... 11
2.5	ฟังก์ชันความเป็นสมาชิกของตัวแปรฟัซซี่ในส่วนเงื่อนไขและส่วนผลลัพธ์ ..... 14
2.6	การเชื่อมโยงประโยคในกฎฟัซซี่ด้วยและ..... 15
2.7	การเชื่อมโยงประโยคในกฎฟัซซี่ด้วยหรือ ..... 15
2.8	รูปแบบฟังก์ชันความเป็นสมาชิก..... 17
2.9	การกำหนดรูปแบบฟังก์ชันความเป็นสมาชิก..... 18
2.10	การแบ่งส่วนตัวแปรฟัซซี่..... 19
2.11	ดีฟัซซี่ฟิเคชันแบบเกณฑ์สูงสุดเมื่อผลการอนุมานที่ได้เป็นฟัซซี่ชนิดลดสัดส่วน..... 20
2.12	ดีฟัซซี่ฟิเคชันแบบเกณฑ์สูงสุดเมื่อผลการอนุมานที่ได้เป็นฟัซซี่ชนิดตัดส่วน..... 20
2.13	การดีฟัซซี่ฟิเคชันด้วยวิธีศูนย์กลางแรงโน้มถ่วง..... 21
3.1	ระบบลูกตุ้มผกผันแบบข้อต่อเดียว..... 24
3.2	โครงสร้างของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว..... 26
3.3	การวิเคราะห์เพื่อกำหนด SIRM สำหรับมุมและความเร็วเชิงมุมของลูกตุ้ม ..... 30
3.4	การวิเคราะห์เพื่อกำหนด SIRM สำหรับตำแหน่งและความเร็วของรถ ..... 30
3.5	ฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM ..... 31
3.6	ฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตใน SIRM..... 31
3.7	ฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัตแต่ละตัว..... 32
3.8	ฟังก์ชันความเป็นสมาชิกของเอาต์พุต..... 33
3.9	การอนุมานด้วยวิธี simplified fuzzy reasoning method ..... 36
3.10	ตัวอย่างการอนุมานกฎฟัซซี่ด้วยวิธี simplified fuzzy reasoning method..... 37
3.11	โครงสร้างตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวเพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน..... 38

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.12 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 1.0 เมตร .....	40
3.13 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร .....	41
3.14 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 0.2 เมตร .....	42
3.15 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 2.2 เมตร .....	42
3.16 ผลตอบสนองเมื่อปรับเปลี่ยนช่วงเวลาแซมปลิงเป็น 0.001 วินาที .....	43
3.17 ผลตอบสนองเมื่อปรับเพิ่มมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม .....	44
3.18 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา และก้านลูกตุ้มยาว 1.0 เมตร .....	45
3.19 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น -2.0 เมตร .....	46
3.20 ผลตอบสนองมุมเริ่มต้นเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร .....	46
3.21 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 2.0 เมตร .....	47
3.22 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม .....	49
3.23 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม .....	49
3.24 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม .....	50
3.25 ฟังก์ชันความเป็นสมาชิกรูปประฆังคว่ำของตัวแปรอินพุตแต่ละตัวใน SIRM .....	51
3.26 ฟังก์ชันความเป็นสมาชิกรูปประฆังคว่ำของตัวแปรพลวัต .....	51
3.27 ผลตอบสนองเมื่อรูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูปประฆังคว่ำ .....	52
3.28 ฟังก์ชันความเป็นสมาชิกรูปสี่เหลี่ยมคางหมูของตัวแปรอินพุตแต่ละตัวใน SIRM .....	52
3.29 ฟังก์ชันความเป็นสมาชิกรูปสี่เหลี่ยมคางหมูของตัวแปรพลวัต .....	52
3.30 ผลตอบสนองเมื่อรูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูปสี่เหลี่ยมคางหมู .....	53
3.31 ฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM ที่มี 5 ฟัชชีเซต .....	54
3.32 ฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตของ SIRM ที่มี 5 ฟัชชีเซต .....	54
3.33 ฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัตที่มี 5 ฟัชชีเซต .....	55
3.34 ฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตของตัวแปรพลวัตที่มี 5 ฟัชชีเซต .....	55
3.35 ผลตอบสนองเมื่อเพิ่มการแบ่งส่วนตัวแปรฟัชชีเป็น 5 ฟัชชีเซต .....	57
4.1 องค์ประกอบของโครงข่ายประสาทเทียม .....	61
4.2 กระบวนการประมวลผลของโครงข่ายประสาทเทียม .....	62

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3	การเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม.....63
4.4	โครงสร้างของตัวควบคุมนิวโร-ฟิชชี.....66
4.5	โครงข่ายประสาทเทียมหลายชั้นแบบป้อนไปหน้า.....68
4.6	โครงสร้างของตัวควบคุมนิวโร-ฟิชชีเพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน.....72
4.7	อัตราการเรียนรู้เมื่อ $\eta = 0.05$ .....74
4.8	อัตราการเรียนรู้เมื่อ $\eta = 0.10$ .....74
4.9	อัตราการเรียนรู้เมื่อ $\eta = 0.25$ .....75
4.10	อัตราการเรียนรู้เมื่อ $\eta = 0.50$ .....76
4.11	ผลการทดสอบค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม.....76
4.12	ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 1.0 เมตร.....79
4.13	ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร.....79
4.14	ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 0.2 เมตร.....80
4.15	ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 2.0 เมตร.....80
4.16	ผลตอบสนองเมื่อปรับเปลี่ยนช่วงเวลาแซมปลิงเป็น 0.001 วินาที.....81
4.17	ผลตอบสนองเมื่อปรับเปลี่ยนมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม.....81
4.18	ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร.....82
4.19	ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา และก้านลูกตุ้มยาว 1.0 เมตร.....83
4.20	ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น -2.0 เมตร.....83
4.21	ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น -1.5 เมตร.....84
4.22	เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม.....86
4.23	เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม.....87
4.24	เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม.....88
5.1	การสุ่มคำตอบ $s_0$ ในพื้นที่การค้นหาค่า $R$ .....92
5.2	คำตอบพื้นที่รอบข้าง $s_0$ .....93
5.3	กำหนดค่าใหม่ให้กับคำตอบพื้นที่รอบข้างที่ดีที่สุด.....93
5.4	กำหนดค่าใหม่ให้กับ $s_0$ .....94

## สารบัญรูป (ต่อ)

รูปที่	หน้า
5.5 การค้นหาคำตอบในรอบต่อไป .....	94
5.6 การเดินย้อนรอยในกระบวนการค้นหาแบบตามูเชิงปรับตัว.....	95
5.7 การค้นหาแบบตามูเชิงปรับตัว.....	96
5.8 การค้นหาแบบตามูธรรมชาติ .....	96
5.9 โครงสร้างของตัวควบคุมนิวโร-ตามู-พีซซี่ .....	97
5.10 การสร้างเสถียรภาพให้ระบบลูกค่อมผกผันด้วยตัวควบคุมนิวโร-ตามู-พีซซี่.....	98
5.11 ผลการทดสอบเมื่อจำนวนคำตอบรอบข้างเป็น 40 และรัศมีการค้นหาเป็น 0.2.....	101
5.12 ผลการทดสอบเมื่อจำนวนคำตอบรอบข้างเป็น 40 และรัศมีการค้นหาเป็น 0.3 .....	101
5.13 ผลการทดสอบเมื่อจำนวนคำตอบรอบข้างเป็น 50 และรัศมีการค้นหาเป็น 0.3 .....	102
5.14 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกค่อมยาว 1.0 เมตร.....	104
5.15 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร .....	105
5.16 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกค่อมยาว 0.2 เมตร.....	105
5.17 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกค่อมยาว 2.0 เมตร.....	106
5.18 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร .....	106
5.19 ผลตอบสนองเมื่อปรับเปลี่ยนมวลของก้านลูกค่อมเป็น 0.5 กิโลกรัม .....	107
5.20 ผลตอบสนองเมื่อปรับลดช่วงเวลาแซมปลิงเป็น 0.001 วินาที .....	108
5.21 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา.....	108
5.22 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น -2.0 เมตร.....	109
5.23 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น -1.5 เมตร....	109
5.24 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกค่อม .....	112
5.25 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกค่อม.....	113
5.26 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกค่อม.....	114
6.1 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 1 ของตัวควบคุมแบบต่างๆ.....	120
6.2 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 2 ของตัวควบคุมแบบต่างๆ.....	120
6.3 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 3 ของตัวควบคุมแบบต่างๆ.....	121
6.4 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 4 ของตัวควบคุมแบบต่างๆ.....	121



## สารบัญรูป (ต่อ)

รูปที่	หน้า
6.5 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 5 ของตัวควบคุมแบบต่างๆ.....	122
6.6 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม .....	125
6.7 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม.....	126
6.8 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม.....	127
6.9 ขอบเขตการสร้างเสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม .....	128
6.10 ขอบเขตการสร้างเสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม....	129
6.11 ขอบเขตการสร้างเสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม...130	130
6.12 ผลทดสอบสมรรถนะของตัวควบคุมพีชชีแบบกฎอินพุตเดียวที่เวลา 2 วินาที.....	131
6.13 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-พีชชีที่เวลา 2 วินาที .....	132
6.14 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-ตาบู-พีชชีที่เวลา 2 วินาที.....	132
6.15 ผลทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 2 วินาที.....	134
6.16 ผลทดสอบสมรรถนะของตัวควบคุมพีชชีแบบกฎอินพุตเดียวที่เวลา 10 วินาที.....	135
6.17 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-พีชชีที่เวลา 10 วินาที .....	136
6.18 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-ตาบู-พีชชีที่เวลา 10 วินาที.....	136
6.19 ผลทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 10 วินาที.....	137
6.20 ผลการจำลองสถานการณ์ที่ 1 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม .....	139
6.21 ผลการจำลองสถานการณ์ที่ 2 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม .....	139
6.22 ผลการจำลองสถานการณ์ที่ 3 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม .....	140
6.23 ผลการจำลองสถานการณ์ที่ 4 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม .....	140
6.24 ผลการจำลองสถานการณ์ที่ 5 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม .....	141
6.25 ก) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-พีชชีเมื่อปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม และความยาวก้านลูกตุ้ม.....	142
ข) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-ตาบู-พีชชีเมื่อทดสอบในลักษณะเดียวกัน ...	142
6.26 ก) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-พีชชี เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้น ของตัวรถและความยาวก้านลูกตุ้ม.....	143
ข) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-ตาบู-พีชชีเมื่อทดสอบในลักษณะเดียวกัน ...	143

## สารบัญรูป (ต่อ)

รูปที่

หน้า

- 6.27 ก) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-พีซซี เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้น  
ของตัวรถและมุมเริ่มต้นของลูกตุ้ม .....144
- ข) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-ตาบู-พีซซีเมื่อทดสอบในลักษณะเดียวกัน ...144

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญของปัญหา

ในปัจจุบันวิธีการควบคุมแบบดั้งเดิม (conventional control) ยังมีความหลากหลายในทางปฏิบัติ (King, 1999) เช่น งานในระบบอุตสาหกรรมการผลิตนิยมใช้ตัวควบคุมแบบพีไอดี (PID controller) การออกแบบตัวควบคุมดังกล่าวจะอาศัยการสังเกตจากพฤติกรรมพลวัตของระบบที่ทำการควบคุม ในกระบวนการควบคุมจะทำการปรับค่าต่างๆ โดยอาศัยการประมาณจากพลวัตของระบบ อย่างไรก็ตามในปัจจุบันระบบอุตสาหกรรมการผลิตสมัยใหม่มีความซับซ้อนมากขึ้น ต้องการความยืดหยุ่น ตลอดจนคุณภาพของผลิตภัณฑ์ อันเป็นปัจจัยสำคัญในการแข่งขันทางการตลาด รวมทั้งการเปลี่ยนแปลงของสถานะแวดล้อมซึ่งสามารถเกิดขึ้นได้ตลอดเวลาในระบบอุตสาหกรรมการผลิต ด้วยเหตุนี้ทำให้ตัวควบคุมแบบดั้งเดิมไม่สามารถครอบคลุมความต้องการดังกล่าวได้

จากข้อจำกัดของวิธีการควบคุมแบบดั้งเดิม จึงได้มีความพยายามที่จะค้นคว้าพัฒนาวิธีการควบคุมแบบใหม่ขึ้น เพื่อควบคุมระบบที่มีความซับซ้อน ความไม่แน่นอน ความไม่เป็นเชิงเส้น และไม่เสถียร เป็นต้น วิธีการนี้จะแตกต่างจากวิธีการควบคุมแบบดั้งเดิม โดยอาศัยข้อมูลความรู้และประสบการณ์การทำงานของมนุษย์เป็นพื้นฐานในการควบคุม เรียกวิธีการควบคุมแบบนี้ว่า การควบคุมแบบชาญฉลาด (intelligent control) (King, 1999) ตัวอย่างเช่น ตัวควบคุมฟัซซี่ลอจิก และโครงข่ายประสาทเทียม เป็นต้น วิธีการเหล่านี้มีความสามารถในการหาความสัมพันธ์ระหว่างอินพุตและเอาต์พุตจากข้อมูลที่ได้จากการสังเกต รวมทั้งไม่จำเป็นต้องอาศัยแบบจำลองที่แม่นยำของกระบวนการที่ทำการควบคุม ประกอบกับมีความสามารถในการเรียนรู้ และตัดสินใจ (Meireles, Almeida, and Simoes, 2003) ปัจจุบันจึงได้มีการนำวิธีการควบคุมแบบชาญฉลาดมาประยุกต์ใช้อย่างกว้างขวาง เช่น ระบบขนส่งมวลชน (Sujitjorn, Mellitt, and Rabukwella, 1987) ระบบเครื่องปรับอากาศ (Iokibe, Tobi, and Araki, 1999) หุ่นยนต์ (Hagras and Sobh, 2002) เป็นต้น สำหรับในงานวิจัยนี้จะทำการศึกษาคู่ควบคุมฟัซซี่แบบกฎอินพุตเดียว (Yi and Yubazaki, 2000) ซึ่งเป็นวิธีการที่ช่วยลดจำนวนกฎฟัซซี่แบบดั้งเดิม เนื่องจากรูปแบบของการอนุมานฟัซซี่แบบดั้งเดิมนั้นจะนำตัวแปรอินพุตทุกตัวมาจัดไว้ในส่วนของเงื่อนไข (antecedent) ของกฎฟัซซี่แต่ละกฎ ทำให้จำนวนของกฎฟัซซี่เพิ่มขึ้นเป็นแบบเอ็กโพเนนเชียลตามจำนวนของ

อินพุตที่เพิ่มขึ้น (Yi and Yubazaki, 2000) รวมทั้งเป็นการยากในการกำหนดกฎฟuzzyแต่ละกฎให้เหมาะสม ในการแก้ปัญหากลกฎฟuzzyที่มีจำนวนมากจะใช้การอนุมานฟuzzyแบบกฎอินพุตเดียว ซึ่งในแต่ละกฎฟuzzyตรงส่วนของเงื่อนไขจะมีตัวแปรอินพุตเพียงตัวเดียวเท่านั้น ทำให้จำนวนกฎและพารามิเตอร์ต่างๆ ลดลง เป็นการง่ายต่อการกำหนดกฎฟuzzy (Cavalcante and Hirota, 1998) ในการปฏิบัติดังกล่าวใช้ค่าระดับความสำคัญพลวัต (dynamic importance degree: DID) เป็นตัวบ่งชี้ความสำคัญของอินพุตแต่ละตัว ซึ่งค่าระดับความสำคัญพลวัตเหล่านี้ได้จากผลรวมของค่าฐานและค่าพลวัต

เมื่อพิจารณาวิธีการควบคุมฟuzzyแบบกฎอินพุตเดียวตามแนวทางของ (Yi and Yubazaki, 2000) พบว่าการหาค่าระดับความสำคัญพลวัตประกอบด้วยหลายกระบวนการ ตัวอย่างเช่น การสุ่มค่าลองผิดลองถูกเพื่อหาค่าพารามิเตอร์ควบคุมสำหรับอินพุตแต่ละตัว การตั้งกฎฟuzzyและการอนุมานฟuzzyเพื่อหาค่าตัวแปรพลวัต เป็นต้น ดังนั้นในงานวิจัยนี้จะทำการพัฒนาโครงสร้างใหม่สำหรับวิธีการควบคุมแบบชาญฉลาด โดยใช้ตัวควบคุมฟuzzyแบบกฎอินพุตเดียวผสมผสานกับโครงข่ายประสาทเทียม (artificial neural network) เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับสถานการณ์ที่ควบคุม โดยค่าเหล่านี้ได้จากการเรียนรู้ของโครงข่ายประสาทเทียม เนื่องจากโครงข่ายประสาทเทียมมีความสามารถในการเรียนรู้และเหมาะสมสำหรับกระบวนการที่มีความไม่เป็นเชิงเส้นและมีความซับซ้อน (Thapa, Jones, and Zhu, 2000) โดยจะใช้การเรียนรู้แบบแพร่กลับ (backpropagation algorithm) ซึ่งมีการนำมาประยุกต์ใช้งานอย่างแพร่หลาย (Jin, Li, Wei, and Zhen, 2000) นอกจากนี้ยังใช้วิธีการค้นหาแบบตามูเชิงปรับตัวเข้าช่วยในการหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม ซึ่งใช้โครงข่ายหลายชั้นแบบป้อนไปหน้า (multilayer feedforward network) ที่พิจารณาค่าความคลาดเคลื่อนแพร่กลับจากเอาต์พุตของระบบ (Cui and Shin, 1993) ค่าพารามิเตอร์ดังกล่าวจะทำให้กระบวนการเรียนรู้มีประสิทธิภาพยิ่งขึ้น เนื่องจากวิธีการค้นหาแบบตามูเชิงปรับตัวมีความสามารถในการค้นหาค่าที่เหมาะสมได้อย่างรวดเร็ว (Puangdownreong, Areerak, Srikaew, Sujitjorn, and Totarong, 2002)

สำหรับในงานวิจัยนี้เพื่อแสดงให้เห็นถึงสมรรถนะของโครงสร้างการควบคุมแบบใหม่ที่นำเสนอ จะทำการสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผันแบบข้อต่อเดียวที่มีส่วนประกอบเป็นลูกตุ้มต่อกันกับรถ ระบบดังกล่าวนี้ได้รับการนำมาพิจารณาเพราะเป็นระบบที่มีความไม่เป็นเชิงเส้น ขาดเสถียรภาพ และควบคุมได้ยาก โดยพิจารณาควบคุมทั้งมุมลูกตุ้มผกผันและตำแหน่งตัวรถ งานวิจัยที่เกี่ยวกับการสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผันโดยใช้วิธีการควบคุมแบบชาญฉลาด ได้แก่ การปรับค่าอัตราขยายของตัวควบคุมฟuzzyด้วยตัวควบคุมนิวโร (neuro-PID control) (Omatu, Fujinaka, and Yoshioka, 2000) การใช้ตัวควบคุมแบบลำดับขั้น (hierarchical controller)

เพื่อสร้างฐานความรู้ฟัซซี่โดยอีดโนมัดติ (Kandadai and Tien, 1996) การใช้ตัวควบคุมฟัซซี่โดยฐานกฎได้มาจากโครงข่ายประสาทเทียม (Kyung and Lee, 1993) การใช้วิธีการไม่เชิงเส้นแบบเหมาะที่สุด (nonlinear optimization method) สำหรับฝึกสอนตัวควบคุมฟัซซี่ (Sakai and Takahama, 2001) รวมทั้งการใช้ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว (Single Input Rule Modules: SIRMs) (Yi and Yubazaki, 2000) โดยปรับเปลี่ยนค่าระดับความสำคัญพลวัตไปตามสถานการณ์ที่ทำให้การควบคุม ซึ่งจะพิจารณาควบคุมมุมของลูกตุ้มเป็นอันดับแรก เมื่อลูกตุ้มอยู่ในตำแหน่งเกือบสมดุลแล้วจะทำการควบคุมตำแหน่งของตัวรถเป็นลำดับต่อมา จากผลการจำลองสถานการณ์ขั้นต้นพบว่า การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวยังคงใช้เวลามาก ดังนั้นในงานวิจัยนี้จะพยายามลดเวลาในการเข้าสู่เสถียรภาพให้เร็วขึ้นกว่าเดิม โดยพัฒนาโครงสร้างใหม่ ซึ่งจะใช้ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวผสมผสานกับโครงข่ายประสาทเทียม รวมทั้งนำวิธีการค้นหาแบบตามูเชิงปรับตัวเข้าช่วยค้นหาพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม ทั้งนี้เพื่อช่วยให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวสามารถเข้าสู่เสถียรภาพได้ภายในระยะเวลาอันสั้น

## 1.2 วัตถุประสงค์การวิจัย

- เพื่อศึกษาและพัฒนาวิธีการควบคุมสำหรับระบบที่ขาดเสถียรภาพ และมีความไม่เป็นเชิงเส้น โดยทำการทดสอบวิธีการควบคุมที่นำเสนอด้วยระบบลูกตุ้มผกผันแบบข้อต่อเดียว
- เพื่อสร้างเสถียรภาพของระบบลูกตุ้มผกผันโดยใช้ตัวควบคุมแบบชาญฉลาด ได้แก่ ฟัซซี่ลอจิกผสมผสานกับโครงข่ายประสาทเทียม รวมทั้งวิธีการค้นหาแบบตามูเชิงปรับตัวโดยทำการควบคุมทั้งตำแหน่งมุมลูกตุ้มผกผันและตำแหน่งตัวรถ
- เพื่อปรับปรุงสมรรถนะของการสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผัน โดยพิจารณาเวลาในการเข้าสู่เสถียรภาพเมื่อเปรียบเทียบกับวิธีการควบคุมของงานวิจัยที่มีปรากฏมาก่อนแล้ว (Yi and Yubazaki, 2000)

## 1.3 ข้อตกลงเบื้องต้น

- ไม่พิจารณาแรงเสียดทานในขณะที่ทำการควบคุมระบบลูกตุ้มผกผัน
- ตัวรถของระบบลูกตุ้มผกผันสามารถเคลื่อนที่ไปมาบนรางลื่นได้อย่างอิสระ โดยตำแหน่งของตัวรถกำหนดให้อยู่ในช่วง  $[-2.4, +2.4]$  เมตร เมื่อพิจารณาจากจุดกึ่งกลางของรางลื่น
- ระบบลูกตุ้มผกผันแบบข้อต่อเดียวจะได้รับการพิจารณาว่าเข้าสู่เสถียรภาพอย่างสมบูรณ์แล้ว เมื่อตัวแปรสถานะต่างๆ เป็นไปตามข้อกำหนดต่อไปนี้ มุมของลูกตุ้มเข้าสู่  $\pm 0.1$  องศา ความเร็ว

เชิงมุมของลูกตุ้มเข้าสู่  $\pm 0.1$  องศาต่อวินาที ตำแหน่งของตัวรถเข้าสู่  $\pm 0.01$  เมตร และความเร็วของตัวรถเข้าสู่  $\pm 0.01$  เมตรต่อวินาที (Yi and Yubazaki, 2000)

#### 1.4 ขอบเขตของการวิจัย

- จำลองสถานการณ์การสร้างเสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยใช้ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวตามแนวทางของ (Yi and Yubazaki, 2000) และศึกษาวิจัยเพิ่มเติมถึงผลกระทบของรูปทรงฟังก์ชันความเป็นสมาชิกและการแบ่งส่วนตัวแปรฟัซซี่ของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว

- จำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ โดยใช้การเรียนรู้ของโครงข่ายประสาทเทียมร่วมกับตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว เพื่อทำการปรับค่าระดับความสำคัญพลวัตให้เหมาะกับแต่ละสถานการณ์ ซึ่งแต่เดิมได้จากผลการอนุมานฟัซซี่และค่าพารามิเตอร์ควบคุม ซึ่งค่าพารามิเตอร์ควบคุมเหล่านี้ได้มาจากการลองผิดลองถูก โดยค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ซึ่งได้แก่ ค่าน้ำหนักและค่าไบแอสได้มาจากวิธีการสุ่มค่า

- จำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ตาบูกู-ฟัซซี่ โดยใช้การเรียนรู้ของโครงข่ายประสาทเทียมร่วมกับตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว เมื่อนำวิธีการค้นหาแบบตาบูกูเชิงปรับตัวเข้าช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม ทั้งนี้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันจะใช้ค่าพารามิเตอร์เริ่มต้นที่ได้จากการค้นหาเพียงชุดเดียวตลอดการจำลองสถานการณ์

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถเป็นแนวทางในการพัฒนาตัวควบคุมแบบชาญฉลาด เพื่อทำการควบคุมระบบต่างๆ ในทางอุตสาหกรรมที่มีความไม่เป็นเชิงเส้น ขาดเสถียรภาพ และควบคุมได้ยาก เช่นเดียวกับระบบลูกตุ้มผกผันแบบข้อต่อเดียวต่อไปได้

- สามารถปรับปรุงวิธีการสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยใช้ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวร่วมกับโครงข่ายประสาทเทียม และการค้นหาแบบตาบูกูเชิงปรับตัว เพื่อช่วยให้เวลาในการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียวรวดเร็วยิ่งขึ้น

- ได้โครงสร้างของตัวควบคุมแบบใหม่สำหรับวิธีการควบคุมแบบชาญฉลาด ซึ่งช่วยลดจำนวนกฎฟัซซี่ในตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว รวมทั้งสามารถสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น

## 1.6 การจัดรูปเล่มวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย 7 บท และ 10 ภาคผนวก บทที่ 1 เป็นบทนำกล่าวถึงความสำคัญของปัญหา วัตถุประสงค์ ข้อตกลงเบื้องต้น ขอบเขตของงานวิจัย และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย รวมทั้งแนะนำเนื้อหาเบื้องต้นของวิทยานิพนธ์ฉบับนี้ ส่วนบทอื่นๆ ประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 2 กล่าวถึงระบบควบคุมแบบฟัซซี่ ทฤษฎีฟัซซี่เซต องค์ประกอบของระบบควบคุมแบบฟัซซี่ รวมทั้งการออกแบบระบบควบคุมแบบฟัซซี่

บทที่ 3 กล่าวถึงตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว หลักการทำงานของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว การจำลองสถานการณ์เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว รวมทั้งได้ดำเนินการปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิก และเพิ่มจำนวนการแบ่งส่วนตัวแปรฟัซซี่

บทที่ 4 กล่าวถึงตัวควบคุมนิวโร-ฟัซซี่ ซึ่งเป็นการผสมผสานระหว่างตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวและโครงข่ายประสาทเทียม หลักการทำงานของตัวควบคุมนิวโร-ฟัซซี่ การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว รวมถึงได้ดำเนินการทดสอบค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียม เพื่อพิจารณาปริมาณการเข้าสู่เสถียรภาพของระบบดังกล่าวเมื่อปรับเปลี่ยนสถานการณ์ที่ใช้ในการทดสอบ

บทที่ 5 กล่าวถึงตัวควบคุมนิวโร-ตาบู-ฟัซซี่ วิธีการค้นหาแบบตาบูเชิงปรับตัว หลักการทำงานของตัวควบคุมนิวโร-ตาบู-ฟัซซี่ ผลการทดสอบเพื่อเลือกค่าพารามิเตอร์ของวิธีการค้นหาแบบตาบูเชิงปรับตัว การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว พร้อมทั้งได้ทดสอบค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมจากวิธีการค้นหาแบบตาบูเชิงปรับตัว เพื่อพิจารณาปริมาณการสร้างเสถียรภาพให้ระบบในแต่ละสถานการณ์ทดสอบที่กำหนดขึ้น

บทที่ 6 กล่าวถึงการอภิปรายสมรรถนะของตัวควบคุมแบบต่างๆ ได้แก่ ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว ตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบู-ฟัซซี่ เมื่อปรับเปลี่ยนสถานการณ์การควบคุม ซึ่งได้ดำเนินการปรับเปลี่ยนความยาวก้านลูกตุ้ม มุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถ โดยพิจารณาเวลาในการเข้าสู่เสถียรภาพของระบบ รวมทั้งดำเนินการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อมีการรบกวนจากภายนอก

บทที่ 7 เป็นบทสรุปและข้อเสนอแนะ

ภาคผนวก ก. รายละเอียดการหาแบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยวิธีการของลากรางจ์

ภาคผนวก ข. รายละเอียดการหาแบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบสองข้อต่อ ด้วยวิธีการของลากรางจ์

ภาคผนวก ก. รายละเอียดการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อ ด้วยตัวควบคุมนิวโร-ตาบิว-ฟัซซี่

ภาคผนวก ง. โครงข่ายประสาทเทียมและการเรียนรู้แบบแพร่กลับ

ภาคผนวก จ. เวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบ โดยใช้ตัวควบคุมแบบต่างๆ

ภาคผนวก ฉ. รายละเอียดโปรแกรม MATLAB ที่ใช้สำหรับสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว

ภาคผนวก ช. รายละเอียด โปรแกรมภาษาซี ที่ใช้สำหรับสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่

ภาคผนวก ซ. รายละเอียด โปรแกรมภาษาซี ที่ใช้สำหรับสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ตาบิว-ฟัซซี่

ภาคผนวก ฅ. รายละเอียดโปรแกรมภาษาซี ที่ใช้สำหรับสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อ ด้วยตัวควบคุมนิวโร-ตาบิว-ฟัซซี่

ภาคผนวก ฉ. บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา



## บทที่ 2

### ระบบควบคุมแบบฟัซซี่

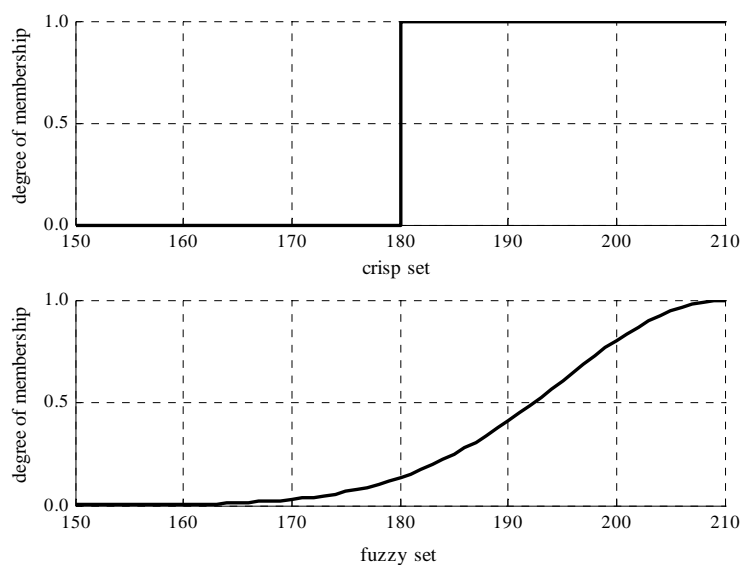
#### 2.1 บทนำ

การวิเคราะห์และออกแบบเพื่อควบคุมระบบด้วยวิธีการดั้งเดิม (conventional control) เช่น การควบคุมแบบอมตะ (classical control) หรือการควบคุมสมัยใหม่ (modern control) ต้องการแบบจำลองที่แม่นยำของกระบวนการที่ควบคุม รวมทั้งต้องมีความรู้ความเข้าใจเกี่ยวกับทฤษฎีการควบคุมที่ใช้เป็นอย่างดี แต่เพราะมนุษย์มิได้มีความรู้ลึกซึ้งเกี่ยวกับกายภาพของทุกระบบ จึงไม่สามารถหาแบบจำลองทางคณิตศาสตร์ (mathematical models) ให้กับระบบทุกประเภทได้อย่างแม่นยำ ในกรณีที่ระบบมีความซับซ้อนมากๆ การหาแบบจำลองทางคณิตศาสตร์ต้องใช้เวลานานและยุ่งยาก อาจทำให้ความถูกต้องเที่ยงตรงลดลง เนื่องจากแบบจำลองทางคณิตศาสตร์เป็นเสมือนหัวใจของการวิเคราะห์และออกแบบระบบด้วยวิธีการแบบอมตะ ดังนั้นระบบที่ขาดแบบจำลองทางคณิตศาสตร์ที่แม่นยำย่อมทำให้วิธีการแบบอมตะขาดประสิทธิภาพไปด้วย การแก้ปัญหาข้อจำกัดของวิธีการควบคุมแบบดั้งเดิมแนวทางหนึ่ง คือ การใช้วิธีการควบคุมแบบชาญฉลาด (intelligent control) ตัวอย่างเช่น ตัวควบคุมฟัซซี่ลอจิก (fuzzy logic controller) ซึ่งเกิดจากแนวคิดและทฤษฎีฟัซซี่เซตของ Lotfi A. Zadeh วิธีการนี้จะอาศัยข้อมูลความรู้และประสบการณ์การทำงานของมนุษย์เป็นพื้นฐานในการควบคุม ไม่จำเป็นต้องพึ่งพาแบบจำลองทางคณิตศาสตร์ที่แม่นยำของระบบที่ควบคุม ทำให้สามารถประยุกต์วิธีการดังกล่าวเพื่อควบคุมระบบที่มีความซับซ้อนหรือคลุมเครือ (vague, ill-define) และมีความไม่เป็นเชิงเส้นได้อย่างมีประสิทธิภาพ ในการวิเคราะห์และออกแบบเพื่อควบคุมระบบที่มีความซับซ้อนด้วยตัวควบคุมแบบฟัซซี่จะแตกต่างจากวิธีการควบคุมแบบดั้งเดิม กล่าวคือ ในการดำเนินงานจะใช้ตัวแปรทางภาษา (linguistic variables) แทนการใช้ตัวแปรเชิงเลข (numerical variables) และบ่งบอกลักษณะความสัมพันธ์ระหว่างตัวแปรต่างๆ ด้วยประโยคเงื่อนไขแบบฟัซซี่ (fuzzy conditional statements) เป็นต้น

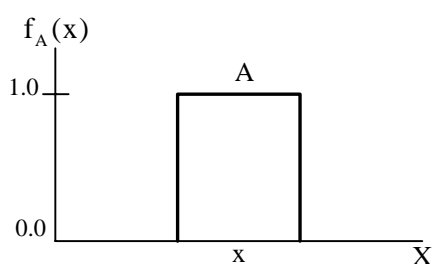
เนื้อหาในบทที่ 2 นี้กล่าวถึง ระบบควบคุมแบบฟัซซี่ ทฤษฎีฟัซซี่เซต อธิบายองค์ประกอบของระบบควบคุมแบบฟัซซี่ รวมถึงแนวทางการออกแบบ การกำหนดฟังก์ชันความเป็นสมาชิกให้กับตัวแปรอินพุตและเอาต์พุตของระบบ การกำหนดกฎฟัซซี่ รูปแบบการอนุมานฟัซซี่ และเทคนิคของการดีฟัซซิฟิเคชันของระบบควบคุมแบบฟัซซี่ ทั้งนี้เพื่อให้เกิดความเข้าใจพื้นฐานเกี่ยวกับการควบคุมระบบแบบฟัซซี่

## 2.2 ทฤษฎีฟัซซีเซต

ฟัซซีลอจิกถูกพัฒนาขึ้นโดย Lotfi A. Zadeh เมื่อปี ค.ศ. 1965 ซึ่งมีรากฐานอยู่บนทฤษฎีฟัซซีเซต ซึ่งฟัซซีเซตเป็นกลุ่มของสิ่งของหรือวัตถุที่มีการเปลี่ยนแปลงความเป็นสมาชิกในเซตอย่างค่อยเป็นค่อยไป หรืออาจกล่าวได้ว่าค่าที่แสดงความเป็นสมาชิกมีลักษณะต่อเนื่อง (สราวุฒิสุจิตจร, 2538) เมื่อ 0 หมายถึง การไม่เป็นสมาชิกในเซต และ 1 หมายถึง การเป็นสมาชิกในเซตอย่างสมบูรณ์ โดยจะมีการกำหนดค่าความเป็นสมาชิกผ่านฟังก์ชันความเป็นสมาชิก (membership function) เพื่อระบุค่าระดับความเป็นสมาชิก (degree of membership) ของสมาชิกใดๆ ในเซตที่อยู่ในขอบเขตของเอกภพสัมพัทธ์ (universe of discourse) ที่พิจารณา ในขณะที่เซตปกติธรรมดา (crisp set) จะพิจารณาสิ่งของหรือวัตถุใดๆ ว่าเป็นสมาชิกหรือไม่เป็นสมาชิกของเซตเท่านั้น ค่าที่ใช้แสดงลักษณะสมบัติของสิ่งของดังกล่าวมีเพียง 0 หรือ 1 ซึ่ง 0 หมายถึง การไม่เป็นสมาชิกในเซต และ 1 หมายถึง การเป็นสมาชิกในเซต ดังนั้นจะเห็นได้ว่าฟัซซีเซตมีความแตกต่างไปจากเซตปกติธรรมดาในเรื่องการให้ค่าที่แสดงลักษณะสมบัติความเป็นสมาชิกและความไม่เป็นสมาชิกในเซต ซึ่งแสดงได้ดังรูปที่ 2.1 โดยเป็นเซตของช่วงความสูงของมนุษย์จาก 150 เซ็นติเมตร ถึง 210 เซ็นติเมตร เมื่อพิจารณาเซตปกติธรรมดาจะเห็นได้ว่า ช่วงความสูง 180 เซ็นติเมตร ถึง 210 เซ็นติเมตร เป็นสมาชิกของเซตอย่างสมบูรณ์ ส่วนที่เหลือนอกช่วงดังกล่าวไม่เป็นสมาชิกอย่างสิ้นเชิง เมื่อพิจารณาฟัซซีเซต พบว่า ช่วงความสูงจาก 150 เซ็นติเมตร ถึง 210 เซ็นติเมตร แสดงความเป็นสมาชิกอย่างค่อยเป็นค่อยไป โดยมีระดับความเป็นสมาชิกอยู่ในช่วง 0 ถึง 1



รูปที่ 2.1 ความแตกต่างของความเป็นสมาชิกระหว่างเซตปกติธรรมดากับฟัซซีเซต



รูปที่ 2.2 ฟังก์ชันลักษณะเฉพาะของเซตปกติธรรมดา

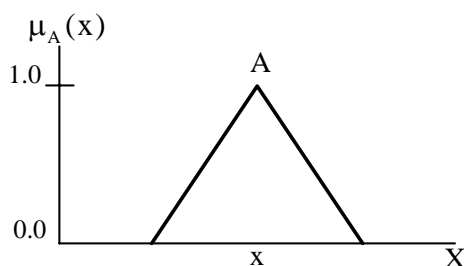
ถ้าให้  $X$  เป็นเอกภพสัมพัทธ์ที่พิจารณา ส่วนสมาชิกในเอกภพสัมพัทธ์กำหนดให้เป็น  $x$  จากรูปที่ 2.2 จะได้ว่าเซตปกติธรรมดาหรือคริปส์เซต  $A$  ของ  $X$  คือ ฟังก์ชัน  $f_A(x)$  เรียกว่า ฟังก์ชันลักษณะเฉพาะ (characteristic function) ของ  $x$  ในเซต  $A$  เมื่อพิจารณาสมการที่ (2-1) พบว่า ในเซตปกติธรรมดาจะมีลอจิกเพียงสองค่าคือ 0 และ 1 โดยที่ 0 หมายถึง การไม่เป็นสมาชิกในเซต และ 1 หมายถึง การเป็นสมาชิกในเซตอย่างสมบูรณ์

$$f_A(x) : X \rightarrow 0, 1 \quad (2-1)$$

โดยที่

$$f_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

ในกรณีของฟัชซีเซตจะเป็นแบบลอจิกหลายค่า (multivalued logic) ซึ่งอยู่ในช่วง 0 ถึง 1 โดยยอมรับการเป็นสมาชิกของสมาชิกที่มีลักษณะตามเซตเพียงบางส่วน ทั้งนี้จะมีการเปลี่ยนแปลงทีละน้อยระหว่างการมีคุณสมบัติของการเป็นสมาชิกอย่างครบถ้วนกับไม่มีคุณสมบัติของการเป็นสมาชิกเลย ดังตัวอย่างที่แสดงในรูปที่ 2.3 ซึ่งฟังก์ชันความเป็นสมาชิกมีลักษณะเป็นรูปสามเหลี่ยม ดังนั้น ระดับความเป็นสมาชิกของ  $x$  จึงอยู่ในช่วง 0 ถึง 1 โดยในช่วงแรกค่าระดับความเป็นสมาชิกจะค่อยๆ เพิ่มจากระดับ 0 จนกระทั่งค่าระดับความเป็นสมาชิกของ  $x$  มีค่าเป็น 1 จากนั้นค่าระดับความเป็นสมาชิกจึงค่อยๆ ลดลงจนกระทั่งมีค่าเป็น 0 ซึ่งจะเห็นได้ว่าค่าระดับความเป็นสมาชิกของฟัชซีเซตจะมีการเปลี่ยนแปลงค่าในลักษณะค่อยเป็นค่อยไป ทั้งนี้ค่าระดับความเป็นสมาชิกของ  $x$  ที่ตำแหน่งต่างๆ สามารถคำนวณได้โดยอาศัยฟังก์ชันของรูปสามเหลี่ยม



รูปที่ 2.3 ฟังก์ชันความเป็นสมาชิกของฟัซซีเซต

เมื่อพิจารณารูปที่ 2.3 จะได้ว่าฟัซซีเซต A อยู่ในเอกภพสัมพัทธ์ X ซึ่งค่าระดับความเป็นสมาชิกของแต่ละสมาชิก x ในเอกภพสัมพัทธ์ X แทนด้วยฟังก์ชัน  $\mu_A(x)$  เรียกว่า ฟังก์ชันความเป็นสมาชิก (membership function) ของเซต A ซึ่งแสดงได้ด้วยสมการที่ (2-2)

$$\mu_A(x) : X \rightarrow [0,1] \quad (2-2)$$

โดยที่

$\mu_A(x) = 1$  ถ้า x เป็นสมาชิกในเซต A อย่างสมบูรณ์

$\mu_A(x) = 0$  ถ้า x ไม่เป็นสมาชิกในเซต A

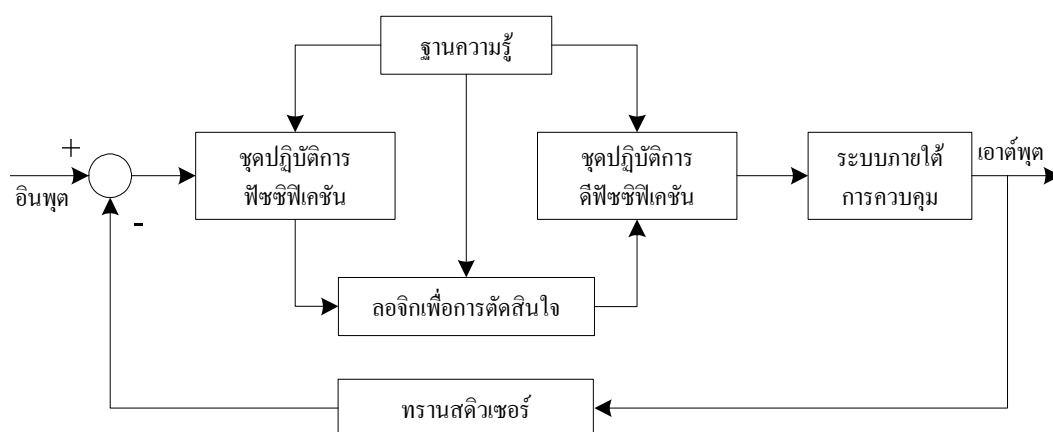
$0 < \mu_A(x) < 1$  ถ้า x เป็นสมาชิกในเซต A เพียงบางส่วน

ดังนั้นฟัซซีเซต A ในเอกภพสัมพัทธ์ X สามารถแสดงในลักษณะของคู่อันดับได้ดังสมการที่ (2-3)

$$A = \{(x, \mu_A(x)); x \in X, \mu_A(x) \in [0,1]\} \quad (2-3)$$

### 2.3 องค์ประกอบของระบบควบคุมแบบฟัซซี

โครงสร้างพื้นฐานของระบบควบคุมแบบฟัซซีแสดงได้ด้วยแผนภาพในรูปที่ 2.4 โดยสัญญาณความคลาดเคลื่อนระหว่างอินพุตกับเอาต์พุตจะเข้าสู่ตัวควบคุมฟัซซี จากนั้นสัญญาณเอาต์พุตจากตัวควบคุมฟัซซีจะถูกป้อนเข้าสู่ระบบที่ต้องการควบคุม จนกระทั่งได้สัญญาณเอาต์พุตของระบบ ซึ่งตัวควบคุมฟัซซีประกอบด้วยชุดปฏิบัติการฟัซซิฟิเคชัน (fuzzification unit) ฐานความรู้ (knowledge base) ตรรกะเพื่อการตัดสินใจ (decision making logic) และชุดปฏิบัติการดีฟัซซิฟิเคชัน (defuzzification unit) ดังมีรายละเอียดต่อไปนี้



รูปที่ 2.4 โครงสร้างพื้นฐานของระบบควบคุมแบบฟัซซี

**ชุดปฏิบัติการฟัซซีฟิเคชัน** ทำหน้าที่แปลงสัญญาณข้อมูลทางด้านอินพุตแบบธรรมดาเป็น ตัวแปรทางภาษาหรือตัวแปรฟัซซี โดยอยู่ในรูปของค่าระดับความเป็นสมาชิกแต่ละฟัซซีเซ็ท บางครั้งอาจรวมชุดวัดสัญญาณด้วยทรานสดิวเซอร์และชุดเปรียบเทียบสัญญาณไว้ในส่วนที่เป็นชุดปฏิบัติการฟัซซีฟิเคชันนี้ด้วย เนื่องจากตัวแปรอินพุตแต่ละตัวที่วัดได้จากเครื่องมือวัดอาจอยู่นอก ขอบเขตเอกภพสัมพัทธ์ที่กำหนด ดังนั้นต้องดำเนินการแปลงค่าตัวแปรอินพุตเหล่านี้ให้อยู่ใน ขอบเขตเอกภพสัมพัทธ์ที่กำหนดด้วยค่าสเกล (scaling) หรือตัวปรับค่า (gain factor) เรียก ขั้นตอนนี้ว่า การทำนอร์มัลไลซ์ (normalization) ตัวอย่างเช่น สัญญาณความเร็วจากมอเตอร์ กระแสตรงที่วัดได้อยู่ในช่วง  $[-1000, +1000]$  รอบต่อนาที ถ้ากำหนดให้เอกภพสัมพัทธ์อยู่ระหว่าง  $[-1, +1]$  ดังนั้นต้องทำการคูณค่าแรงดันที่วัดได้ด้วย  $(1/1000)$  รอบต่อนาที เป็นต้น

**ฐานความรู้** ประกอบด้วยส่วนที่เป็นฐานข้อมูล (data base) และฐานกฎการควบคุม (control rule base) ส่วนที่เป็นฐานข้อมูลนั้นให้นิยามต่างๆ ที่ใช้กำหนดกฎการควบคุมนั้น รวมทั้งเป็นศูนย์รวมของกฎการควบคุมแบบตัวแปรทางภาษา ซึ่งแสดงถึงแนวทางการควบคุมให้ ระบบมีพฤติกรรมตามที่ต้องการ

**ลอจิกเพื่อการตัดสินใจ** เป็นส่วนที่ให้การเลียนแบบการตัดสินใจของมนุษย์ โดยอาศัย แนวคิดทางคณิตศาสตร์ และการอนุมานแบบฟัซซี

**ชุดปฏิบัติการดีฟัซซีฟิเคชัน** ทำหน้าที่แปลงข้อมูลแบบฟัซซีไปเป็นตัวแปรธรรมดา (conventional variable) หรือคริปส์เอาต์พุตที่สมนัยกัน เพื่อให้ระบบภายใต้การควบคุมสามารถ เข้าใจสัญญาณดังกล่าวได้ ข้อมูลแบบฟัซซีนี้เป็นข้อมูลที่ได้มาจากการอนุมานกฎการควบคุมแบบ ฟัซซี เนื่องจากขอบเขตเอกภพสัมพัทธ์ของเอาต์พุตในฟังก์ชันความเป็นสมาชิกอาจยังไม่อยู่ใน ขอบเขตที่นำไปใช้งานได้ ดังนั้นต้องทำการแปลงค่าเอาต์พุตในเอกภพสัมพัทธ์ให้อยู่ในขอบเขต

เอาต์พุตจริงที่ใช้ควบคุมระบบ เรียกขั้นตอนนี้ว่า การคืนอร์มัลไลซ์ (denormalization) ตัวอย่างเช่น เอกภพสัมพัทธ์ของเอาต์พุตในฟังก์ชันความเป็นสมาชิกอยู่ในช่วง  $[-1, +1]$  แต่สัญญาณแรงดันที่ป้อนให้มอเตอร์กระแสตรงมีขอบเขตอยู่ในช่วง  $[-12, +12]$  โวลต์ ดังนั้นต้องทำการคูณค่าเอาต์พุตในเอกภพสัมพัทธ์ด้วย 12 ก่อนส่งไปควบคุมระบบจริง เป็นต้น

## 2.4 การออกแบบระบบควบคุมแบบฟัซซี่

การทำงานของตัวควบคุมแบบฟัซซี่จะอาศัยคณิตศาสตร์แบบฟัซซี่ และตัวแปรฟัซซี่เมื่อต้องเกี่ยวข้องกับระบบทางกายภาพที่ไม่เป็นฟัซซี่ ตัวแปรต่างๆ ที่ไม่เป็นฟัซซี่ ได้แก่ เอาต์พุตของกระบวนการ หรือระบบภายใต้การควบคุม อินพุตอ้างอิง เป็นต้น โดยผ่านกระบวนการทำให้เป็นฟัซซี่ด้วยฟัซซี่โอเพอเรเตอร์ (fuzzy operator) ที่เรียกว่า ฟัซซี่ฟายเออร์ (fuzzifier) ผลของการอนุมานที่ได้จากตัวควบคุมแบบฟัซซี่จะนำไปควบคุมกระบวนการ โดยอยู่ในรูปของตัวแปรฟัซซี่ ซึ่งระบบหรือกระบวนการภายใต้การควบคุมนั้นไม่สามารถเข้าใจได้ ตัวแปรฟัซซี่ดังกล่าวจึงต้องผ่านกระบวนการทำให้ไม่เป็นฟัซซี่ด้วยโอเพอเรเตอร์ชนิดดีฟัซซี่ฟายเออร์ (defuzzifier) หรือดีฟัซซี่ฟิเคชันโอเพอเรเตอร์ (defuzzification operator)

การออกแบบตัวควบคุมฟัซซี่จะต้องคำนึงถึงพารามิเตอร์ต่างๆ ต่อไปนี้

- 1) วิธีการฟัซซี่ฟิเคชันและการตีความของฟัซซี่ฟายเออร์
- 2) ฐานข้อมูล ซึ่งจะต้องคำนึงถึง
  - เอกภพสัมพัทธ์ การกำหนดการแบ่งส่วน และการนอร์มัลไลซ์
  - การกำหนดช่วงอินพุต เอาต์พุต ให้เหมาะสมสอดคล้องกับการแบ่งส่วนตัวแปรแบบฟัซซี่
  - ความครบถ้วน
  - ลักษณะของฟังก์ชันความเป็นสมาชิก
- 3) ฐานกฎ ซึ่งจะต้องคำนึงถึง
  - การเลือกตัวแปรสถานะ (อินพุต) และตัวแปรควบคุม (เอาต์พุต) สำหรับกฎการควบคุม
  - แหล่งที่มาและสังเคราะห์กฎการควบคุม
  - ชนิดของกฎการควบคุมแบบฟัซซี่
  - ความครบถ้วน ความสอดคล้องกันอย่างเหมาะสม
- 4) ลอจิกเพื่อการตัดสินใจ ซึ่งจะต้องคำนึงถึงสิ่งต่อไปนี้
  - นิยามของฟัซซี่อิมพลิเคชัน (fuzzy implication)
  - การตีความค่าเชื่อมประโยคแบบ and

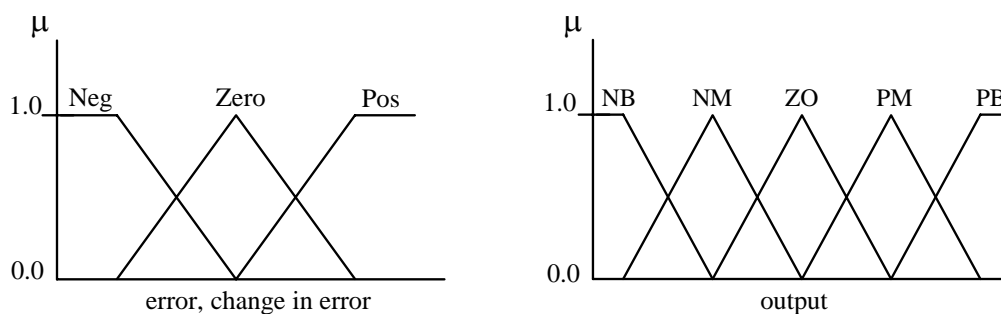
- การตีความค่าเชื่อมประโยคแบบ also
  - นิยามของคอมโพสิชันนัลโอเพอเรเตอร์ (compositional operator)
  - กลไกการอนุมาน (inference mechanism)
- 5) วิธีการตีฟัซซีฟิเคชันและการตีความตีฟัซซีฟายเออร์

## 2.5 กฎฟัซซี

การควบคุมระบบแบบฟัซซี เป็นการวิเคราะห์และแก้ไขปัญหาเกี่ยวกับระบบที่มีความซับซ้อนโดยอาศัยคณิตศาสตร์แบบฟัซซี อาศัยกระบวนการตัดสินใจและความคิดอ่านของมนุษย์เป็นพื้นฐานในการควบคุม โดยแสดงความสัมพันธ์ระหว่างสถานะต่างๆ ในระบบพลวัตในรูปแบบประโยคฟัซซี ซึ่งมีลักษณะเป็นประโยคเงื่อนไข ถ้า... แล้ว... (if... then...) เรียกประโยคเงื่อนไขเหล่านี้ว่า กฎฟัซซี (fuzzy rule) ในการดำเนินงานจะใช้ตัวแปรทางภาษาแทนการใช้ตัวแปรเชิงเลข ซึ่งตัวแปรทางภาษาเป็นตัวแปรที่มีการให้ค่าเป็นคำหรือประโยค เพื่อบ่งบอกลักษณะของสิ่งของใดๆ ซึ่งมีค่าตามค่าให้ค่าของฟัซซีเซต ตัวอย่างเช่น ถ้าความเร็วรอบของมอเตอร์มีค่ามาก แล้ว ลดแรงดันที่ให้กับมอเตอร์ ซึ่งตัวแปรทางภาษาหรือตัวแปรฟัซซี คือความเร็วรอบของมอเตอร์ และแรงดันที่ให้กับมอเตอร์ เป็นต้น

กฎฟัซซีในประโยคเงื่อนไข ถ้า...แล้ว... ประกอบด้วยส่วนของตัวแปรเงื่อนไข (antecedent) และส่วนของตัวแปรผลลัพธ์ (consequent) ซึ่งในแต่ละส่วนอาจมีได้หลายตัวแปร โดยเชื่อมประโยคด้วยโอเพอเรเตอร์ and หรือ or เพื่อใช้ในการแก้ปัญหาสำหรับควบคุมระบบอินพุตเดี่ยว เดียวเอาต์พุตเดี่ยว (SISO) และระบบหลายอินพุตหลายเอาต์พุต (MIMO) กฎการควบคุมฟัซซีสามารถกำหนดขึ้นด้วยการเขียนเป็นประโยคแสดงอาการของการพยายามควบคุมกับผลที่จะเกิดขึ้นกับการตอบสนองของระบบ โดยอาศัยความรู้ความชำนาญและประสบการณ์ของมนุษย์ในการกำหนดกฎฟัซซีเพื่อควบคุมระบบ

ตัวอย่างการกำหนดกฎฟัซซี เมื่อตัวแปรฟัซซีในส่วนเงื่อนไขประกอบด้วยสองอินพุต ได้แก่ สัญญาณความคลาดเคลื่อน (error) และการเปลี่ยนแปลงของสัญญาณความคลาดเคลื่อน (change in error) โดยเชื่อมประโยคด้วยโอเพอเรเตอร์ and ซึ่งตัวแปรในส่วนผลลัพธ์กำหนดเป็นสัญญาณเอาต์พุต (output) ทั้งนี้ฟังก์ชันความเป็นสมาชิกของตัวแปรฟัซซีในส่วนเงื่อนไขและตัวแปรในส่วนผลลัพธ์แสดงได้ดังรูปที่ 2.5 โดยรูปแบบฟังก์ชันความเป็นสมาชิกของตัวแปรในส่วนเงื่อนไขและส่วนผลลัพธ์ของกฎฟัซซีที่ปรากฏในรูปดังกล่าวประกอบด้วยรูปสามเหลี่ยม และรูปสี่เหลี่ยมคางหมู ซึ่งจำนวนการแบ่งส่วนตัวแปรฟัซซีสำหรับฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตในส่วนเงื่อนไขประกอบด้วย 3 ฟัซซีเซต และฟังก์ชันความเป็นสมาชิกในส่วนผลลัพธ์ประกอบด้วย 5 ฟัซซีเซต



รูปที่ 2.5 ฟังก์ชันความเป็นสมาชิกของตัวแปรฟัซซีในส่วนเงื่อนไขและส่วนผลลัพธ์

โดยที่ตัวแปรทางภาษาสำหรับแต่ละฟัซซีเซตที่ปรากฏในรูปดังกล่าวมีความหมายดังต่อไปนี้ คือ Neg (negative) Zero (zero) และ Pos (positive) เป็นตัวแปรทางภาษาในส่วนเงื่อนไขของกฎฟัซซีสำหรับฟัซซีเซตของสัญญาณความคลาดเคลื่อน และการเปลี่ยนแปลงของสัญญาณความคลาดเคลื่อน ในขณะที่ NB (negative big) NM (negative medium) ZO (zero) PM (positive medium) และ PB (positive big) เป็นตัวแปรทางภาษาในส่วนผลลัพธ์ของกฎฟัซซีสำหรับฟัซซีเซตของเอาต์พุต ทั้งนี้การกำหนดกฎฟัซซีแสดงได้ดังตารางที่ 2.1

ตารางที่ 2.1 การกำหนดกฎฟัซซีสำหรับตัวแปรฟัซซีที่กำหนดขึ้น

error	change in error		
	Neg	Zero	Pos
Neg	NB	NM	ZO
Zero	NM	ZO	PM
Pos	ZO	PM	PB

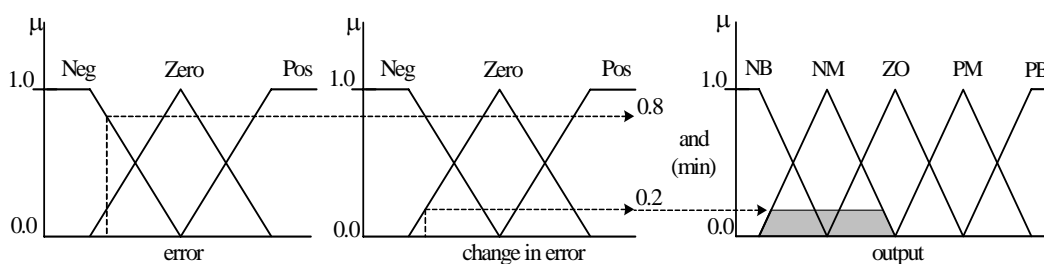
เมื่อพิจารณาการกำหนดกฎฟัซซีในตารางที่ 2.1 สามารถนำมาเขียนกฎฟัซซีได้จำนวน 9 กฎ ดังต่อไปนี้

- กฎที่ 1 ถ้า error เป็น Neg และ change in error เป็น Neg แล้ว output เป็น NB
- กฎที่ 2 ถ้า error เป็น Neg และ change in error เป็น Zero แล้ว output เป็น NM
- กฎที่ 3 ถ้า error เป็น Neg และ change in error เป็น Pos แล้ว output เป็น ZO
- กฎที่ 4 ถ้า error เป็น Zero และ change in error เป็น Neg แล้ว output เป็น NM
- กฎที่ 5 ถ้า error เป็น Zero และ change in error เป็น Zero แล้ว output เป็น ZO

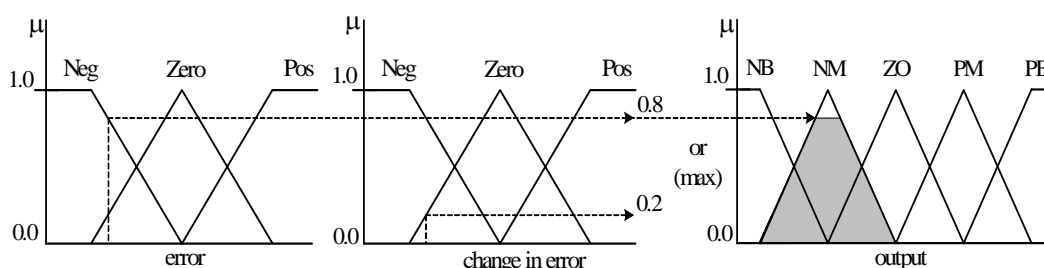


กฎที่ 6 ถ้า error เป็น Zero และ change in error เป็น Pos แล้ว output เป็น PM  
 กฎที่ 7 ถ้า error เป็น Pos และ change in error เป็น Neg แล้ว output เป็น ZO  
 กฎที่ 8 ถ้า error เป็น Pos และ change in error เป็น Zero แล้ว output เป็น PM  
 กฎที่ 9 ถ้า error เป็น Pos และ change in error เป็น Pos แล้ว output เป็น PB

เมื่อพิจารณาในส่วนของค่าเชื่อมประโยค and จะเป็นการเลือกค่าระดับความเป็นสมาชิกที่น้อยที่สุดจากตัวแปรอินพุตทุกตัว ส่วน or หมายถึง การเลือกค่าระดับความเป็นสมาชิกของตัวแปรที่มากที่สุด ตัวอย่างเช่น ถ้า error เป็น Neg และ change in error เป็น Zero แล้ว output เป็น NM สามารถอธิบายได้ดังรูปที่ 2.6 และกฎฟuzzy ถ้า error เป็น Neg หรือ change in error เป็น Zero แล้ว output เป็น NM แสดงได้ดังรูปที่ 2.7 ตามลำดับ



รูปที่ 2.6 การเชื่อมประ โยคในกฎฟuzzy ด้วยและ



รูปที่ 2.7 การเชื่อมประ โยคในกฎฟuzzy ด้วยหรือ

ในระบบควบคุมที่พิจารณาตัวแปรสถานะของระบบหลายตัวสามารถกำหนดตัวแปรอินพุตหรือตัวแปรฟuzzy ในส่วนของเงื่อนไขของกฎฟuzzy ได้หลายตัวเช่นกัน ซึ่งมีรูปแบบการเขียนกฎฟuzzy ได้ดังนี้

if  $x_1 \in \mu_1$  and  $x_2 \in \mu_2$  and ...  $x_k \in \mu_k$ , then ...

โดยที่  $x_i (i = 1, \dots, k)$  คือ ตัวแปรฟัซซี่ในส่วนเงื่อนไขของกฎฟัซซี่ และ  $\mu_i (i = 1, \dots, k)$  เป็นค่าระดับความเป็นสมาชิกในฟัซซี่เซต ส่วนการกำหนดส่วนผลลัพธ์ในกฎฟัซซี่ สามารถพิจารณาได้ 3 แนวทาง (Filippidis, Jain, and Silva, 1999) ดังนี้คือ

1) แบบจำลองของแมมดานี (Mamdani model)

การกำหนดกฎฟัซซี่ในส่วนผลลัพธ์ตามแนวทางของแมมดานี สามารถพิจารณาได้จากสมการที่ (2-4) โดยที่  $y$  คือ ตัวแปรเอาต์พุต และ  $A$  คือ ตัวแปรภาษาของฟัซซี่เซตที่ปรากฏในฟังก์ชันความเป็นสมาชิกที่กำหนด ตัวอย่างเช่น  $y = \text{big}$  หรือ  $y = \text{negative small}$  เป็นต้น การกำหนดกฎฟัซซี่ในลักษณะนี้เหมาะสำหรับการใช้งานกับระบบผู้เชี่ยวชาญ (expert systems) ซึ่งต้องอาศัยความเชี่ยวชาญและประสบการณ์ในการกำหนดกฎฟัซซี่เหล่านี้

$$y = A \quad (2-4)$$

2) แบบจำลองของ TSK (TSK model)

แบบจำลองของ TSK (Takagi-Sugeno-Kang) กำหนดให้ค่าของตัวแปรผลลัพธ์ในกฎฟัซซี่เป็นฟังก์ชันของตัวแปรอินพุต กล่าวคือ ผลลัพธ์ของตัวแปรเอาต์พุตเป็นการผสมผสานเชิงเส้น (linear combination) ของน้ำหนักตัวแปรอินพุตแต่ละตัว ดังแสดงได้ด้วยสมการที่ (2-5) โดยที่  $a_0$  และ  $a_i$  เป็นค่าคงที่ และ  $x_i$  แทนด้วยตัวแปรอินพุต

$$y = a_0 + \sum a_i x_i \quad (2-5)$$

3) แบบจำลองฟัซซี่อย่างง่าย (simplified fuzzy model)

การกำหนดกฎฟัซซี่โดยใช้แบบจำลองอย่างง่าย สามารถพิจารณาได้จากสมการที่ (2-6) โดยที่  $c$  เป็นค่าคงที่ ดังนั้น การเขียนกฎฟัซซี่ในลักษณะนี้ในส่วนตัวแปรผลลัพธ์อาจกำหนดให้เป็นจำนวนจริงเดี่ยว (singleton) ที่มีค่าคงที่ การกำหนดกฎฟัซซี่อย่างง่ายนี้เป็นกรณีพิเศษของแบบจำลองแมมดานี และ TSK ข้อดีของวิธีการนี้คือง่ายต่อการออกแบบกฎฟัซซี่ในส่วนผลลัพธ์ รวมทั้งช่วยให้ง่ายต่อการอนุมานกฎฟัซซี่ด้วย

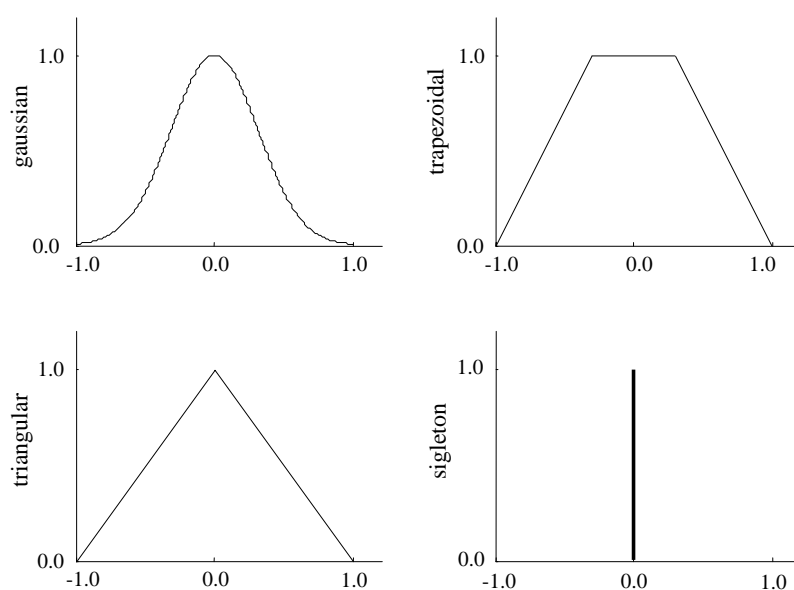
$$y = c \quad (2-6)$$

## 2.6 การอนุมานฟัซซี่

การอนุมาน (inference) หมายถึง การเอ็กซ์ทิวชันกฎ ในระบบควบคุมแบบฟัซซี่พบว่า เทคนิคการอนุมานมีสองลักษณะ (สราวุฒิ สุจิตจร, 2538) คือ ลักษณะที่หนึ่ง เป็นการอนุมานกฎ การควบคุมแยกจากกันแต่ละกฎ โดยทำการอนุมานทุกกฎ และนำผลการอนุมานที่ได้จากแต่ละกฎ มารวมกันด้วยการใช้เทคนิคเฉพาะ เช่นการหาค่าเฉลี่ยแบบฟัซซี่ เป็นต้น ลักษณะที่สอง เป็นการอนุมานแบบรวมกฎ เทคนิคการอนุมานนี้ใช้วิธีการคำนวณเมทริกซ์ความสัมพันธ์ (relational matrix) ซึ่งเป็นตัวแทนของการอนุมานกฎทุกกฎ แต่การอนุมานในลักษณะนี้ไม่เป็นที่นิยม เพราะสิ้นเปลืองเนื้อที่หน่วยความจำ นอกจากนี้พบว่ถ้าจำนวนกฎมีมาก เมทริกซ์ความสัมพันธ์ที่คำนวณได้จะมีความคลุมเครือมาก ทำให้ผลการอนุมานขาดความแม่นยำ

## 2.7 การกำหนดฟังก์ชันความเป็นสมาชิก

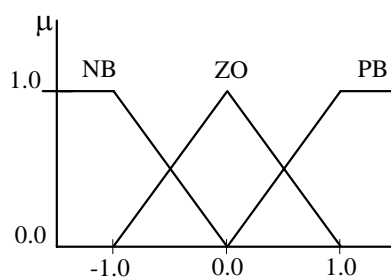
การเลือกฟังก์ชันความเป็นสมาชิกที่เหมาะสมนั้น สามารถกำหนดได้จากข้อมูลความรู้ และประสบการณ์ของมนุษย์โดยตรง หรือที่เรียกว่าเป็นวิธีแบบศึกษาสำนึก (heuristic approach) ค่าแสดงความเป็นสมาชิกที่ได้จึงมีลักษณะเป็นแบบอัตวิสัย (subjective) ขึ้นอยู่กับผู้ออกแบบแต่ละคน อีกลักษณะหนึ่งอาจกำหนดรูปแบบของฟังก์ชันความเป็นสมาชิก โดยอาศัยผลการวิเคราะห์เชิงคำนวณที่เกี่ยวข้องกับกลุ่มวัตถุ หรือสิ่งของที่ต้องการให้มีการแบ่งแยกความเป็นสมาชิก และไม่เป็น



รูปที่ 2.8 รูปแบบฟังก์ชันความเป็นสมาชิก

สมาชิกในฟัซซีเซ็ท วิธีการเช่นนี้เรียกว่าแบบวัตถุวิสัย (objective) รูปแบบฟังก์ชันความเป็นสมาชิกในฟัซซีเซ็ทอาจมีรูปร่างเช่นไรก็ได้ เช่น รูปสามเหลี่ยม รูปสี่เหลี่ยมคางหมู รูปประฆังคว่ำ เป็นต้น รวมทั้งอาจมีรูปแบบเป็นจำนวนจริงเดี่ยวยก็ได้ ดังแสดงในรูปที่ 2.8

ในการควบคุมระบบสามารถกำหนดฟังก์ชันความเป็นสมาชิกได้หลายรูปแบบขึ้นอยู่กับความเหมาะสมและประสิทธิภาพของผู้ออกแบบเป็นสำคัญ นอกจากนี้อาจนำวิธีการแบบชาญฉลาดเข้าช่วยในการปรับช่วงกว้างของฟัซซีเซ็ท การแบ่งส่วนฟัซซีเซ็ท รวมทั้งกำหนดรูปแบบฟังก์ชันความเป็นสมาชิกให้เหมาะสม เช่น ใช้จินเนติกอัลกอริทึม เป็นต้น รูปที่ 2.9 แสดงฟังก์ชันความเป็นสมาชิกที่ประกอบด้วยรูปสามเหลี่ยมและรูปสี่เหลี่ยมคางหมู โดยตัวแปรทางภาษาสำหรับแต่ละฟัซซีเซ็ทแทนด้วย NB (negative big) ZO (zero) และ PB (positive big) และตารางที่ 2.2 แสดงขอบเขตของฟัซซีเซ็ทในฟังก์ชันความเป็นสมาชิก ซึ่งสามารถพิจารณาขอบเขตของแต่ละฟัซซีเซ็ทได้จากรูปที่ 2.9 ซึ่งเอกภพสัมพัทธ์กำหนดให้อยู่ในช่วง  $[-1, +1]$



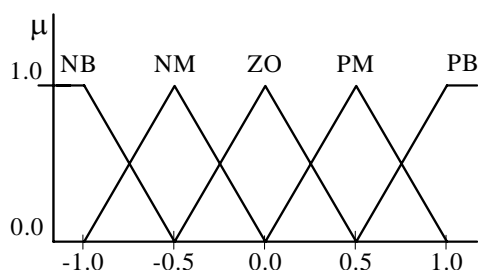
รูปที่ 2.9 การกำหนดรูปแบบฟังก์ชันความเป็นสมาชิก

ตารางที่ 2.2 ขอบเขตฟัซซีเซ็ทของฟังก์ชันความเป็นสมาชิก

เอกภพสัมพัทธ์ที่นอร์มัลไลซ์แล้ว	เซ็กเมนต์ที่นอร์มัลไลซ์แล้ว	ฟัซซีเซ็ทมูลฐาน
[-1.0, +1.0]	น้อยกว่าศูนย์	NB
	[-1.0, +1.0]	ZO
	มากกว่าศูนย์	PB

นอกจากรูปแบบของฟังก์ชันความเป็นสมาชิกแล้ว สิ่งที่ต้องพิจารณาอีกประการคือ การแบ่งส่วนตัวแปรฟัซซีเซ็ทนั้นคือ จำนวนฟัซซีเซ็ทในเอกภพสัมพัทธ์ที่กำหนด รูปที่ 2.10 แสดงการแบ่งส่วนตัวแปรฟัซซีเซ็ทเพื่อเพิ่มจำนวนฟัซซีเซ็ท ซึ่งเดิมประกอบด้วย 3 ฟัซซีเซ็ท ดังแสดงในรูปที่ 2.9 หลังจากthatเพิ่มการแบ่งส่วนตัวแปรฟัซซีเซ็ทแล้ว ประกอบด้วย 5 ฟัซซีเซ็ทดังนี้คือ NB (negative big)

NM (negative medium) ZO (zero) PM (positive medium) และ PB (positive big) ตารางที่ 2.3 แสดงขอบเขตของฟัซซี่เซตหลังจากที่เพิ่มการแบ่งส่วนฟัซซี่ ทั้งนี้เอกภพสัมพัทธ์ที่พิจารณา ยังคงกำหนดให้อยู่ในช่วง  $[-1, +1]$  เช่นกัน



รูปที่ 2.10 การแบ่งส่วนตัวแปรฟัซซี่

ตารางที่ 2.3 ขอบเขตฟัซซี่เซตจากการแบ่งส่วนตัวแปรฟัซซี่

เอกภพสัมพัทธ์ที่นอร์มัลไลซ์แล้ว	เซกเมนต์ที่นอร์มัลไลซ์แล้ว	ฟัซซี่เซตมูลฐาน
[-1.0, +1.0]	น้อยกว่า -0.5	NB
	[-1.0, 0.0]	NM
	[-0.5, +0.5]	ZO
	[0.0, 1.0]	PM
	มากกว่า +0.5	PB

## 2.8 เทคนิคการดีฟัซซิฟิเคชัน

เอาต์พุตจากกระบวนการอนุมานจะอยู่ในรูปของฟัซซี่เซตซึ่งเป็นการบ่งบอกถึงการกระจายความเป็นไปได้ (possibility distribution) ของการควบคุมที่จะส่งไปยังกระบวนการ ในการสั่งการควบคุมแก่กระบวนการต้องการสัญญาณสั่งการที่ไม่ใช่ฟัซซี่ (nonfuzzy or crisp control signal) ดังนั้นกระบวนการดีฟัซซิฟิเคชันเพื่อเปลี่ยนฟัซซี่เซตไปเป็นเอาต์พุตที่ไม่ใช่ฟัซซี่จึงเป็นสิ่งจำเป็น ดังแสดงในสมการที่ (2-7)

$$x_0 = D(x) = \text{defuzzifier}(x) \quad (2-7)$$

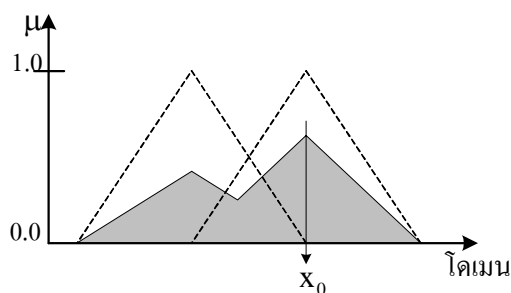
เมื่อ  $x_0$  คือ สัญญาณการควบคุมแบบธรรมดา

$x$  คือ ฟัซซี่เซตของคำสั่งการควบคุม

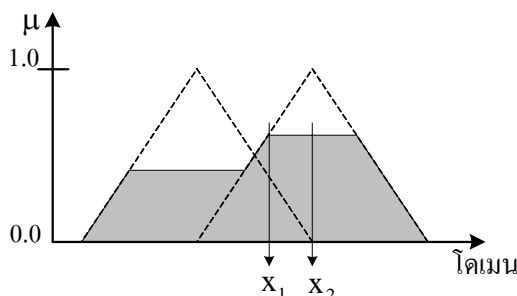
D( $\cdot$ ) คือ ดีฟัซซิฟิเคชันโอเปอเรเตอร์ (defuzzification operator)

เทคนิคการดีฟัซซิฟิเคชันที่ได้รับการใช้งานมีหลากหลายเทคนิค ต่างก็ให้เอาต์พุต  $x_0$  ที่ไม่เท่ากัน การเลือกใช้เทคนิคใดนั้นขึ้นอยู่กับผู้ออกแบบระบบควบคุมเป็นหลัก ในการพิจารณาเลือกใช้มักคำนึงถึงความยุ่งยากซับซ้อนในการคำนวณ ตัวอย่างเทคนิคดีฟัซซิฟิเคชันที่ได้รับการใช้งานอย่างกว้างขวาง ได้แก่

1) วิธีแบบเกณฑ์สูงสุด (max criterion) วิธีการนี้จะให้เอาต์พุตที่มีค่าการกระจายความเป็นไปได้สูงสุดเป็นค่าที่นำไปสั่งการควบคุม ในกรณีที่ผลการอนุมานให้ฟัซซีเซตแทนการกระจายความเป็นไปได้ที่เป็นแบบลดสัดส่วน (scaled fuzzy sets) ดังรูปที่ 2.11 ผลการดีฟัซซิฟิเคชันจะให้ค่า  $x_0$  ในโดเมน หากว่าการอนุมานกฎการควบคุมให้ฟัซซีเซตแบบตัดส่วน (clipped fuzzy set) อาจทำให้มีค่าความเป็นไปได้สูงสุดหลายค่าดังแสดงในรูปที่ 2.12 ในกรณีดังกล่าวการหาผลดีฟัซซิฟิเคชันจะเลือกค่าในโดเมนที่มีค่าความเป็นสมาชิกสูงสุดค่าแรกที่ค้นพบจากการค้นหาค่าในโดเมนจากน้อยไปมาก ดังนั้น จากรูปที่ 2.12 จะได้  $x_0 = x_1$



รูปที่ 2.11 ดีฟัซซิฟิเคชันแบบเกณฑ์สูงสุดเมื่อผลการอนุมานที่ได้เป็นฟัซซีเซตลดสัดส่วน



รูปที่ 2.12 ดีฟัซซิฟิเคชันแบบเกณฑ์สูงสุดเมื่อผลการอนุมานที่ได้เป็นฟัซซีเซตตัดส่วน

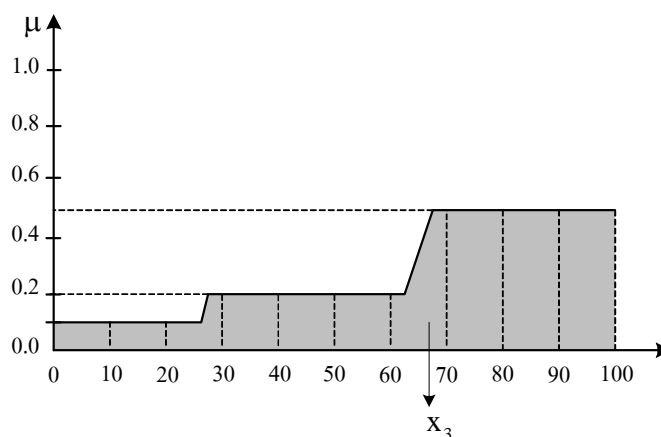
2) วิธีหาค่าเฉลี่ยของค่าสูงสุด (mean of maximum) เป็นการนำค่าเฉลี่ยของค่า  $x$  ต่างๆ ในโดเมนที่พบว่าค่าระดับความเป็นสมาชิก หรือ  $\mu$  มีค่าสูงสุด จะได้ผลการดีฟัซซิฟิเคชันดังสมการที่ (2-8) โดยที่ค่า  $k$  คือจำนวนช่วงของค่า  $x$  ในโดเมนที่มีค่า  $\mu$  สูงสุด ดังนั้นจากรูปที่ 2.12 ผลการดีฟัซซิฟิเคชันจะได้  $x_0 = x_2$

$$x_0 = \sum_{j=1}^k \frac{x_j}{k} \quad (2-8)$$

3) วิธีแบบศูนย์กลางพื้นที่ (center of area) วิธีการนี้จะให้ผลการดีฟัซซิฟิเคชันจากการคำนวณศูนย์กลางแรงโน้มถ่วง (center of gravity) หรือ COG ของการกระจายความเป็นไปได้ แทนสัญญาณตั้งการควบคุม บางครั้งอาจเรียกเทคนิคการหาเซนทรอยด์ (centroid technique) ผลการดีฟัซซิฟิเคชันพิจารณาได้จากสมการที่ (2-9) โดยที่  $n$  เป็นจำนวนการแบ่งส่วนเอาต์พุต

$$x_0 = \frac{\sum_{j=1}^n \mu(x_j) \cdot x_j}{\sum_{j=1}^n \mu(x_j)} \quad (2-9)$$

ตัวอย่างการหาผลการดีฟัซซิฟิเคชันด้วยวิธีศูนย์กลางแรงโน้มถ่วง อธิบายได้ดังรูปที่ 2.13 ซึ่งกำหนดให้เอกภพสัมพัทธ์ของเอาต์พุตอยู่ในช่วง  $[0, 100]$  จะได้ว่า  $x_0 = x_3$  และสามารถแสดงการคำนวณได้ดังต่อไปนี้



รูปที่ 2.13 การดีฟัซซิฟิเคชันด้วยวิธีศูนย์กลางแรงโน้มถ่วง

$$x_3 = \frac{(0 + 10 + 20) \times 0.1 + (30 + 40 + 50 + 60) \times 0.2 + (70 + 80 + 90 + 100) \times 0.5}{0.1 + 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 + 0.5 + 0.5 + 0.5 + 0.5}$$

$$= 67.4$$

ในวิธีการดีฟัซซิฟิเคชันเหล่านี้ ได้มีผู้ทำการวิเคราะห์แล้วพบว่าวิธีหาค่าเฉลี่ยของค่าสูงสุด ให้ผลที่ดีต่อสมรรถนะของระบบ เมื่อเปรียบเทียบกับการใช้วิธีเกณฑ์สูงสุด และเมื่อเปรียบเทียบทั้งสามวิธีจะได้ว่า วิธีหาค่าเฉลี่ยของค่าสูงสุดส่งผลให้ระบบมีสมรรถนะชั่วคราวที่เหนือกว่า ส่วนวิธีแบบศูนย์กลางของพื้นที่ส่งผลที่ดีกว่าต่อสมรรถนะในสภาวะคงตัว

## 2.9 สรุป

ในบทนี้ได้กล่าวถึงระบบควบคุมแบบฟัซซี่ ซึ่งเหมาะสำหรับควบคุมระบบที่ไม่สามารถหาแบบจำลองทางคณิตศาสตร์ได้โดยง่าย โดยได้กล่าวถึงทฤษฎีฟัซซี่เซต เพื่อให้เห็นความแตกต่างระหว่างฟัซซี่เซตและเซตปกติธรรมดา รวมทั้งได้อธิบายโครงสร้างพื้นฐานของระบบควบคุมแบบฟัซซี่ กระบวนการออกแบบตัวควบคุมแบบฟัซซี่ การกำหนดกฎฟัซซี่ การอนุมานฟัซซี่ การกำหนดฟังก์ชันความเป็นสมาชิก เทคนิคการดีฟัซซิฟิเคชัน ทั้งนี้เพื่อมีความรู้ความเข้าใจเกี่ยวกับการควบคุมระบบด้วยฟัซซี่ และเป็นพื้นฐานในการออกแบบตัวควบคุมเพื่อสร้างเสถียรภาพให้ระบบ ลูกคัมผกผันแบบข้อต่อเดียวที่จะได้กล่าวถึงในบทที่ 3 ต่อไป



## บทที่ 3

### ตัวควบคุมฟuzzyแบบกฎอินพุตเดียว

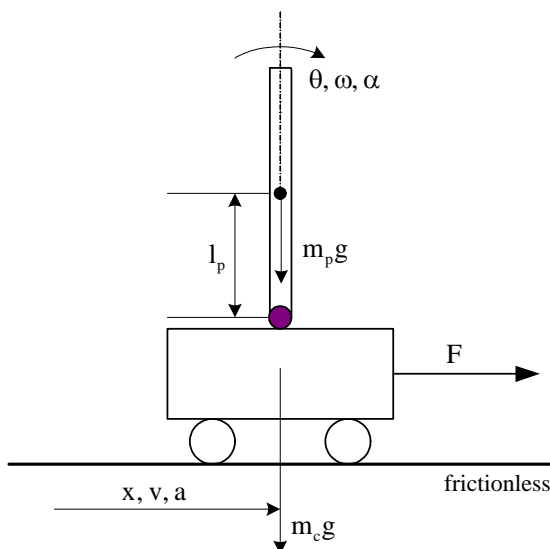
#### 3.1 บทนำ

การออกแบบตัวควบคุมฟuzzyแบบเดิม ดังได้กล่าวในบทที่ 2 พบว่าจะมีความยุ่งยากมากขึ้นเมื่อระบบที่ต้องการควบคุมมีความซับซ้อนเพิ่มขึ้น และเมื่อตัวแปรสถานะของระบบหรือตัวแปรอินพุตที่พิจารณามีจำนวนมาก ย่อมส่งผลให้จำนวนกฎฟuzzyเพิ่มขึ้นเป็นแบบเอ็กโพเนนเชียลตามจำนวนตัวแปรอินพุตที่เพิ่มขึ้นด้วยเช่นกัน (Yi and Yubazaki, 2000) ทั้งนี้เนื่องจากการออกแบบตัวควบคุมฟuzzyแบบเดิมนั้นจะนำตัวแปรอินพุตทุกตัวเหล่านี้มาจัดวางไว้ในส่วนเงื่อนไขของกฎควบคุมฟuzzy ทำให้เป็นการยากในการกำหนดกฎฟuzzyให้ได้อย่างเหมาะสม และต้องใช้เวลามากในการเอ็กซิกิวต์กฎฟuzzyเหล่านี้ แนวทางหนึ่งที่ช่วยลดปัญหาจำนวนกฎฟuzzyที่เพิ่มมากขึ้นได้รับการนำเสนอโดย Yi และ Yubazaki ซึ่งวิธีการนี้ในส่วนเงื่อนไขของกฎฟuzzyแต่ละกฎจะพิจารณาตัวแปรอินพุตเพียงตัวเดียวเท่านั้น ช่วยให้จำนวนกฎฟuzzyลดลง และง่ายต่อการกำหนดกฎฟuzzy โดยมีค่าระดับความสำคัญพลวัตเป็นตัวบ่งชี้ความสำคัญของอินพุตแต่ละตัว เรียกวิธีการนี้ว่า การควบคุมฟuzzyแบบกฎอินพุตเดียว ซึ่งรายละเอียดจะได้กล่าวต่อไปในบทนี้ วิธีการดังกล่าวได้รับการนำไปประยุกต์เพื่อควบคุมระบบต่างๆ เช่น การสร้างเสถียรภาพให้ระบบลูกคัมผกผันแบบข้อต่อเดียวและแบบสองข้อต่อ เป็นต้น

เนื้อหาในบทนี้กล่าวถึง การออกแบบตัวควบคุมฟuzzyแบบกฎอินพุตเดียว เพื่อสร้างเสถียรภาพให้ระบบลูกคัมผกผันแบบข้อต่อเดียว ซึ่งเป็นตัวอย่างของระบบที่มีความไม่เป็นเชิงเส้น ขาดเสถียรภาพ และควบคุมได้ยาก ลำดับแรกได้กล่าวถึงแบบจำลองทางคณิตศาสตร์ของระบบลูกคัมผกผันแบบข้อต่อเดียวที่ใช้ในการจำลองสถานการณ์ จากนั้นเป็นหลักการทำงานของตัวควบคุมฟuzzyแบบกฎอินพุตเดียว การออกแบบตัวควบคุมดังกล่าวเพื่อสร้างเสถียรภาพให้ระบบ รวมทั้งดำเนินการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกคัมผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนสถานการณ์ที่ควบคุม โดยเปรียบเทียบกับผลการจำลองสถานการณ์ที่ได้จากการสืบค้นข้อมูล เพื่อตรวจสอบความถูกต้องของโปรแกรม MATLAB ที่พัฒนาขึ้น นอกจากนี้ได้ทำการปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิก และเพิ่มจำนวนการแบ่งส่วนตัวแปรฟuzzy เพื่อศึกษาถึงผลกระทบต่อการสร้างเสถียรภาพให้ระบบดังกล่าว โดยดำเนินการเปรียบเทียบผลการจำลองสถานการณ์ที่ได้กับผลเดิมที่ออกแบบไว้ก่อนหน้านี้แล้ว

### 3.2 ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

ระบบลูกตุ้มผกผันแบบข้อต่อเดียวที่พิจารณาแสดงได้ด้วยแผนภาพในรูปที่ 3.1 ระบบดังกล่าวประกอบด้วยรางเลื่อน ตัวรถ ก้านลูกตุ้มผกผัน และแรงที่ใช้ขับเคลื่อน โดยที่ตัวรถสามารถเคลื่อนที่ไปทางด้านซ้ายหรือด้านขวาบนรางเลื่อนอย่างอิสระตามทิศทางแรงกระทำ ก้านลูกตุ้มถูกแขวนติดตรงส่วนกลางของตัวรถ สามารถหมุนได้อย่างอิสระตามทิศทางการเคลื่อนที่ของตัวรถ กำหนดให้ไม่มีแรงเสียดทานเกิดขึ้นกับระบบในขณะที่ทำการควบคุม ในการหาแบบจำลองทางคณิตศาสตร์ของลูกตุ้มผกผันแบบข้อต่อเดียวจะใช้สมการของลากรางจ์ (Lagrange equation) ช่วยวิเคราะห์ (ดังแสดงรายละเอียดในภาคผนวก ก.) ซึ่งแบบจำลองทางคณิตศาสตร์ของระบบดังกล่าวแสดงได้ด้วยสมการที่ (3-1) และ (3-2) ดังนี้คือ



รูปที่ 3.1 ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

$$\alpha = \frac{(m_c + m_p)g \sin\theta - \{F + m_p l_p \omega^2 \sin\theta\} \cos\theta}{\left\{ \frac{4}{3}(m_c + m_p) - m_p (\cos\theta)^2 \right\} l_p} \quad (3-1)$$

$$a = \frac{\frac{4}{3}\{F + m_p l_p \omega^2 \sin\theta\} - m_p g \sin\theta \cos\theta}{\left\{ \frac{4}{3}(m_c + m_p) - m_p (\cos\theta)^2 \right\}} \quad (3-2)$$

โดยที่ ค่าพารามิเตอร์ของระบบลูกตุ้มผกผันแบบข้อต่อเดียว มีดังต่อไปนี้

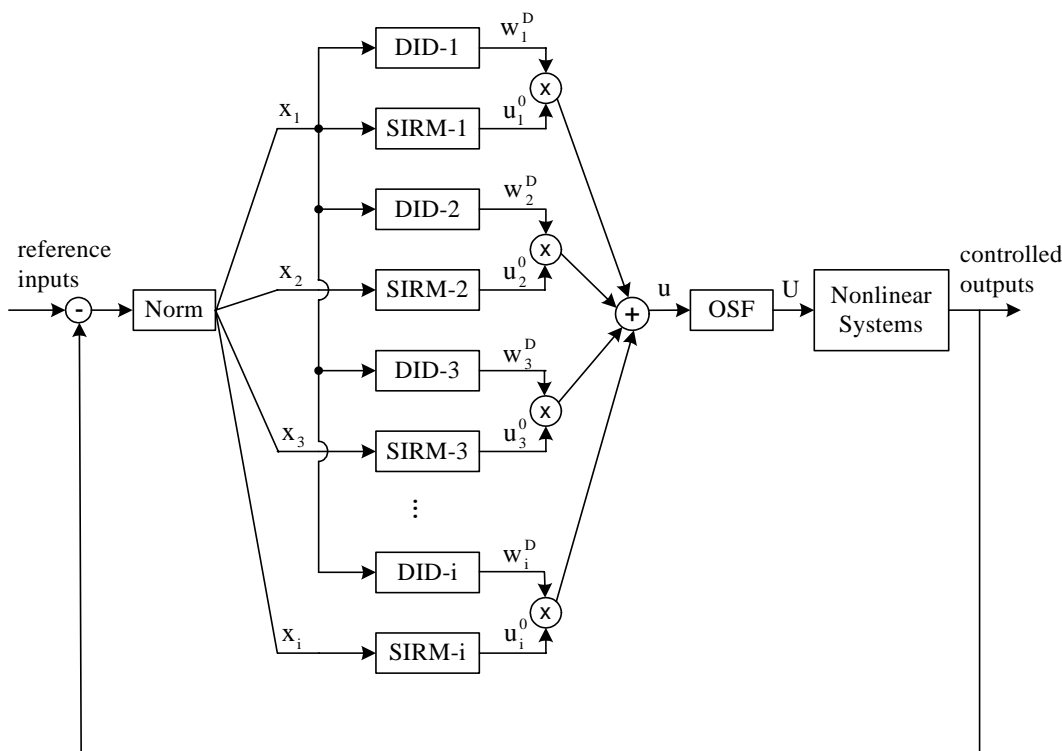
- F คือ แรงขับเคลื่อนในแนวระดับที่กระทำต่อตัวรถ (N)
- $m_c$  คือ มวลของตัวรถ (kg)
- $m_p$  คือ มวลของก้านลูกตุ้มผกผัน (kg)
- g คือ ค่าแรงโน้มถ่วงของโลกเท่ากับ  $9.8 \text{ (m/sec}^2\text{)}$
- $l_p$  คือ ความยาวของก้านลูกตุ้มจากตรงกลางถึงจุดต่อของตัวรถหรือความยาวครึ่งหนึ่งของความยาวก้านทั้งหมด (m)
- $\theta$  คือ มุมของลูกตุ้มผกผันพิจารณาจากตำแหน่งสมมูล (rad)
- $\omega$  คือ ความเร็วเชิงมุมของลูกตุ้มผกผัน (rad/sec)
- $\alpha$  คือ ความเร่งเชิงมุมของลูกตุ้มผกผัน (rad/sec<sup>2</sup>)
- x คือ ตำแหน่งตัวรถพิจารณาจากจุดกึ่งกลางบนรางเลื่อน (m)
- v คือ ความเร็วของตัวรถ (m/sec)
- a คือ ความเร่งของตัวรถ (m/sec<sup>2</sup>)

โดยธรรมชาติระบบลูกตุ้มผกผันเป็นระบบที่มีความไม่เป็นเชิงเส้น และขาดเสถียรภาพ กล่าวคือ หากไม่มีการควบคุมที่เหมาะสมแล้วก้านลูกตุ้มจะตกลงมาด้านล่างตามแรงโน้มถ่วงของโลก ไม่สามารถตั้งตรงที่ตำแหน่งสมมูลได้ รวมทั้งตัวรถจะเคลื่อนที่ออกไปไม่อยู่ในตำแหน่งที่ต้องการ ดังนั้นในงานวิจัยวิทยานิพนธ์นี้จึงนำระบบลูกตุ้มผกผันแบบข้อต่อเดียวมาเป็นตัวอย่างในการทดสอบตัวควบคุมแบบต่างๆ โดยกำหนดให้มวลของตัวรถเป็น 1.0 กิโลกรัม มวลของก้านลูกตุ้มเป็น 0.1 กิโลกรัม และก้านลูกตุ้มยาว 1.0 เมตร นอกจากนี้ทิศทางการหมุนของก้านลูกตุ้มและการเคลื่อนที่ของตัวรถสำหรับใช้ในการจำลองสถานการณ์ได้กำหนดให้ มุมของลูกตุ้มมีค่าเป็นบวกเมื่อก้านลูกตุ้มหมุนในทิศตามเข็มนาฬิกา และเป็นลบเมื่อก้านลูกตุ้มหมุนในทิศทวนเข็มนาฬิกา ในส่วนการเคลื่อนที่ของรถจะเป็นไปตามแนวแรงที่กระทำกับตัวรถ กล่าวคือ ในขณะที่ตัวรถอยู่ที่ตำแหน่งกึ่งกลางของรางเลื่อน ถ้าแรงกระทำมีทิศเป็นบวกจะทำให้ตัวรถเคลื่อนไปทางด้านขวา ถ้าแรงกระทำมีทิศเป็นลบก็จะทำให้รถเคลื่อนที่ไปทางด้านซ้ายของรางเลื่อน ซึ่งในงานวิจัยนี้จะทำการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวโดยพิจารณาควบคุมทั้งมุมลูกตุ้มและตำแหน่งรถไปพร้อมๆ กัน เพื่อให้ก้านลูกตุ้มสามารถตั้งตรงที่ศูนย์องศา และตัวรถหยุดตรงตำแหน่งกึ่งกลางของรางเลื่อน

### 3.3 หลักการของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว

เนื่องจากรูปแบบการอนุมานฟัซซี่แบบดั้งเดิม (conventional fuzzy inference) จะนำตัวแปรอินพุตทุกตัวมาจัดวางไว้ในส่วนเงื่อนไขของกฎฟัซซี่ ทำให้จำนวนกฎฟัซซี่เพิ่มขึ้น และเป็นการยากต่อการกำหนดกฎฟัซซี่ให้ได้อย่างเหมาะสม เพื่อแก้ปัญหาเหล่านี้ได้มีการนำมอดูลฟัซซี่

แบบกฎอินพุตเดี่ยว (single input rule module: SIRM) มาใช้ในการควบคุมระบบที่ประกอบด้วย  $n$  อินพุต และหนึ่งเอาต์พุต (Yi and Yubazaki, 2000) ซึ่งในส่วนเงื่อนไขของกฎฟัซซี่แต่ละกฎมีตัวแปรอินพุตเพียงตัวเดียวเท่านั้น โดยมีค่าระดับความสำคัญพลวัต (dynamic important degree: DID) เป็นตัวบ่งชี้ความสำคัญของอินพุตแต่ละตัว โครงสร้างของตัวควบคุมฟัซซี่แบบกฎอินพุตเดี่ยวสำหรับควบคุมระบบที่มีความไม่เป็นเชิงเส้นแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างของตัวควบคุมฟัซซี่แบบกฎอินพุตเดี่ยว

เมื่อพิจารณารูปที่ 3.2 จะสังเกตเห็นว่า สัญญาณความคลาดเคลื่อนจากการเปรียบเทียบระหว่างสัญญาณอินพุตอ้างอิง (reference inputs) และสัญญาณเอาต์พุตควบคุม (controlled outputs) จะเข้าสู่บล็อกนอร์ม (norm block) เพื่อทำการนอร์มัลไลซ์สัญญาณให้อยู่ในช่วงเอกภพสัมพัทธ์ (universe of discourse) ที่กำหนด จากนั้นตัวแปรอินพุตแต่ละตัวที่ผ่านการนอร์มัลไลซ์แล้วจะเป็นอินพุตให้กับ SIRM ซึ่งการกำหนดกฎฟัซซี่แต่ละกฎจะประกอบด้วยตัวแปรอินพุตเพียงตัวเดียวเท่านั้น รูปแบบการกำหนดกฎฟัซซี่แบบกฎอินพุตเดี่ยวแสดงดังสมการที่ (3-3)

$$\text{SIRM} - i : \{R_i^j : \text{if } x_i = A_i^j \text{ then } u_i = C_i^j\}_{j=1}^{m_i} \tag{3-3}$$

โดยที่ SIRM-i คือ มอดูลฟัซซี่แบบกฎอินพุตเดียวของอินพุตตัวที่ i

$R_i^j$  คือ กฎที่ j ใน SIRM-i

$x_i$  คือ ตัวแปรอินพุตตัวที่ i ในส่วนเงื่อนไขของกฎฟัซซี่

$u_i$  คือ ตัวแปรเอาต์พุตใน SIRM-i

$A_i^j$  คือ ฟังก์ชันความเป็นสมาชิกของ  $x_i$  ในกฎที่ j ของ SIRM-i

$C_i^j$  คือ ฟังก์ชันความเป็นสมาชิกของ  $u_i$  ในกฎที่ j ของ SIRM-i

$i = 1, 2, \dots, n$  คือ จำนวนตัวแปรอินพุตที่พิจารณา

$j = 1, 2, \dots, m_i$  คือ จำนวนกฎใน SIRM-i

ในส่วนการอนุมานฟัซซี่แบบกฎอินพุตเดียวของ SIRM สำหรับอินพุตแต่ละตัว จนกระทั่งได้ผลการอนุมานเป็น  $u_i^0$  จะใช้วิธี simplified fuzzy reasoning method (Mizumoto, 1996) ซึ่งจะได้กล่าวถึงรายละเอียดในหัวข้อ 3.5 โดยที่ฟังก์ชันความเป็นสมาชิกในส่วนผลลัพธ์ของกฎฟัซซี่กำหนดให้เป็นค่าคงที่ (singleton) ดังนั้นสามารถคำนวณผลการอนุมานกฎฟัซซี่ของ SIRM ได้ด้วยสมการที่ (3-4)

$$u_i^0 = \frac{\sum_{j=1}^{m_i} A_i^j(x_i) C_i^j}{\sum_{j=1}^{m_i} A_i^j(x_i)} \quad (3-4)$$

ส่วนประกอบที่สำคัญอีกประการหนึ่งของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว คือ ค่าระดับความสำคัญพลวัต (DID) ซึ่งค่านี้จะเป็นตัวบ่งชี้ความสำคัญของอินพุตแต่ละตัว จากรูปที่ 3.2 พบว่ามีเพียง  $x_i$  เท่านั้นที่ใช้เป็นตัวแปรอินพุตสำหรับกำหนดกฎฟัซซี่ เพื่อหาค่าระดับความสำคัญพลวัตให้กับอินพุตทุกตัว ในการหาค่าระดับความสำคัญพลวัตสามารถคำนวณได้จากความสัมพันธ์ตามสมการที่ (3-5)

$$w_i^D = w_i + B_i \cdot \Delta w_i^0 \quad (3-5)$$

โดยที่  $w_i^D$  คือ ค่าระดับความสำคัญพลวัตของตัวแปรอินพุต i

$w_i$  คือ ค่าคงที่ เรียกว่าค่าฐาน (base value)

$B_i$  คือ ค่าคงที่ เรียกว่าค่าขยาย (breadth value)

$\Delta w_i^0$  คือ ผลการอนุมานกฎฟัซซี่ที่ได้จากตัวแปรพลวัต  $\Delta w_i$

จากสมการที่ (3-5) พจน์แรก คือ ค่าฐาน พจน์หลัง คือ ค่าพลวัต (dynamic value) ซึ่งเกิดจากผลคูณระหว่างค่าขยายกับผลการอนุมานฟัซซี่ของตัวแปรพลวัต โดยค่าฐานและค่าขยายของตัวแปรอินพุตแต่ละตัวจะกำหนดเป็นค่าพารามิเตอร์ควบคุม (control parameters) ซึ่งสามารถหาค่าพารามิเตอร์ควบคุมเหล่านี้ได้จากวิธีการสุ่มค่าลองผิดลองถูก (Yi and Yubazaki, 2000) หรือวิธีค้นหาแบบสุ่มค่าที่เหมาะสมที่สุด (random optimization search) (Yi, Yubazaki, and Hirota, 2001) ส่วนตัวแปรพลวัตของอินพุตแต่ละตัวได้จากการกำหนดกฎฟัซซี่แบบกฎอินพุตเดียว

หลังจากที่ได้ค่า  $u_i^0$  จาก SIRM และค่าระดับความสำคัญพลวัต  $w_i^D$  สำหรับอินพุตแต่ละตัวแล้ว เมื่อพิจารณาแผนภาพโครงสร้างของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวในรูปที่ 3.2 ประกอบจะเห็นว่า  $u$  ซึ่งเป็นสัญญาณควบคุม (control signal) ที่กระทำกับระบบไม่เป็นเชิงเส้น (nonlinear systems) สามารถหาได้จากผลรวมระหว่างผลคูณของผลการอนุมานกฎฟัซซี่  $u_i^0$  กับค่าระดับความสำคัญพลวัต  $w_i^D$  ดังนั้นสามารถเขียนแสดงความสัมพันธ์ได้ดังสมการที่ (3-6)

$$u = \sum_{i=1}^n w_i^D \cdot u_i^0 \quad (3-6)$$

โดยที่  $n$  คือ จำนวนตัวแปรอินพุตทั้งหมดที่พิจารณา แต่ก่อนที่จะทำการควบคุมระบบ จำเป็นต้องปรับค่าสัญญาณควบคุมนี้โดยการคูณด้วยตัวปรับค่าของเอาต์พุต (output scaling factor: OSF) ซึ่งแสดงในสมการที่ (3-7) เพื่อให้ได้สัญญาณควบคุมระบบที่มากพอ ดังนั้นสัญญาณควบคุมจริงที่กระทำกับระบบไม่เป็นเชิงเส้นคือ

$$U = u \cdot OSF \quad (3-7)$$

เมื่อให้สัญญาณควบคุมจริงกับระบบแล้วจะได้สัญญาณเอาต์พุตควบคุม เพื่อนำมาเปรียบเทียบกับสัญญาณอินพุตอ้างอิงอีกครั้ง ซึ่งโดยทั่วไปกำหนดให้อินพุตอ้างอิงมีค่าเป็นศูนย์ ทั้งนี้เนื่องจากต้องการควบคุมให้ตัวแปรสถานะของระบบพลวัตมีค่าเป็นศูนย์ เมื่อสถานการณ์ที่ทำการควบคุมเปลี่ยนแปลงไป ค่าระดับความสำคัญพลวัตต้องแปรเปลี่ยนไปด้วย เพื่อให้ตัวควบคุมยังคงสามารถควบคุมระบบต่อไปได้ ในหัวข้อต่อไปจะเป็นการออกแบบตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว เพื่อให้การควบคุมระบบลูกตุ้มผกผันแบบข้อต่อเดียว ซึ่งมีโครงสร้างตามรูปที่ 3.1 ทั้งนี้เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว กล่าวคือ ตัวควบคุมที่ได้รับการออกแบบต้องสามารถควบคุมให้ก้านลูกตุ้มตั้งตรงที่ตำแหน่งศูนย์องศา พร้อมทั้งควบคุมให้ตัวรถเคลื่อนที่กลับมายังตำแหน่งกึ่งกลางของรางลื่น

### 3.4 การออกแบบตัวควบคุมฟuzzyแบบกฎอินพุตเดียว

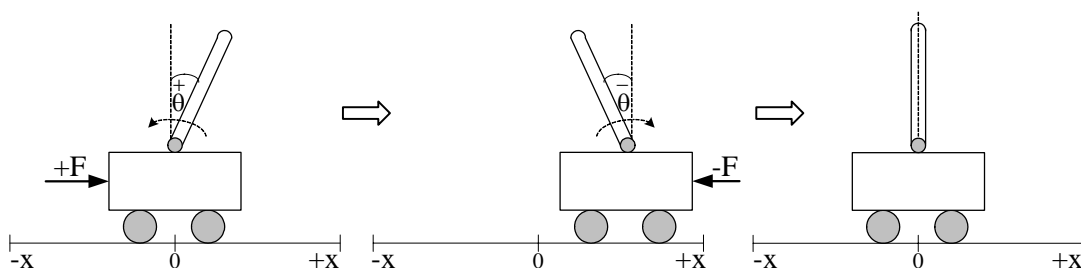
การออกแบบตัวควบคุมฟuzzyแบบกฎอินพุตเดียว เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวในงานวิจัยวิทยานิพนธ์นี้ พิจารณาควบคุมทั้งตำแหน่งมุมของลูกตุ้มผกผันและตำแหน่งของตัวรถ กล่าวคือ ต้องการควบคุมให้ก้านลูกตุ้มผกผันสามารถตั้งตรงที่ตำแหน่งสมดุลหรือที่ศูนย์องศา โดยตัวรถต้องสามารถเคลื่อนที่กลับมายังตำแหน่งเริ่มต้นที่กำหนด นั่นคือตำแหน่งกึ่งกลางของรางลื่น ทั้งนี้จะพิจารณาเวลาในการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียวเป็นปัจจัยบังคับสมรรถนะด้วย ตัวแปรสถานะของระบบที่พิจารณา ได้แก่ มุมของลูกตุ้มผกผัน ( $\theta$ ) ความเร็วเชิงมุมของลูกตุ้ม ( $\omega$ ) ตำแหน่งของรถ ( $x$ ) และความเร็วของรถ ( $v$ ) ในการออกแบบตัวควบคุมต้องการให้ตัวแปรสถานะทุกตัวมีค่าเข้าสู่หาศูนย์ภายในช่วงเวลาหนึ่ง ดังนั้นหลังจากที่ตัวแปรสถานะเหล่านี้ผ่านการนอร์มัลไลซ์ด้วยตัวปรับค่า (scaling factor) ที่กำหนดขึ้นสำหรับตัวแปรสถานะแต่ละตัวแล้ว จะได้ตัวแปรอินพุต  $x_i$  ( $i = 1, 2, 3, 4$ ) ซึ่งแทนด้วยมุมลูกตุ้ม ความเร็วเชิงมุมของลูกตุ้ม ตำแหน่งรถ และความเร็วของตัวรถ ตามลำดับ การกำหนดกฎฟuzzyแบบอินพุตเดียวในแต่ละกฎจะใช้ตัวแปรอินพุตเพียงตัวเดียวเท่านั้น โดยที่ค่าระดับความสำคัญพลวัตสามารถปรับเปลี่ยนได้ตามสถานการณ์ที่ทำการควบคุม เพื่อสร้างแรงขับเคลื่อนที่ที่เหมาะสมต่อการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้

#### 3.4.1 การกำหนดกฎฟuzzyแบบอินพุตเดียว

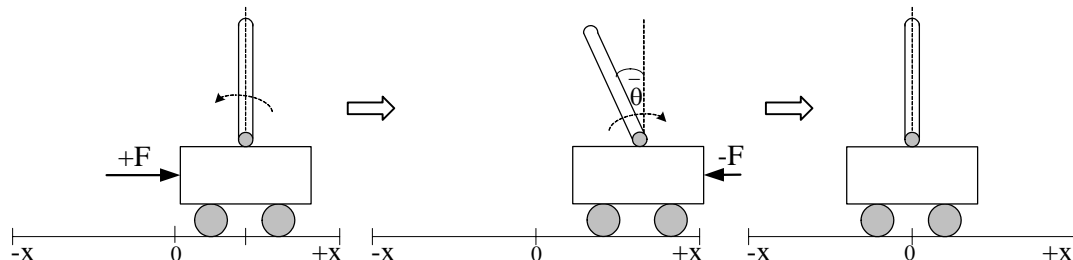
การกำหนดกฎฟuzzyแบบอินพุตเดียว หมายถึง การกำหนด SIRM สำหรับตัวแปรอินพุตแต่ละตัว เพื่อใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ในการดำเนินงานได้พิจารณาเป็นสองส่วน ได้แก่ การกำหนด SIRM สำหรับมุมและความเร็วเชิงมุมของลูกตุ้ม โดยอีกส่วนจะเป็นการกำหนด SIRM สำหรับตำแหน่งและความเร็วของรถ ทั้งนี้จะพิจารณาจากพฤติกรรมตามธรรมชาติของระบบลูกตุ้มผกผัน กล่าวคือ ในกรณีที่มุมและความเร็วเชิงมุมของลูกตุ้มมีค่าเป็นบวก แรงที่กระทำกับระบบก็ควรมีค่าเป็นบวกเพื่อให้ตัวรถเคลื่อนที่ไปทางด้านขวาของรางลื่น ส่งผลให้ก้านลูกตุ้มหมุนกลับในทิศทวนเข็มนาฬิกา มุมและความเร็วเชิงมุมจะมีค่าเป็นบวกน้อยลงจนกระทั่งเป็นศูนย์ ในกรณีที่มุมของลูกตุ้มหมุนเลยตำแหน่งสมดุลที่ศูนย์องศาจนมีค่าเป็นลบ จำเป็นต้องให้แรงกระทำกับระบบที่ทำให้ตัวรถเคลื่อนที่กลับมาทางด้านซ้ายของรางลื่น เพื่อให้ก้านลูกตุ้มหมุนกลับในทิศตามเข็มนาฬิกา จนกระทั่งก้านลูกตุ้มสามารถตั้งตรงอีกครั้งได้ รวมทั้งตัวรถสามารถเคลื่อนที่มายังจุดกึ่งกลางของรางลื่นได้ด้วยเช่นกัน ซึ่งแสดงได้ด้วยรูปที่ 3.3

ในการกำหนด SIRM สำหรับตำแหน่งของตัวรถ และความเร็วของตัวรถ สามารถพิจารณาได้จากรูปที่ 3.4 ซึ่งเป็นกรณีที่ตำแหน่งและความเร็วของรถมีค่าเป็นบวก สมมติให้ก้านลูกตุ้มตั้งตรงที่ตำแหน่งสมดุล ในกรณีนี้จะให้แรงบวกกระทำกับระบบเพื่อให้ตัวรถเคลื่อนที่ไปทางด้านขวาของรางลื่น ส่งผลให้ก้านลูกตุ้มเสียสมดุลหมุนในทิศทวนเข็มนาฬิกา ทำให้มุมลูกตุ้มมี

ค่าเป็นลบ หลังจากนั้นแรงที่กระทำกับระบบจะต้องทำให้ตัวรถเคลื่อนที่ไปทางด้านซ้ายเพื่อทำให้ก้านลูกตุ้มหมุนในทิศตามเข็มนาฬิกา เพื่อกลับมาตั้งตรงที่ตำแหน่งสมดุล และรถเคลื่อนมายังตำแหน่งกึ่งกลางของรางเส้น และเมื่อพิจารณาในกรณีที่ตำแหน่งและความเร็วของรถมีค่าเป็นลบ ก็สามารถวิเคราะห์ได้ในทำนองเดียวกันนี้ คือให้แรงกระทำกับระบบเพื่อทำให้ก้านลูกตุ้มเสียสมดุลก่อน จากนั้นจึงให้แรงกับระบบเพื่อให้ก้านลูกตุ้มกลับมาตั้งตรงและรถเคลื่อนที่กลับมาที่ตำแหน่งกึ่งกลางอีกครั้ง



รูปที่ 3.3 การวิเคราะห์เพื่อกำหนด SIRM สำหรับมุมและความเร็วเชิงมุมของลูกตุ้ม



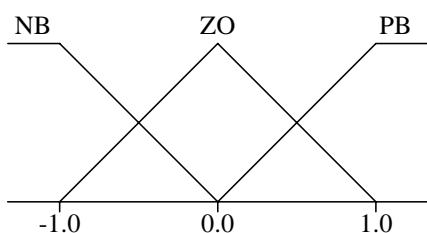
รูปที่ 3.4 การวิเคราะห์เพื่อกำหนด SIRM สำหรับตำแหน่งและความเร็วของรถ

จากการวิเคราะห์พฤติกรรมตามธรรมชาติของระบบลูกตุ้มผกผันแบบข้อต่อเดียวสามารถกำหนดกฎฟัซซี่สำหรับ SIRM ของตัวแปรอินพุตแต่ละตัวได้ดังตารางที่ 3.1 และฟังก์ชันความเป็นสมาชิกในส่วนเงื่อนไขของ SIRM แต่ละตัวแสดงดังรูปที่ 3.5 ซึ่งกำหนดเป็นรูปสามเหลี่ยม และรูปสี่เหลี่ยมคางหมู ประกอบไปด้วย 3 ฟัซซี่เซตคือ NB (negative big) ZO (zero) และ PB (positive big) โดยฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตในส่วนผลลัพธ์ของกฎฟัซซี่แต่ละกฎกำหนดเป็นค่าคงที่จำนวนจริง นั่นคือฟังก์ชันความเป็นสมาชิกมีรูปแบบเป็นจำนวนจริงเดียว ดังปรากฏในรูปที่ 3.6 ทั้งนี้เพื่อเป็นการง่ายต่อการอนุมานกฎฟัซซี่ด้วยวิธี simplified fuzzy reasoning method

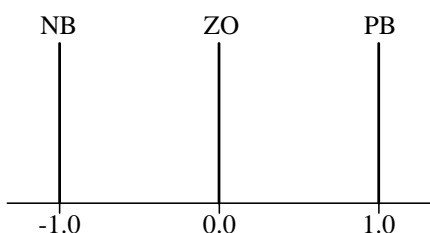


ตารางที่ 3.1 การกำหนดกฎฟัซซี่ใน SIRM

ตัวแปรในส่วนเงื่อนไข $x_i$ ( $i = 1, 2, 3, 4$ )	ตัวแปรในส่วนผลลัพธ์ $f_i$ ( $i = 1, 2, 3, 4$ )
NB	-1.0
ZO	0.0
PB	1.0



รูปที่ 3.5 ฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM



รูปที่ 3.6 ฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตใน SIRM

จากการกำหนดกฎฟัซซี่ตามตารางที่ 3.1 และการกำหนดฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตและตัวแปรเอาต์พุตดังแสดงในรูปที่ 3.5 และ 3.6 เมื่อนำมาเขียนกฎฟัซซี่แบบอินพุตเดียวตามสมการที่ (3-3) จะได้ทั้งสิ้นจำนวน 12 กฎ ดังต่อไปนี้

SIRM-1: กฎที่ 1 ถ้า  $x_1 = \text{NB}$  แล้ว  $f_1 = -1.0$

กฎที่ 2 ถ้า  $x_1 = \text{ZO}$  แล้ว  $f_1 = 0.0$

กฎที่ 3 ถ้า  $x_1 = \text{PB}$  แล้ว  $f_1 = 1.0$

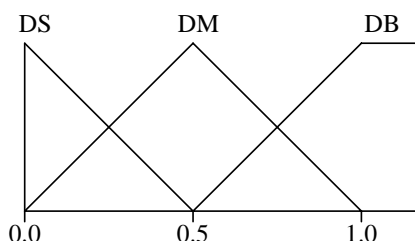
SIRM-2: กฎที่ 4 ถ้า  $x_2 = \text{NB}$  แล้ว  $f_2 = -1.0$

กฎที่ 5 ถ้า  $x_2 = \text{ZO}$  แล้ว  $f_2 = 0.0$

- กฎที่ 6 ถ้า  $x_2 = PB$  แล้ว  $f_2 = 1.0$   
 SIRM-3: กฎที่ 7 ถ้า  $x_3 = NB$  แล้ว  $f_3 = -1.0$   
 กฎที่ 8 ถ้า  $x_3 = ZO$  แล้ว  $f_3 = 0.0$   
 กฎที่ 9 ถ้า  $x_3 = PB$  แล้ว  $f_3 = 1.0$   
 SIRM-4: กฎที่ 10 ถ้า  $x_4 = NB$  แล้ว  $f_4 = -1.0$   
 กฎที่ 11 ถ้า  $x_4 = ZO$  แล้ว  $f_4 = 0.0$   
 กฎที่ 12 ถ้า  $x_4 = PB$  แล้ว  $f_4 = 1.0$

### 3.4.2 การกำหนดค่าระดับความสำคัญพลวัต

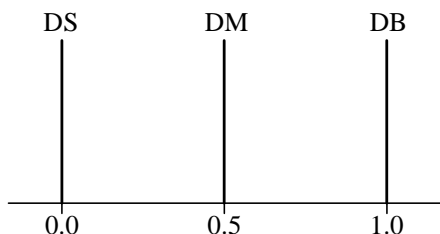
เมื่อพิจารณาสมการที่ (3-5) เพื่อหาค่าระดับความสำคัญพลวัตสำหรับบ่งชี้ความสำคัญของตัวแปรอินพุตแต่ละตัว พบว่าประกอบด้วยค่าตัวแปรพลวัต (dynamic variable) และพารามิเตอร์ควบคุม ได้แก่ ค่าฐาน และค่าขยาย ดังนั้นการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว จำเป็นต้องกำหนดค่าตัวแปรพลวัตและค่าพารามิเตอร์ควบคุมให้ได้อย่างเหมาะสมด้วยเช่นกัน ในการกำหนดค่าเหล่านี้พิจารณาได้จากพฤติกรรมของระบบลูกตุ้มผกผันแบบข้อต่อเดียว ซึ่งจะได้กล่าวถึงรายละเอียดเพื่อหาค่าระดับความสำคัญพลวัตดังต่อไปนี้



รูปที่ 3.7 ฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัตแต่ละตัว

การกำหนดค่าตัวแปรพลวัตให้กับค่าระดับความสำคัญพลวัตสำหรับตัวแปรอินพุตแต่ละตัว สามารถวิเคราะห์ได้จากแนวทางการควบคุมระบบลูกตุ้มผกผันแบบข้อต่อเดียว กล่าวคือในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวนั้น จะพิจารณาควบคุมมุมของลูกตุ้มผกผันเป็นอันดับแรก เมื่อมุมลูกตุ้มเกือบสมดุลแล้วจึงทำการควบคุมตำแหน่งของตัวรถเป็นลำดับต่อมา ในการกำหนดกฎฟัซซีของตัวแปรพลวัตทุกตัวจะพิจารณาขนาดมุมของลูกตุ้มผกผันเป็นตัวแปรอินพุตเพียงตัวเดียวในกฎฟัซซีทุกกฎ ซึ่งทำให้ฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัตแต่ละตัว มีขอบเขตของเอกภพสัมพัทธ์เฉพาะค่าบวกเท่านั้นดังแสดงในรูปที่ 3.7 ซึ่งประกอบด้วยรูปสามเหลี่ยมและรูปสี่เหลี่ยมคางหมู โดยมีฟัซซีเซตได้แก่ DS (degree small) DM (degree medium)

และ DB (degree big) ในส่วนเอาต์พุตของกฎฟัซซีแต่ละกฎจะมีขอบเขตของเอกภพสัมพัทธ์ เฉพาะค่าที่เป็นบวกเช่นกัน โดยอยู่ในช่วง  $[0, 1]$  ซึ่งฟังก์ชันความเป็นสมาชิกของเอาต์พุตแสดงได้ ดังรูปที่ 3.8



รูปที่ 3.8 ฟังก์ชันความเป็นสมาชิกของเอาต์พุต

เนื่องจากการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน จะทำการควบคุมมุมและความเร็วเชิงมุมของลูกตุ้มเป็นลำดับแรก จากนั้นจึงควบคุมตำแหน่งและความเร็วของรถเป็นลำดับต่อมา ดังนั้นการกำหนดกฎฟัซซีจึงพิจารณาเป็นสองส่วน เพื่อให้ได้ค่าระดับความสำคัญพลวัตที่เหมาะสมกับแต่ละสถานการณ์ นั่นคือส่วนของมุมและความเร็วเชิงมุมของลูกตุ้ม และอีกส่วนหนึ่งคือตำแหน่งและความเร็วของรถ ตัวอย่างเช่น ในกรณีที่ขนาดมุมลูกตุ้มมีค่ามาก ค่าระดับความสำคัญพลวัตของมุมและความเร็วเชิงมุมก็ควรจะมีค่ามากด้วย ขณะที่ค่าระดับความสำคัญพลวัตของตำแหน่งและความเร็วรถควรจะมีค่าน้อยๆ และเมื่อกำหนดลูกตุ้มเกือบตั้งตรงได้แล้ว ทำให้ขนาดมุมของลูกตุ้มเล็กลง ดังนั้นค่าระดับความสำคัญพลวัตของมุมและความเร็วเชิงมุมลูกตุ้มควรมีค่าน้อยลง ในขณะที่ค่าระดับความสำคัญพลวัตของตำแหน่งและความเร็วรถควรมีค่าเพิ่มขึ้น เพื่อให้ตัวรถเคลื่อนที่กลับมายังตำแหน่งที่กำหนดได้อีกครั้ง จากการวิเคราะห์ดังกล่าวสามารถกำหนดกฎฟัซซีสำหรับตัวแปรพลวัตของมุมและความเร็วเชิงมุมได้ดังตารางที่ 3.2 ในส่วนกฎฟัซซีสำหรับตำแหน่งและความเร็วรถกำหนดได้ดังตารางที่ 3.3

ตารางที่ 3.2 การกำหนดกฎฟัซซีสำหรับตัวแปรพลวัตของมุมและความเร็วเชิงมุม

ตัวแปรในส่วนเงื่อนไข	ตัวแปรในส่วนผลลัพธ์
$ x_1 $	$\Delta w_i \ (i = 1, 2)$
DS	0.0
DM	0.5
DB	1.0

ตารางที่ 3.3 การกำหนดกฎฟัซซี่สำหรับตัวแปรพลวัตของตำแหน่งและความเร็วรถ

ตัวแปรในส่วนเงื่อนไข	ตัวแปรในส่วนผลลัพธ์
$ x_1 $	$\Delta w_i$ ( $i = 3, 4$ )
DS	1.0
DM	0.5
DB	0.0

จากการกำหนดกฎฟัซซี่ตามตารางที่ 3.2 และ 3.3 ซึ่งพิจารณาตัวแปรอินพุตในส่วนเงื่อนไขของกฎฟัซซี่เพียงตัวเดียวคือ ขนาดมุมของลูกค้อน ( $|x_1|$ ) สามารถเขียนกฎฟัซซี่แบบกฎอินพุตเดียวได้ทั้งสิ้น 12 กฎดังต่อไปนี้

DID-1: กฎที่ 1 ถ้า  $|x_1| = DS$  แล้ว  $\Delta w_1 = 0.0$

กฎที่ 2 ถ้า  $|x_1| = DM$  แล้ว  $\Delta w_1 = 0.5$

กฎที่ 3 ถ้า  $|x_1| = DB$  แล้ว  $\Delta w_1 = 1.0$

DID-2: กฎที่ 4 ถ้า  $|x_1| = DS$  แล้ว  $\Delta w_2 = 0.0$

กฎที่ 5 ถ้า  $|x_1| = DM$  แล้ว  $\Delta w_2 = 0.5$

กฎที่ 6 ถ้า  $|x_1| = DB$  แล้ว  $\Delta w_2 = 1.0$

DID-3: กฎที่ 7 ถ้า  $|x_1| = DS$  แล้ว  $\Delta w_3 = 1.0$

กฎที่ 8 ถ้า  $|x_1| = DM$  แล้ว  $\Delta w_3 = 0.5$

กฎที่ 9 ถ้า  $|x_1| = DB$  แล้ว  $\Delta w_3 = 0.0$

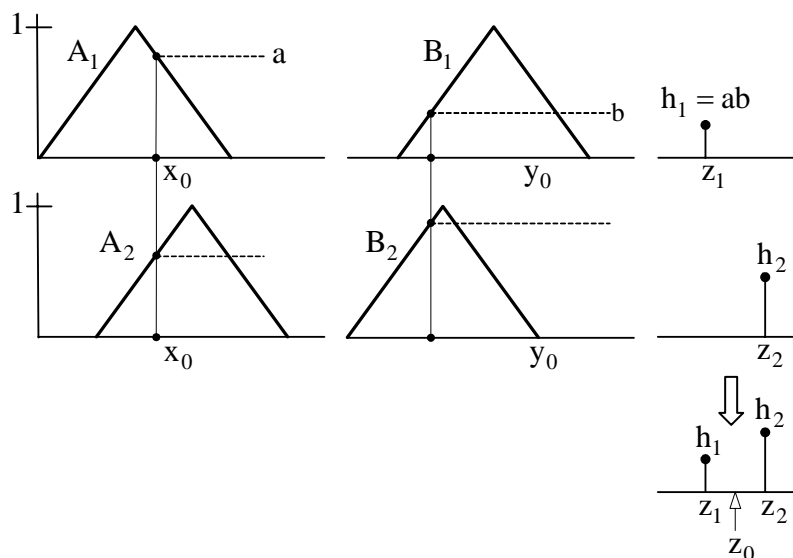
DID-4: กฎที่ 10 ถ้า  $|x_1| = DS$  แล้ว  $\Delta w_4 = 1.0$

กฎที่ 11 ถ้า  $|x_1| = DM$  แล้ว  $\Delta w_4 = 0.5$

กฎที่ 12 ถ้า  $|x_1| = DB$  แล้ว  $\Delta w_4 = 0.0$

เนื่องจากการกำหนดกฎฟัซซี่สำหรับตัวแปรพลวัตเพียงอย่างเดียวอาจไม่เพียงพอต่อการกำหนดลำดับการควบคุมของระบบลูกค้อนผกผันแบบข้อต่อเดียวได้ ดังนั้นจึงต้องมีการกำหนดค่าพารามิเตอร์ควบคุมเพื่อให้ได้ค่าระดับความสำคัญพลวัตที่สามารถครอบคลุมสำหรับทุกสถานการณ์ที่ทำการควบคุม โดยค่าพารามิเตอร์ควบคุมนี้เป็นค่าคงที่ ประกอบด้วยค่าฐาน และค่าขยาย ซึ่งตารางที่ 3.4 แสดงการกำหนดค่าฐานและค่าขยายสำหรับตัวแปรอินพุตแต่ละตัว เพื่อใช้หาค่าระดับความสำคัญพลวัตดังกล่าวตามสมการที่ (3-5) ซึ่งค่าเหล่านี้ได้มาจากการปรับค่าลองผิดลองถูก (Yi and Yubazaki, 2000)





รูปที่ 3.9 การอนุมานด้วยวิธี simplified fuzzy reasoning method

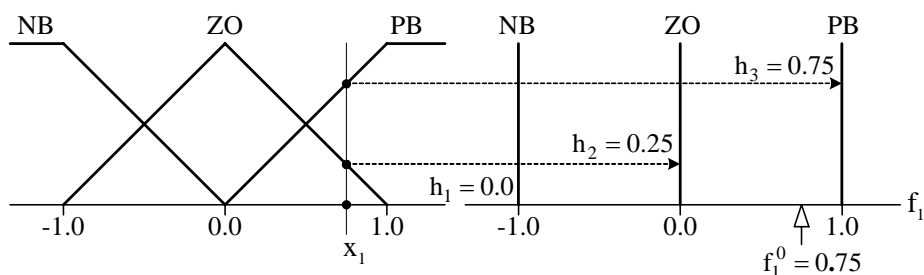
ดังนั้นเมื่อพิจารณาการอนุมานกฎฟัซซี่ที่แสดงในรูปที่ 3.9 ประกอบ จะได้ว่าในกรณีที่มีกฎฟัซซี่ทั้งหมดจำนวน n กฎ สามารถคำนวณเอาต์พุต  $z_0$  ได้จากสมการที่ (3-10)

$$z_0 = \frac{h_1 \cdot z_1 + h_2 \cdot z_2 + \dots + h_n \cdot z_n}{h_1 + h_2 + \dots + h_n} \tag{3-10}$$

เมื่อ  $h_1$  คือ ค่าระดับความเป็นสมาชิกที่ได้จากอินพุต  $x_0$  และ  $y_0$  สำหรับกฎที่ 1 ส่วน  $z_1$  คือ เอาต์พุตที่ได้จากกฎที่ 1 ซึ่งมีค่าเป็นจำนวนจริงเดี่ยว ส่วนตัวแปรอื่นๆ ที่ปรากฏในสมการที่ (3-10) ก็สามารถพิจารณาได้ในลักษณะเดียวกันนี้

เนื่องจากตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวได้กำหนดให้แต่ละกฎฟัซซี่ประกอบด้วยตัวแปรอินพุตเพียงตัวเดียวเท่านั้น ดังนั้นการหาค่าระดับความเป็นสมาชิก  $h_1$  ของเอาต์พุตในแต่ละกฎก็คือ ค่าระดับความเป็นสมาชิกของตัวแปรอินพุตในแต่ละกฎนั่นเอง ตัวอย่างต่อไปนี้เป็นการหาผล การอนุมานกฎฟัซซี่  $f_1^0$  เมื่อตัวแปรอินพุตที่พิจารณา คือ มุมของลูกตุ้มผกผันซึ่งผ่านการนอร์ มัลไลซ์ให้อยู่ในขอบเขตเอกภพสัมพัทธ์ที่กำหนด กฎฟัซซี่แบบกฎอินพุตเดียวสำหรับมุมของลูกตุ้ม ตามที่ได้กำหนดไว้แล้วคือ

- SIRM-1: กฎที่ 1 ถ้า  $x_1 = NB$  แล้ว  $f_1 = -1.0$
- กฎที่ 2 ถ้า  $x_1 = ZO$  แล้ว  $f_1 = 0.0$
- กฎที่ 3 ถ้า  $x_1 = PB$  แล้ว  $f_1 = 1.0$



รูปที่ 3.10 ตัวอย่างการอนุมานกฎฟัซซี่ด้วยวิธี simplified fuzzy reasoning method

จากรูปที่ 3.10 ในกรณีที่ตัวแปรอินพุต  $x_1$  อยู่ตรงตำแหน่งดังกล่าว จะเข้าเงื่อนไขในกฎที่ 2 และกฎที่ 3 เท่านั้น ดังนั้นค่าระดับความเป็นสมาชิกของฟัซซี่เซต NB ในกฎที่ 1 จึงมีค่าเป็นศูนย์ กล่าวคือ  $h_1 = 0.0$ ,  $h_2 = 0.25$  และ  $h_3 = 0.75$  เมื่อคำนวณผลการอนุมานกฎฟัซซี่ด้วยวิธี simplified fuzzy reasoning method จะได้ผลตามสมการที่ (3-11)

$$f_1^0 = \frac{h_1 \cdot f_{1,1} + h_2 \cdot f_{1,2} + h_3 \cdot f_{1,3}}{h_1 + h_2 + h_3} \tag{3-11}$$

$$f_1^0 = \frac{(0.0) \cdot (-1.0) + (0.25) \cdot (0.0) + (0.75) \cdot (1.0)}{0.0 + 0.25 + 0.75}$$

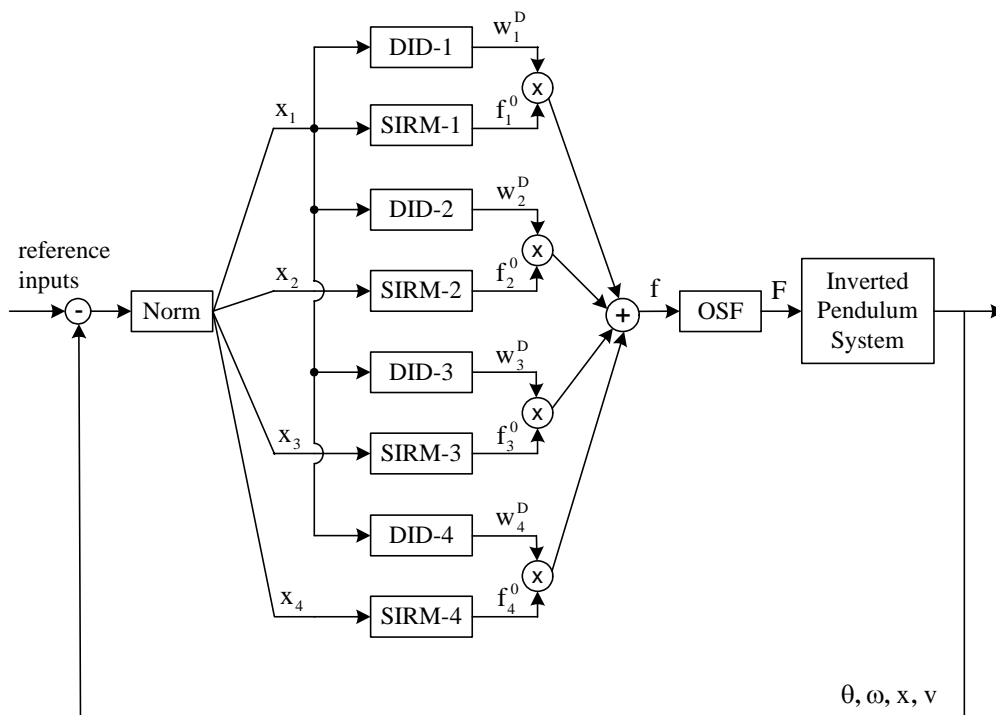
$$f_1^0 = 0.75$$

เมื่อพิจารณาสมการที่ (3-11) พบว่า การคำนวณหาผลการอนุมานฟัซซี่ด้วยวิธี simplified fuzzy reasoning method เป็นวิธีที่ง่าย ไม่ยุ่งยากซับซ้อน ในกรณีที่พิจารณาตัวแปรอินพุตอื่นๆ ของระบบลูกตุ้มผกผันแบบข้อต่อเดียวใน SIRM ก็สามารรถคำนวณหาผลการอนุมานฟัซซี่ได้ในลักษณะเดียวกันนี้ รวมทั้งการหาผลการอนุมานฟัซซี่ของตัวแปรพลวัต เพื่อหาค่าระดับความสำคัญพลวัตของตัวแปรอินพุตแต่ละตัวด้วย

### 3.6 การทำงานของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว

ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวเพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวมีโครงสร้างดังที่แสดงด้วยแผนภาพในรูปที่ 3.11 ตัวแปรสถานะ  $\theta$ ,  $\omega$ ,  $x$  และ  $v$  หมายถึง มุมของลูกตุ้ม ความเร็วเชิงมุมของลูกตุ้ม ตำแหน่งรถ และความเร็วยของรถ ตามลำดับ ตัวแปรสถานะทั้งสี่ตัวนี้จะถูกป้อนกลับเพื่อเปรียบเทียบกับค่าอินพุตอ้างอิง เนื่องจากการสร้างเสถียรภาพให้ระบบ

ลูกตุ้มผกผันแบบข้อต่อเดียว ต้องการให้ตัวแปรสถานะทุกตัวมีค่าเป็นศูนย์ ดังนั้นค่าอินพุตอ้างอิง  
 สำหรับใช้ในการเปรียบเทียบจึงกำหนดให้มีค่าเป็นศูนย์ทั้งหมด



รูปที่ 3.11 โครงสร้างตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวเพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน

ค่าความคลาดเคลื่อนของตัวแปรสถานะจากการเปรียบเทียบระหว่างเอาต์พุตกับอินพุต  
 อ้างอิงจะถูกป้อนเข้าสู่บล็อกนอร์ม เพื่อทำการปรับค่าด้วยตัวปรับค่าที่กำหนด ทำให้ตัวแปร  
 สถานะเหล่านี้มีค่าอยู่ในช่วงขอบเขตเอกภพสัมพัทธ์ที่ต้องการ หลังจากทำการนอร์มัลไลซ์แล้ว  
 จะได้ตัวแปรอินพุต  $x_1, x_2, x_3$  และ  $x_4$  จากตัวแปรสถานะ  $\theta, \omega, x$  และ  $v$  ตามลำดับ โดยตัว  
 ปรับค่าของมุมลูกตุ้ม ความเร็วเชิงมุมลูกตุ้ม ตำแหน่งรถ และความเร็วยาน กำหนดให้มีค่าเป็น  
 30 องศา 100 องศาต่อวินาที 2.4 เมตร และ 1.0 เมตรต่อวินาที ตามลำดับ จากรูปที่ 3.11 ตัวแปร  
 อินพุต  $x_i$  ( $i = 1, 2, 3, 4$ ) จะป้อนเข้าสู่บล็อก SIRM- $i$  ในส่วนของค่าระดับความสำคัญพลวัตใน  
 บล็อก DID- $i$  นั้นจะพิจารณาตัวแปรอินพุตเพียงตัวเดียวคือ ขนาดมุมของลูกตุ้ม จากนั้นทำการ  
 อนุมานกฎฟัซซี่ด้วยวิธี simplified fuzzy reasoning method จนกระทั่งได้เอาต์พุตในแต่ละ  
 บล็อก เมื่อพิจารณาจากแผนภาพ พบว่าเอาต์พุต  $f$  เกิดจากผลรวมของผลคูณระหว่างเอาต์พุตจาก  
 บล็อก SIRM- $i$  และเอาต์พุตจากบล็อก DID- $i$  ดังแสดงด้วยสมการที่ (3-12) แต่ก่อนจะให้เอาต์พุต  
 $f$  เป็นแรงขับเคลื่อนกระทำกับระบบ จะต้องทำการปรับค่าเอาต์พุตนี้ด้วยตัวปรับค่าทางด้านเอาต์พุต



หรือ OSF โดยที่ค่า OSF นี้กำหนดให้มีค่าเท่ากับสิบเท่าของผลรวมระหว่างมวลของตัวรถ ( $m_c$ ) และมวลของก้านลูกตุ้ม ( $m_p$ ) โดย  $F$  คือ แรงขับเคลื่อนจริงที่กระทำกับตัวรถในแนวระดับ ดังแสดงด้วยสมการที่ (3-13) และ (3-14)

$$f = \sum_{i=1}^4 w_i^D \cdot f_i^0 \quad (3-12)$$

$$F = f \cdot OSF \quad (3-13)$$

โดยที่

$$OSF = 10 \cdot (m_c + m_p) \quad (3-14)$$

### 3.7 การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

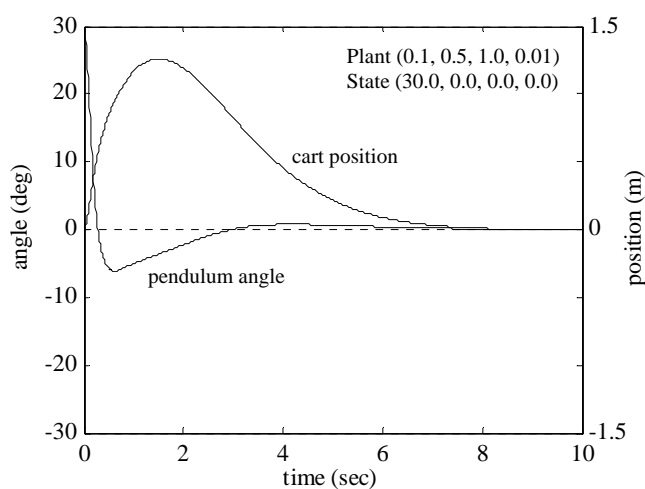
เพื่อทดสอบประสิทธิภาพของตัวควบคุมพีชชีแบบกฎอินพุตเดียว ได้ดำเนินการจำลองสถานการณ์เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยใช้แบบจำลองทางคณิตศาสตร์ของระบบดังที่ได้แสดงไว้ในสมการที่ (3-1) และ (3-2) ซึ่งค่าพารามิเตอร์ต่างๆ ของระบบลูกตุ้มผกผันแบบข้อต่อเดียวที่ปรากฏในสมการดังกล่าว กำหนดให้มวลของก้านลูกตุ้มผกผันเป็น 0.1 กิโลกรัม มวลของตัวรถเป็น 1.0 กิโลกรัม และก้านลูกตุ้มยาว 1.0 เมตร นั่นคือ ความยาวครึ่งหนึ่งของก้านลูกตุ้ม ( $l_p$ ) เป็น 0.5 เมตร ทั้งนี้กำหนดให้ความยาวของรางลื่นที่ตัวรถสามารถเคลื่อนที่ไปมาได้โดยอิสระอยู่ในช่วงขอบเขต  $[-2.4, +2.4]$  เมตร ถ้าตัวรถเคลื่อนที่เกินขอบเขตดังกล่าว จะพิจารณาว่าตัวควบคุมพีชชีแบบกฎอินพุตเดียวไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้

นอกจากนี้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวจะได้รับการพิจารณาว่าเข้าสู่เสถียรภาพอย่างสมบูรณ์แล้วก็ต่อเมื่อตัวแปรสถานะทุกตัวเป็นไปตามข้อกำหนดดังนี้คือ มุมลูกตุ้มเข้าสู่  $\pm 0.1$  องศา ความเร็วเชิงมุมของลูกตุ้มเข้าสู่  $\pm 0.1$  องศาต่อวินาที ตำแหน่งของตัวรถเข้าสู่  $\pm 0.01$  เมตร และความเร็วของตัวรถเข้าสู่  $\pm 0.01$  เมตรต่อวินาที และเพื่อตรวจสอบความถูกต้องของโปรแกรม MATLAB ที่พัฒนาขึ้น ได้ดำเนินการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนสถานการณ์ที่ควบคุม ดังปรากฏในตารางที่ 3.5 ทั้งนี้เพื่อพิจารณาเปรียบเทียบผลการจำลองสถานการณ์ที่ได้กับผลงานวิจัยอ้างอิง (Yi and Yubazaki, 2000)

ตารางที่ 3.5 สถานการณ์ที่กำหนดขึ้นเพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

สถานการณ์ที่	มุมเริ่มต้น ของลูกตุ้ม (องศา)	ตำแหน่งเริ่มต้น ของตัวรถ (เมตร)	ความยาวของ ก้านลูกตุ้ม (เมตร)	มวลของ ก้านลูกตุ้ม (กิโลกรัม)	ช่วงเวลา แซมปลิ่ง (วินาที)
1	30.0	0.0	1.0	0.1	0.01
2	0.0	2.0	1.0	0.1	0.01
3	30.0	0.0	2.2	0.1	0.01
4	30.0	0.0	0.2	0.1	0.01
5	30.0	0.0	0.2	0.1	0.001
6	30.0	0.0	1.0	0.5	0.01

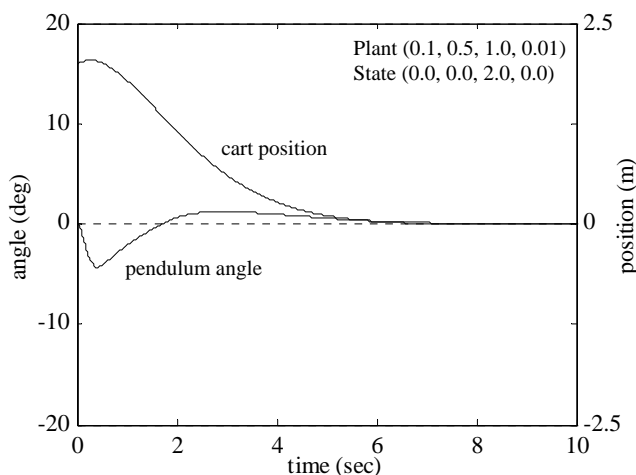
สถานการณ์แรก กำหนดให้มุมเริ่มต้นของลูกตุ้มผกผันเป็น 30 องศา นั่นคือก้านลูกตุ้มเอียงทำมุม 30 องศากับแนวตั้งฉากในทิศทางบวก โดยที่ตัวแปรสถานะอื่นๆ กำหนดให้มีค่าเป็นศูนย์ ซึ่งตำแหน่งเริ่มต้นของตัวรถอยู่ที่จุดกึ่งกลางของรางเลื่อน และก้านลูกตุ้มยาว 1.0 เมตร ผลการจำลองสถานการณ์แสดงในรูปที่ 3.12 โดยตัวเลขที่ปรากฏใน Plant (0.1, 0.5, 1.0, 0.01) ซึ่งแทรกเป็นส่วนหนึ่งของรูปนั้น หมายถึง มวลของก้านลูกตุ้ม ความยาวครึ่งหนึ่งของก้านลูกตุ้ม มวลของตัวรถ และช่วงเวลาแซมปลิ่ง ตามลำดับ และตัวเลขใน State (30.0, 0.0, 0.0, 0.0) ซึ่งแทรกเป็นส่วนหนึ่งของรูปเช่นกัน หมายถึง มุมเริ่มต้นของลูกตุ้ม ความเร็วเชิงมุมเริ่มต้นของลูกตุ้ม ตำแหน่งเริ่มต้นของตัวรถ และความเร็วเริ่มต้นของตัวรถ ตามลำดับ เมื่อพิจารณาผลการจำลองสถานการณ์จะสังเกตเห็นว่า



รูปที่ 3.12 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 1.0 เมตร

กรณีที่มุมเริ่มต้นของลูกตุ้มมีค่ามากในทิศทางบวก ตัวรถจะเคลื่อนที่ออกจากจุดกึ่งกลางของรางลื่นไปทางด้านขวา เพื่อให้ลูกตุ้มหมุนกลับในทิศทางเข็มนาฬิกา เป็นผลให้มุมของลูกตุ้มเป็นบวกลดลง จนกระทั่งลูกตุ้มหมุนมาอยู่ในตำแหน่งทิศทางลบ จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ทำให้ลูกตุ้มหมุนกลับในทิศทางเข็มนาฬิกา จนกระทั่งสามารถตั้งตรงที่ศูนย์องศา และตัวรถหยุดที่ตำแหน่งกึ่งกลางของรางลื่น ซึ่งใช้เวลาในการเข้าสู่เสถียรภาพทั้งสิ้น 8.24 วินาที

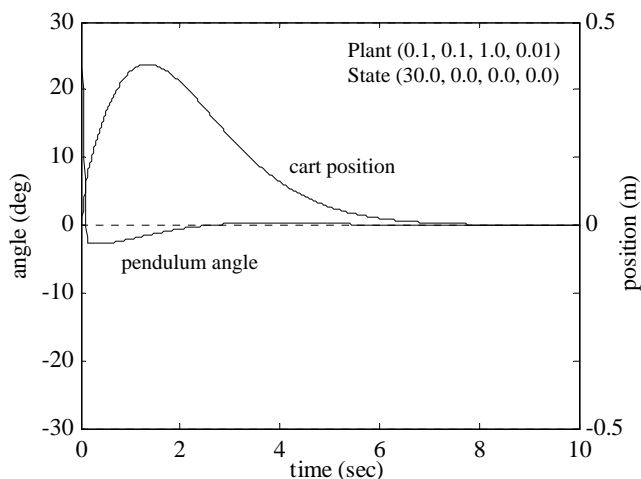
สถานการณ์ต่อมาเป็นการกำหนดให้ตำแหน่งเริ่มต้นของตัวรถอยู่ห่างจากจุดกึ่งกลางของรางลื่นไปทางด้านขวาเป็นระยะทาง 2.0 เมตร โดยลูกตุ้มตั้งตรงที่ศูนย์องศา และก้านลูกตุ้มยังคงยาว 1.0 เมตรเท่าเดิม ผลการจำลองสถานการณ์แสดงในรูปที่ 3.13 ซึ่งใช้เวลาในการสร้างเสถียรภาพประมาณ 7.19 วินาที เมื่อพิจารณาผลการจำลองสถานการณ์ที่ได้พบว่า ในขณะที่ตัวควบคุมดำเนินการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันนั้น ตัวรถจะเคลื่อนที่ไปทางด้านขวาเล็กน้อย เพื่อให้ก้านลูกตุ้มเสียดสมดุลแล้วหมุนในทิศทางเข็มนาฬิกา ทำให้มุมของลูกตุ้มมีค่าเป็นลบ จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ส่งผลให้ลูกตุ้มหมุนกลับในทิศทางเข็มนาฬิกา ทำให้มุมของลูกตุ้มมีค่าเป็นบวกเล็กน้อย จนกระทั่งลูกตุ้มสามารถกลับมาตั้งตรงที่ตำแหน่งสมดุลได้อีกครั้งพร้อมกันนั้นตัวรถก็เคลื่อนที่กลับมายังกึ่งกลางของรางลื่นด้วยเช่นกัน



รูปที่ 3.13 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร

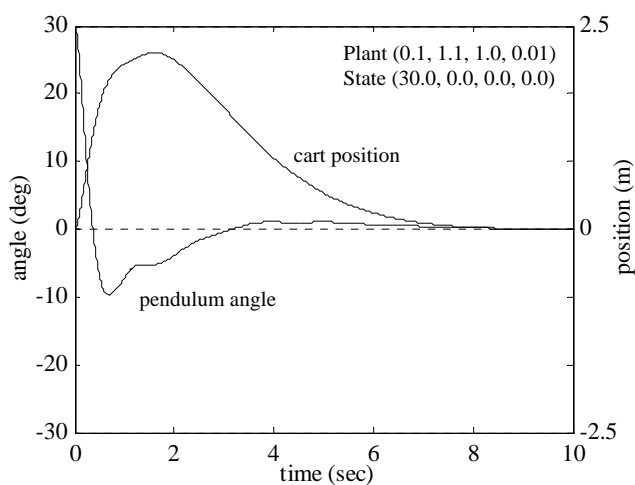
สถานการณ์ที่ 3 เป็นการจำลองสถานการณ์เช่นเดียวกันกับสถานการณ์แรก แต่ได้ทำการปรับเปลี่ยนความยาวของก้านลูกตุ้มให้สั้นลงเป็น 0.2 เมตร ผลการจำลองสถานการณ์แสดงได้ด้วยรูปที่ 3.14 ซึ่งจะสังเกตเห็นว่า ผลการจำลองสถานการณ์ที่ได้เป็นไปในแนวทางเดียวกันกับกรณีที่ก้านลูกตุ้มยาว 1.0 เมตร แต่ในกรณีที่ก้านลูกตุ้มยาว 0.2 เมตร ตัวรถจะเคลื่อนที่ออกห่างจากกึ่งกลาง

ของรางลื่นเป็นระยะทางที่น้อยกว่า จากนั้นจึงเคลื่อนที่กลับมาหยุดที่ตำแหน่งกึ่งกลางอีกครั้ง ซึ่งตัวควบคุมพีชซีแบบกฎอินพุตเดียวใช้เวลาในการสร้างเสถียรภาพให้กับระบบประมาณ 6.59 วินาที



รูปที่ 3.14 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 0.2 เมตร

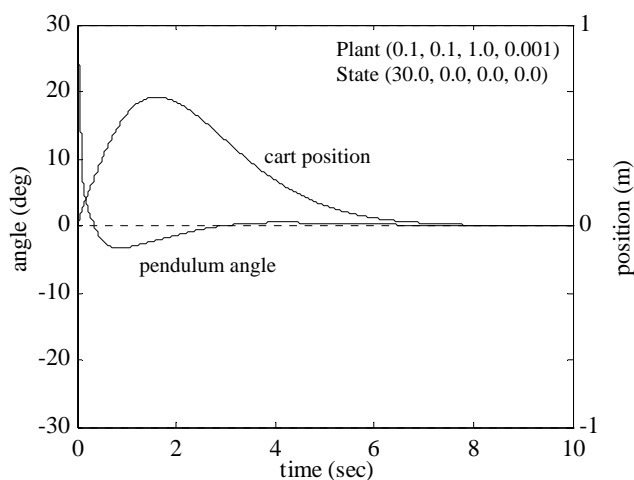
สถานการณ์ที่ 4 เป็นการจำลองสถานการณ์ เมื่อทำการปรับเพิ่มความยาวของก้านลูกตุ้มเป็น 2.2 เมตร โดยค่าต่างๆ ที่ใช้ในการจำลองสถานการณ์กำหนดให้มีค่าเช่นเดียวกับสถานการณ์แรก ผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบแสดงได้ดังรูปที่ 3.15 ซึ่งจะสังเกตเห็นว่า ผลการจำลองสถานการณ์ดังกล่าวยังคงเป็นไปในแนวทางเดียวกันกับสถานการณ์แรก ดังปรากฏในรูป



รูปที่ 3.15 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 2.2 เมตร

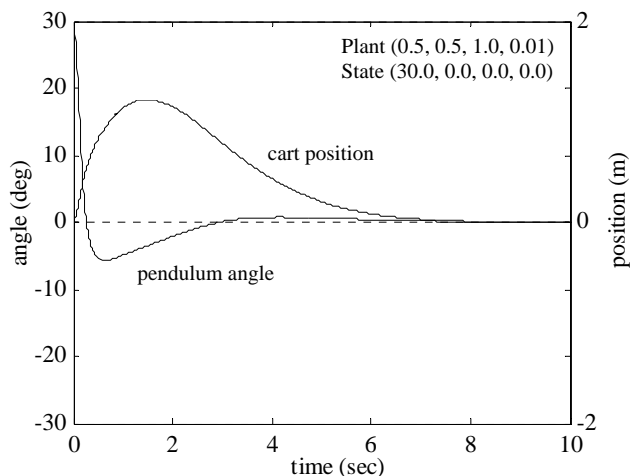
ที่ 3.12 แต่ตัวรถจะเคลื่อนที่ออกจากตำแหน่งกึ่งกลางของรางลื่นเป็นระยะทางที่มากกว่าจนเกือบถึงปลายสุดของรางลื่น ส่งผลให้ต้องใช้เวลาในการสร้างเสถียรภาพให้แกระบบมากกว่าด้วยเช่นกัน กล่าวคือใช้เวลาประมาณ 9.09 วินาที

สถานการณ์ที่ 5 เป็นการจำลองสถานการณ์เมื่อปรับเปลี่ยนช่วงเวลาแซมปลิงเป็น 0.001 วินาที โดยที่มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร และก้านลูกตุ้มยาว 0.2 เมตร ผลการจำลองสถานการณ์แสดงได้ด้วยรูปที่ 3.16 โดยใช้เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวทั้งสิ้นประมาณ 7.409 วินาที ซึ่งผลการจำลองสถานการณ์ที่ได้มีลักษณะคล้ายคลึงกับผลการจำลองสถานการณ์ที่ปรากฏในรูปที่ 3.14 ซึ่งเป็นกรณีที่ใช้ช่วงเวลาแซมปลิงเป็น 0.01 วินาที



รูปที่ 3.16 ผลตอบสนองเมื่อปรับเปลี่ยนช่วงเวลาแซมปลิงเป็น 0.001 วินาที

สถานการณ์สุดท้ายเป็นการจำลองสถานการณ์ เมื่อปรับเพิ่มมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม โดยที่มุมเริ่มต้นยังคงเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถอยู่ตรงกึ่งกลางของรางลื่น และก้านลูกตุ้มยาว 1.0 เมตร ผลการจำลองสถานการณ์แสดงได้ดังรูปที่ 3.17 ซึ่งผลการจำลองสถานการณ์ดังกล่าวใกล้เคียงกับผลการจำลองสถานการณ์ที่ปรากฏในรูปที่ 3.12 ซึ่งเป็นกรณีที่มวลของก้านลูกตุ้มเป็น 0.1 กิโลกรัม จากผลการจำลองสถานการณ์ที่ได้แสดงให้เห็นว่า แม้มวลของก้านลูกตุ้มจะเพิ่มขึ้นจากเดิมถึงห้าเท่า จะส่งผลต่อการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันเพียงเล็กน้อยเท่านั้น และเมื่อพิจารณาเวลาในการสร้างเสถียรภาพให้ระบบพบว่าใกล้เคียงกันด้วย กล่าวคือ เมื่อมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม ใช้เวลาประมาณ 8.17 วินาที ในขณะที่มวลของก้านลูกตุ้มเป็น 0.1 กิโลกรัม ใช้เวลาในการเข้าสู่เสถียรภาพทั้งสิ้น 8.24 วินาที



รูปที่ 3.17 ผลตอบสนองเมื่อปรับเพิ่มมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม

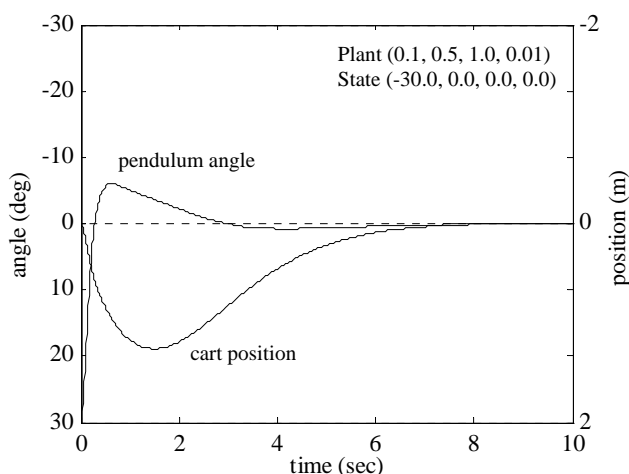
จากผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมพีซีซีแบบกฎอินพุตเดียวตามสถานการณ์ต่างๆ ที่กำหนดขึ้นทั้ง 6 สถานการณ์ เมื่อพิจารณาเปรียบเทียบผลการจำลองสถานการณ์ที่ได้และเวลาที่ใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวจากโปรแกรม MATLAB ที่พัฒนาขึ้นกับผลงานวิจัยอ้างอิง (Yi and Yubazaki, 2000) พบว่า มีความใกล้เคียงกันมากดังอาจพิจารณารายละเอียดได้จากตารางที่ 3.6 ซึ่งเป็นการเปรียบเทียบเวลาในการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันทั้ง 6 สถานการณ์ ดังนั้นโปรแกรมที่พัฒนาขึ้นจึงมีความถูกต้องและน่าเชื่อถือ สามารถนำไปใช้สำหรับการพัฒนาตัวควบคุมแบบใหม่ที่น่าเสนอในงานวิจัยวิทยานิพนธ์นี้ได้ต่อไป

ตารางที่ 3.6 ผลการเปรียบเทียบเวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียวจากโปรแกรมที่พัฒนาขึ้นและผลงานวิจัยอ้างอิง

สถานการณ์ที่	เวลาในการเข้าสู่เสถียรภาพ (วินาที)		ค่าคลาดเคลื่อน (วินาที)
	ผลงานวิจัยอ้างอิง	โปรแกรมที่พัฒนาขึ้น	
1	8.24	8.24	+0.00
2	7.16	7.19	-0.03
3	6.65	6.59	+0.06
4	9.04	9.09	-0.05
5	7.411	7.409	+0.002
6	8.18	8.17	+0.01

เนื่องจากการจำลองสถานการณ์ทั้ง 6 สถานการณ์ดังกล่าวข้างต้น พิจารณาเฉพาะกรณีที่มุมของลูกตุ้มและตำแหน่งเริ่มต้นของตัวรถมีค่าในทิศทางบวกเท่านั้น ดังนั้นเพื่อทดสอบประสิทธิภาพของตัวควบคุมพีชชีแบบกฎอินพุตเดียว จึงได้ดำเนินการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อพิจารณาให้มุมเริ่มต้นของลูกตุ้มอยู่ในทิศทางลบ และตำแหน่งเริ่มต้นของตัวรถอยู่ทางด้านซ้ายของรางเลื่อน นอกจากนี้ยังได้จำลองสถานการณ์เมื่อมุมเริ่มต้นของลูกตุ้มไม่ได้อยู่ที่ตำแหน่งสมดุลศูนย์องศา รวมทั้งตำแหน่งเริ่มต้นของตัวรถก็ไม่ได้อยู่ตรงกึ่งกลางของรางเลื่อน ผลการจำลองสถานการณ์ดังกล่าวอธิบายได้ดังนี้คือ

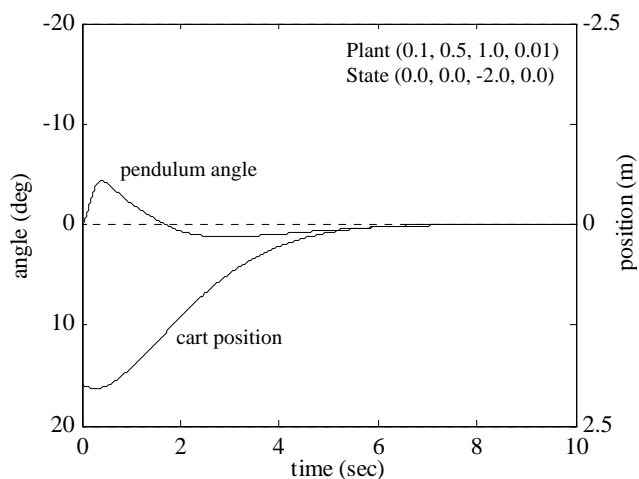
ผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวเมื่อมุมเริ่มต้นเป็น  $-30$  องศา และก้านลูกตุ้มยาว 1.0 เมตร โดยที่ตำแหน่งเริ่มต้นของตัวรถอยู่ตรงกึ่งกลางของรางเลื่อน แสดงได้ดังรูปที่ 3.18 จากผลการจำลองสถานการณ์ที่ได้พบว่า ตัวรถจะเคลื่อนที่ออกจากจุดเริ่มต้นไปทางด้านซ้าย เพื่อให้ลูกตุ้มหมุนในทิศตามเข็มนาฬิกา เป็นผลให้มุมของลูกตุ้มมีค่า



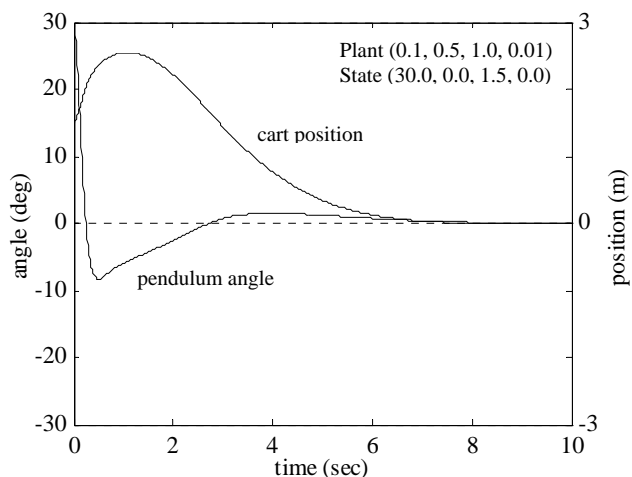
รูปที่ 3.18 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น  $-30$  องศา และก้านลูกตุ้มยาว 1.0 เมตร

เป็นบวก จากนั้นตัวรถจะเคลื่อนที่กลับมาทางด้านขวา ทำให้ลูกตุ้มหมุนกลับในทิศทวนเข็มนาฬิกา จนกระทั่งตัวรถเคลื่อนมาหยุดที่กึ่งกลางของรางเลื่อนและก้านลูกตุ้มตั้งตรงที่ศูนย์องศาได้อีกครั้ง ซึ่งใช้เวลาทั้งสิ้นประมาณ 8.24 วินาที ทั้งนี้เมื่อเปรียบเทียบผลการจำลองสถานการณ์ที่ได้กับผลการจำลองสถานการณ์ที่ 1 ดังปรากฏในรูปที่ 3.12 ซึ่งเป็นกรณีที่มุมเริ่มต้นของลูกตุ้มเป็นบวก 30 องศา พบว่าตัวควบคุมพีชชีแบบกฎอินพุตเดียวใช้เวลาในการสร้างเสถียรภาพให้ระบบเท่ากัน และจะเห็นได้ว่าพลวัตของระบบไม่แตกต่างกันเพียงแต่กระบวนการต่างๆ ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันเกิดขึ้นในทิศทางตรงข้ามกันเท่านั้น

เมื่อกำหนดให้ตำแหน่งเริ่มต้นของตัวรถอยู่ที่ตำแหน่ง -2.0 เมตร โดยที่มุมเริ่มต้นของลูกตุ้มเป็นศูนย์องศา และก้านลูกตุ้มยาว 1.0 เมตร ผลการจำลองสถานการณ์แสดงได้ดังรูปที่ 3.19 พบว่าระบบใช้เวลาในการเข้าสู่เสถียรภาพประมาณ 7.19 วินาที ซึ่งเท่ากับกรณีที่ตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร เพียงแต่พลวัตที่เกิดขึ้นมีทิศทางตรงข้ามกันเท่านั้น



รูปที่ 3.19 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น -2.0 เมตร



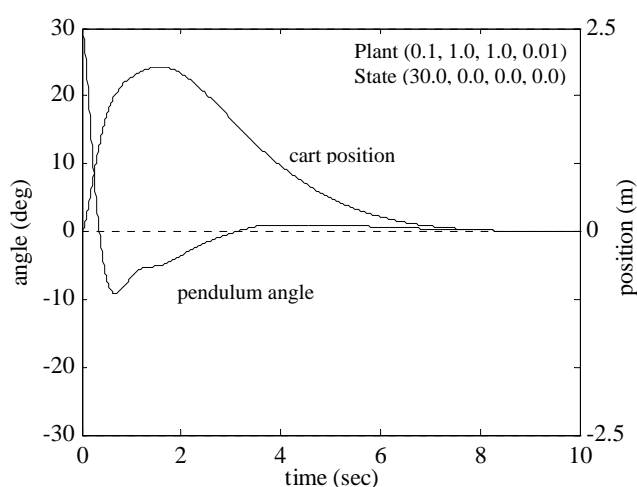
รูปที่ 3.20 ผลตอบสนองมุมเริ่มต้นเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร

การจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน เมื่อกำหนดให้ตำแหน่งเริ่มต้นของตัวรถอยู่ห่างจากจุดกึ่งกลางของรางเลื่อนไปทางด้านขวาเป็นระยะทาง 1.5 เมตร และมุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ผลการจำลองสถานการณ์ดังกล่าวแสดงในรูปที่ 3.20 ซึ่งตัวรถจะ



เคลื่อนที่ต่อไปทางด้านขวาเพื่อทำให้ลูกตุ้มหมุนในทิศทวนเข็มนาฬิกา จากนั้นตัวรถจึงเคลื่อนที่ กลับมาทางด้านซ้ายส่งผลให้ลูกตุ้มหมุนในทิศตามเข็มนาฬิกา จนกระทั่งตัวรถเคลื่อนที่มายัง ตำแหน่งกึ่งกลางของรางเลื่อน และก้านลูกตุ้มตั้งตรงที่ตำแหน่งสมดุล แต่เมื่อพิจารณาระยะทางที่ตัว รถเคลื่อนที่ พบว่าเป็นระยะทางประมาณ 2.55 เมตร ซึ่งเกินขอบเขตความยาวของรางเลื่อนที่กำหนดไว้ คือ  $[-2.4, +2.4]$  เมตร ดังนั้น ในกรณีนี้ตัวควบคุมพีชซีแบบกฎอินพุตเดียวไม่สามารถสร้าง เสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้

ส่วนผลการจำลองสถานการณ์ที่ปรากฏดังรูปที่ 3.21 กำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร โดยที่ก้านลูกตุ้มยาว 2.0 เมตร ทั้งนี้เพื่อเป็น สถานการณ์สำหรับใช้ทดสอบตัวควบคุมแบบต่างๆ ที่พัฒนาขึ้น ดังจะได้กล่าวถึงรายละเอียดในบท ต่อๆ ไป โดยตัวควบคุมพีชซีแบบกฎอินพุตเดียวสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบ ข้อต่อเดียวได้ภายในเวลา 8.95 วินาที เมื่อพิจารณาผลการจำลองสถานการณ์ที่ได้ พบว่า ใช้เวลาใน การเข้าสู่เสถียรภาพใกล้เคียงกับผลการจำลองสถานการณ์ในรูปที่ 3.15 ซึ่งเป็นกรณีที่ก้านลูกตุ้ม ยาว 2.2 เมตร



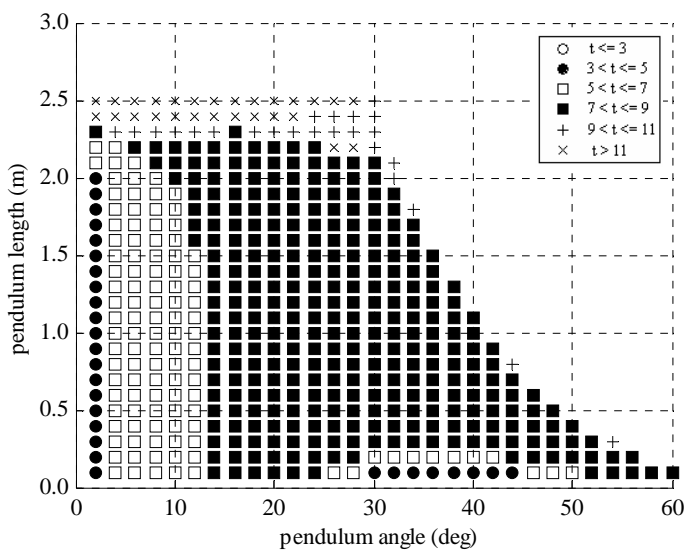
รูปที่ 3.21 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 2.0 เมตร

เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมพีชซีแบบ กฎอินพุตเดียว เมื่อปรับเปลี่ยนสถานการณ์ที่ควบคุมสามารถพิจารณาได้จากตารางที่ 3.7 ซึ่ง สถานการณ์ต่างๆ ที่ปรากฏในตารางดังกล่าวจะใช้เป็นสถานการณ์ทดสอบสำหรับพิจารณา เปรียบเทียบเวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมแบบ ต่างๆ ต่อไป

ตารางที่ 3.7 เวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมพีชชีแบบกฎอินพุตเดียว

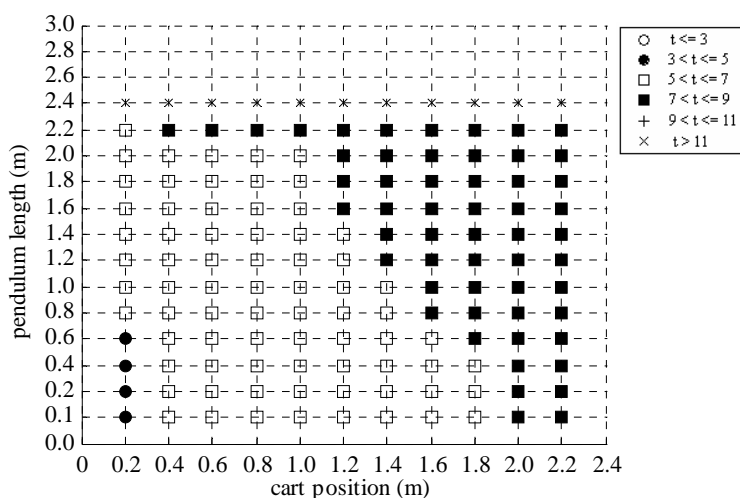
สถานการณ์	เวลาในการเข้าสู่เสถียรภาพ (วินาที)
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	8.24
มุมเริ่มต้นของลูกตุ้มเป็น 0.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	7.19
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 0.2 เมตร	6.59
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 2.0 เมตร	8.95
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	-

เพื่อทดสอบสมรรถนะของตัวควบคุมพีชชีแบบกฎอินพุตเดียว รวมทั้งพิจารณาขอบเขตการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมพีชชีแบบกฎอินพุตเดียว ได้ดำเนินการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบ เมื่อปรับเปลี่ยนตำแหน่งมุมเริ่มต้นของลูกตุ้มพร้อมกับปรับเปลี่ยนความยาวของก้านลูกตุ้ม โดยที่ตัวรถอยู่ที่กึ่งกลางของรางลื่น ทั้งนี้ในการจำลองสถานการณ์ได้ปรับเพิ่มมุมลูกตุ้มครั้งละ 2 องศา โดยเริ่มจากมุมเริ่มต้นของลูกตุ้มเป็น 2 องศา จนกระทั่งถึง 60 องศา และที่ทุกตำแหน่งมุมเริ่มต้นของลูกตุ้มที่เปลี่ยนแปลงไป จะทำการปรับความยาวก้านลูกตุ้มเพิ่มขึ้นครั้งละ 0.1 เมตร โดยเริ่มจาก 0.1 เมตร จนกระทั่งถึง 3.0 เมตร ผลการทดสอบเมื่อพิจารณาเวลาในการเข้าสู่เสถียรภาพของระบบแสดงได้ดังรูปที่ 3.22 โดยสัญลักษณ์ต่างๆ ที่ปรากฏในรูปแทนด้วยช่วงเวลาที่ตัวควบคุมพีชชีแบบกฎอินพุตเดียวใช้ในการสร้างเสถียรภาพให้ระบบ ซึ่งได้แสดงรายละเอียดไว้ดังรูป ตัวอย่างเช่น กรณีมุมเริ่มต้นเป็น 30 องศา ก้านลูกตุ้มยาว 1.0 เมตร ใช้เวลาเข้าสู่เสถียรภาพ 8.24 วินาที ซึ่งอยู่ในช่วง  $7 < t \leq 9$  วินาที ดังนั้นจึงแทนด้วยสัญลักษณ์ ■ ส่วนกรณีที่มุมเริ่มต้นเป็น 30 องศา ก้านลูกตุ้มยาว 0.2 เมตร ใช้เวลาเข้าสู่เสถียรภาพประมาณ 6.59 วินาที ดังนั้นจึงแทนด้วยสัญลักษณ์ เป็นต้น ส่วนบริเวณที่ไม่มีสัญลักษณ์ใดปรากฏ หมายถึง ตัวควบคุมพีชชีแบบกฎอินพุตเดียวไม่สามารถสร้างเสถียรภาพให้ระบบได้



รูปที่ 3.22 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม

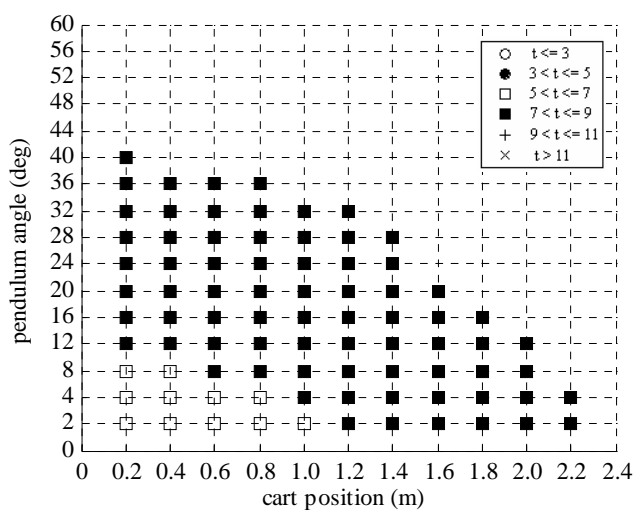
จากนั้นได้ดำเนินการทดสอบความสมรรถนะของตัวควบคุมดังกล่าว ด้วยการปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับปรับเปลี่ยนความยาวก้านลูกตุ้ม โดยที่มุมเริ่มต้นของลูกตุ้มเป็นศูนย์องศา ทั้งนี้จะทำการปรับเพิ่มตำแหน่งของตัวรถครั้งละ 0.2 เมตร โดยเริ่มจากตำแหน่งเริ่มต้นของตัวรถเป็น 0.2 เมตร จนกระทั่งถึง 2.4 เมตร พร้อมกันนี้ได้ทำการปรับเพิ่มความยาวก้านลูกตุ้มครั้งละ 0.2 เมตร โดยเริ่มจากก้านลูกตุ้มยาว 0.1 เมตร จนกระทั่งก้านลูกตุ้มยาวถึง 3.0 เมตร ผลการทดสอบแสดงดังรูปที่ 3.23 โดยสัญลักษณ์ต่างๆ ที่ปรากฏในรูปแทนด้วยช่วงเวลาในการสร้างเสถียร



รูปที่ 3.23 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม

ภาพให้ระบบตามที่ได้กำหนดไว้เช่นเดียวกับรูปที่ 3.22 ตัวอย่างเช่น กรณีที่ตำแหน่งเริ่มต้นของรถเป็น 2.0 เมตร และก้านลูกตุ้มยาว 1.0 เมตร ใช้เวลาในการสร้างเสถียรภาพประมาณ 7.19 วินาที ดังนั้นจึงแทนด้วยสัญลักษณ์ ■ เป็นต้น

นอกจากนี้ได้ทำการทดสอบตัวควบคุมพีชชีแบบกฎอินพุตเดียว โดยดำเนินการปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม พร้อมทั้งปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถ โดยที่ก้านลูกตุ้มยาว 1.0 เมตร เวลาที่ตัวควบคุมดังกล่าวใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวแสดงได้ดังรูปที่ 3.24 โดยจะสังเกตเห็นว่า ตัวควบคุมพีชชีแบบกฎอินพุตเดียวใช้เวลาในการสร้างเสถียรภาพให้ระบบประมาณ 9 วินาที ซึ่งในกรณีที่ตำแหน่งเริ่มต้นของตัวรถอยู่ใกล้กับกึ่งกลางของรางลื่นคือที่ระยะ 0.2 เมตร สามารถปรับเพิ่มมุมเริ่มต้นของลูกตุ้มได้ถึง 40 องศา แต่เมื่อตำแหน่งเริ่มต้นของตัวรถอยู่ห่างจากกึ่งกลางของรางลื่นมากขึ้น เช่นที่ระยะ 2.2 เมตร พบว่า ถ้ามุมเริ่มต้นของลูกตุ้มมากกว่า 4 องศา ตัวควบคุมพีชชีแบบกฎอินพุตเดียวจะไม่สามารถสร้างเสถียรภาพให้แก่ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้



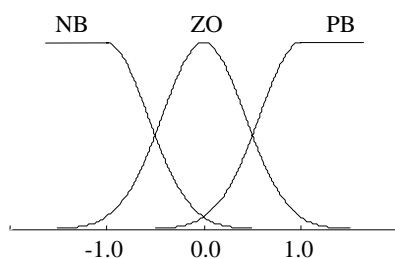
รูปที่ 3.24 เวลาเข้าสู่เสถียรเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม

### 3.8 การปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิก

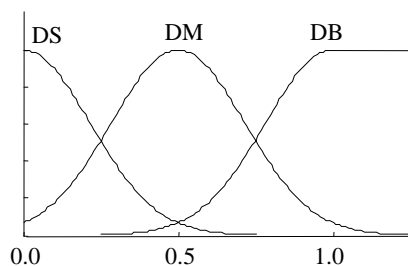
เพื่อศึกษาถึงผลกระทบของรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM และรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัตของตัวควบคุมพีชชีแบบกฎอินพุตเดียว ที่ส่งผลต่อการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ดังนั้นจึงได้ดำเนินการปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูปประฆังคว่ำ และรูปสี่เหลี่ยมคางหมู ทั้งนี้ในส่วนฟังก์ชันความเป็นสมาชิกของเอาต์พุตสำหรับ SIRM และตัวแปรพลวัตยังคงรูปแบบเป็นจำนวน

จริงเดี่ยวเช่นเดิม รวมถึงการตั้งกฎการควบคุม กระบวนการฟuzzyฟิเคชันและดีฟuzzyฟิเคชัน นอกจากนี้ค่าพารามิเตอร์ต่างๆ ของตัวควบคุมฟuzzyแบบกฎอินพุตเดี่ยวยังคงเดิม เพียงแต่ปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตในกฎฟuzzyเท่านั้น

เมื่อปรับฟังก์ชันความเป็นสมาชิกเป็นรูปประฆังคว่ำ ดังปรากฏในรูปที่ 3.25 และ 3.26 ซึ่งแสดงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM และของตัวแปรพลวัตตามลำดับ โดยการแบ่งส่วนตัวแปรฟuzzyประกอบด้วย 3 ฟuzzyเซต ได้แก่ NB (negative big) ZO (zero about) และ PB (positive big) ซึ่งเป็นตัวแปรภาษาสำหรับ SIRM โดยที่ DS (degree small) DM (degree medium) และ DB (degree big) เป็นตัวแปรภาษาสำหรับตัวแปรพลวัต

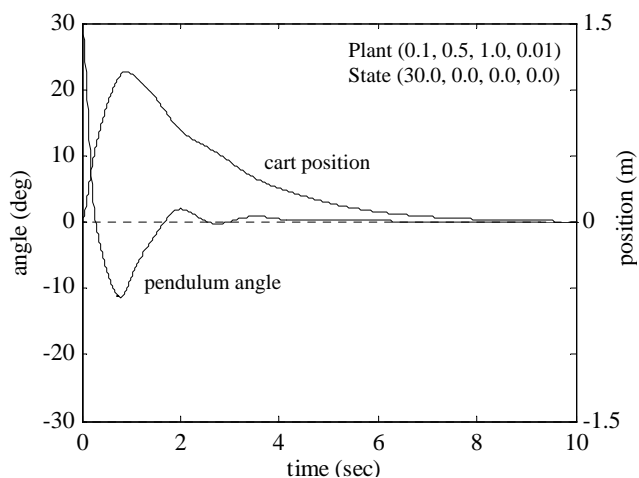


รูปที่ 3.25 ฟังก์ชันความเป็นสมาชิกรูปประฆังคว่ำของตัวแปรอินพุตแต่ละตัวใน SIRM



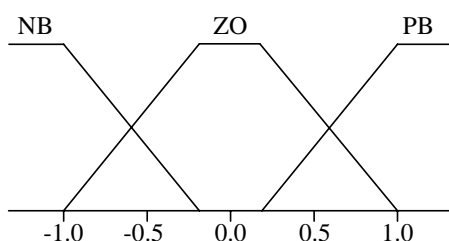
รูปที่ 3.26 ฟังก์ชันความเป็นสมาชิกรูปประฆังคว่ำของตัวแปรพลวัต

เมื่อกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร และก้านลูกตุ้มยาว 1.0 เมตร ผลการจำลองสถานการณ์แสดงได้ดังรูปที่ 3.27 ซึ่งจะเห็นได้ว่าตัวรถจะเคลื่อนที่กลับมาอยู่ที่กลางของรางลื่นอย่างช้าๆ ทำให้ตัวควบคุมฟuzzyแบบกฎอินพุตเดี่ยวใช้เวลาในการสร้างเสถียรภาพให้ระบบทั้งสิ้น 9.15 วินาที ซึ่งยาวนานกว่ากรณีใช้ฟังก์ชันความเป็นสมาชิกเป็นรูปสามเหลี่ยม ดังได้แสดงผลการจำลองสถานการณ์ไว้ในรูปที่ 3.12 ซึ่งระบบลูกตุ้มผกผันแบบข้อต่อเดี่ยวสามารถเข้าสู่เสถียรภาพได้ภายในเวลา 8.24 วินาที

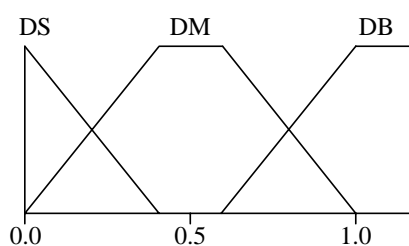


รูปที่ 3.27 ผลตอบสนองเมื่อรูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูปประฆังคว่ำ

จากนั้นได้ดำเนินการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน โดยใช้ฟังก์ชันความเป็นสมาชิกรูปสี่เหลี่ยมคางหมู ซึ่งฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM แสดงได้ด้วยรูปที่ 3.28 และฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัตปรากฏดังรูปที่ 3.29 ตามลำดับ ทั้งนี้ฟังก์ชันความเป็นสมาชิกดังกล่าวยังคงประกอบด้วย 3 ฟังก์ชันเซตดั้งเดิม

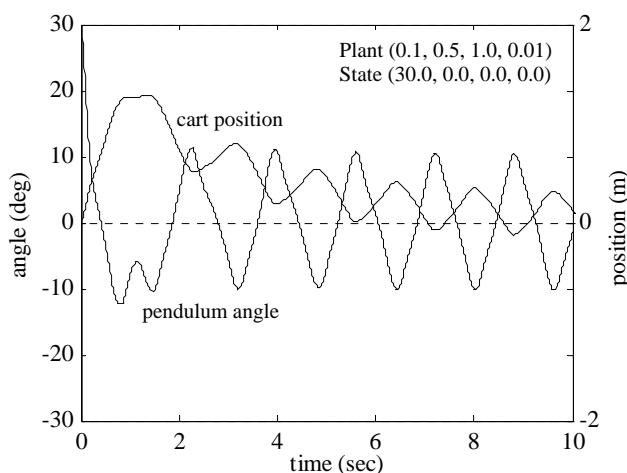


รูปที่ 3.28 ฟังก์ชันความเป็นสมาชิกรูปสี่เหลี่ยมคางหมูของตัวแปรอินพุตแต่ละตัวใน SIRM



รูปที่ 3.29 ฟังก์ชันความเป็นสมาชิกรูปสี่เหลี่ยมคางหมูของตัวแปรพลวัต

ผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว หลังจากปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกเป็นรูปสี่เหลี่ยมคางหมู แสดงได้ดังรูปที่ 3.30 ซึ่งกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร และก้านลูกตุ้มยาว 1.0 เมตร จากผลการจำลองสถานการณ์แสดงให้เห็นว่า ตัวควบคุมฟuzzy แบบกฎอินพุตเดียวไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ เนื่องจากตัวรถจะเคลื่อนที่กลับไปกลับมาทั้งทางด้านซ้ายและด้านขวา ส่งผลให้ลูกตุ้มหมุนกลับไปกลับมาในทิศทางเข็มนาฬิกาสลับกับทวนเข็มนาฬิกาด้วยเช่นกัน ทำให้มุมของลูกตุ้มเกิดการแกว่งกวัดรอบๆ ตำแหน่งสมดุลอยู่ตลอดเวลา และไม่สามารถตั้งตรงที่ตำแหน่งสมดุลศูนย์องศาได้ รวมทั้งไม่สามารถทำให้ตัวรถเคลื่อนที่กลับมายังตำแหน่งกึ่งกลางของรางเดินได้ด้วยเช่นกัน

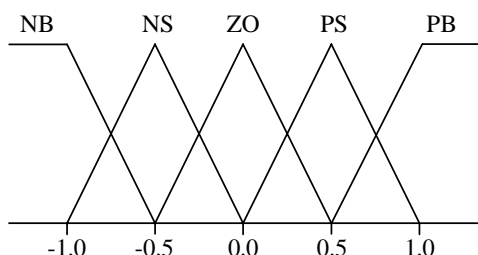


รูปที่ 3.30 ผลตอบสนองเมื่อรูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูปสี่เหลี่ยมคางหมู

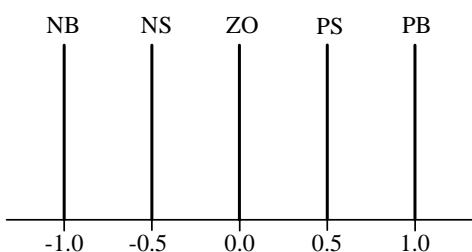
### 3.9 การเพิ่มการแบ่งส่วนตัวแปรฟuzzy

เพื่อศึกษาถึงผลกระทบจากการแบ่งส่วนตัวแปรฟuzzy ของตัวควบคุมฟuzzy แบบกฎอินพุตเดียว ต่อการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ได้ดำเนินการเพิ่มการแบ่งส่วนตัวแปรฟuzzy โดยประกอบด้วย 5 ฟuzzy เซต ซึ่งเดิมได้ออกแบบไว้ประกอบด้วย 3 ฟuzzy เซต ทั้งนี้ได้ทำการทดสอบโดยกำหนดให้รูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูปสามเหลี่ยมเช่นเดิม โดยยังคงกำหนดให้ค่าพารามิเตอร์ต่างๆ ของตัวควบคุมฟuzzy แบบกฎอินพุตเดียวมีค่าเท่าเดิม

ฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM ประกอบด้วยตัวแปรทางภาษา NB (negative big) NS (negative small) ZO (zero about) PS (positive small) และ PB (positive big) ดังแสดงในรูปที่ 3.31 ซึ่งกำหนดให้ขอบเขตของเอกภพสัมพัทธ์อยู่ในช่วง  $[-1, +1]$



รูปที่ 3.31 ฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM ที่มี 5 ฟัซซีเซต



รูปที่ 3.32 ฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตของ SIRM ที่มี 5 ฟัซซีเซต

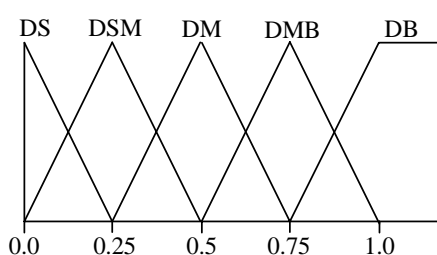
ส่วนฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตประกอบด้วย 5 ฟัซซีเซตด้วยเช่นกัน ดังแสดงในรูปที่ 3.32 ซึ่งแทนด้วยตัวแปรภาษาดังนี้คือ NB (negative big) NS (negative small) ZO (zero about) PS (positive small) และ PB (positive big) เมื่อทำการเพิ่มการแบ่งส่วนตัวแปรฟัซซี จำเป็นต้องปรับเปลี่ยนการกำหนดกฎฟัซซีที่ใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันด้วย ดังแสดงในตารางที่ 3.8 ซึ่งการกำหนดกฎฟัซซีเหล่านี้สามารถพิจารณาได้จากการวิเคราะห์ธรรมชาติของระบบลูกตุ้มผกผันแบบข้อต่อเดียว

ตารางที่ 3.8 การกำหนดกฎฟัซซีใน SIRM เมื่อเพิ่มการแบ่งส่วนตัวแปรฟัซซี

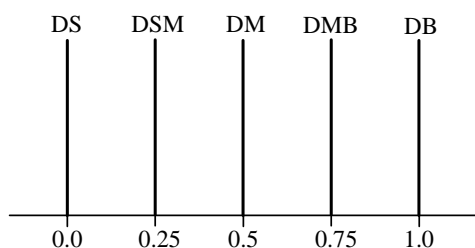
ตัวแปรในส่วนเงื่อนไข $x_i$ ( $i = 1, 2, 3, 4$ )	ตัวแปรในส่วนผลลัพธ์ $f_i$ ( $i = 1, 2, 3, 4$ )
NB	-1.0
NS	-0.5
ZO	0.0
PS	0.5
PB	1.0



ส่วนฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัต หลังจากที่ได้ทำการเพิ่มจำนวนการแบ่งส่วนตัวแปรฟuzzy เป็น 5 ฟuzzy เซต แสดงได้ดังที่ 3.33 โดยที่รูปทรงของฟังก์ชันความเป็นสมาชิกเป็นรูปสามเหลี่ยม ซึ่งแทนด้วยตัวแปรทางภาษาคำนี้คือ DS (degree small) DSM (degree small-medium) DM (degree medium) DMB (degree medium-big) และ DB (degree big) ในส่วนฟังก์ชันความเป็นสมาชิกของเอาต์พุตแสดงดังรูปที่ 3.34 ซึ่งประกอบด้วย 5 ฟuzzy เซตเช่นกัน โดยมีรูปแบบฟังก์ชันความเป็นสมาชิกเป็นจำนวนจริงเดี่ยว ทั้งนี้ตัวแปรทางภาษาที่ปรากฏในรูปดังกล่าวมีความหมายเช่นเดียวกับรูปที่ 3.33



รูปที่ 3.33 ฟังก์ชันความเป็นสมาชิกของตัวแปรพลวัตที่มี 5 ฟuzzy เซต



รูปที่ 3.34 ฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตของตัวแปรพลวัตที่มี 5 ฟuzzy เซต

สำหรับการกำหนดกฎฟuzzy ของตัวควบคุมฟuzzy แบบกฎอินพุตเดี่ยว เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดี่ยว จะพิจารณาตัวแปรสถานะเพียงตัวเดียวเท่านั้น นั่นคือ ขนาดมุมของลูกตุ้มผกผัน โดยที่ค่าฐานและค่าขยายของตัวควบคุมฟuzzy แบบกฎอินพุตเดี่ยวกำหนดตามตารางที่ 3.4 ซึ่งหลังจากเพิ่มจำนวนการแบ่งส่วนตัวแปรฟuzzy ย่อมส่งผลให้การกำหนดกฎฟuzzy เปลี่ยนแปลงไปด้วยเช่นกัน โดยตารางที่ 3.9 แสดงการกำหนดกฎฟuzzy ให้กับตัวแปรพลวัตเมื่อพิจารณามุมและความเร็วเชิงมุมของลูกตุ้ม ในขณะที่ตารางที่ 3.10 เป็นการกำหนดกฎฟuzzy ให้กับตัวแปรพลวัต เมื่อพิจารณาค่าแรงแท่งและความเร็วของตัวรถ

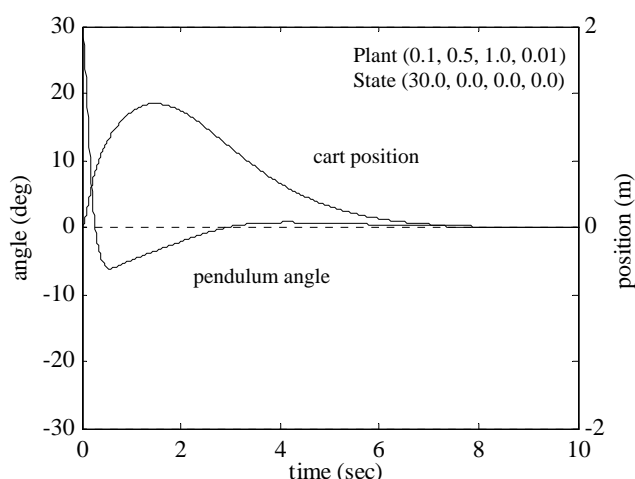
ตารางที่ 3.9 การกำหนดกฎฟิชชีสำหรับตัวแปรพลวัตของมุมและความเร็วเชิงมุม

ตัวแปรในส่วนเงื่อนไข $ x_1 $	ตัวแปรในส่วนผลลัพธ์ $\Delta w_i (i = 1, 2)$
DS	0.00
DSM	0.25
DM	0.50
DMB	0.75
DB	1.00

ตารางที่ 3.10 การกำหนดกฎฟิชชีสำหรับตัวแปรพลวัตของตำแหน่งและความเร็วรถ

ตัวแปรในส่วนเงื่อนไข $ x_1 $	ตัวแปรในส่วนผลลัพธ์ $\Delta w_i (i = 3, 4)$
DS	1.00
DSM	0.75
DM	0.50
DMB	0.25
DB	0.00

ผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว หลังจากเพิ่มจำนวนการแบ่งส่วนตัวแปรฟิชชีที่ได้ดังรูปที่ 3.35 โดยกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถอยู่ที่กึ่งกลางของรางลื่น และก้านลูกตุ้มยาว 1.0 เมตร เมื่อพิจารณาผลการจำลองสถานการณ์ พบว่า ตัวควบคุมฟิชชีแบบกฎอินพุตเดียวใช้เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวประมาณ 8.20 วินาที ซึ่งใกล้เคียงกับผลการจำลองสถานการณ์ที่ปรากฏในรูปที่ 3.12 โดยเป็นกรณีที่ดำเนินการแบ่งส่วนตัวแปรฟิชชีประกอบด้วย 3 ฟิชชีเซต ซึ่งใช้เวลาในการเข้าสู่เสถียรภาพทั้งสิ้น 8.24 วินาที นอกจากนี้ เมื่อพิจารณาเปรียบเทียบพลวัตของระบบลูกตุ้มผกผันจากทั้งสองกรณี จะได้ว่า กระบวนการต่างๆ ในการสร้างเสถียรภาพให้ระบบดังกล่าวมีความคล้ายคลึงกันอย่างมาก อย่างไรก็ตามถ้าพิจารณาจำนวนกฎฟิชชีที่ใช้พบว่า กรณีที่ทำการแบ่งส่วนตัวแปรฟิชชีเพียง 3 ฟิชชีเซต จะมีจำนวนกฎฟิชชีรวมทั้งสิ้น 24 กฎ ในขณะที่เมื่อทำการเพิ่มการแบ่งส่วนตัวแปรฟิชชีเป็น 5 ฟิชชีเซต พบว่า จำนวนกฎฟิชชีเพิ่มขึ้นเป็น 40 กฎ ดังนั้น อาจกล่าวได้ว่า การเพิ่มจำนวนการแบ่งส่วนตัวแปรฟิชชีจะส่งผลให้จำนวนกฎฟิชชีเพิ่มขึ้น โดยเวลาที่ใช้สำหรับสร้างเสถียรภาพให้ระบบไม่แตกต่างจากเดิมมากนัก



รูปที่ 3.35 ผลตอบสนองเมื่อเพิ่มการแบ่งส่วนตัวแปรฟิซซีเป็น 5 ฟิซซีเซ็ท

### 3.10 สรุป

การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวโดยใช้ตัวควบคุมฟิซซีแบบกฎอินพุตเดียว ซึ่งประกอบด้วย SIRM และค่าระดับความสำคัญพลวัต สามารถช่วยลดจำนวนกฎฟิซซีเมื่อเปรียบเทียบกับตัวควบคุมฟิซซีแบบดั้งเดิม โดยใช้จำนวนกฎฟิซซีรวมทั้งสิ้น 24 กฎ เนื่องจากจะนำตัวแปรอินพุตเพียงตัวเดียวเท่านั้นมาจัดวางในส่วนเงื่อนไขของกฎฟิซซีแต่ละกฎ และเป็นการง่ายในการกำหนดกฎฟิซซีให้ได้อย่างเหมาะสม จากผลการจำลองสถานการณ์แสดงให้เห็นว่าตัวควบคุมฟิซซีแบบกฎอินพุตเดียวสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ แม้ทำการปรับเปลี่ยนสถานการณ์ในการควบคุม เช่น เปลี่ยนความยาวของก้านลูกตุ้ม ตำแหน่งเริ่มต้นของตัวรถ มวลของก้านลูกตุ้ม เป็นต้น ทั้งนี้ได้พิจารณาควบคุมทั้งตำแหน่งของตัวรถ และตำแหน่งมุมของลูกตุ้มผกผัน และเมื่อเปรียบเทียบเวลาในการเข้าสู่เสถียรภาพของระบบที่ได้จากการจำลองสถานการณ์พบว่าใกล้เคียงกับเวลาในผลงานวิจัยอ้างอิง ดังนั้นอาจกล่าวได้ว่าโปรแกรมสำหรับใช้จำลองสถานการณ์ที่พัฒนาขึ้นมีความถูกต้องและน่าเชื่อถือ นอกจากนี้ได้จำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยการปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูปประฆังคว่ำ และรูปสี่เหลี่ยมคางหมู เพื่อศึกษาถึงกระทบต่อการสร้างเสถียรภาพให้ระบบของตัวควบคุมฟิซซีแบบกฎอินพุตเดียว พบว่า การปรับเปลี่ยนฟังก์ชันความเป็นสมาชิกดังกล่าวไม่สามารถช่วยลดเวลาในการเข้าสู่เสถียรภาพของระบบให้น้อยลงได้ รวมทั้งได้ดำเนินการเพิ่มจำนวนการแบ่งส่วนตัวแปรฟิซซี ซึ่งจากผลการจำลองสถานการณ์แสดงให้เห็นว่า เวลาในการสร้างเสถียรภาพให้แก่ระบบใกล้เคียงกับการแบ่งส่วนตัวแปรฟิซซีเดิมที่ได้ออกแบบไว้ก่อนหน้านี้ และทำให้จำนวนกฎฟิซซีเพิ่มขึ้นตามจำนวนการแบ่งส่วนตัวแปรฟิซซีด้วย ดังนั้นในการพัฒนาตัวควบคุม

เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว จะพิจารณาใช้รูปแบบฟังก์ชันความเป็นสมาชิกและจำนวนการแบ่งส่วนตัวแปรฟuzzyซึ่งตามผลงานวิจัยอ้างอิง อย่างไรก็ตาม เมื่อพิจารณาผลการจำลองสถานการณ์ของตัวควบคุมฟuzzyแบบกฎอินพุตเดียว เมื่อปรับเปลี่ยนสถานการณ์ในการควบคุม พบว่า ตัวควบคุมดังกล่าวใช้เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวค่อนข้างมาก ดังนั้นในงานวิจัยวิทยานิพนธ์นี้จึงได้พยายามพัฒนาวิธีการควบคุมแบบใหม่เพื่อให้สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้รวดเร็วยิ่งขึ้น เรียกตัวควบคุมแบบใหม่ที่พัฒนาขึ้นนี้ว่า ตัวควบคุมนิวโร-ฟuzzy ซึ่งจะได้อธิบายรายละเอียดในบทที่ 4 ต่อไป

## บทที่ 4

### ตัวควบคุมนิวโร-ฟัซซี่

#### 4.1 บทนำ

การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวดังได้กล่าวรายละเอียดในบทที่ 3 แล้วนั้น พบว่าสามารถช่วยลดจำนวนกฎฟัซซี่และง่ายต่อการกำหนดกฎฟัซซี่ให้ได้อย่างเหมาะสม แต่จากการพิจารณาผลการจำลองสถานการณ์เมื่อปรับเปลี่ยนสถานการณ์ในการควบคุม แสดงให้เห็นว่า ตัวควบคุมดังกล่าวจำเป็นต้องใช้เวลาค่อนข้างมากในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ดังนั้นในงานวิจัยวิทยานิพนธ์นี้จึงได้พยายามพัฒนาวิธีการควบคุมแบบใหม่ เพื่อให้สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้รวดเร็วยิ่งขึ้น เรียกวิธีการควบคุมแบบใหม่ที่นำเสนอชื่อว่า ตัวควบคุมนิวโร-ฟัซซี่ (neuro-fuzzy controller) ซึ่งเป็นการผสมผสานระหว่างตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวและโครงข่ายประสาทเทียม โดยค่าระดับความสำคัญพลวัตสำหรับตัวแปรอินพุตแต่ละตัวได้มาจากการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่งมีลักษณะเป็นโครงข่ายหลายชั้นแบบป้อนไปหน้า (multilayer feedforward network) ทั้งนี้อัลกอริทึมที่ใช้ในกระบวนการเรียนรู้เพื่อปรับค่าระดับความสำคัญพลวัตแต่ละตัวให้เหมาะกับสถานการณ์ที่ดำเนินการควบคุมเป็นแบบแพร่กลับ (backpropagation learning algorithm) ซึ่งเป็นวิธีการเรียนรู้ที่นิยมใช้กันอย่างกว้างขวาง

เนื้อหาในบทนี้กล่าวถึง ความรู้พื้นฐานด้านโครงข่ายประสาทเทียม องค์ประกอบของโครงข่ายประสาทเทียม และกระบวนการเรียนรู้แบบแพร่กลับ จากนั้นได้นำเสนอหลักการของตัวควบคุมนิวโร-ฟัซซี่ การออกแบบตัวควบคุมนิวโร-ฟัซซี่ การทำงานของตัวควบคุมนิวโร-ฟัซซี่ การเลือกค่าอัตราการเรียนรู้ที่เหมาะสม รวมทั้งได้ดำเนินการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ เมื่อปรับเปลี่ยนสถานการณ์ในการควบคุม ทั้งนี้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ได้แก่ ค่าน้ำหนัก และค่าไบแอส ได้มาจากการสุ่มค่า ซึ่งในการสร้างเสถียรภาพให้ระบบยังคงใช้ค่าพารามิเตอร์เริ่มต้นชุดเดียวกันนี้ ถึงแม้ว่าสถานการณ์ในการควบคุมจะเปลี่ยนแปลงไป จากนั้นได้ศึกษาถึงผลกระทบจากค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ที่มีผลต่อการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยทดสอบสุ่มค่าพารามิเตอร์เริ่มต้นเหล่านี้ในการจำลองสถานการณ์ เพื่อพิจารณาปริมาณการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว

## 4.2 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม คือ โครงสร้างของหน่วยประมวลผลจำนวนมาก ที่ถูกจำลองขึ้นตามอย่างโครงสร้างของระบบประสาทของสมองมนุษย์ การสร้างแบบจำลองของโครงข่ายประสาทขึ้นมาเป็นโครงข่ายประสาทเทียมนั้น ทำได้โดยการพยายามทำความเข้าใจกระบวนการทำงานของสมองแล้วพยายามอธิบายการทำงานนั้นด้วยแบบจำลองทางคณิตศาสตร์ ซึ่งโครงข่ายประสาทเทียมมีข้อดีหลายประการ ได้แก่

1) มีความสามารถในการเรียนรู้และปรับค่าน้ำหนักในการเชื่อมโยงไปตามสถานการณ์ที่กำหนดหรือตามสภาวะแวดล้อมที่เปลี่ยนแปลงไป

2) มีความสามารถในการประมาณค่าฟังก์ชันต่อเนื่องไม่เป็นเชิงเส้น ดังนั้นจึงสามารถนำไปใช้กับระบบที่ไม่เป็นเชิงเส้นได้

3) แบบจำลองของโครงข่ายประสาทเทียมประกอบด้วยเซลล์ประสาทเทียมจำนวนมากมาเป็นโครงข่ายแบบขนานและเชื่อมโยงกันด้วยค่าน้ำหนักที่สามารถปรับตัวได้ เมื่อเกิดความผิดพลาดในบางเซลล์ประสาททำให้ไม่ส่งผลกระทบต่อสมรรถนะโดยรวมทั้งหมด หรือที่เรียกว่า ความทนทานต่อความผิดพลาด (fault tolerance)

4) เนื่องจากโครงข่ายประสาทเทียมสามารถมีได้หลายอินพุตและหลายเอาต์พุตดังนั้นจึงง่ายต่อการนำไปประยุกต์ใช้กับระบบหลายตัวแปร (multivariable system)

5) ปัจจุบันได้มีการพัฒนาโครงข่ายประสาทเทียมในรูปแบบอุปกรณ์ฮาร์ดแวร์ ซึ่งมีความสามารถในการประมวลผลจำนวนมากๆ และเรียนรู้ได้อย่างรวดเร็ว จึงเหมาะสำหรับประยุกต์ใช้กับระบบที่มีความซับซ้อนและต้องการความเร็วในการประมวลผล

เนื่องจากโครงข่ายประสาทเทียมมีข้อดีหลายประการดังได้กล่าวมาแล้วข้างต้น จึงมีการนำมาประยุกต์ในงานประเภทต่างๆ อย่างแพร่หลาย ตัวอย่างเช่น ตัวควบคุมนิวโร-พีไอดี (neuro-PID controller) ซึ่งเป็นการนำโครงข่ายประสาทเทียมมาใช้ในงานควบคุมเพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน (Omatu, Fujinaka, and Yoshioka, 2000) โดยโครงข่ายประสาทเทียมจะทำหน้าที่เรียนรู้เพื่อปรับค่าอัตราขยายของตัวควบคุมพีไอดีโดยอัตโนมัติ รวมถึงการใช้โครงข่ายประสาทเทียมสำหรับควบคุมระบบอนุกรม (Cui and Shin, 1993) การรู้จำรูปแบบ (pattern recognition) (Toh, 2003) เป็นต้น นอกจากนี้ยังได้มีการนำโครงข่ายประสาทเทียมมาประยุกต์ใช้งานร่วมกับวิธีชาลูลาดแบบอื่นๆ เช่น การให้โครงข่ายประสาทเทียมปรับฟังก์ชันความเป็นสมาชิกของตัวควบคุมแบบฟัซซี่ รวมทั้งกำหนดกฎฟัซซี่โดยอัตโนมัติ (Di and Katsunori, 1997) การใช้จินเนติกอัลกอริทึมเข้ามามีส่วนร่วมในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม (Sun, Ren, Wu, Ning, and Wang, 2003) เป็นต้น

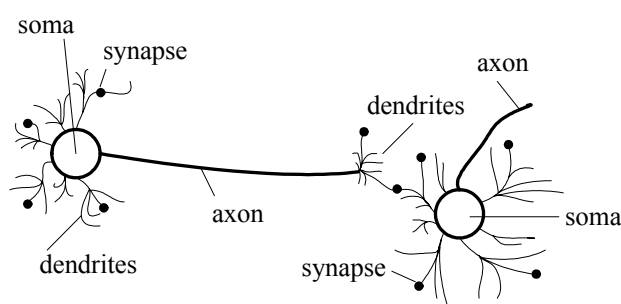
#### 4.2.1 องค์ประกอบของโครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมได้ถูกพัฒนาขึ้นโดยอาศัยหลักการทำงานของสมองมนุษย์ ดังนั้นความเข้าใจในคุณลักษณะเชิงกายภาพ และเชิงพฤติกรรมขององค์ประกอบต่างๆ ในสมองจึงเป็นสิ่งจำเป็นอย่างยิ่ง ซึ่งสมองประกอบด้วยหน่วยประมวลผลพื้นฐานที่เรียกว่าเซลล์ประสาท (neuron) ที่มีการเชื่อมโยงกันเป็นจำนวนมาก ดังนั้นเซลล์ประสาทแต่ละเซลล์ก็คือ หน่วยประมวลผลอย่างง่าย ๆ ซึ่งรับสัญญาณและรวมสัญญาณที่ถูกส่งมาจากเซลล์ประสาทอื่นๆ แต่ละเซลล์ประสาทในสมองของมนุษย์มีโครงสร้างที่ง่ายไม่ซับซ้อนประกอบด้วยองค์ประกอบหลัก 3 ส่วน ดังแสดงในรูปที่ 4.1 ได้แก่

1) ตัวเซลล์ซึ่งเรียกว่า โซมา (soma) มีลักษณะเป็นรูปทรงพีระมิด หรือทรงกระบอก ทำหน้าที่ในการรวมสัญญาณที่เข้ามาในเซลล์จากเดนไดรต์

2) เดนไดรต์ (dendrite) คือ เส้นใยบางๆ ที่เซลล์ประสาทใช้รับสัญญาณไฟฟ้าเข้าสู่เซลล์ แต่ละเซลล์ประสาทจะมีเดนไดรต์จำนวนมากจัดตัวเป็นลักษณะเหมือนกิ่งไม้ โดยที่เดนไดรต์จะแตกแขนงเข้าไปยังเซลล์ประสาทอื่นรอบๆ โซมา เพื่อรับสัญญาณจากเซลล์ประสาทอื่นในสมอง

3) แอกซอน (axon) คือ สายส่งผ่านสัญญาณทรงกระบอกขนาดยาวและใหญ่ที่เซลล์ประสาทใช้เป็นทางส่งสัญญาณไปยังเซลล์ประสาทอื่นๆ ส่วนปลายของแอกซอนจะแตกออกเป็นกิ่งก้านย่อยๆ โดยที่ส่วนปลายของแต่ละกิ่งก้านเหล่านี้มีลักษณะเป็นปม และจะไปจ่ออยู่จนเกือบสัมผัสกับปลายของเดนไดรต์ของเซลล์ประสาทอื่น



รูปที่ 4.1 องค์ประกอบของโครงข่ายประสาทเทียม

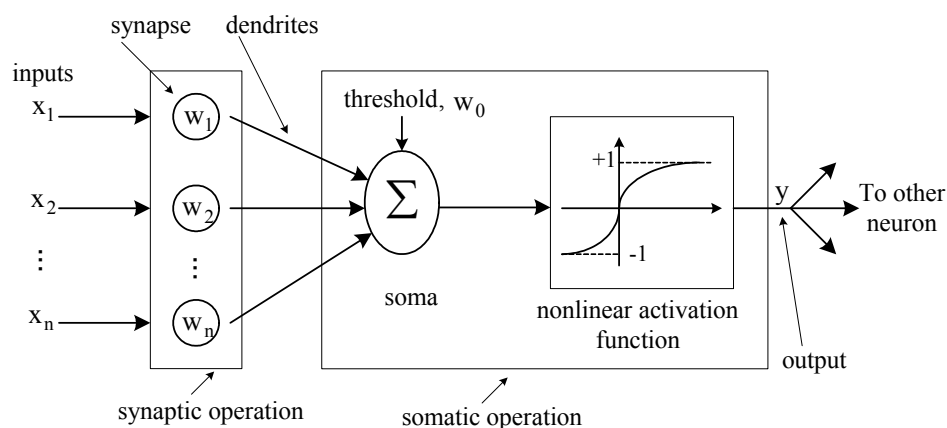
บริเวณที่เป็นรอยต่อระหว่างปลายของแอกซอนกับปลายของเดนไดรต์เรียกว่าไซแนปส์ (synapse) สัญญาณไฟฟ้าที่ถูกส่งมาถึงปลายของแอกซอนจะกระตุ้นให้เกิดการส่งผ่านสัญญาณในเชิงเคมีผ่านไซแนปส์ สัญญาณเชิงดังกล่าวจะถูกเดนไดรต์ตีความเป็นสัญญาณไฟฟ้าเข้าสู่เซลล์ประสาท เมื่อผลรวมของสัญญาณสูงถึงระดับจุดเปลี่ยนที่จะกระตุ้นเซลล์ประสาท ตัวเซลล์

ประสาทจะให้กำเนิดศักย์กระทำ (action potential) แต่ถ้าผลรวมดังกล่าวไม่ถึงระดับจุดเปลี่ยน สัญญาณที่มีอยู่จะสลายไปอย่างรวดเร็วและไม่ทำให้เกิดการกระตุ้นใดๆ ดังนั้นคุณลักษณะสำคัญของไซแนปส์คือ ความแรงของสัญญาณที่ถูกส่งผ่านซึ่งขึ้นอยู่กับความเหนียวแน่นของการเชื่อมต่อ โดยสัญญาณนี้อาจมีสภาพเป็นสัญญาณกระตุ้น (excitatory) หรือสัญญาณยับยั้ง (inhibitory) ก็ได้

สิ่งสำคัญของโครงข่ายประสาทเทียม คือ การจำลองไซแนปส์ แม็คคัลลอคและพิตส์ (McCulloch and Pitts) เสนอให้ใช้ตัวแปรที่เรียกว่า ค่าน้ำหนัก (weight) ในการจำลองไซแนปส์ หากค่าน้ำหนักนี้มีขนาดใหญ่ ความเหนียวแน่นของรอยต่อไซแนปส์จะมีค่าสูง นั่นคือส่งผ่านสัญญาณได้มาก หากค่าน้ำหนักนี้มีขนาดเล็กก็หมายความว่าสัญญาณจะส่งผ่านรอยต่อไซแนปส์ได้น้อย นอกจากนี้ ความเป็นบวกหรือลบของค่าน้ำหนักก็มีความหมายเช่นกัน กล่าวคือ ค่าน้ำหนักมีค่าเป็นบวกจะเป็นสัญญาณกระตุ้น แต่หากค่าน้ำหนักมีค่าเป็นลบจะเป็นสัญญาณยับยั้ง ดังนั้นจะเห็นได้ว่าการประมวลผลของโครงข่ายประสาทเทียมอาศัยหลักการประมวลผลของสมอง ซึ่งมีองค์ประกอบต่างๆ ที่ทำหน้าที่คล้ายคลึงกันดังแสดงในตารางที่ 4.1 และสามารถอธิบายการประมวลผลดังกล่าวได้ดังรูปที่ 4.2

ตารางที่ 4.1 องค์ประกอบของสมองและโครงข่ายประสาทเทียม

สมอง	โครงข่ายประสาทเทียม
ตัวเซลล์	เซลล์ประสาทเทียม
เดนไดรต์	อินพุต
แอกซอน	เอาต์พุต
ไซแนปส์	ค่าน้ำหนัก

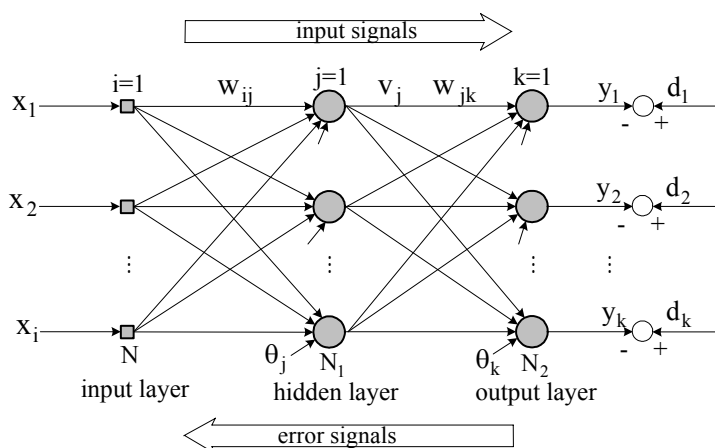


รูปที่ 4.2 กระบวนการประมวลผลของโครงข่ายประสาทเทียม



#### 4.2.2 การเรียนรู้แบบแพร่กลับ

การเรียนรู้ของโครงข่ายประสาทเทียมด้วยวิธีแพร่กลับค่าความคลาดเคลื่อนนั้น ประกอบด้วยสองกระบวนการสำคัญคือ การป้อนสัญญาณอินพุตแบบไปหน้า (feedforward) และแพร่กลับค่าความคลาดเคลื่อนแบบย้อนกลับ (feedback) ในลักษณะขั้นต่อขั้น ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 การเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม

กระบวนการเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม ที่ใช้ฟังก์ชันกระตุ้นแบบซิกมอยด์ (ดูรายละเอียดในภาคผนวก ง.) สามารถสรุปได้ 7 ขั้นตอน ดังนี้คือ

ขั้นตอนที่ 1 กำหนดค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ได้แก่ ค่าน้ำหนัก และไบแอส โดยการสุ่มค่า ซึ่งกำหนดให้มีค่าน้อยๆ อาจอยู่ในช่วง  $[-1, +1]$

ขั้นตอนที่ 2 คำนวณเอาต์พุตจากชั้นซ่อนเร้น

$$v_j(n) = f(\text{net}_j(n)) = \frac{1}{1 + \exp(-\text{net}_j(n))} \quad (4-1)$$

โดยที่

$$\text{net}_j(n) = \sum_{i=1}^N w_{ij} \cdot x_i + \theta_j \quad \text{เมื่อ } j = 1, 2, 3, \dots, N_1$$

ขั้นตอนที่ 3 คำนวณค่าเอาต์พุตจากชั้นเอาต์พุต

$$y_k(n) = f(\text{net}_k(n)) = \frac{1}{1 + \exp(-\text{net}_k(n))} \quad (4-2)$$

โดยที่

$$\text{net}_k(n) = \sum_{j=1}^{N_1} w_{jk} \cdot v_j + \theta_k \quad \text{เมื่อ } k = 1, 2, 3, \dots, N_2$$

ขั้นตอนที่ 4 ปรับค่าน้ำหนักจากชั้นซ่อนเร้นไปยังชั้นเอาต์พุต

$$w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n) \quad (4-3)$$

โดยที่

$$\Delta w_{jk}(n) = \eta_1 \cdot \delta_k \cdot v_k \quad \text{และ} \quad \delta_k = (d_k(n) - y_k(n)) \cdot y_k(n) \cdot (1 - y_k(n))$$

ขั้นตอนที่ 5 ปรับค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้น

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \quad (4-4)$$

โดยที่

$$\Delta w_{ij}(n) = \eta_2 \cdot \delta_j \cdot x_i \quad \text{และ} \quad \delta_j = \sum_k \delta_k \cdot w_{jk} \cdot v_j(n) \cdot (1 - v_j(n))$$

ขั้นตอนที่ 6 ปรับค่าไบแอสที่ชั้นซ่อนเร้นและชั้นเอาต์พุต

$$\theta_j(n+1) = \theta_j(n) + \eta_3 \delta_j(n) \quad (4-5)$$

$$\theta_k(n+1) = \theta_k(n) + \eta_4 \delta_k(n) \quad (4-6)$$

ขั้นตอนที่ 7 ตรวจสอบเงื่อนไขการหยุดเรียนรู้ ถ้ายังไม่พบให้  $n = n + 1$  แล้วกลับไปขั้นตอนที่ 2

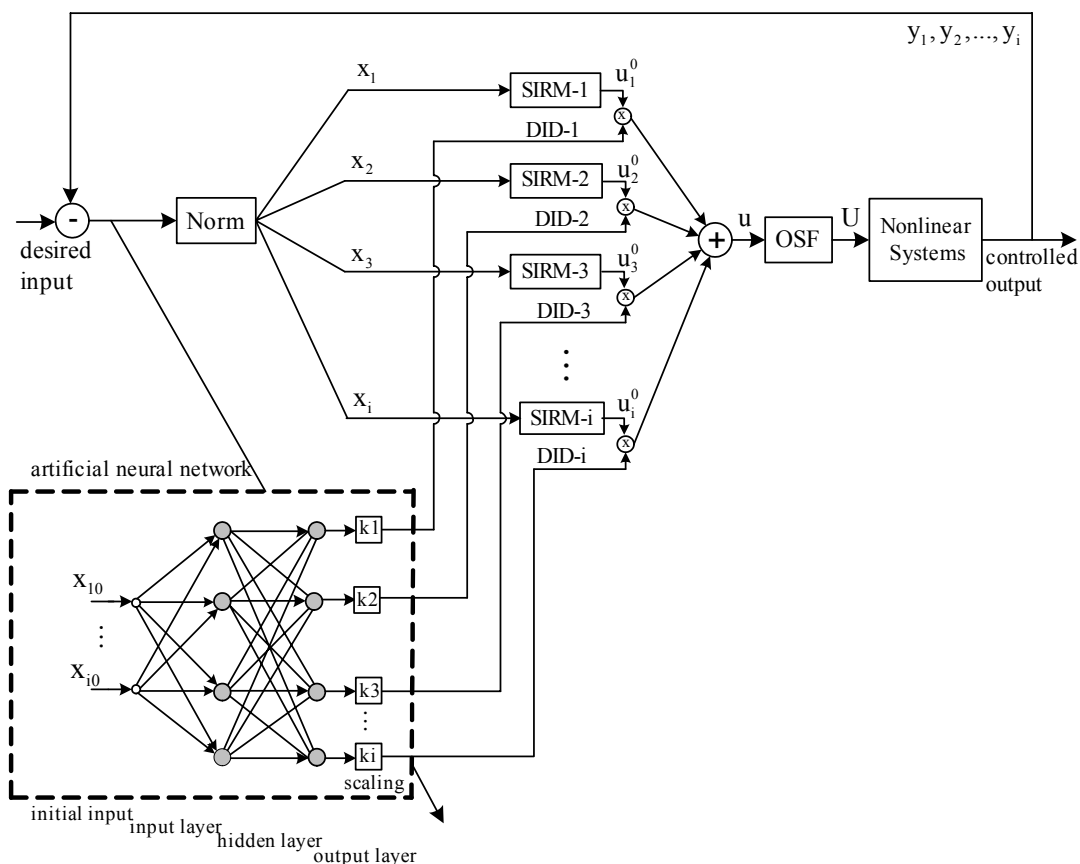
การกำหนดค่าอัตราการเรียนรู้  $\eta$ ,  $\eta_2$ ,  $\eta_3$  และ  $\eta_4$  โดยทั่วไปแล้วจะกำหนดให้มีค่าเป็นบวก อยู่ในช่วง  $[0, 1]$  ซึ่งค่าพารามิเตอร์เหล่านี้มีผลต่อการเรียนรู้ของโครงข่ายประสาทเทียม กล่าวคือ ถ้ากำหนดให้ค่าอัตราการเรียนรู้มีค่าน้อยเกินไป จะส่งผลให้เกิดการเปลี่ยนแปลงค่าน้ำหนักและค่าไบแอสของโครงข่ายประสาทเทียมเพียงเล็กน้อย ผลตอบสนองที่ได้จึงมีความเรียบ (smooth) แต่การเรียนรู้ของโครงข่ายจะดำเนินไปอย่างช้าๆ ด้วยเช่นกัน ทำให้ต้องใช้เวลาสำหรับการเรียนรู้ค่อนข้างมากจึงจะได้เอาต์พุตของโครงข่ายประสาทเทียมใกล้เคียงกับเอาต์พุตที่ต้องการ ในทางตรงกันข้าม ถ้ากำหนดให้ค่าอัตราการเรียนรู้มีค่ามากเกินไป ก็จะส่งผลให้โครงข่ายประสาทเทียมสามารถเรียนรู้ได้อย่างรวดเร็ว เนื่องจากค่าน้ำหนักและค่าไบแอสได้รับการปรับค่าต่างไปจากเดิมมาก แต่อาจทำให้ขาดเสถียรภาพได้ เนื่องจากมีการแกว่งกวัด (oscillation) เกิดขึ้น วิธีการหนึ่ง ที่ช่วยให้โครงข่ายประสาทเทียมเรียนรู้ได้รวดเร็วขึ้น และหลีกเลี่ยงการขาดเสถียรภาพ คือ การเพิ่มพจน์โมเมนตัม (momentum term) (Haykin, 1994) ดังแสดงด้วยสมการที่ (4-7) ซึ่งเป็นตัวอย่างการปรับแก้ค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น ในทำนองเดียวกันการปรับแก้ค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุตก็สามารถดำเนินการได้ในลักษณะเช่นเดียวกันนี้

$$\Delta w_{ij}(n) = \eta \cdot \delta_j(n) \cdot x_i(n) + \alpha \cdot \Delta w_{ij}(n-1) \quad (4-7)$$

โดยที่  $\alpha$  คือ ค่าคงที่โมเมนตัม (momentum constant) ซึ่งโดยทั่วไปแล้วจะกำหนดให้มีค่าเป็นบวกอยู่ในช่วง  $[0, 1]$

### 4.3 หลักการของตัวควบคุมนิวโร-ฟัซซี่

โครงสร้างของตัวควบคุมนิวโร-ฟัซซี่ ที่พัฒนาขึ้นแสดงได้ดังรูปที่ 4.4 ซึ่งจะสังเกตเห็นว่า ตัวควบคุมดังกล่าวประกอบด้วยมอดูลฟัซซี่แบบกฎอินพุตเดี่ยว (SIRM) และค่าระดับความสำคัญพลวัต (DID) เช่นเดียวกับตัวควบคุมฟัซซี่แบบกฎอินพุตเดี่ยว แต่จะแตกต่างกันในส่วนของการหาค่าระดับความสำคัญพลวัต กล่าวคือ ตัวควบคุมฟัซซี่แบบกฎอินพุตเดี่ยว สามารถหาค่านี้ได้จากผลรวมของค่าฐานกับผลคูณระหว่างค่าขยาย และผลการอนุมานกฎฟัซซี่ของตัวแปรพลวัต ดังได้อธิบายไว้ในบทที่ 3 ในขณะที่ตัวควบคุมนิวโร-ฟัซซี่สามารถหาค่าระดับความสำคัญพลวัตได้จากการเรียนรู้ของโครงข่ายประสาทเทียม โดยในส่วนของมอดูลฟัซซี่แบบกฎอินพุตเดี่ยว การกำหนดกฎฟัซซี่แต่ละกฎยังคงประกอบด้วยตัวแปรอินพุตเพียงตัวเดียวเท่านั้น



รูปที่ 4.4 โครงสร้างของตัวควบคุมนิวโร-ฟัซซี่

การหาค่าระดับความสำคัญพลวัตของตัวควบคุมนิวโร-ฟัซซี่ พิจารณาได้ด้วยวิธีการเรียนรู้แบบแพร่กลับค่าความคลาดเคลื่อน โดยสัญญาณอินพุตจะเข้าสู่โครงข่ายประสาทเทียมที่ชั้นอินพุต และผ่านกระบวนการเรียนรู้ เพื่อปรับค่าน้ำหนักและค่าไบแอสของโครงข่ายประสาทเทียม จนกระทั่งค่าความคลาดเคลื่อนลดลง โดยค่าระดับความสำคัญพลวัตจะได้จากชั้นเอาต์พุตของโครงข่ายประสาทเทียม แต่เนื่องจากฟังก์ชันกระตุ้นที่ใช้สำหรับการเรียนรู้แบบแพร่กลับเป็นแบบซิกมอยด์ ซึ่งเอาต์พุตมีขอบเขตอยู่ในช่วง  $[0, 1]$  ดังนั้นจึงจำเป็นต้องยกระดับเอาต์พุตนี้ด้วยตัวปรับค่า (scaling) เพื่อให้ได้ค่าระดับความสำคัญพลวัตที่สามารถสร้างเสถียรภาพให้ระบบที่มีความไม่เป็นเชิงเส้นได้ อย่างไรก็ตาม ในการสร้างเสถียรภาพให้ระบบดังกล่าวนี้ ไม่สามารถกำหนดค่าระดับความสำคัญพลวัตที่ต้องการสำหรับใช้เปรียบเทียบกับค่าระดับความสำคัญพลวัตที่ได้จากเอาต์พุตของโครงข่ายประสาทเทียม ดังนั้นจึงจำเป็นต้องใช้ค่าอื่นที่ทราบค่าและสามารถทดแทนกันได้ โดยในวิทยานิพนธ์นี้ได้นำสัญญาณเอาต์พุตควบคุมมาเปรียบเทียบกับสัญญาณอินพุตที่ต้องการ เพื่อคำนวณหาค่าความคลาดเคลื่อนที่ใช้ในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม

หลังจากที่ได้ค่า  $u_i^0$  จากมอดูลพีชชีแบบกฎอินพุตเดียว SIRM-i และค่าระดับความสำคัญพลวัต DID-i จากการเรียนรู้ของโครงข่ายประสาทเทียมสำหรับอินพุตแต่ละตัวแล้ว สัญญาณควบคุม  $u$  ที่กระทำกับระบบไม่เป็นเชิงเส้น สามารถหาได้จากผลรวมของผลคูณระหว่างผลลัพธ์การอนุมานพีชชีและค่าระดับความสำคัญพลวัตของอินพุตแต่ละตัว ซึ่งสามารถแสดงความสัมพันธ์ได้ดังสมการที่ (4-8)

$$u = \sum_{i=1}^n \text{DID-}i \cdot u_i^0 \quad (4-8)$$

โดยที่  $n$  คือ จำนวนอินพุตทั้งหมดที่พิจารณา

อย่างไรก็ตามก่อนที่จะทำการควบคุมระบบ จำเป็นต้องปรับค่าสัญญาณควบคุมนี้ให้มีค่ามากพอ โดยการคูณด้วยตัวปรับค่าทางด้านเอาต์พุต (OSF) ดังนั้นสัญญาณควบคุมจริงที่กระทำกับระบบที่มีความไม่เป็นเชิงเส้น แสดงได้ดังสมการที่ (4-9)

$$U = u \cdot \text{OSF} \quad (4-9)$$

เมื่อสัญญาณควบคุมจริงกระทำกับระบบ จะได้สัญญาณเอาต์พุตควบคุม ซึ่งนำมาใช้เปรียบเทียบกับสัญญาณอินพุตที่ต้องการ โดยผลรวมของค่าความคลาดเคลื่อนของตัวแปรสถานะแต่ละตัวที่พิจารณาจะถูกนำไปใช้ในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม เพื่อปรับค่าระดับความสำคัญพลวัตแต่ละตัวให้เหมาะสมกับสถานการณ์ที่ทำการควบคุม กล่าวคือเมื่อสถานการณ์ที่ต้องการควบคุมเปลี่ยนแปลงไป โครงข่ายประสาทเทียมต้องสามารถเรียนรู้และปรับค่าระดับความสำคัญพลวัตให้แปรเปลี่ยนไปอย่างเหมาะสมด้วย ทั้งนี้เพื่อให้ตัวควบคุมนิวโร-พีชชียังคงสามารถควบคุมระบบต่อไปได้

#### 4.4 การออกแบบตัวควบคุมนิวโร-พีชชี

การออกแบบตัวควบคุมนิวโร-พีชชี เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวนั้น ต้องการให้ลูกตุ้มมีสมดุลโดยสามารถตั้งตรงขึ้นได้ที่ตำแหน่งศูนย์กลาง และตัวรถต้องสามารถเคลื่อนที่กลับมาหยุดนิ่งที่ตำแหน่งกึ่งกลางของรางลื่นในช่วงระยะเวลาอันสั้น กล่าวคือในการออกแบบตัวควบคุมต้องการให้ตัวแปรสถานะทุกตัวที่พิจารณามีค่าเข้าสู่ศูนย์ ซึ่งตัวแปรสถานะทั้งหมดที่พิจารณาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ได้แก่ มุมของลูกตุ้ม ความเร็วเชิงมุมของลูกตุ้ม ตำแหน่งของตัวรถ และความเร็วของตัวรถ โดยหลังจากที่ตัวแปรสถานะ

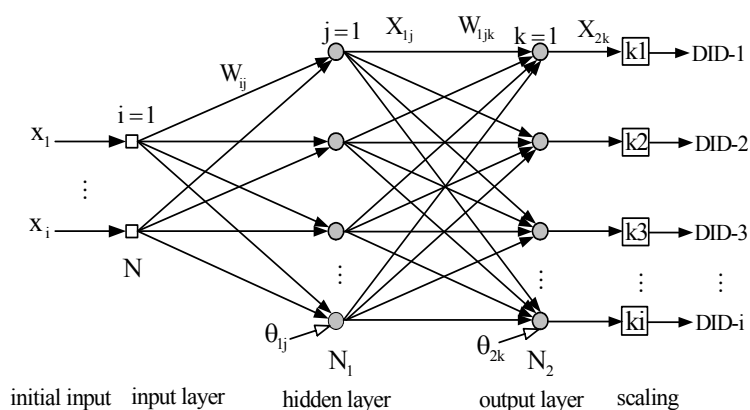
เหล่านี้ผ่านการนอร์มัลไลซ์ด้วยตัวปรับค่าที่กำหนดขึ้นสำหรับตัวแปรสถานะแต่ละตัวแล้ว จะได้ตัวแปรอินพุต  $x_i$  ( $i = 1, 2, 3, 4$ ) ซึ่งแทนด้วย มุมลูกตุ้ม ความเร็วเชิงมุมของลูกตุ้ม ตำแหน่งรถ และความเร็วยานตามลำดับโดยตัวแปรอินพุตเหล่านี้จะถูกใช้เป็นอินพุตให้กับกฎฟuzzy ซึ่งแต่ละกฎ จากนั้นโครงข่ายประสาทเทียมจะเรียนรู้ เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะกับตัวแปรอินพุตแต่ละตัว จนกระทั่งสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้

#### 4.4.1 การกำหนดค่า SIRM

การกำหนด SIRM สำหรับตัวแปรอินพุตแต่ละตัวนั้นจะพิจารณาให้กฎฟuzzy ซึ่งแต่ละกฎประกอบด้วยตัวแปรอินพุตเพียงตัวเดียวเท่านั้น โดยฟังก์ชันความเป็นสมาชิกสำหรับตัวแปรอินพุตแต่ละตัว และฟังก์ชันความเป็นสมาชิกของเอาต์พุตจะกำหนดให้มีลักษณะเช่นเดียวกับตัวควบคุมฟuzzy แบบกฎอินพุตเดียว รวมถึงการกำหนดคกฎฟuzzy ซึ่งแต่ละกฎ ซึ่งอาศัยการวิเคราะห์จากธรรมชาติของระบบลูกตุ้มผกผันแบบข้อต่อเดียวในการหาความสัมพันธ์ระหว่างตัวแปรอินพุตกับแรงเอาต์พุตที่กระทำกับตัวรถในแนวระดับ ซึ่งได้อธิบายรายละเอียดไว้แล้วในบทที่ 3

#### 4.4.2 การปรับค่าระดับความสำคัญพลวัตโดยใช้โครงข่ายประสาทเทียม

กลไกที่สำคัญในการสร้างเสถียรภาพให้กับระบบ คือ การควบคุมแรงที่กระทำกับตัวรถให้พอเหมาะ ซึ่งแรงดังกล่าวจะต้องมีความพอดีจึงจะสามารถสร้างเสถียรภาพให้ระบบได้ ดังนั้นหลังจากที่ได้ผลการอนุมานฟuzzy ของตัวแปรอินพุตแต่ละตัวแล้ว จำเป็นต้องบ่งชี้ความสำคัญของตัว



รูปที่ 4.5 โครงข่ายประสาทเทียมหลายชั้นแบบป้อนไปหน้า

แปรอินพุตเหล่านี้ด้วยค่าระดับความสำคัญพลวัต หรือ DID เพื่อให้ได้แรงกระทำที่สามารถสร้างเสถียรภาพให้กับระบบต่อไปได้ เมื่อสถานการณ์ในการควบคุมเปลี่ยนไป ค่าระดับความสำคัญพลวัตเหล่านี้ต้องสามารถปรับเปลี่ยนเพื่อรักษาเสถียรภาพให้แก่ระบบ วิธีการนี้จะอาศัยการเรียนรู้

จากโครงข่ายประสาทเทียมเพื่อปรับค่าระดับความสำคัญพลวัต โดยอัลกอริทึมที่ใช้ในกระบวนการเรียนรู้เป็นแบบแพร่กลับ แผนภาพแทนกระบวนการเรียนรู้ของโครงข่ายประสาทเทียมแสดงไว้ในรูปที่ 4.5 ที่เราอาจเรียกแผนภาพเช่นนี้ว่า โครงข่ายประสาทเทียมหลายชั้นแบบป้อนไปหน้า

โครงข่ายประสาทเทียมที่ใช้สำหรับสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน ได้รับการออกแบบให้มีชั้นซ่อนเร้นหนึ่งชั้นประกอบด้วยสี่เซลล์ประสาทเทียม โดยที่ชั้นอินพุตมีสองเซลล์ประสาทเทียมเพื่อรับสัญญาณอินพุตสองค่า ได้แก่ ค่ามุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถ ส่วนที่ชั้นเอาต์พุตประกอบด้วยสี่เซลล์ประสาทเทียม สำหรับค่าระดับความสำคัญพลวัตทั้งสี่ค่า โดยฟังก์ชันกระตุ้นที่ใช้เป็นแบบซิกมอยด์ เนื่องจากในขณะที่สร้างเสถียรภาพให้ระบบลูกตุ้มผกผันไม่สามารถระบุค่าระดับความสำคัญพลวัตที่ต้องการได้ ดังนั้นจะพิจารณาค่าความคลาดเคลื่อนที่แพร่กลับจากเอาต์พุตของระบบ (Cui and Shin, 1993) กระบวนการเรียนรู้ของโครงข่ายประสาทเทียมสามารถสรุปได้ 4 ขั้นตอน โดยพิจารณารูปที่ 4.5 ประกอบ ดังนี้คือ

ขั้นตอนที่ 1 สุ่มค่าพารามิเตอร์เริ่มต้นที่ใช้ในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม ได้แก่ ค่าน้ำหนักและค่าไบแอส โดยกำหนดให้มีค่าน้อยๆ อยู่ในช่วง  $[-1, +1]$

ขั้นตอนที่ 2 คำนวณเอาต์พุตจากฟังก์ชันกระตุ้นสำหรับชั้นซ่อนเร้นและชั้นเอาต์พุต

- เอาต์พุตของชั้นซ่อนเร้น;  $X_{1j}$

$$X_{1j}(t) = \frac{1}{1 + \exp(-O_{1j} - \theta_{1j})}$$

โดยที่

$$O_{1j} = \sum_{i=1}^N W_{ij} X_i, \quad j=1,2,\dots,N_1$$

- เอาต์พุตของชั้นเอาต์พุต;  $X_{2k}$

$$X_{2k}(t) = \frac{1}{1 + \exp(-O_{2k} - \theta_{2k})}$$

โดยที่

$$O_{2k} = \sum_{j=1}^{N_1} W_{1jk} X_{1j}(t), \quad k=1,2,\dots,N_2$$

ขั้นตอนที่ 3 ปรับค่าน้ำหนักและไบแอส

- ค่าน้ำหนักจากชั้นซ่อนเร้นไปชั้นเอาต์พุต;  $W_{1jk}$

$$W_{1jk}(t + \Delta t) = W_{1jk}(t) + \Delta W_{1jk}(t)$$

โดยที่

$$\Delta W_{1jk}(t) = \eta_1 \delta_{1k} X_{1j}(t) + \alpha \cdot \Delta W_{1jk}(t - \Delta t)$$

และ

$$\delta_{1k} = (X_{2kd}(t) - X_{2k}(t)) X_{2k}(t) (1 - X_{2k}(t)) \quad (4-10)$$

- ค่าน้ำหนักจากชั้นอินพุตไปชั้นซ่อนเร้น;  $W_{ij}$

$$W_{ij}(t + \Delta t) = W_{ij}(t) + \Delta W_{ij}(t)$$

โดยที่

$$\Delta W_{ij}(t) = \eta_j \delta_j X_i(t) + \alpha \cdot \Delta W_{ij}(t - \Delta t)$$

และ

$$\delta_j = \left[ \sum_{k=1}^{N_2} \delta_{1k} W_{1jk}(t + \Delta t) \right] X_{1j}(t) (1 - X_{1j}(t))$$

- ค่าไบแอสที่ชั้นเอาต์พุต;  $\theta_{2k}$  และค่าไบแอสที่ชั้นซ่อนเร้น;  $\theta_{1j}$

$$\theta_{2k}(t + \Delta t) = \theta_{2k}(t) + \eta_{1\theta} \delta_{1k}$$

$$\theta_{1j}(t + \Delta t) = \theta_{1j}(t) + \eta_{\theta} \delta_j$$



ขั้นตอนที่ 4 ปรับเพิ่มจำนวนรอบการเรียนรู้ และวนรอบไปยังขั้นตอนที่ 2 ในกรณีที่ ไม่ตรงตามเงื่อนไขที่กำหนดไว้สำหรับยุคกระบวนการเรียนรู้

โดยที่  $\eta$ ,  $\eta_1$ ,  $\eta_0$  และ  $\eta_{10}$  คือค่าอัตราการเรียนรู้ และ  $\alpha$  คือค่าคงที่โมเมนตัม เมื่อพิจารณารูปที่ 4.5 พบว่าเอาต์พุตที่ออกจากโครงข่ายประสาทเทียมคือ  $X_{2k}$  จะมีค่าอยู่ในช่วง  $[0, 1]$  เนื่องจากฟังก์ชันกระตุ้นที่ใช้เป็นแบบซิกมอยด์ ดังนั้นจะต้องคูณค่านี้ด้วยตัวปรับค่า (scaling) ซึ่งกำหนดให้ตัวปรับค่าทุกตัวมีค่าเป็น 10 เพื่อยกระดับให้ได้ค่าระดับความสำคัญพลวัตที่สามารถสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยอยู่ในช่วง  $[0, 10]$  นอกจากนี้เมื่อพิจารณาสมการที่ (4-10) จะพบว่าค่า  $X_{2kd}$  ซึ่งเป็นเอาต์พุตที่ต้องการของโครงข่ายประสาทเทียมไม่สามารถทราบค่าได้ ดังนั้น  $X_{2kd}$  ในสมการดังกล่าวต้องถูกแทนที่ด้วยตัวแปรอื่นที่ทราบค่า ในวิทยานิพนธ์นี้ได้นำเสนอตัวแปรเอาต์พุตของระบบมาใช้งานดังความสัมพันธ์ในสมการที่ (4-11) ซึ่งปรับปรุงมาจากแนวทางของ (Cui and Shin, 1993)

$$(X_{2kd}(t) - X_{2k}(t)) = (y_{kd}(t) - y_k(t)) \cdot D \quad (4-11)$$

เมื่อแทนค่า  $(X_{2kd}(t) - X_{2k}(t))$  ในสมการที่ (4-11) ลงในสมการที่ (4-10) จะได้ว่า

$$\delta_{lk} = (y_{kd}(t) - y_k(t)) \cdot D \cdot X_{2k}(t)(1 - X_{2k}(t)) \quad (4-12)$$

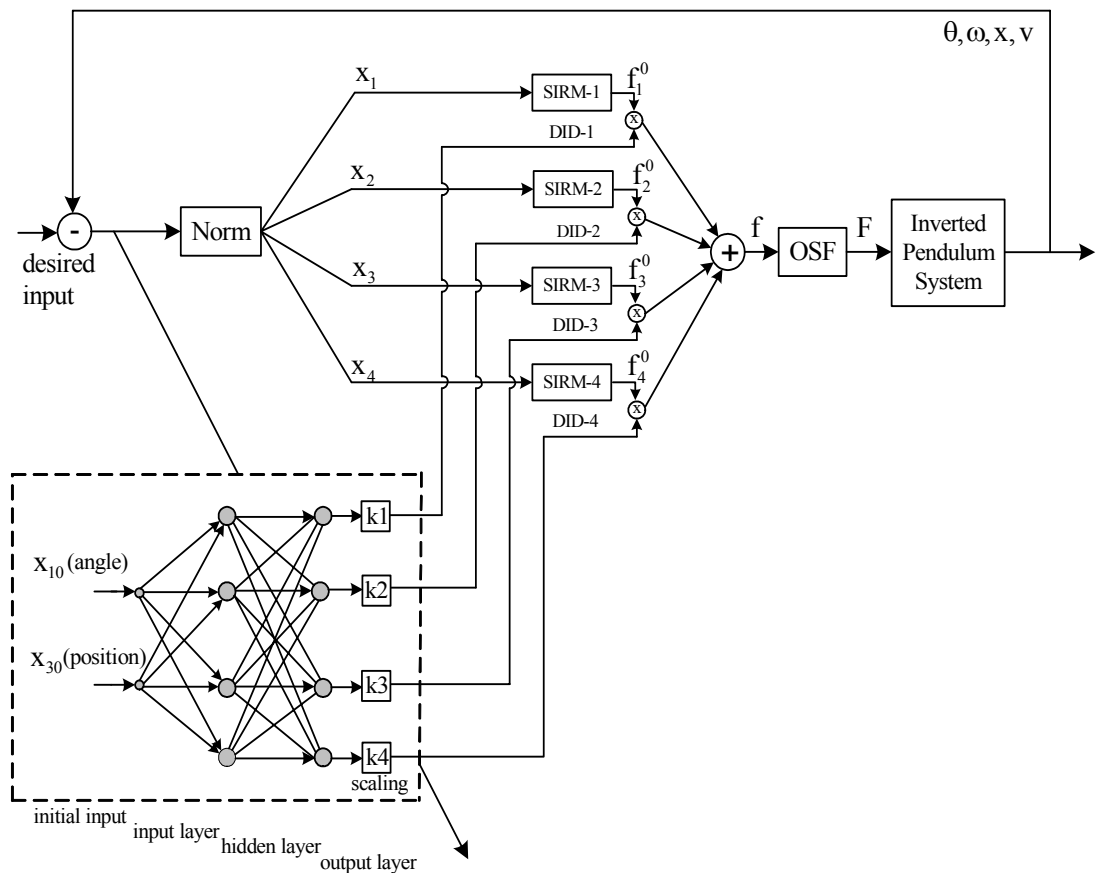
โดยที่

$$D = \text{sign} \left( \frac{\partial y_k(t)}{\partial X_{2k}(t)} \right)$$

และ  $y_{kd}$  คือ ค่าเอาต์พุตที่ต้องการของระบบ

#### 4.5 การทำงานของตัวควบคุมนิวโร-ฟิซซี

การสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟิซซี มีโครงสร้างดังแสดงด้วยแผนภาพในรูปที่ 4.6 ซึ่งตัวแปรสถานะของระบบลูกตุ้มแบบข้อต่อเดียวที่พิจารณามี 4 ตัว ได้แก่ มุมลูกตุ้ม ( $\theta$ ) ความเร็วเชิงมุมของลูกตุ้ม ( $\omega$ ) ตำแหน่งของตัวรถ ( $x$ ) และความเร็วของตัวรถ ( $v$ ) ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันนั้นต้องการให้ลูกตุ้มตั้งตรงที่ตำแหน่งศูนย์องศา และตัวรถเคลื่อนที่กลับมาหยุดที่จุดกึ่งกลางของรางเดิน ดังนั้นจึงต้องทำการควบคุมเพื่อให้ตัวแปรสถานะทุกตัวมีค่าเป็นศูนย์ทั้งหมด



รูปที่ 4.6 โครงสร้างของตัวควบคุมนิวโร-ฟัซซีเพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน

เมื่อพิจารณารูปที่ 4.6 จะสังเกตเห็นว่าค่าตัวแปรสถานะเหล่านี้จะถูกป้อนกลับ เพื่อเปรียบเทียบกับค่าที่ต้องการ โดยค่าความคลาดเคลื่อนที่ได้จากการเปรียบเทียบดังกล่าวจะเป็นอินพุตป้อนเข้าสู่บล็อกนอร์ม (norm) เพื่อทำการปรับค่าตัวแปรสถานะแต่ละตัวด้วยตัวปรับค่าอินพุต (input scaling factor) ซึ่งกำหนดให้ตัวปรับค่าสำหรับมุมลูกตุ้ม ความเร็วเชิงมุมของลูกตุ้ม ตำแหน่งของตัวรถ และความเร็วของตัวรถ มีค่าเป็น 30 องศา 100 องศาต่อวินาที 2.4 เมตร และ 1 เมตรต่อวินาที ตามลำดับ จากนั้นตัวแปรสถานะเหล่านี้จะถูกแปลงเป็นอินพุต  $x_i$  ( $i = 1, 2, 3, 4$ ) เพื่อป้อนเข้าสู่ส่วนของ SIRM- $i$  หลังจากทำการอนุมานฟัซซีแล้วจะได้  $f_i^0$  เพื่อคูณกับค่า DID- $i$  ซึ่งค่าระดับความสำคัญพลวัตแต่ละตัวได้มาจากการเรียนรู้ของโครงข่ายประสาทเทียม โดยที่อินพุตของโครงข่ายประสาทเทียม คือ มุมเริ่มต้นของลูกตุ้มและตำแหน่งเริ่มต้นของตัวรถ โครงข่ายประสาทเทียมจะเรียนรู้โดยการปรับค่าน้ำหนักและค่าไบแอสจากค่าความคลาดเคลื่อนเอาต์พุตของระบบ จนกระทั่งสามารถสร้างเสถียรภาพให้กับระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ ซึ่งเอาต์พุต  $f$  เกิดจากผลรวมของผลคูณระหว่าง SIRM และ DID ของอินพุตแต่ละตัว แต่ก่อนจะเป็นแรงจริงที่

กระทำกับระบบลูกตุ้มผกผันต้องทำการปรับค่าด้วยตัวปรับค่าของเอาต์พุต (OSF) ซึ่งกำหนดให้มีขนาดเป็นสิบเท่าของผลรวมระหว่างมวลของตัวรถและมวลของก้านลูกตุ้มผกผัน ดังแสดงด้วยสมการที่ (4-13)

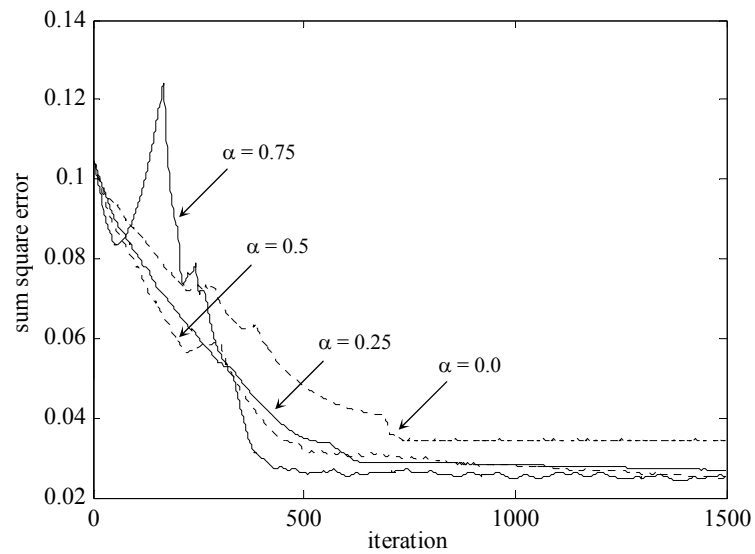
$$\text{OSF} = 10 \cdot (m_c + m_p) \quad (4-13)$$

#### 4.6 การทดสอบเพื่อเลือกค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม

เนื่องจากค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม เป็นพารามิเตอร์ที่สำคัญในกระบวนการเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม ซึ่งใช้ในการคำนวณเพื่อปรับค่าน้ำหนักและค่าไบแอส ทำให้มีผลต่อการเรียนรู้และการลดลงของค่าความคลาดเคลื่อน ดังนั้นในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวด้วยตัวควบคุมนิวโร-ฟัซซี่ จำเป็นต้องกำหนดค่าพารามิเตอร์เหล่านี้ให้เหมาะสม ในการทดสอบเพื่อเลือกค่าอัตราการเรียนรู้ได้กำหนดให้  $\eta$ ,  $\eta_1$ ,  $\eta_0$  และ  $\eta_{10}$  มีค่าเท่ากันทุกตัว โดยแทนด้วยตัวแปร  $\eta$  กล่าวคือ ดำเนินการทดสอบที่ค่า  $\eta = 0.05$ ,  $\eta = 0.1$ ,  $\eta = 0.25$  และ  $\eta = 0.5$  ตามลำดับ รวมทั้งทำการปรับเปลี่ยนค่าคงที่โมเมนตัม จนกระทั่งได้ค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัมที่เหมาะสมสำหรับกระบวนการเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม

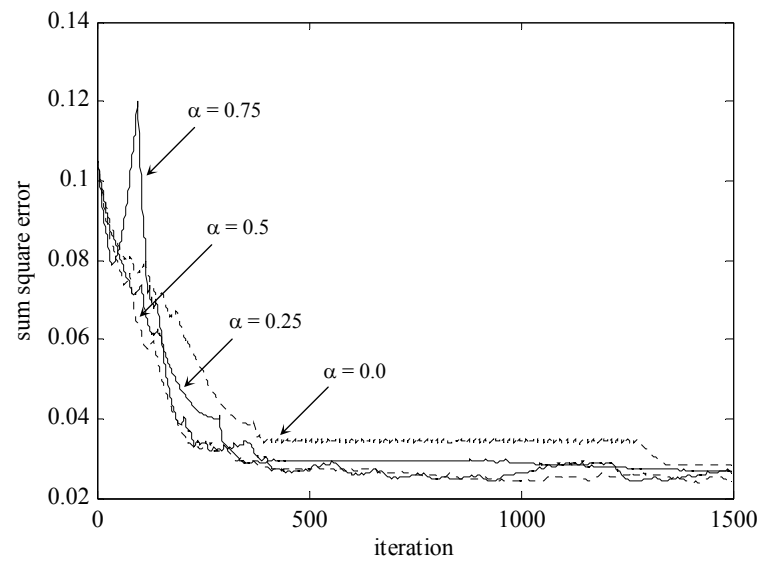
ในการจำลองสถานการณ์เพื่อเลือกค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม ดำเนินการทดสอบเมื่อมุมเริ่มต้นของลูกตุ้มมีค่าเป็น 30.0 องศา และก้านลูกตุ้มยาว 1.0 เมตร โดยที่ตำแหน่งเริ่มต้นของตัวรถอยู่ที่กึ่งกลางของรางลื่น นั่นคือที่ระยะ 0.0 เมตร ซึ่งค่ามุมเริ่มต้นและตำแหน่งเริ่มต้นของรถ จะเป็นสัญญาณอินพุตให้กับชั้นอินพุตของโครงข่ายประสาทเทียม กำหนดให้จำนวนรอบที่ใช้ในกระบวนการเรียนรู้คือ 1500 รอบ ทั้งนี้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมซึ่งได้แก่ ค่าน้ำหนักและค่าไบแอส ได้จากการสุ่มค่าเริ่มต้นชุดเดียวกัน ผลการทดสอบเลือกค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม แสดงได้ดังรูปที่ 4.7, 4.8, 4.9 และ 4.10

เมื่อพิจารณารูปที่ 4.7 ซึ่งเป็นการทดสอบ เมื่อค่าอัตราการเรียนรู้มีค่าน้อยๆ คือ  $\eta = 0.05$  พบว่า ในกรณีที่ค่าคงที่โมเมนตัมมีค่าเป็นศูนย์ การปรับค่าน้ำหนักและค่าไบแอสของโครงข่ายประสาทเทียมจะขึ้นอยู่กับค่าอัตราการเรียนรู้เพียงค่าเดียวเท่านั้น ซึ่งจะสังเกตเห็นว่าเมื่อ  $\alpha = 0.0$  การลดลงของค่าความคลาดเคลื่อนมีลักษณะราบเรียบและลดลงอย่างช้าๆ จนกระทั่งคงที่ จากนั้นดำเนินการปรับเปลี่ยนค่าคงที่โมเมนตัมให้เพิ่มขึ้นเป็น  $\alpha = 0.25$ ,  $\alpha = 0.5$  และ  $\alpha = 0.75$  ตามลำดับ ผลการทดสอบแสดงให้เห็นว่า เมื่อค่าคงที่โมเมนตัมมีค่าสูงขึ้นลักษณะการลดลงของค่าความคลาดเคลื่อนจะเป็นไปอย่างรวดเร็ว แต่จะมีการแกว่งกวัดเกิดขึ้นมาก โดยเฉพาะอย่างยิ่งที่  $\alpha = 0.75$



รูปที่ 4.7 อัตราการเรียนรู้เมื่อ  $\eta = 0.05$

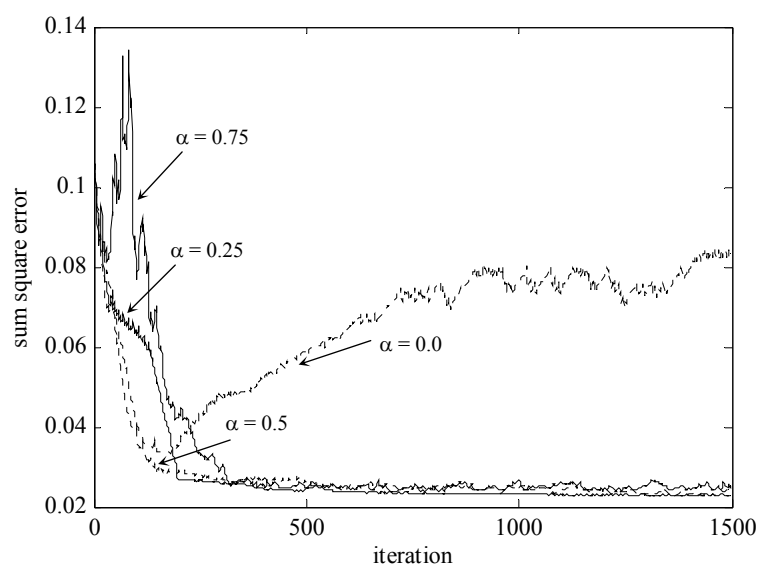
ซึ่งค่าความคลาดเคลื่อนมีค่าสูงขึ้นในช่วงแรก แต่เมื่อจำนวนรอบในการเรียนรู้เพิ่มขึ้น ค่าความคลาดเคลื่อนจึงลดลงและใกล้เคียงกับกรณี  $\alpha = 0.25$  และ  $\alpha = 0.5$  ดังนั้น จากผลการทดสอบพบว่า ค่าคงที่โมเมนต์ที่ดีที่สุดสำหรับค่าอัตราการเรียนรู้  $\eta = 0.05$  คือ  $\alpha = 0.25$  เนื่องจากการลดลงของค่าความคลาดเคลื่อนยังคงราบเรียบไม่มีการแกว่งกวัดเกิดขึ้น



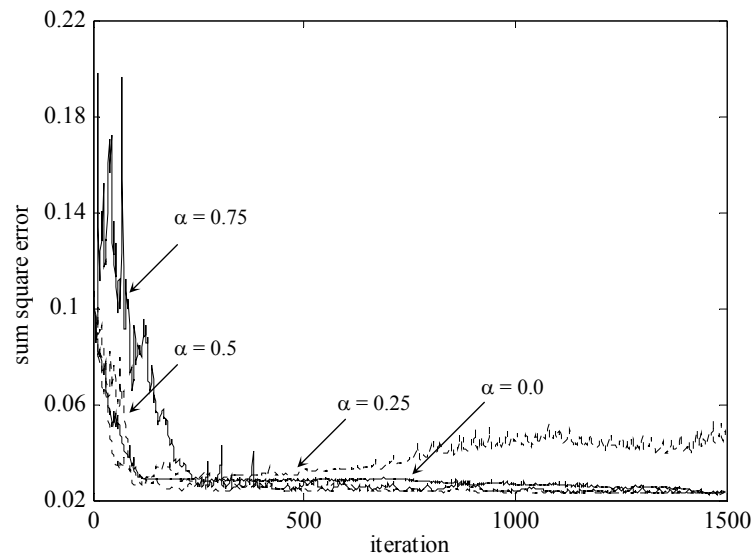
รูปที่ 4.8 อัตราการเรียนรู้เมื่อ  $\eta = 0.10$

ผลการทดสอบในรูปที่ 4.8 เป็นกรณี เมื่อค่าอัตราการเรียนรู้มีค่าเพิ่มขึ้นเป็น 0.10 หลังจากที่ได้ดำเนินการปรับเปลี่ยนค่าคงที่โมเมนตัม พบว่า การลดลงของค่าความคลาดเคลื่อนยังคงมีการแกว่งกวัดเกิดขึ้น โดยเฉพาะอย่างยิ่งเมื่อค่าคงที่โมเมนตัมมีค่าเพิ่มขึ้น อย่างไรก็ตาม จากผลการทดสอบจะสังเกตเห็นว่า ในกรณีที่ค่าคงที่โมเมนตัมมีค่าเป็นศูนย์ จะมีการแกว่งกวัดเล็กน้อยเกิดขึ้นอยู่ช่วงหนึ่ง แต่หลังจากที่จำนวนรอบในการเรียนรู้เพิ่มขึ้น การแกว่งกวัดนี้จะหายไปรวมทั้งค่าความคลาดเคลื่อนก็ลดลงด้วย นอกจากนี้จะพบว่าหลังจากโครงข่ายประสาทเทียมเรียนรู้จนกระทั่งครบ 1500 รอบแล้ว ค่าความคลาดเคลื่อนที่ได้จากค่าคงที่โมเมนตัมทุกตัวมีค่าใกล้เคียงกัน ดังนั้นเมื่อพิจารณาผลการทดสอบเพื่อเลือกค่าคงที่โมเมนตัมที่ดีที่สุด พบว่า ค่าคงที่โมเมนตัม  $\alpha = 0.25$  ให้ผลการทดสอบที่ดีที่สุด แม้ว่าจะมีการแกว่งกวัดเกิดขึ้นบ้างแต่ก็ไม่มากนักเมื่อเปรียบเทียบกับค่าคงที่โมเมนตัมค่าอื่น

เมื่อพิจารณารูปที่ 4.9 และ 4.10 พบว่าผลการทดสอบเป็นไปในแนวทางเดียวกัน กล่าวคือ เมื่อค่าอัตราการเรียนรู้มีค่าสูงขึ้น การลดลงของค่าความคลาดเคลื่อนจะมีการแกว่งกวัดเกิดขึ้นมาก โดยเฉพาะอย่างยิ่งเมื่อค่าคงที่โมเมนตัมมีค่าเพิ่มขึ้น แต่หลังจากที่โครงข่ายประสาทเทียมเรียนรู้จนครบจำนวนรอบที่กำหนดแล้ว ค่าความคลาดเคลื่อนดังกล่าวจะมีค่าใกล้เคียงกัน นอกจากนี้จะสังเกตเห็นว่าค่าคงที่โมเมนตัมบางค่าจะส่งผลให้ค่าความคลาดเคลื่อนเพิ่มสูงขึ้นได้ในระหว่างการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่งเหตุการณ์เช่นนี้อาจทำให้ได้ค่าระดับความสำคัญพลวัตที่ไม่สามารถสร้างเสถียรภาพให้ระบบลูกคุ่มผกผันแบบข้อต่อเดียวได้

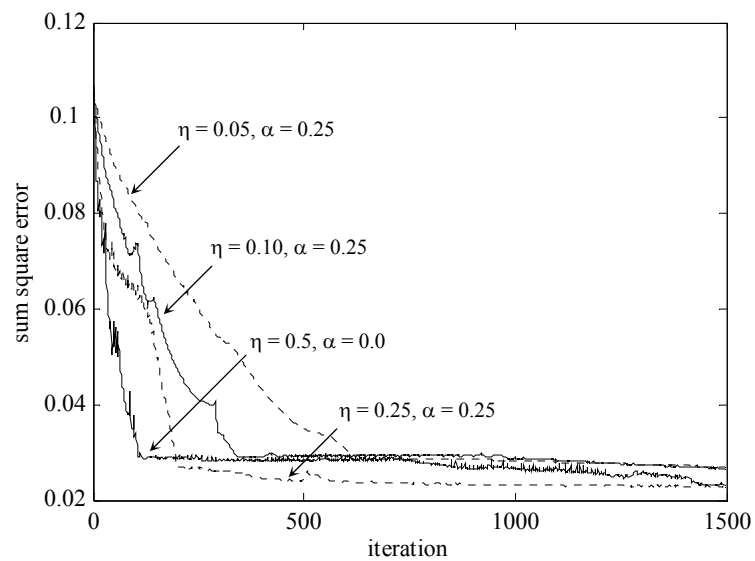


รูปที่ 4.9 อัตราการเรียนรู้เมื่อ  $\eta = 0.25$



รูปที่ 4.10 อัตราการเรียนรู้เมื่อ  $\eta = 0.50$

รูปที่ 4.11 เป็นการนำผลทดสอบที่ให้ผลดีที่สุดจากรูปที่ 4.7, 4.8, 4.9, 4.10 มาเปรียบเทียบกันอีกครั้ง เพื่อพิจารณาเลือกค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัมที่เหมาะสมสำหรับการเรียนรู้ของโครงข่ายประสาทเทียม จากผลการทดสอบแสดงให้เห็นว่า เมื่อค่าอัตราการเรียนรู้มีค่าเพิ่มขึ้น การแกว่งกวัดของค่าความคลาดเคลื่อนจะมากขึ้น โดยในช่วงแรกค่าความคลาดเคลื่อนดังกล่าวจะลดลง



รูปที่ 4.11 ผลการทดสอบค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม

อย่างรวดเร็ว นอกจากนี้จะเห็นว่า เมื่อสิ้นสุดกระบวนการเรียนรู้ค่าความคลาดเคลื่อนที่ได้จากค่า อัตราการเรียนรู้และค่าคงที่โมเมนตัมแต่ละค่าจะใกล้เคียงกัน ดังนั้นจึงควรเลือกค่าอัตราการเรียนรู้ และค่าคงที่โมเมนตัมที่ทำให้การลดลงของค่าความคลาดเคลื่อนมีความราบเรียบมากที่สุด นั่นคือ ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวด้วยตัวควบคุมนิวโร-ฟัซซี่ กำหนดให้ ค่าอัตราการเรียนรู้  $\eta = 0.05$  และค่าคงที่โมเมนตัม  $\alpha = 0.25$

#### 4.7 การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

เพื่อทดสอบประสิทธิผลของตัวควบคุมนิวโร-ฟัซซี่ ที่พัฒนาขึ้น ได้ดำเนินการจำลอง สถานการณ์เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยใช้แบบจำลองทาง คณิตศาสตร์และพารามิเตอร์ของระบบลูกตุ้มผกผัน เช่นเดียวกันกับกรณีใช้ตัวควบคุมฟัซซี่แบบกฎ อินพุตเดียว โดยระบบลูกตุ้มผกผันจะได้รับการพิจารณาว่าเข้าสู่เสถียรภาพ เมื่อตัวแปรสถานะที่ใช้ ในการควบคุมเป็นไปตามข้อกำหนดต่อไปนี้ คือ มุมลูกตุ้มเข้าสู่  $\pm 0.1$  องศา ความเร็วเชิงมุมของ ลูกตุ้มเข้าสู่  $\pm 0.1$  องศาต่อวินาที ตำแหน่งรถเข้าสู่  $\pm 0.01$  เมตร และความเร็วของรถเข้าสู่  $\pm 0.01$  เมตรต่อวินาที ในการทดสอบการสร้างเสถียรภาพให้ระบบตัวควบคุมนิวโร-ฟัซซี่ ได้ดำเนินการ จำลองสถานการณ์เมื่อปรับเปลี่ยนสถานการณ์ที่ควบคุม ตัวอย่างเช่น ปรับเปลี่ยนความยาวของก้าน ลูกตุ้ม ตำแหน่งเริ่มต้นของตัวรถ และมุมเริ่มต้นของลูกตุ้ม เป็นต้น

โครงข่ายประสาทเทียมที่ใช้ในกระบวนการเรียนรู้ มีลักษณะเป็น โครงข่ายหลายชั้นแบบ ป้อนไปหน้า ประกอบด้วยชั้นซ่อนเร้นเพียงชั้นเดียว โดยในชั้นอินพุตมี 2 เซลล์ประสาทเทียม สำหรับรับสัญญาณอินพุต ซึ่งกำหนดให้เป็นค่ามุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถ เข้าสู่โครงข่ายประสาทเทียม ส่วนชั้นซ่อนเร้นและชั้นเอาต์พุตประกอบด้วย 4 เซลล์ประสาทเทียม สำหรับเรียนรู้เพื่อปรับค่าระดับความสำคัญพลวัตทั้ง 4 ค่า ดังนั้น ในโครงข่ายประสาทเทียมจะ ประกอบด้วยค่าพารามิเตอร์เริ่มต้น ซึ่งได้แก่ ค่าน้ำหนัก และค่าไบแอส รวมทั้งสิ้น 32 ค่า ดังแสดง ในตารางที่ 4.2 ซึ่งค่าพารามิเตอร์เริ่มต้นเหล่านี้ได้มาจากการสุ่มค่าโดยมีขอบเขตอยู่ใน  $[-1, +1]$  เมื่อพิจารณาค่าพารามิเตอร์เริ่มต้นที่ปรากฏในตารางที่ 4.2 จะได้ว่า  $W_{11}$  หมายถึง ค่าน้ำหนักที่ เชื่อมต่อระหว่างเซลล์ประสาทเทียมที่ 1 ของชั้นอินพุต ไปยังเซลล์ประสาทเทียมที่ 1 ของชั้นซ่อนเร้น ในขณะที่  $W_{111}$  แทนด้วย ค่าน้ำหนักที่เชื่อมต่อระหว่างเซลล์ประสาทเทียมที่ 1 ของชั้นซ่อนเร้น ไปยังเซลล์ประสาทเทียมที่ 1 ของชั้นเอาต์พุต ส่วน  $\theta_{11}$  หมายถึง ค่าไบแอสสำหรับเซลล์ประสาทเทียม ที่ 1 ของชั้นซ่อนเร้น และ  $\theta_{21}$  คือ ค่าไบแอสสำหรับเซลล์ประสาทเทียมที่ 1 ของชั้นเอาต์พุต โดยใน การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ จะใช้ค่าพารามิเตอร์ เริ่มต้นเหล่านี้เพียงชุดเดียวตลอดไป แม้ว่าปรับเปลี่ยนสถานการณ์ที่ควบคุม

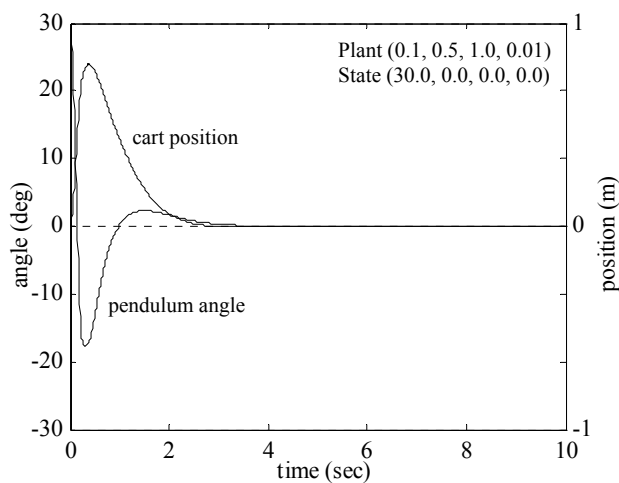
ตารางที่ 4.2 พารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการสุ่มค่า

ค่าที่	พารามิเตอร์	ค่าที่สุ่มได้	ค่าที่	พารามิเตอร์	ค่าที่สุ่มได้
1	$W_{11}$	-0.83093	17	$W_{121}$	-0.93860
2	$W_{12}$	0.91589	18	$W_{122}$	-0.26511
3	$W_{13}$	0.30674	19	$W_{123}$	0.55187
4	$W_{14}$	0.84039	20	$W_{124}$	-0.36729
5	$W_{21}$	0.22568	21	$W_{131}$	-0.45940
6	$W_{22}$	-0.50487	22	$W_{132}$	0.76574
7	$W_{23}$	0.55065	23	$W_{133}$	-0.92248
8	$W_{24}$	0.61486	24	$W_{134}$	0.75701
9	$\theta_{11}$	0.53551	25	$W_{141}$	0.26609
10	$\theta_{12}$	0.39366	26	$W_{142}$	-0.90399
11	$\theta_{13}$	0.09812	27	$W_{143}$	0.01846
12	$\theta_{14}$	0.26164	28	$W_{144}$	-0.95105
13	$W_{111}$	0.52367	29	$\theta_{21}$	0.86425
14	$W_{112}$	0.64592	30	$\theta_{22}$	0.36119
15	$W_{113}$	0.25419	31	$\theta_{23}$	-0.87970
16	$W_{114}$	-0.23637	32	$\theta_{24}$	-0.66771

รูปที่ 4.12 เป็นผลการจำลองสถานการณ์ เมื่อมุมเริ่มต้นของลูกตุ้มเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถอยู่ตรงจุดกึ่งกลางของรางเส้น โดยที่ก้านลูกตุ้มยาว 1.0 เมตร ทั้งนี้กำหนดให้ตัวแปรอินพุตอื่นๆ มีค่าเป็นศูนย์ เมื่อพิจารณาผลการจำลองสถานการณ์ที่ได้พบว่า เนื่องจากมุมเริ่มต้นมีค่ามากในทิศทางบวก ดังนั้นตัวรถจะเคลื่อนที่ออกจากจุดกึ่งกลางของรางเส้นไปทางด้านขวา ทำให้ลูกตุ้มหมุนกลับในทิศทวนเข็มนาฬิกา ส่งผลให้มุมของลูกตุ้มมีค่าเป็นบวก ลดลงจนกระทั่งมุมลูกตุ้มมีทิศทางลบ จากนั้นตัวรถจะเคลื่อนที่กลับมาทางด้านซ้ายเพื่อทำให้ลูกตุ้มหมุนกลับในทิศตามเข็มนาฬิกา จนกระทั่งก้านลูกตุ้มตั้งตรงที่ตำแหน่งสมดุลและตัวรถเคลื่อนที่กลับมายังกึ่งกลางของรางเส้นได้อีกครั้ง ซึ่งตัวควบคุมนิวโร-ฟัซซีใช้เวลาในการสร้างเสถียรภาพให้แก่ระบบประมาณ 3.95 วินาที

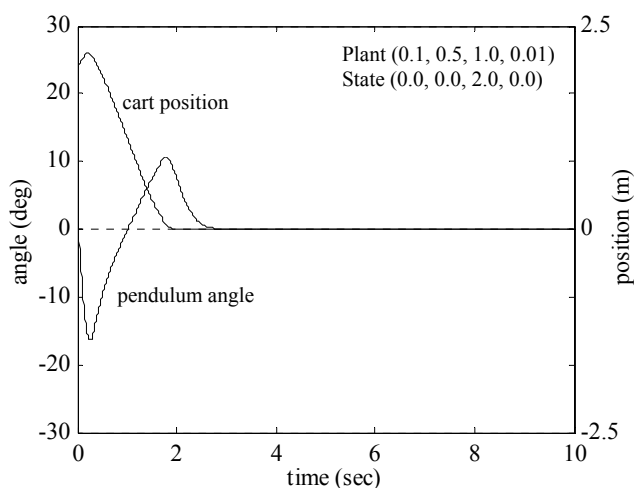
รูปที่ 4.13 แสดงผลการจำลองสถานการณ์ เมื่อตำแหน่งเริ่มต้นของรถไม่อยู่ที่กึ่งกลางของรางเส้น กล่าวคือ กำหนดให้ตำแหน่งเริ่มต้นของตัวรถอยู่ห่างจากจุดกึ่งกลางของรางเส้นไปทางขวาเป็นระยะทาง 2 เมตร โดยที่มุมเริ่มต้นของลูกตุ้มผกผันเป็นศูนย์องศา นั่นคือ ก้านลูกตุ้มอยู่ในสภาวะ





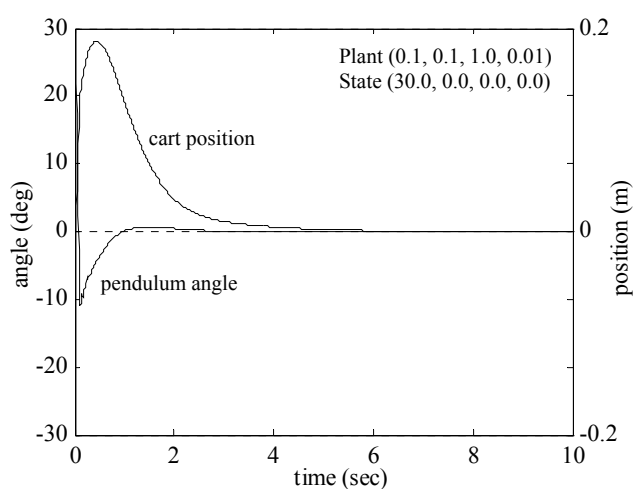
รูปที่ 4.12 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 1.0 เมตร

ตั้งตรงที่ตำแหน่งสมดุล ทั้งนี้ก้านลูกตุ้มยาว 1.0 เมตร เมื่อพิจารณาผลการจำลองสถานการณ์ที่ปรากฏในรูปที่ 4.13 พบว่า ตัวรถจะเคลื่อนที่ไปทางด้านขวาเล็กน้อย เพื่อให้มุมลูกตุ้มเสียสมดุลหมุนกลับมาในทิศทางเข็มนาฬิกา ส่งผลให้มุมของลูกตุ้มมีค่าเป็นลบ จากนั้นตัวรถจะเคลื่อนที่กลับทางด้านซ้ายมายังตำแหน่งกึ่งกลางของรางลื่น พร้อมทั้งทำให้ลูกตุ้มหมุนกลับในทิศทางตามเข็มนาฬิกา ส่งผลให้มุมของลูกตุ้มมีค่าเป็นบวกใกล้เคียงกับตำแหน่งสมดุล ซึ่งตัวรถจะเคลื่อนที่เพียงเล็กน้อย จนกระทั่งลูกตุ้มสามารถตั้งตรงได้อีกครั้ง พร้อมกันนี้ตัวรถก็สามารถกลับมายังกึ่งกลางของรางลื่น ซึ่งใช้เวลาในการเข้าสู่เสถียรภาพประมาณ 3.16 วินาที

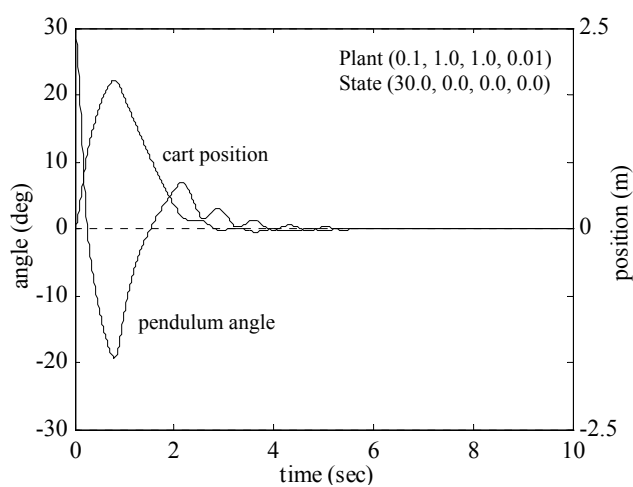


รูปที่ 4.13 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร

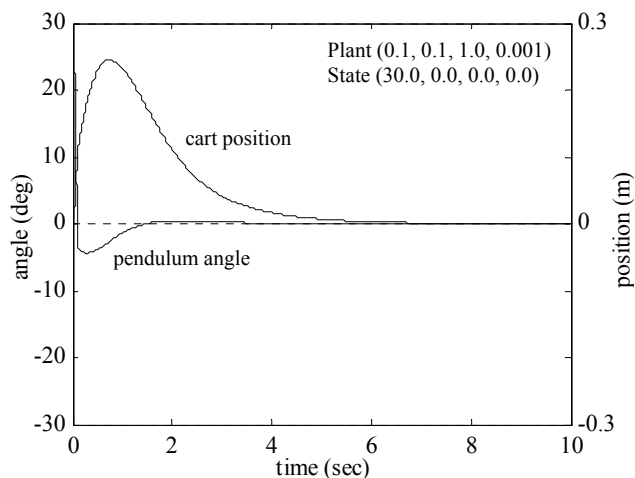
รูปที่ 4.14 และ 4.15 เป็นผลการจำลองสถานการณ์เมื่อทำการปรับเปลี่ยนความยาวของก้านลูกตุ้มผกผันให้สั้นและยาวขึ้นกว่าเดิมคือ 0.2 เมตร และ 2.0 เมตร ตามลำดับ โดยที่มุมเริ่มต้นของลูกตุ้มยังคงเป็น 30 องศา และตัวรถอยู่ที่ตำแหน่งกึ่งกลางของรางเดิน พบว่าผลการจำลองสถานการณ์ที่ได้ยังคงเป็นไปในแนวทางเดียวกันกับกรณีเมื่อก้านลูกตุ้มยาว 1.0 เมตร โดยใช้เวลาในการสร้างเสถียรภาพให้ระบบประมาณ 3.19 และ 9.20 วินาที ตามลำดับ อย่างไรก็ตาม เมื่อพิจารณารูปที่ 4.15 ซึ่งเป็นกรณีที่ก้านลูกตุ้มมีความยาวเพิ่มขึ้น จะสังเกตเห็นว่า มุมของลูกตุ้มและตำแหน่งของตัวรถจะมีการแกว่งกวัดเกิดขึ้นเล็กน้อยก่อนเข้าสู่เสถียรภาพ



รูปที่ 4.14 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 0.2 เมตร

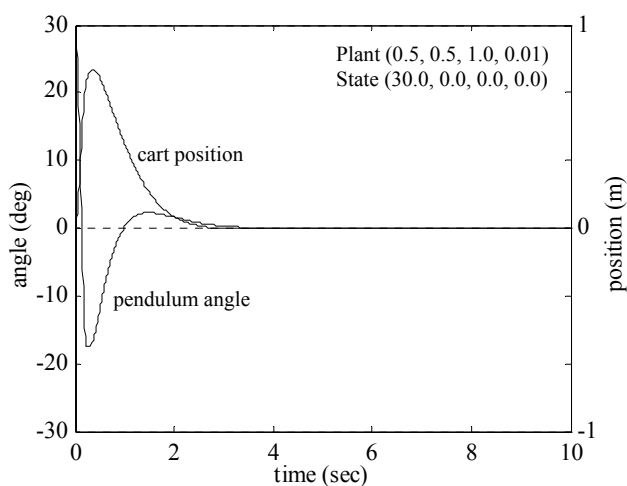


รูปที่ 4.15 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 2.0 เมตร



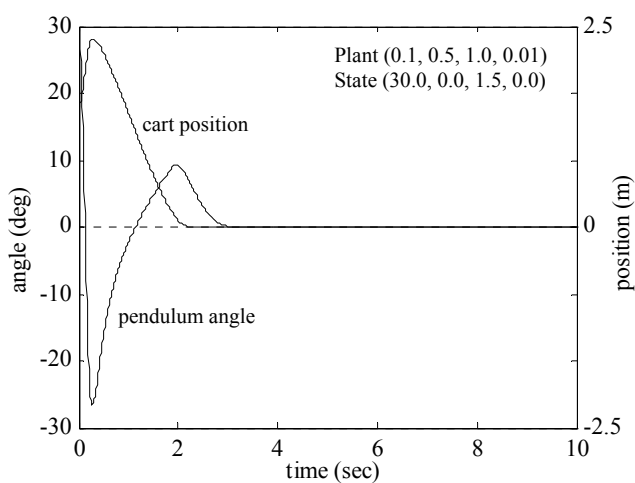
รูปที่ 4.16 ผลตอบสนองเมื่อปรับเปลี่ยนช่วงเวลาแซมปลิงเป็น 0.001 วินาที

รูปที่ 4.16 เป็นผลการจำลองสถานการณ์ เมื่อกำหนดเงื่อนไขในการจำลองสถานการณ์ เช่นเดียวกับรูปที่ 4.14 กล่าวคือ มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถอยู่ที่กึ่งกลางของรางเลื่อน และก้านลูกตุ้มยาว 0.2 เมตร แต่ได้ปรับเปลี่ยนช่วงเวลาแซมปลิงในการจำลองสถานการณ์เป็น 0.001 วินาที ซึ่งพบว่า ตัวควบคุมนิวโร-ฟิซใช้เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวประมาณ 4.570 วินาที และผลการจำลองสถานการณ์ที่ได้ยังคงเป็นไปในแนวทางเดียวกันกับกรณีที่ช่วงเวลาแซมปลิงมีค่าเป็น 0.01 วินาที ซึ่งผลการจำลองสถานการณ์ดังกล่าวปรากฏดังรูปที่ 4.14



รูปที่ 4.17 ผลตอบสนองเมื่อปรับเพิ่มมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม

สถานการณ์ต่อมาเป็นการจำลองสถานการณ์ เมื่อปรับเพิ่มมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม โดยมุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร และก้านลูกตุ้มยาว 1.0 เมตร ผลการจำลองสถานการณ์ดังกล่าวแสดงได้ดังรูปที่ 4.17 ซึ่งใช้เวลาในการสร้างเสถียรภาพให้ระบบประมาณ 3.92 วินาที เมื่อพิจารณาผลการจำลองสถานการณ์ที่ได้เปรียบเทียบกับกรณีที่มีมวลของก้านลูกตุ้มเป็น 0.1 กิโลกรัม ซึ่งแสดงผลการจำลองสถานการณ์ได้ดังรูปที่ 4.12 พบว่า ผลการจำลองสถานการณ์ทั้งสองกรณีมีลักษณะที่คล้ายคลึงกัน รวมทั้งสามารถเข้าสู่เสถียรภาพได้ภายในระยะเวลาที่ใกล้เคียงกัน

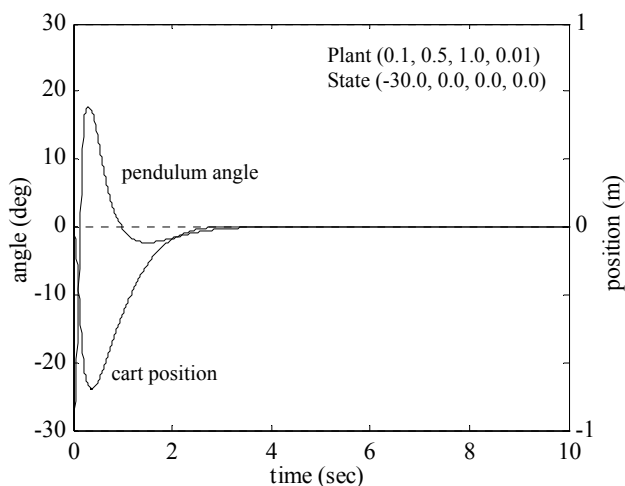


รูปที่ 4.18 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร

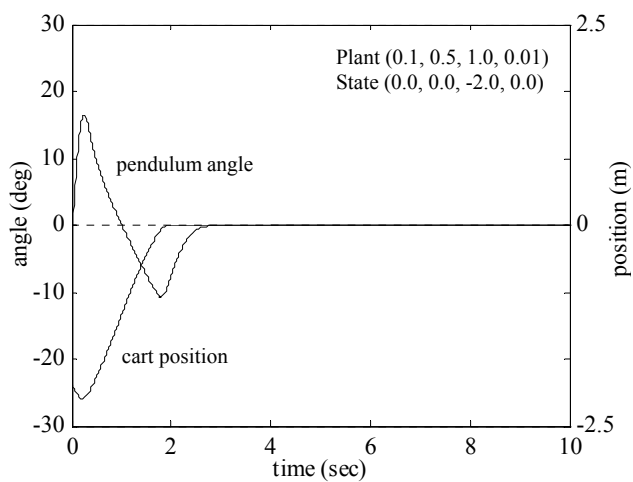
รูปที่ 4.18 เป็นผลจำลองสถานการณ์ เมื่อกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถอยู่ห่างจากจุดกึ่งกลางของรางเลื่อนไปทางด้านขวาเป็นระยะทาง 1.5 เมตร จากผลการจำลองสถานการณ์ที่แสดงในรูปที่ 4.18 แสดงให้เห็นว่าตัวรถจะเคลื่อนที่ต่อไปทางด้านขวา เพื่อให้ลูกตุ้มหมุนกลับในทิศทวนเข็มนาฬิกา จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ทำให้ลูกตุ้มหมุนในทิศตามเข็มนาฬิกา จนกระทั่งตัวรถเคลื่อนที่กลับมาหยุดที่ตำแหน่งกึ่งกลางของรางเลื่อน พร้อมทั้งทำให้ก้านลูกตุ้มสามารถตั้งตรงที่ตำแหน่งศูนย์องศาได้อีกครั้ง ซึ่งใช้เวลาในการสร้างเสถียรภาพทั้งสิ้น 4.06 วินาที

จากนั้นได้จำลองสถานการณ์ โดยกำหนดให้มุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถ มีทิศทางตรงกันข้ามกับที่ได้จำลองสถานการณ์ไว้ก่อนแล้ว ทั้งนี้ยังคงกำหนดค่าอื่นๆ เหมือนเดิมทุกประการ ซึ่งจะทำการจำลองสถานการณ์ 3 กรณี โดยกรณีแรกกำหนดให้มุมเริ่มต้นลูกตุ้มเป็น -30 องศา ตำแหน่งเริ่มต้นตัวรถเป็นศูนย์เมตร กรณีต่อมากำหนดให้ตำแหน่งเริ่มต้นตัวรถ

เป็น -2.0 เมตร โดยมุมเริ่มต้นลูกตุ้มเป็นศูนย์องศา ส่วนกรณีสุดท้ายกำหนดให้มุมเริ่มต้นลูกตุ้มเป็น -30 องศา ตำแหน่งเริ่มต้นตัวรถเป็น -1.5 เมตร ซึ่งผลการจำลองสถานการณ์ดังกล่าวแสดงได้ดังรูปที่ 4.19, 4.20 และ 4.21 ตามลำดับ โดยทุกกรณีกำหนดให้ก้านลูกตุ้มยาว 1.0 เมตร

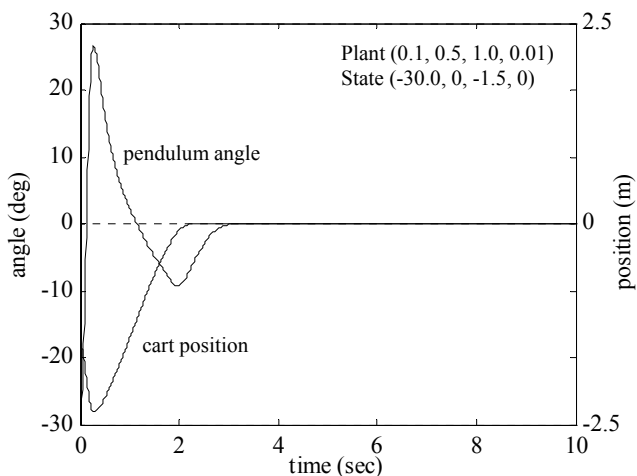


รูปที่ 4.19 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา และก้านลูกตุ้มยาว 1.0 เมตร



รูปที่ 4.20 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น -2.0 เมตร

เมื่อพิจารณาผลการจำลองสถานการณ์ในทุกกรณี พบว่า ไม่แตกต่างกับกรณีที่กำหนดให้มุมเริ่มต้นลูกตุ้มและตำแหน่งเริ่มต้นตัวรถมีทิศทางเป็นบวก เพียงแต่กระบวนการต่างๆ ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวเกิดขึ้นในทิศทางตรงข้ามกันเท่านั้น นอกจากนี้เมื่อ



รูปที่ 4.21 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น -1.5 เมตร

พิจารณาเวลาในการเข้าสู่เสถียรภาพของระบบจะได้ว่า กรณีที่กำหนดให้มุมเริ่มต้นลูกตุ้มและตำแหน่งเริ่มต้นตัวรถมีทิศทางตรงข้ามกับที่กำหนดไว้เดิม จะใช้เวลาในการเข้าสู่เสถียรภาพเท่ากัน ตัวอย่างเช่น กรณีมุมเริ่มต้นลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นตัวรถเป็น 1.5 เมตร กับกรณีที่มุมเริ่มต้นลูกตุ้มเป็น -30 องศา ตำแหน่งเริ่มต้นตัวรถเป็น -1.5 เมตร จะได้ว่า ตัวควบคุมนิวโร-ฟัซซี่ใช้เวลาในการสร้างเสถียรภาพให้ระบบเท่ากันคือ 4.06 วินาที เป็นต้น ส่วนผลการจำลองสถานการณ์ที่ปรากฏในรูปที่ 4.19 และ 4.20 พบว่า ระบบลูกตุ้มผกผันแบบข้อต่อเดียวสามารถเข้าสู่เสถียรภาพได้ภายในเวลา 3.95 และ 3.16 วินาที ตามลำดับ

จากการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ เมื่อปรับเปลี่ยนสถานการณ์ที่ควบคุมแตกต่างกัน พบว่า ตัวควบคุมนิวโร-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาอันสั้น โดยเมื่อสถานการณ์ในการควบคุมเปลี่ยนแปลงไป โครงข่ายประสาทเทียมยังคงสามารถเรียนรู้ เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะกับสถานการณ์ที่แปรเปลี่ยน ซึ่งตารางที่ 4.3 แสดงค่าระดับความสำคัญพลวัตที่ได้จากการเรียนรู้ของโครงข่ายประสาทเทียม โดยที่ DID-1, DID-2, DID-3 และ DID-4 หมายถึงค่าระดับความสำคัญพลวัตของมุมลูกตุ้ม ความเร็วเชิงมุมของลูกตุ้ม ตำแหน่งตัวรถ และความเร็วของตัวรถ ตามลำดับ เมื่อพิจารณาค่าระดับความสำคัญพลวัตที่ปรากฏในตารางที่ 4.3 จะสังเกตเห็นว่า เมื่อสถานการณ์ในการควบคุมเปลี่ยนแปลงไป ค่าระดับความสำคัญพลวัตเหล่านี้ก็จะแปรเปลี่ยนไปด้วย นอกจากนี้ กรณีที่กำหนดให้มุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถ มีทิศทางตรงกันข้ามกับที่กำหนดไว้เดิม จะได้ว่า ค่าระดับความสำคัญพลวัตที่ได้จากการเรียนรู้ของโครงข่ายประสาทเทียมยังคงมีค่าเท่ากัน

ตารางที่ 4.3 ค่าระดับความสำคัญพลวัตของตัวควบคุมนิวโร-พีซซี่ในสถานการณ์ต่างๆ

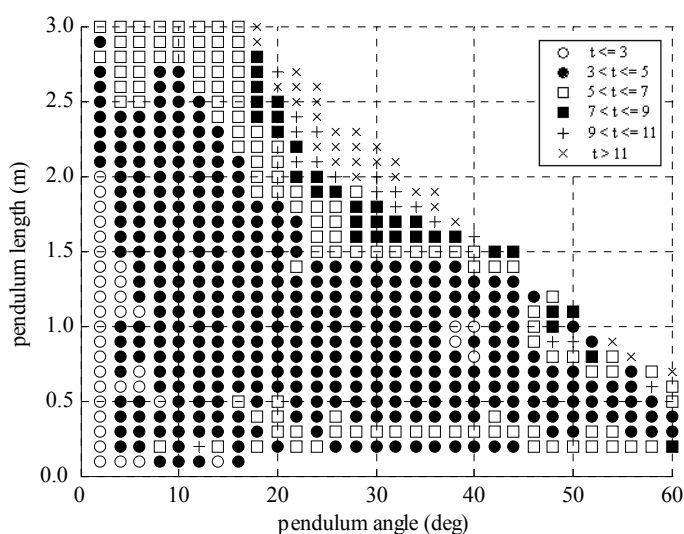
สถานการณ์	ค่าระดับความสำคัญพลวัต			
	DID-1	DID-2	DID-3	DID-4
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.5964	5.1130	5.8458	2.5733
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 0.2 เมตร	5.6961	4.0936	2.4867	1.8806
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 2.0 เมตร	7.1303	6.5145	3.9356	2.3777
มุมเริ่มต้น 0 องศา ตำแหน่งเริ่มต้น 2.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.4701	5.8249	4.3502	2.9421
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 1.5 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.0634	5.7231	3.1463	2.4189
มุมเริ่มต้น -30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.5964	5.1130	5.8458	2.5733
มุมเริ่มต้น 0 องศา ตำแหน่งเริ่มต้น -2.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.4701	5.8249	4.3502	2.9421
มุมเริ่มต้น -30 องศา ตำแหน่งเริ่มต้น -1.5 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.0634	5.7231	3.1463	2.4189

เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-พีซซี่ แสดงได้ดังตารางที่ 4.4 ซึ่งเวลาเข้าสู่เสถียรภาพตามที่ปรากฏในสถานการณ์ต่างๆ พิจารณาได้จากผลการจำลองสถานการณ์ที่ได้แสดงไว้ในรูปที่ 4.12, 4.13, 4.14, 4.15 และ 4.18 ตามลำดับ ซึ่งจะสังเกตเห็นได้ว่า ตัวควบคุมนิวโร-พีซซี่ สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ครบทั้ง 5 สถานการณ์ภายในระยะเวลาอันสั้น

ตารางที่ 4.4 เวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมนิวโร-ฟัซซี่

สถานการณ์	เวลาในการเข้าสู่เสถียรภาพ (วินาที)
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	3.95
มุมเริ่มต้นของลูกตุ้มเป็น 0.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	3.16
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 0.2 เมตร	3.19
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 2.0 เมตร	9.20
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	4.06

นอกจากนี้ได้ทำการทดสอบการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ เมื่อปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้มผกผันพร้อมกับปรับความยาวก้านลูกตุ้ม

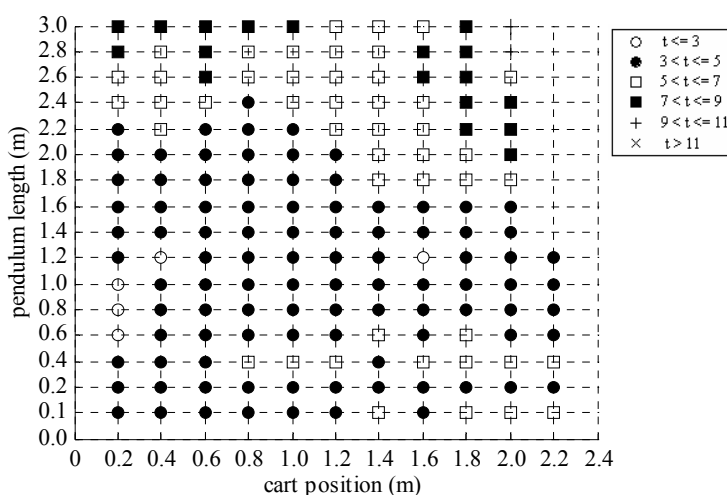


รูปที่ 4.22 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม



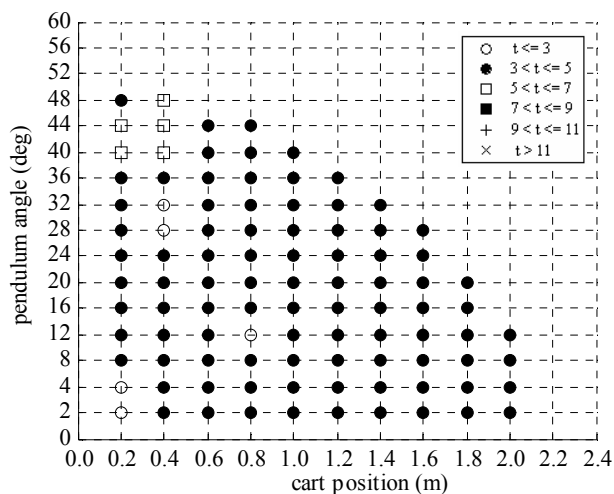
ทั้งนี้กำหนดให้ตำแหน่งเริ่มต้นของรถเป็นศูนย์เมตร ในการทดสอบดังกล่าวจะดำเนินการปรับเพิ่มมุมเริ่มต้นของลูกตุ้มครั้งละ 2 องศา โดยเริ่มจาก 2 องศา จนกระทั่งถึง 60 องศา และทุกตำแหน่งมุมเริ่มต้นของลูกตุ้มที่แปรเปลี่ยนไป จะทำการปรับเพิ่มความยาวก้านลูกตุ้มครั้งละ 0.1 เมตร โดยเริ่มจาก 0.1 เมตร จนกระทั่งถึง 3.0 เมตร เมื่อพิจารณาผลการทดสอบในรูปที่ 4.22 จะสังเกตเห็นว่า โดยส่วนใหญ่แล้ว ตัวควบคุมนิวโร-ฟัซซี่ใช้เวลาในการสร้างเสถียรภาพให้ระบบได้ภายในเวลา 5 วินาที รวมทั้งสามารถสร้างเสถียรภาพให้ระบบเมื่อแปรเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้มได้ในขอบเขตที่กว้าง

ต่อมาดำเนินการปรับเปลี่ยนตำแหน่งเริ่มต้นของรถและความยาวก้านลูกตุ้ม โดยกำหนดมุมเริ่มต้นลูกตุ้มเป็นศูนย์องศา ทั้งนี้จะทำการปรับตำแหน่งเริ่มต้นของรถครั้งละ 0.2 เมตร โดยเริ่มจากตำแหน่ง 0.2 เมตร จนกระทั่งถึงปลายสุดของรางเลื่อน นั่นคือที่ระยะ 2.4 เมตร และปรับเพิ่มความยาวก้านลูกตุ้มครั้งละ 0.2 เมตร โดยเริ่มที่ก้านลูกตุ้มยาว 0.1 เมตร จนกระทั่งถึง 3.0 เมตร ซึ่งเวลาที่ใช้สร้างเสถียรภาพให้แก่ระบบลูกตุ้มผกผันแบบข้อต่อเดียวแสดงได้ดังรูปที่ 4.23 โดยจะสังเกตเห็นได้ว่า ถ้าตำแหน่งเริ่มต้นของตัวรถเป็น 2.4 เมตร ตัวควบคุมนิวโร-ฟัซซี่ จะไม่สามารถสร้างเสถียรภาพให้ระบบได้ เนื่องจากตัวรถจะเคลื่อนที่เกินความยาวของรางเลื่อนที่กำหนดไว้



รูปที่ 4.23 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม

นอกจากนี้ได้ดำเนินการทดสอบการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน ด้วยตัวควบคุมนิวโร-ฟัซซี่ โดยปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้มพร้อมกับปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถ เมื่อกำหนดให้ก้านลูกตุ้มยาว 1.0 เมตร เวลาที่ตัวควบคุมนิวโร-ฟัซซี่ใช้สำหรับสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวแสดงได้ดังรูปที่ 4.24



รูปที่ 4.24 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม

เมื่อพิจารณาผลการทดสอบตัวควบคุมนิวโร-ฟัซซี่ ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยดำเนินการปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม ตำแหน่งเริ่มต้นของตัวรถ รวมทั้งความยาวก้านลูกตุ้ม ดังที่ปรากฏในรูปที่ 4.22, 4.23 และ 4.24 อาจกล่าวได้ว่า ตัวควบคุมนิวโร-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบได้ในระยะเวลาอันสั้น รวมทั้งยังสามารถสร้างเสถียรภาพให้ระบบดังกล่าวได้ แม้จะแปรเปลี่ยนสถานการณ์เริ่มต้นของระบบในขอบเขตที่กว้าง

#### 4.8 การทดสอบสุ่มค่าพารามิเตอร์เริ่มต้นให้โครงข่ายประสาทเทียม

เนื่องจากค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ได้แก่ ค่าน้ำหนัก และค่าไบแอส จะมีอิทธิพลต่อการเรียนรู้ของโครงข่ายประสาทเทียม กล่าวคือ จะมีผลต่อการลดลงของค่าความคลาดเคลื่อน ซึ่งการปรับค่าพารามิเตอร์เริ่มต้นเหล่านี้จะขึ้นอยู่กับค่าอนุพันธ์ของฟังก์ชันกระตุ้นที่ใช้ ดังนั้นควรหลีกเลี่ยงค่าพารามิเตอร์เริ่มต้นที่ทำให้ฟังก์ชันกระตุ้นและอนุพันธ์ของฟังก์ชันกระตุ้นมีค่าเป็นศูนย์ เพราะไม่ทำให้เกิดการเรียนรู้เพื่อปรับค่าน้ำหนักและค่าไบแอส นอกจากนี้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมไม่ควรมีค่ามากหรือน้อยเกินไป เนื่องจากอาจทำให้เกิดปรากฏการณ์อิ่มตัว (saturation) รวมทั้งส่งผลให้การเรียนรู้ดำเนินไปได้ช้า โดยทั่วไปแล้วการสุ่มค่าพารามิเตอร์เริ่มต้นสำหรับกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม จะกำหนดให้อยู่ในช่วง  $[-1, +1]$  ซึ่งค่าพารามิเตอร์เริ่มต้นเหล่านี้สามารถเป็นได้ทั้งค่าบวกและค่าลบ

ในการทดสอบสุ่มค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียม กำหนดให้พารามิเตอร์เหล่านี้มีค่าอยู่ในช่วง  $[-1, +1]$  ซึ่งในแต่ละครั้งของการสุ่มจะได้ค่าน้ำหนักและค่าไบแอสเริ่มต้นทั้งสิ้น 32 ค่า จากนั้นจึงจำลองสถานการณ์เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน

ทั้งหมด 5 กรณี ดังปรากฏในตารางที่ 4.5 โดยสุ่มค่าพารามิเตอร์เริ่มต้นเหล่านี้จำนวน 4000 ครั้ง เพื่อพิจารณาจำนวนครั้งที่ตัวควบคุมนิวโร-ฟิชซึ่งสามารถสร้างเสถียรภาพให้ระบบได้

ตารางที่ 4.5 การทดสอบสุ่มค่าพารามิเตอร์เริ่มต้นให้โครงข่ายประสาทเทียม

กรณีที่	สถานการณ์	ปริมาณการเข้าสู่เสถียรภาพ
1	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	67.830%
2	มุมเริ่มต้น 0 องศา ตำแหน่งเริ่มต้น 2.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	59.525%
3	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 0.2 เมตร	32.825%
4	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 2.0 เมตร	36.825%
5	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 1.5 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	17.925%

เมื่อพิจารณาปริมาณการเข้าสู่เสถียรภาพดังแสดงในตารางที่ 4.5 จะสังเกตเห็นว่า ในแต่ละกรณีมีปริมาณการเข้าสู่เสถียรภาพที่ไม่สูงมากนัก ตัวอย่างเช่น กรณีแรกคิดเป็น 67.830% หมายความว่า เมื่อทำการสุ่มค่าพารามิเตอร์เริ่มต้นจำนวน 4000 ครั้ง ตัวควบคุมนิวโร-ฟิชซึ่งสามารถสร้างเสถียรภาพได้จำนวน 2713 ครั้ง ส่วนกรณีสุดท้ายคิดเป็น 17.925% หมายความว่า พารามิเตอร์เริ่มต้นชุดเดียวกันนี้ส่งผลให้ตัวควบคุมนิวโร-ฟิชซึ่งสร้างเสถียรภาพให้ระบบได้เพียง 717 ครั้ง ดังนั้น ถ้าค่าพารามิเตอร์เริ่มต้นที่ได้จากการสุ่มค่าไม่เหมาะสม โครงข่ายประสาทเทียมอาจจะไม่สามารถปรับค่าระดับความสำคัญพลวัตให้เหมาะกับสถานการณ์ที่ควบคุมได้ และค่าพารามิเตอร์เริ่มต้นชุดเดียวกันก็อาจไม่สามารถสร้างเสถียรภาพให้ระบบได้ทุกสถานการณ์ ดังนั้นกล่าวได้ว่า ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมเป็นปัจจัยที่สำคัญประการหนึ่งที่มีผลต่อการเรียนรู้ เพื่อปรับค่าระดับความสำคัญพลวัต ซึ่งจะส่งผลกระทบต่อกรการสร้างเสถียรภาพให้ระบบ ลูกตุ้มผกผันแบบข้อต่อเดียวของตัวควบคุมนิวโร-ฟิช

## 4.9 สรุป

การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ เป็นการผสมผสานระหว่างตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวและการเรียนรู้ของโครงข่ายประสาทเทียม โดยค่าระดับความสำคัญพลวัตได้จากการเรียนรู้แบบแพร่กลับค่าความคลาดเคลื่อน เมื่อสถานการณ์ในการสร้างเสถียรภาพให้ระบบเปลี่ยนแปลงไป โครงข่ายประสาทเทียมต้องสามารถเรียนรู้ จนกระทั่งได้ค่าระดับความสำคัญพลวัตที่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ ทั้งนี้ในกระบวนการเรียนรู้ไม่สามารถระบุค่าระดับความสำคัญพลวัตที่ต้องการ สำหรับเปรียบเทียบกับค่าเอาต์พุตที่ได้จากโครงข่ายประสาทเทียม ดังนั้นจำเป็นต้องใช้ค่าตัวแปรอื่นที่ทราบค่า โดยในงานวิจัยวิทยานิพนธ์นี้ได้เสนอให้พิจารณาเอาต์พุตของระบบในการคำนวณหาค่าความคลาดเคลื่อนที่ใช้ในกระบวนการเรียนรู้ จากผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนสถานการณ์การควบคุมแตกต่างกัน แสดงให้เห็นว่าตัวควบคุมนิวโร-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาอันสั้น รวมทั้งสามารถแปรเปลี่ยนมุมเริ่มต้นลูกตุ้ม ตำแหน่งเริ่มต้นของตัวรถ และความยาวก้านลูกตุ้ม ได้ในขอบเขตที่กว้าง โดยที่ตัวควบคุมนิวโร-ฟัซซี่ ยังคงรักษาเสถียรภาพให้แก่ระบบได้

อย่างไรก็ตาม ค่าพารามิเตอร์เริ่มต้นที่ใช้ในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม ได้มาจากวิธีการสุ่มค่า ซึ่งอาจทำให้ได้ค่าระดับความสำคัญพลวัตที่ไม่เหมาะสมกับสถานการณ์ที่ควบคุม โดยเฉพาะอย่างยิ่งเมื่อระบบที่ต้องการควบคุมมีความซับซ้อนมากขึ้น ค่าพารามิเตอร์เริ่มต้นที่ต้องดำเนินการสุ่มค่าให้กับโครงข่ายประสาทเทียมย่อมมีจำนวนเพิ่มมากขึ้นด้วย ซึ่งมีโอกาสที่ค่าพารามิเตอร์เริ่มต้นเหล่านี้ไม่สร้างเสถียรภาพให้กับระบบได้ ดังนั้นในงานวิจัยวิทยานิพนธ์นี้จึงได้นำวิธีการค้นหาแบบตาบู่เชิงปรับตัวเข้าช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียมแทนการสุ่มค่า เรียกวิธีการนี้ว่า ตัวควบคุมนิวโร-ตาบู่-ฟัซซี่ ซึ่งจะได้อธิบายรายละเอียดในบทที่ 5 ต่อไป

## บทที่ 5

### ตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่

#### 5.1 บทนำ

จากการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ ดังได้อธิบายรายละเอียดไว้ในบทที่ 4 พบว่า เมื่อทำการทดสอบลุ่มค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียม ซึ่งได้แก่ ค่าน้ำหนักและค่าไบแอสเริ่มต้น เป็นจำนวน 4000 ครั้ง หลังจากที่น่าค่าพารามิเตอร์เริ่มต้นเหล่านี้ไปใช้ในกระบวนการเรียนรู้แบบแพร่กลับ เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะกับการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวในแต่ละสถานการณ์ โดยยังคงใช้ค่าพารามิเตอร์เริ่มต้นชุดเดียวกันแม้สถานการณ์ในการควบคุมจะเปลี่ยนแปลงไป เมื่อพิจารณาผลการทดสอบ พบว่า จำนวนครั้งจากการลุ่มค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียมที่สามารถสร้างเสถียรภาพให้ระบบได้ในแต่ละสถานการณ์มีปริมาณไม่มากนัก ซึ่งอาจกล่าวได้ว่า ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมเป็นปัจจัยสำคัญประการหนึ่งที่มีผลกระทบต่อกรสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวของตัวควบคุมนิวโร-ฟัซซี่ ดังนั้นในงานวิจัยวิทยานิพนธ์นี้ จึงได้นำวิธีการค้นหาแบบตาบูลูเชิงปรับตัวเข้าช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม แทนที่การลุ่มค่า เพื่อสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้อย่างรวดเร็ว และลดปริมาณการขาดเสถียรภาพของระบบอันเนื่องมาจากค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมไม่เหมาะสม เรียกวิธีการควบคุมแบบใหม่ที่นำเสนอในวิทยานิพนธ์นี้ว่า ตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ (neuro-tabu-fuzzy controller) ซึ่งเป็นการผสมผสานระหว่างตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว และการเรียนรู้ของโครงข่ายประสาทเทียม โดยได้นำวิธีการค้นหาแบบตาบูลูเชิงปรับตัวเข้ามาช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

เนื้อหาในบทนี้กล่าวถึง วิธีการค้นหาแบบตาบูลูเชิงปรับตัว หลักการทำงานของตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว รวมถึงการทดสอบเพื่อเลือกค่าพารามิเตอร์สำหรับวิธีการค้นหาแบบตาบูลูเชิงปรับตัว ได้แก่ จำนวนคำตอบรอบข้างในการค้นหา (number of neighbors) และรัศมีการค้นหา (radius) เพื่อช่วยให้การค้นหาแบบตาบูลูเชิงปรับตัวสามารถค้นหาคำตอบได้รวดเร็วยิ่งขึ้น จากนั้นจึงนำค่าพารามิเตอร์เริ่มต้นที่ค้นหาได้นี้ไปใช้

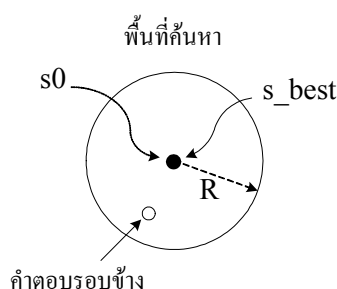
ในการจำลองสถานการณ์เพื่อสร้างเสถียรภาพให้ระบบดังกล่าว เมื่อปรับเปลี่ยนสถานการณ์ในการควบคุม นอกจากนี้ยังได้ดำเนินการทดสอบตัวควบคุมนิวโร-ตาบิว-ฟิชซี โดยนำค่าพารามิเตอร์เริ่มต้นที่ได้จากการค้นหาแบบตาบิวเชิงปรับตัวจำนวน 4000 ครั้ง ไปจำลองสถานการณ์เมื่อสถานการณ์การควบคุมแปรเปลี่ยนไป เพื่อพิจารณาจำนวนครั้งที่ตัวควบคุมดังกล่าวยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้

## 5.2 การค้นหาแบบตาบิวเชิงปรับตัว

เนื่องจากวิธีการค้นหาแบบตาบิวเชิงปรับตัว (adaptive tabu search) (Puangdownreong, Kulworawanichpong, and Sujitjorn, 2004) ได้รับการปรับปรุงมาจากวิธีการค้นหาแบบตาบิวธรรมดา (conventional tabu search) ดังนั้น เพื่อเป็นพื้นฐานความเข้าใจเกี่ยวกับการค้นหาแบบตาบิวเชิงปรับตัว จึงกล่าวถึงวิธีการค้นหาแบบตาบิวธรรมดาเป็นลำดับแรก ซึ่งการค้นหาดังกล่าวเป็นขั้นตอนวิธีการคิดที่นำมาประยุกต์ในการหาคำตอบที่ดีที่สุดเชิงผสมผสาน (combinatorial optimization) ได้อย่างมีประสิทธิภาพ และมีความสามารถในการหลีกเลี่ยงคำตอบที่ดีที่สุดแบบวงแคบเฉพาะถิ่น (local optimum) รวมทั้งยังสามารถค้นหาคำตอบต่อไปจนกระทั่งได้คำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุดแบบวงกว้าง (global optimum) ดังนั้นวิธีการค้นหาแบบตาบิวจึงถูกนำมาใช้ในการแก้ปัญหา ระบบที่ต้องการหาค่าที่เหมาะสมที่สุดได้อย่างไม่ยุ่งยากมากนัก ขั้นตอนในการนำเอาวิธีการค้นหาแบบตาบิวธรรมดาไปประยุกต์ใช้สามารถสรุปได้ดังนี้

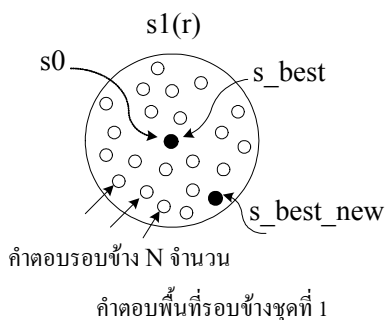
ขั้นตอนที่ 1 กำหนดให้ count เป็นจำนวนรอบในการค้นหา และ count\_max เป็นจำนวนรอบสูงสุดที่จะทำการค้นหา

ขั้นตอนที่ 2 ทำการสุ่มคำตอบเริ่มต้น  $s_0$  ภายในพื้นที่การค้นหาคำตอบรอบข้างในรัศมี  $R$  โดยขณะนี้  $s_0$  มีสถานะเป็นคำตอบที่ดีที่สุดแบบวงแคบเฉพาะถิ่น กำหนดให้ค่า  $s_0 = s_{best}$  ดังแสดงในรูปที่ 5.1



รูปที่ 5.1 การสุ่มคำตอบ  $s_0$  ในพื้นที่การค้นหารัศมี  $R$

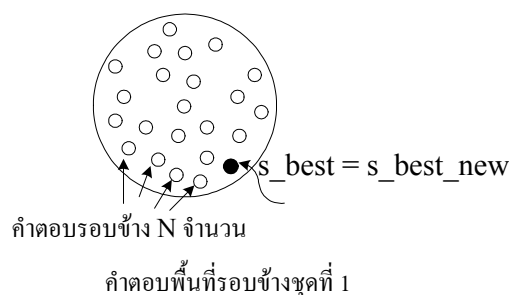
ขั้นตอนที่ 3 ทำการสุ่มเลือกคำตอบใหม่  $N$  คำตอบ โดยทำการสุ่มรอบๆ คำตอบ  $s_0$  ภายในพื้นที่การค้นหา กำหนดให้  $s_1(r)$  เป็นชุดของคำตอบที่ประกอบไปด้วยคำตอบใหม่ทั้งหมด  $N$  คำตอบ ซึ่งแสดงได้ด้วยรูปที่ 5.2



รูปที่ 5.2 คำตอบพื้นที่รอบข้าง  $s_0$

ขั้นตอนที่ 4 ทำการประเมินค่าของคำตอบที่อยู่ภายใน  $s_1(r)$  ด้วยฟังก์ชันวัตถุประสงค์ แล้วเลือกคำตอบที่ดีที่สุดจากคำตอบทั้งหมดที่อยู่ภายใน  $s_1(r)$  และกำหนดให้คำตอบนั้นเป็น  $s\_best\_new$  ดังแสดงในรูปที่ 5.2

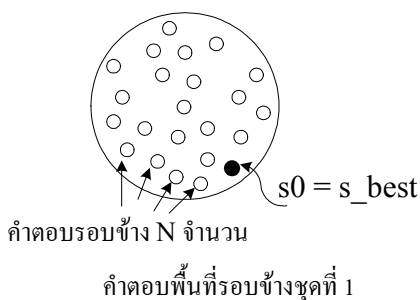
ขั้นตอนที่ 5 ถ้า  $s\_best\_new$  ดีกว่า  $s\_best$  แล้ว กำหนดให้  $s\_best = s\_best\_new$  ดังพิจารณาได้จากรูปที่ 5.3 แต่ถ้าไม่มีคำตอบใดที่ดีกว่า  $s\_best$  ให้ไปยังขั้นตอนที่ 6



รูปที่ 5.3 กำหนดค่าใหม่ให้กับคำตอบพื้นที่รอบข้างที่ดีที่สุด

ขั้นตอนที่ 6 กำหนดค่า  $s_0 = s\_best$  ซึ่งแสดงได้ด้วยรูปที่ 5.4

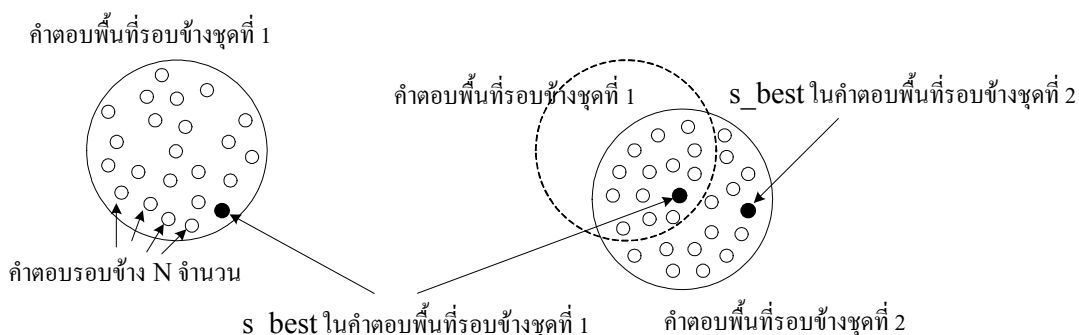
ขั้นตอนที่ 7 ถ้า  $s\_best$  ไม่อยู่ใน Neighbor List ให้ทำการบันทึก  $s\_best\_new$  ลงใน Neighbor List



รูปที่ 5.4 กำหนดค่าใหม่ให้กับ  $s_0$

ขั้นตอนที่ 8 ถ้า  $count > count\_max$  ให้ยุติการค้นหา ค่าตอบที่ได้จะเป็นค่าตอบที่ดีที่สุดตั้งแต่ทำการค้นหา

ขั้นตอนที่ 9 ถ้า  $count < count\_max$  ให้กลับไปเริ่มที่ขั้นตอนที่ 2 ใหม่ และทำการค้นหาจนกระทั่งถึงเป้าหมายที่ตั้งไว้ ซึ่งแสดงได้ดังรูปที่ 5.5

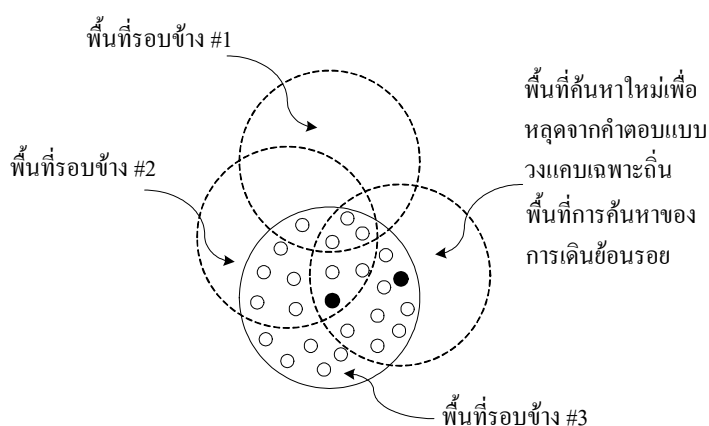


รูปที่ 5.5 การค้นหาค่าตอบในรอบต่อไป

อย่างไรก็ตาม การค้นหาแบบตามูธรรมดานั้นยังมีประสิทธิภาพไม่เพียงพอต่อการนำไปประยุกต์ใช้งานในหลายๆ กรณี ตัวอย่างเช่น การค้นหาค่าพารามิเตอร์ของระบบที่มีความซับซ้อนมากขึ้น รวมถึงกรณีที่มีค่าพารามิเตอร์ที่ต้องการค้นหาเป็นจำนวนมาก ทำให้ต้องใช้เวลาในการค้นหาค่าพารามิเตอร์เหล่านี้เพิ่มขึ้น ดังนั้นจึงได้มีการนำเอาโครงสร้างของการค้นหาแบบตามูธรรมดามาทำการปรับปรุง เพื่อให้สามารถค้นหาค่าตอบที่ดีที่สุดได้อย่างมีประสิทธิภาพและรวดเร็วยิ่งขึ้น เรียกวิธีการที่ได้รับการปรับปรุงนี้ว่า การค้นหาแบบตามูเชิงปรับตัว ซึ่งได้ทำการเพิ่มสองกลไกเข้าไปในการค้นหาแบบตามูธรรมดา ได้แก่ การเดินย้อนรอย และการปรับรัศมีการค้นหา ซึ่งมีรายละเอียดดังนี้คือ



การเดินย้อนรอย (back tracking) เป็นขั้นตอนที่อนุญาตให้ระบบทำการกลับไปค้นหาพื้นที่คำตอบเก่าที่เคยถูกค้นหามาแล้ว คำตอบที่ดีกว่าจะถูกเลือกจากพื้นที่คำตอบที่ถูกย้อนรอยนี้ รายละเอียดของการย้อนรอยแสดงได้ดังรูปที่ 5.6 โดยการกำหนดคำตอบใหม่ให้เป็นจุดเริ่มต้นในการค้นหา พื้นที่ในการค้นหาใหม่ก็จะเกิดขึ้นด้วย นั่นหมายความว่า การค้นหามีโอกาสที่จะหลุดออกจากคำตอบที่เป็นแบบวงแคบเฉพาะถิ่น คำตอบใหม่ที่ถูกใช้เป็นจุดเริ่มต้นนี้ไม่จำเป็นจะต้องเป็นคำตอบที่ดีที่สุดในพื้นที่ปัจจุบัน การเดินย้อนรอยจึงเป็นกลไกหนึ่งที่สามารถเพิ่มประสิทธิภาพในการหลุดจากคำตอบที่เป็นแบบวงแคบเฉพาะถิ่นได้



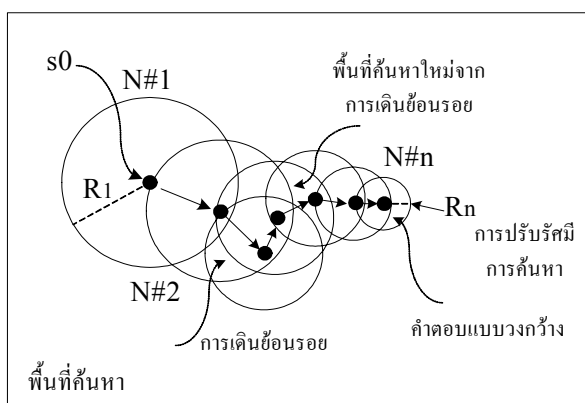
รูปที่ 5.6 การเดินย้อนรอยในกระบวนการค้นหาแบบตามูเชิงปรับตัว

การปรับรัศมีการค้นหา (adaptive radius) กระบวนการปรับรัศมีการค้นหาจะทำการลดรัศมีการค้นหาในระหว่างดำเนินกระบวนการ ซึ่งการลดรัศมีการค้นหาจะดำเนินไปเรื่อยๆ จนกระทั่งการค้นหาเข้าใกล้คำตอบที่ดีที่สุดแบบวงกว้าง ถ้ากำหนดรัศมีการค้นหาที่กว้างจะให้ผลของการค้นหาที่หยาบ ซึ่งอาจจะมีผลให้การค้นหาพลาดคำตอบที่ต้องการได้ ในทางตรงกันข้าม ถ้ารัศมีการค้นหาที่มีขนาดเล็ก การค้นหาคำตอบจะต้องใช้เวลานานขึ้น แต่ถ้ารัศมีการค้นหาเล็กมากๆ อาจจะทำให้ไม่สามารถครอบคลุมพื้นที่ของคำตอบที่ต้องการค้นหาได้ ดังนั้นการปรับรัศมีการค้นหาให้เหมาะสมกับสถานการณ์การค้นหาคำตอบ จึงสามารถเพิ่มประสิทธิภาพของการค้นหาได้ ในกระบวนการค้นหาแบบตามูเชิงปรับตัว จึงได้มีการปรับรัศมีการค้นหาในระหว่างการค้นหาคำตอบให้เหมาะสม โดยการใช้ค่าประเมินที่ได้จากฟังก์ชันวัตถุประสงค์เป็นองค์ประกอบในการพิจารณาปรับรัศมีการค้นหา กล่าวคือ เมื่อคำตอบปัจจุบันให้ค่าการประเมินที่ดีขึ้น รัศมีการค้นหาจะถูกปรับลดลงตามความสัมพันธ์ดังแสดงในสมการที่ (5-1)

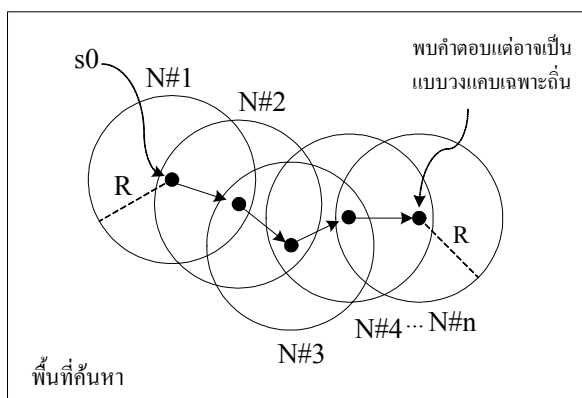
$$\text{radius}_{\text{new}} = \frac{\text{radius}_{\text{old}}}{\text{DF}} \tag{5-1}$$

โดยที่ DF คือ ตัวประกอบของการลดรัศมี (decreasing factor) ซึ่งการกำหนดค่าที่เหมาะสมของ DF จะแตกต่างกันและขึ้นอยู่กับปัญหาที่ต้องการค้นหาคำตอบด้วย

โดยปกติการค้นหาแบบตามูเชิงปรับตัว ได้เพิ่มกลไกการเดินย้อนรอยเข้าไปในขั้นตอนที่ 6 และขั้นตอนที่ 7 ของการค้นหาแบบตามูธรรมดา ในขณะที่กลไกการปรับรัศมีการค้นหาจะใช้ในช่วงท้ายสุดของขั้นตอนที่ 7 ซึ่งความแตกต่างระหว่างการค้นหาแบบตามูเชิงปรับตัว และการค้นหาแบบตามูธรรมดาแสดงดังรูปที่ 5.7 และ 5.8 ตามลำดับ โดยจะเห็นได้ว่าการค้นหาแบบตามูธรรมดาจะค้นหาคำตอบไปเรื่อยๆ โดยไม่มีกระบวนการเดินย้อนรอยและการปรับรัศมีการค้นหาซึ่งคำตอบที่ได้อาจเป็นแบบวงแคบเฉพาะถิ่น



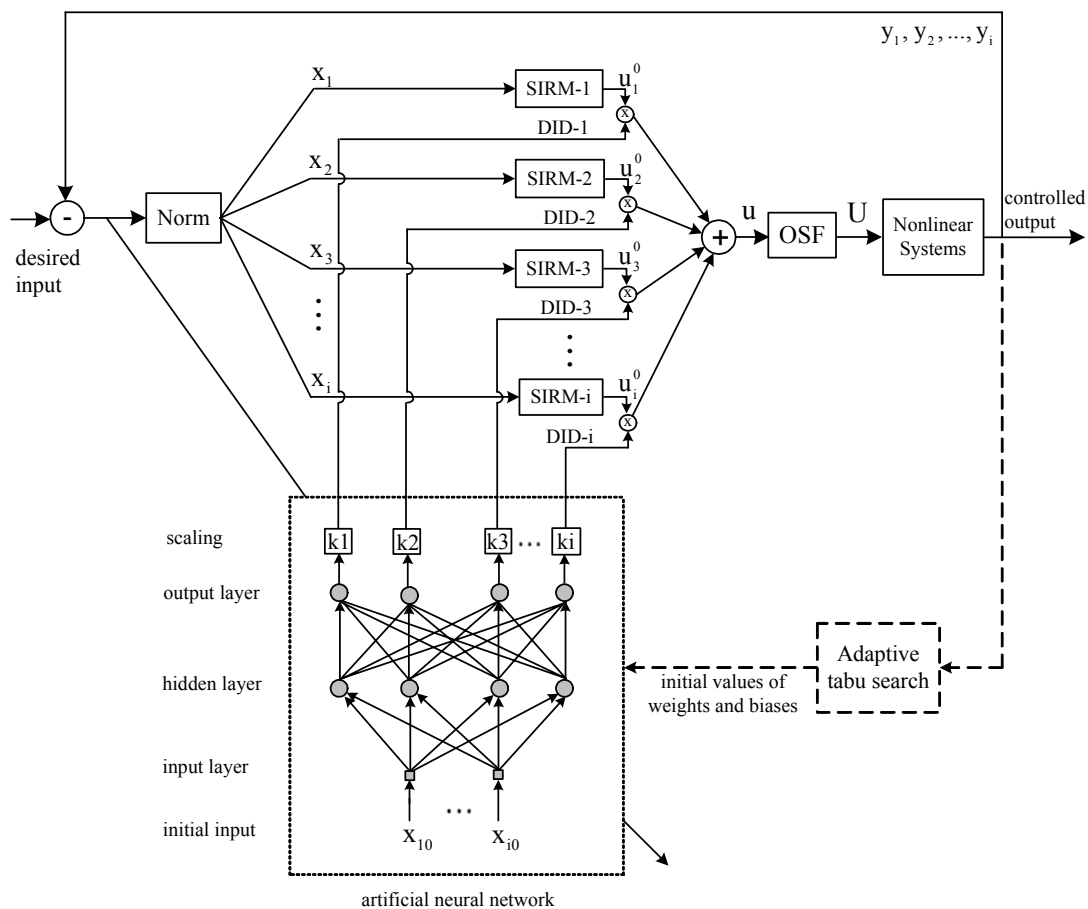
รูปที่ 5.7 การค้นหาแบบตามูเชิงปรับตัว



รูปที่ 5.8 การค้นหาแบบตามูธรรมดา

### 5.3 หลักการของตัวควบคุมนิวโร-ตาบู-ฟิชชี

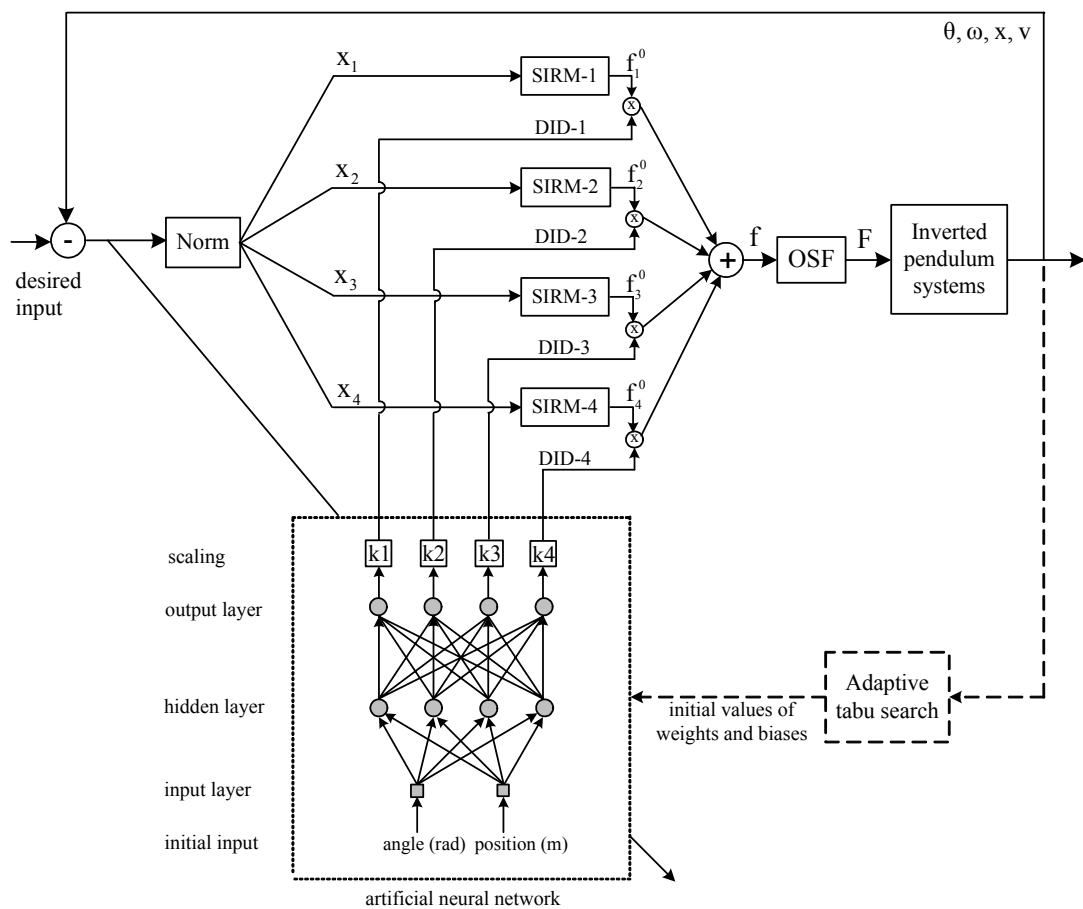
ตัวควบคุมนิวโร-ตาบู-ฟิชชี มีหลักการที่คล้ายคลึงกับตัวควบคุมนิวโร-ฟิชชี ซึ่งได้อธิบายรายละเอียดไว้ในบทที่ 4 เพียงแต่มีการนำวิธีการค้นหาแบบตาบูเชิงปรับตัวเข้ามาช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม หรืออาจกล่าวได้ว่า ตัวควบคุมนิวโร-ตาบู-ฟิชชี เป็นวิธีการควบคุมที่ปรับปรุงมาจากตัวควบคุมนิวโร-ฟิชชี เพื่อหลีกเลี่ยงผลกระทบจากค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการสุ่มค่า ซึ่งมีผลต่อการสร้างเสถียรภาพให้กับระบบที่มีความไม่เป็นเชิงเส้น โครงสร้างของตัวควบคุมนิวโร-ตาบู-ฟิชชี แสดงดังรูปที่ 5.9 ซึ่งโครงสร้างดังกล่าวประกอบด้วยมอดูลฟิชชีแบบกฎอินพุตเดียว การเรียนรู้ของโครงข่ายประสาทเทียมเพื่อปรับค่าระดับความสำคัญพลวัต โดยค่าพารามิเตอร์เริ่มต้นที่ได้จากการค้นหาแบบตาบูเชิงปรับตัวนี้ จะถูกใช้เป็นค่าเริ่มต้นให้กับโครงข่ายประสาทเทียมตลอดไป แม้ว่าสถานการณ์ในการควบคุมจะถูกเปลี่ยนแปลงไป



รูปที่ 5.9 โครงสร้างของตัวควบคุมนิวโร-ตาบู-ฟิชชี

### 5.4 การทำงานของตัวควบคุมนิวโร-ตาบู-ฟัซซี่

การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ตาบู-ฟัซซี่ แสดงดังรูปที่ 5.10 โดยตัวแปรสถานะของระบบที่พิจารณา ได้แก่ มุมของลูกตุ้ม ( $\theta$ ) ความเร็วเชิงมุมของลูกตุ้ม ( $\omega$ ) ตำแหน่งของตัวรถ ( $x$ ) และความเร็วของตัวรถ ( $v$ ) หลังจากที่ตัวแปรสถานะเหล่านี้ถูกนอร์มัลไลซ์ด้วยตัวปรับค่าที่กำหนดขึ้น ซึ่งมีค่าเช่นเดียวกับตัวปรับค่าที่ใช้ในตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว และตัวควบคุมนิวโร-ฟัซซี่ ดังได้กล่าวรายละเอียดไว้ในบทที่ 3 และ 4 แล้วนั้น จะได้ตัวแปรอินพุต  $x_1$  ถึง  $x_4$  ตามลำดับ โดยในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ต้องการให้ตัวแปรอินพุตทุกตัวเหล่านี้เข้าสู่ศูนย์ ในส่วนของมอดูลฟัซซี่แบบกฎอินพุตเดียว การกำหนดกฎฟัซซี่แต่ละกฎยังคงพิจารณาตัวแปรอินพุตเพียงตัวเดียวเท่านั้น ซึ่งตัวแปร  $f_i^0$  ( $i = 1, 2, 3, 4$ ) ที่ปรากฏในรูปที่ 5.10 หมายถึง ผลการอนุมานกฎฟัซซี่เมื่อพิจารณาตัวแปรอินพุต  $x_i$  ด้วยวิธี simplified fuzzy reasoning method



รูปที่ 5.10 การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันด้วยตัวควบคุมนิวโร-ตาบู-ฟัซซี่

เมื่อพิจารณาโครงสร้างของตัวควบคุมนิวโร-ตาบู-ฟิซซีที่ใช้สร้างเสถียรภาพให้ระบบลูกตุ้ม ผกผันแบบข้อต่อเดียว ดังปรากฏในรูปที่ 5.10 จะพบว่า ค่าระดับความสำคัญพลวัตสำหรับอินพุตแต่ละตัวได้มาจากการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่งประกอบด้วยชั้นซ่อนเร้นเพียงชั้นเดียว โดยชั้นอินพุตประกอบด้วยสองเซลล์ประสาทเทียมสำหรับรับสัญญาณอินพุต ซึ่งในงานวิจัยวิทยานิพนธ์นี้พิจารณาเริ่มต้นของลูกตุ้มในหน่วยเรเดียน และตำแหน่งของรถในหน่วยเมตร เป็นสัญญาณอินพุตให้กับชั้นอินพุต ทั้งนี้ค่าอัตราการเรียนรู้ ( $\eta$ ) และค่าคงที่โมเมนตัมยัง ( $\alpha$ ) คงมีค่าเป็น 0.05 และ 0.25 ตามลำดับ ซึ่งค่าเหล่านี้ได้จากการทดสอบดังได้อธิบายไว้ในบทที่ 4

องค์ประกอบที่สำคัญประการหนึ่งของตัวควบคุมนิวโร-ตาบู-ฟิซซี คือ การค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม ซึ่งได้แก่ ค่าน้ำหนักและค่าไบแอส ด้วยวิธีการค้นหาแบบตาบูเซิงปรับตัว ซึ่งในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว จะดำเนินการค้นหาพารามิเตอร์เริ่มต้นที่เหมาะสมเพียงชุดเดียว จากนั้นเมื่อสถานการณ์การควบคุมเปลี่ยนแปลงไป โครงข่ายประสาทเทียมจะใช้ค่าที่ค้นหาได้นี้เป็นค่าพารามิเตอร์เริ่มต้นตลอดไปสำหรับกระบวนการเรียนรู้แบบแพร่กลับ เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมแต่ละสถานการณ์ที่แปรเปลี่ยนไป

จากรูปที่ 5.10 จะสังเกตเห็นว่าแรงกระทำ  $f$  ได้มาจากผลรวมของผลคูณระหว่างผลการอนุมานกฎฟิซซีกับค่าระดับความสำคัญพลวัตของตัวแปรอินพุตแต่ละตัว ซึ่งแสดงความสัมพันธ์ดังกล่าวด้วยสมการที่ (5-2) แต่เนื่องจากแรงกระทำที่ได้นี้ยังไม่เพียงพอกับการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน ดังนั้นจึงต้องทำการยกระดับแรงกระทำนี้ด้วยตัวปรับค่า (OSF) ซึ่งกำหนดให้มีค่าเป็นสิบเท่าของผลรวมระหว่างมวลของตัวรถกับมวลของก้านลูกตุ้มผกผัน จนกระทั่งได้แรงจริงกระทำต่อตัวรถในแนวระดับ ซึ่งแสดงได้ดังสมการที่ (5-3)

$$f = \sum_{i=1}^4 f_i^0 \cdot DID - i \quad (5-2)$$

$$F = f \cdot OSF \quad (5-3)$$

โดยที่  $OSF = 10 \cdot (m_c + m_p)$

เมื่อ  $F$  คือ แรงจริงที่กระทำกับตัวรถในแนวระดับ

$m_c$  คือ มวลของตัวรถ

$m_p$  คือ มวลของก้านลูกตุ้มผกผัน

## 5.5 การพิจารณาเลือกค่าพารามิเตอร์ของการค้นหาแบบตามูขีงปรับตัว

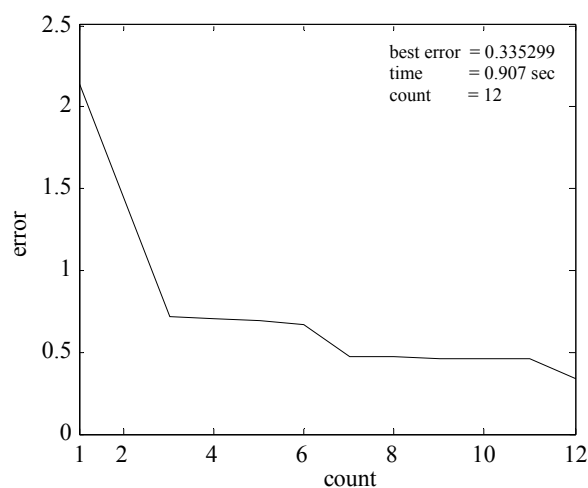
เนื่องจากวิธีการค้นหาแบบตามูขีงปรับตัว ประกอบด้วยกลไกที่สำคัญสองประการได้แก่ การเดินข้อนรอย และการปรับรัศมีการค้นหา ดังนั้นรัศมีการค้นหาจึงเป็นปัจจัยหนึ่งที่จะส่งผลกระทบต่อการค้นหาค่าพารามิเตอร์ที่ต้องการ นอกจากนี้จำนวนคำตอบรอบข้างในการค้นหาก็มีผลต่อการค้นหาด้วยเช่นกัน กล่าวคือถ้ากำหนดให้รัศมีการค้นหาและจำนวนคำตอบรอบข้างในการค้นหาให้มีค่ามากหรือน้อยเกินไป ย่อมส่งผลกระทบต่อระยะเวลาในการค้นหา รวมทั้งอาจทำให้ไม่สามารถพบคำตอบที่ต้องการได้

ในการทดสอบเพื่อวินิจฉัยเลือกค่าพารามิเตอร์ของวิธีการค้นหาแบบตามูขีงปรับตัว จะดำเนินการปรับจำนวนคำตอบรอบข้าง 3 กรณีด้วยกัน ได้แก่ 30, 40 และ 50 โดยในแต่ละกรณีจะดำเนินการปรับรัศมีการค้นหาเป็น 0.1, 0.2 และ 0.3 ตามลำดับ เมื่อพิจารณาโครงข่ายประสาทเทียมที่ปรากฏในรูปที่ 5.10 จะได้ว่าค่าน้ำหนักและค่าไบแอสเริ่มต้นที่ต้องการค้นหาประกอบด้วย 32 ค่า ทั้งนี้สถานการณ์ที่ใช้ในการทดสอบกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ก้านลูกตุ้มยาว 1.0 เมตร โดยที่ตัวรถอยู่ตรงกึ่งกลางของรางลื่น ซึ่งเงื่อนไขในการยุติการค้นหาคำตอบกำหนดไว้ว่า กระบวนการค้นหาดำเนินมาจนครบจำนวนรอบสูงสุดที่ตั้งไว้ คือ 2500 รอบ หรือค่าความคลาดเคลื่อนที่ประเมินได้จากฟังก์ชันวัตถุประสงค์มีค่าน้อยกว่า 0.35 ซึ่งค่านี้ยังไม่ใช่ค่าที่ดีที่สุด เพียงแต่เป็นค่าที่กำหนดขึ้นเพื่อการทดสอบเท่านั้น (รายละเอียดของฟังก์ชันวัตถุประสงค์ได้แสดงไว้ในภาคผนวก ข.) ผลการทดสอบดังกล่าวพิจารณาได้จากตารางที่ 5.1 ซึ่งแสดงเวลาที่ใช้ในการค้นหาจำนวนรอบที่เจอคำตอบ และค่าความคลาดเคลื่อนที่น้อยที่สุด

ตารางที่ 5.1 การทดสอบเลือกค่าพารามิเตอร์ของวิธีการค้นหาแบบตามูขีงปรับตัว

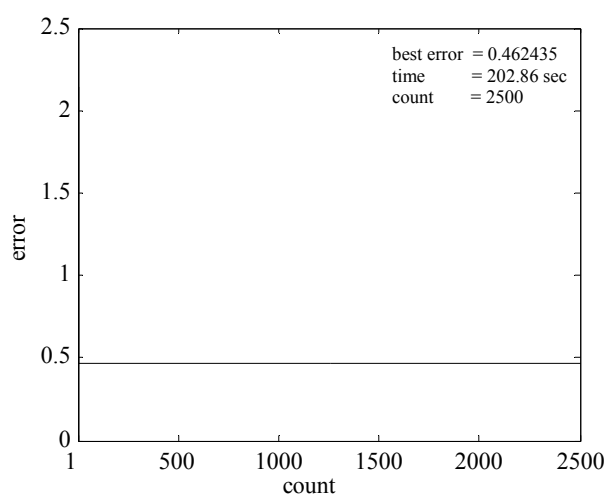
number of neighbor	radius	time (sec)	count	best error
30	0.1	120.91	1819	0.317376
	0.2	143.44	2196	0.322995
	0.3	161.56	2500	0.386682
40	0.1	74.95	902	0.349332
	0.2	0.907	12	0.335291
	0.3	202.86	2500	0.462435
50	0.1	60.92	573	0.310324
	0.2	168.57	1605	0.326617
	0.3	131.75	1264	0.329180

เมื่อพิจารณาผลการทดสอบในตารางที่ 5.1 จะพบว่ากรณีจำนวนคำตอบรอบข้างในการค้นหาเป็น 40 และรัศมีการค้นหาเป็น 0.2 จะเจอคำตอบได้รวดเร็วที่สุด กล่าวคือ ใช้เวลาเพียง 0.907 วินาที โดยเจอคำตอบในรอบที่ 12 ซึ่งค่าความคลาดเคลื่อนน้อยที่สุดเป็น 0.335291 ผลการทดสอบดังกล่าวแสดงได้ด้วยรูปที่ 5.11



รูปที่ 5.11 ผลการทดสอบเมื่อจำนวนคำตอบรอบข้างเป็น 40 และรัศมีการค้นหาเป็น 0.2

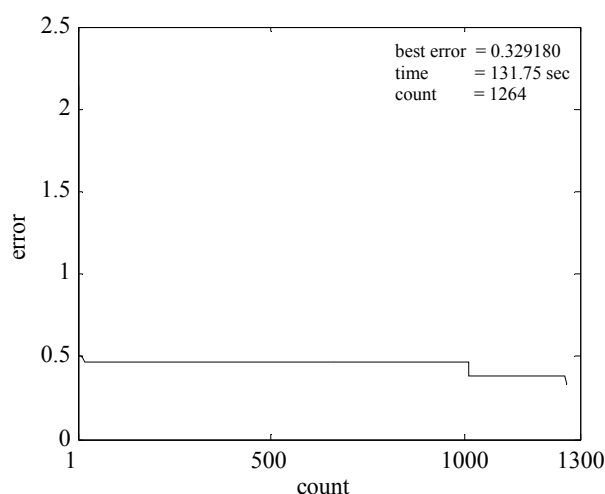
นอกจากนี้จะสังเกตเห็นว่า มีบางกรณีที่มีการค้นหาแบบตาบุงเชิงปรับตัวไม่สามารถค้นเจอคำตอบที่ต้องการได้ ดังเช่น กรณีที่มีจำนวนคำตอบรอบข้างในการค้นหาเป็น 40 และรัศมีการค้นหา



รูปที่ 5.12 ผลการทดสอบเมื่อจำนวนคำตอบรอบข้างเป็น 40 และรัศมีการค้นหาเป็น 0.3

เป็น 0.3 ซึ่งจะเห็นได้ว่าแม้ทำการค้นหาจนครบ 2500 รอบแล้ว ค่าความคลาดเคลื่อนก็ไม่ลดต่ำลงตามต้องการ ดังแสดงได้ดังรูปที่ 5.12 ส่วนอีกกรณีที่คล้ายคลึงคือ เมื่อจำนวนคำตอบรอบข้างในการค้นหาเป็น 30 และรัศมีการค้นหาเป็น 0.3 จะพบว่าค่าความคลาดเคลื่อนที่น้อยที่สุดมีค่าเป็น 0.386682 ซึ่งเป็นค่าที่สูงเกินกว่าค่าที่กำหนด แม้ว่าการค้นหาแบบตาบู่เชิงปรับตัวจะดำเนินการค้นหาจนครบ 2500 รอบแล้วก็ตาม

ส่วนกรณีอื่นๆ ที่ปรากฏในตารางที่ 5.1 เป็นกรณี que การค้นหาแบบตาบู่เชิงปรับตัวสามารถค้นหาค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ที่ทำให้ค่าความคลาดเคลื่อนที่ได้จากฟังก์ชันวัตถุประสงค์มีค่าน้อยกว่า 0.35 แต่อาจต้องใช้จำนวนรอบในการค้นหาที่มากขึ้น ซึ่งส่งผลให้ต้องใช้เวลาในการค้นหาเพิ่มขึ้นด้วยเช่นกัน รูปที่ 5.13 เป็นตัวอย่างผลการทดสอบ กรณีที่จำนวนคำตอบรอบข้างในการค้นหาเป็น 50 และรัศมีการค้นหาเป็น 0.3 ซึ่งจะยุติการค้นหาเมื่อกระบวนการค้นหาคำเนินไปจนกระทั่งถึงรอบที่ 1264 ค่าความคลาดเคลื่อนจะลดลงเหลือ 0.329180 โดยใช้เวลาในการค้นหาทั้งสิ้น 131.75 วินาที



รูปที่ 5.13 ผลการทดสอบเมื่อจำนวนคำตอบรอบข้างเป็น 50 และรัศมีการค้นหาเป็น 0.3

จากผลการทดสอบเพื่อเลือกค่าพารามิเตอร์ของวิธีการค้นหาแบบตาบู่เชิงปรับตัว แสดงให้เห็นว่า เมื่อกำหนดให้จำนวนคำตอบรอบข้างในการค้นหาเป็น 40 และรัศมีการค้นหาเป็น 0.2 จะส่งผลให้การค้นหาแบบตาบู่เชิงปรับตัวสามารถเจอคำตอบได้รวดเร็วที่สุด ค่าพารามิเตอร์เริ่มต้นที่ได้จากการค้นหาดังกล่าว แสดงได้ดังตารางที่ 5.2 ซึ่งพารามิเตอร์  $W_{11}$  หมายถึง ค่าน้ำหนักที่เชื่อมโยงระหว่างเซลล์ประสาทเทียมตัวที่ 1 ของชั้นอินพุตไปยังเซลล์ประสาทเทียมตัวที่ 1 ของชั้นซ่อนเร้น ในขณะที่พารามิเตอร์  $W_{111}$  หมายถึง ค่าน้ำหนักที่เชื่อมโยงระหว่างเซลล์ประสาทเทียมตัว



ที่ 1 ของชั้นซ่อนเร้นไปยังเซลล์ประสาทเทียมตัวที่ 1 ของชั้นเอาต์พุต นอกจากนี้พารามิเตอร์  $\theta_{11}$  แทนด้วยค่าไบแอสที่เซลล์ประสาทเทียมตัวที่ 1 ของชั้นซ่อนเร้น ดังนั้น พารามิเตอร์  $\theta_{21}$  จึงแทนด้วยค่าไบแอสที่เซลล์ประสาทเทียมตัวที่ 1 ของชั้นเอาต์พุต เมื่อพิจารณาผลการค้นหาที่ปรากฏในตารางที่ 5.2 จะสังเกตเห็นว่า ค่าพารามิเตอร์เริ่มต้นเหล่านี้มีทั้งค่าบวกและค่าลบ โดยมีขอบเขตอยู่ในช่วง  $[-1, +1]$

ตารางที่ 5.2 พารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการค้นหาแบบตามูเชิงปรับตัว

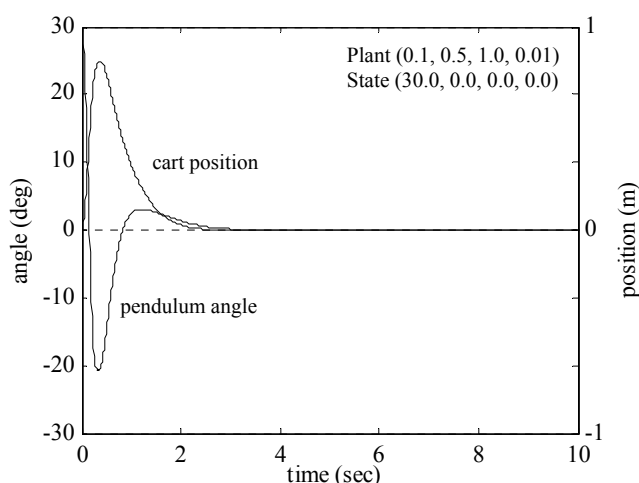
ค่าที่	พารามิเตอร์	ค่าที่ค้นหาได้	ค่าที่	พารามิเตอร์	ค่าที่ค้นหาได้
1	$W_{11}$	0.05828	17	$W_{121}$	-0.15122
2	$W_{12}$	0.70909	18	$W_{122}$	-0.03689
3	$W_{13}$	-0.52011	19	$W_{123}$	0.37800
4	$W_{14}$	0.47337	20	$W_{124}$	-0.82972
5	$W_{21}$	-0.32569	21	$W_{131}$	0.95896
6	$W_{22}$	-0.64555	22	$W_{132}$	0.07480
7	$W_{23}$	0.51375	23	$W_{133}$	-0.99927
8	$W_{24}$	-0.43546	24	$W_{134}$	-0.53417
9	$\theta_{11}$	-0.75969	25	$W_{141}$	-0.07386
10	$\theta_{12}$	-0.83846	26	$W_{142}$	-0.13282
11	$\theta_{13}$	0.85940	27	$W_{143}$	0.42250
12	$\theta_{14}$	-0.69855	28	$W_{144}$	-0.83317
13	$W_{111}$	-0.75564	29	$\theta_{21}$	-0.13758
14	$W_{112}$	-0.32193	30	$\theta_{22}$	0.17864
15	$W_{113}$	0.06556	31	$\theta_{23}$	-0.17942
16	$W_{114}$	-0.66350	32	$\theta_{24}$	-0.24478

## 5.6 การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

เพื่อทดสอบสมรรถนะและประสิทธิผลของตัวควบคุมนิวโร-ตามู-ฟิชชี ที่ปรับปรุงมาจากตัวควบคุมนิวโร-ฟิชชี จึงได้ดำเนินการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนสถานการณ์ที่ควบคุม ทั้งนี้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ใช้ในกระบวนการเรียนรู้เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะกับแต่ละสถานการณ์ พิจารณาได้จากตารางที่ 5.2 ซึ่งสถานการณ์ที่ใช้สำหรับการค้นหาค่าพารามิเตอร์เริ่มต้นดังกล่าวให้กับโครงข่าย

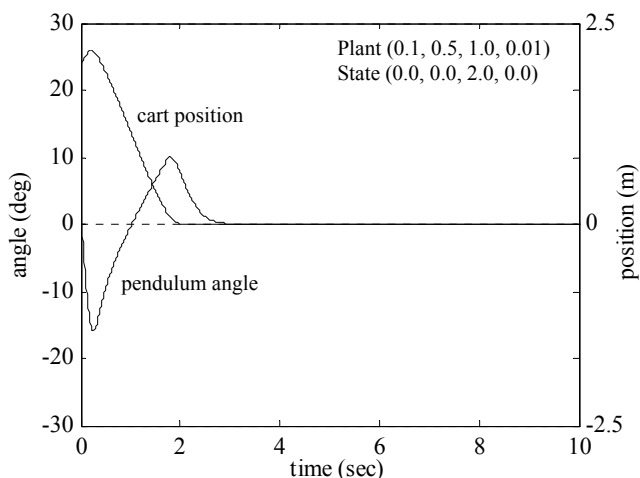
ประสาทเทียม กำหนดให้มุมเริ่มต้นของลูกตุ้มผกผันเป็น 30 องศา ก้านลูกตุ้มยาว 1.0 เมตร โดยตัวรถอยู่ที่ตำแหน่งกึ่งกลางของรางลื่น ซึ่งเมื่อสถานการณ์ในการควบคุมเปลี่ยนแปลงไป ตัวควบคุมนิวโร-ตาบิว-ฟิชซึ่งยังคงใช้ค่าพารามิเตอร์เริ่มต้นชุดเดียวกันนี้เป็นค่าเริ่มต้นให้กับโครงข่ายประสาทเทียมตลอดไป โดยกระบวนการเรียนรู้แบบแพร่กลับจะปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับแต่ละสถานการณ์ควบคุม

รูปที่ 5.14 เป็นผลการจำลองสถานการณ์เมื่อ มุมเริ่มต้นเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร โดยที่ก้านลูกตุ้มยาว 1.0 เมตร จากผลการจำลองสถานการณ์แสดงให้เห็นว่า ตัวรถจะเคลื่อนที่ไปทางด้านขวาของรางลื่น ส่งผลให้ลูกตุ้มหมุนในทิศทวนเข็มนาฬิกา ทำให้มุมของลูกตุ้มมีค่าเป็นลบ จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ทำให้ลูกตุ้มในทิศตามเข็มนาฬิกา กลับมาตั้งตรงที่ตำแหน่งสมดุล พร้อมทั้งตัวรถก็สามารถเคลื่อนที่กลับมายังตำแหน่งเริ่มต้นได้อีกครั้ง ซึ่งใช้เวลาในการสร้างเสถียรภาพทั้งสิ้น 3.58 วินาที



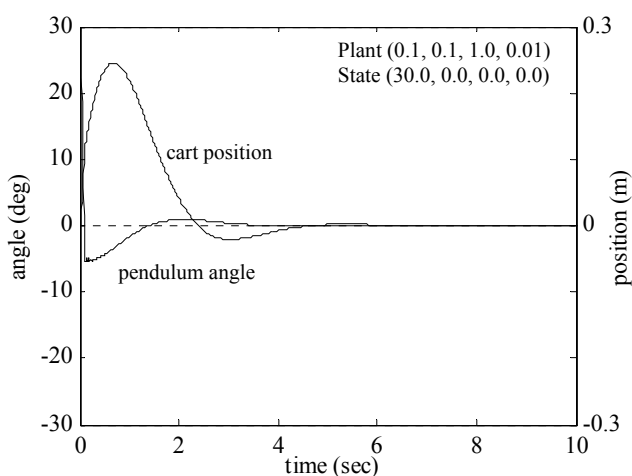
รูปที่ 5.14 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 1.0 เมตร

รูปที่ 5.15 เป็นผลการจำลองสถานการณ์เมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร โดยที่ก้านลูกตุ้มตั้งตรงที่ตำแหน่งสมดุลศูนย์องศา จากผลการจำลองสถานการณ์จะสังเกตเห็นว่า ตัวรถจะเคลื่อนที่ต่อไปทางด้านขวาล็กน้อย เพื่อให้ก้านลูกตุ้มหมุนในทิศทวนเข็มนาฬิกา ทำให้มุมของลูกตุ้มเป็นลบ จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ส่งผลให้ลูกตุ้มหมุนกลับในทิศตามเข็มนาฬิกาด้วยเช่นกัน จนกระทั่งลูกตุ้มสามารถตั้งตรงที่ตำแหน่งสมดุลได้อีกครั้ง พร้อมกันนี้ตัวรถก็เคลื่อนที่กลับมายังกึ่งกลางของรางลื่น ซึ่งใช้เวลาในการเข้าสู่เสถียรภาพทั้งสิ้น 3.30 วินาที

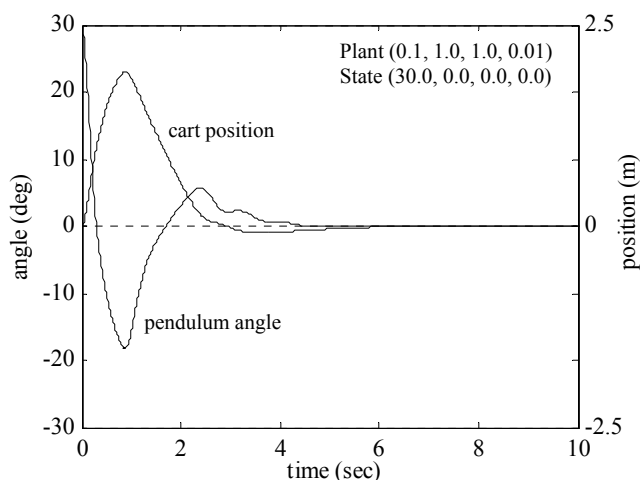


รูปที่ 5.15 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร

รูปที่ 5.16 เป็นผลการจำลองสถานการณ์ เมื่อปรับลดความยาวก้านลูกตุ้มเป็น 0.2 เมตร โดยที่มุมเริ่มต้นของลูกตุ้มยังคงเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถอยู่ที่กึ่งกลางของรางเลื่อน ซึ่งตัวควบคุมนิวโร-ตาบิว-ฟัซซี่ใช้เวลาในการสร้างเสถียรภาพให้ระบบประมาณ 4.48 วินาที ส่วนรูปที่ 5.17 เป็นผลการจำลองสถานการณ์ เมื่อปรับเพิ่มความยาวของก้านลูกตุ้มเป็น 2.0 เมตร ซึ่งจะได้ว่าตัวควบคุมดังกล่าวใช้เวลาสร้างเสถียรภาพให้ระบบทั้งสิ้น 6.31 วินาที นอกจากนี้ เมื่อพิจารณาผลการจำลองสถานการณ์ที่ปรากฏ พบว่ากระบวนการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวยังคงมีลักษณะเช่นเดียวกับผลการจำลองสถานการณ์ในรูปที่ 5.14

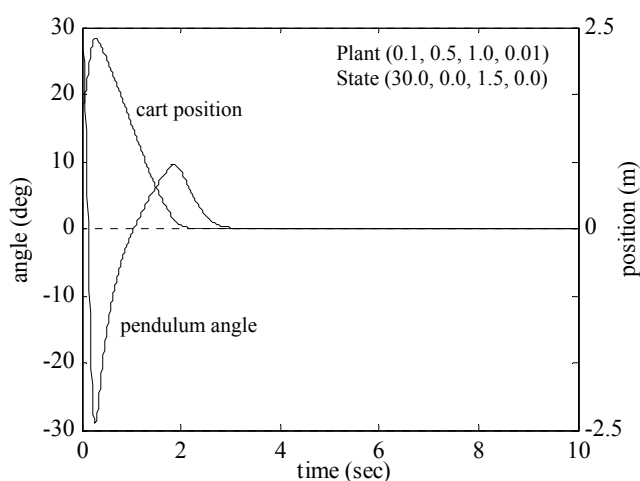


รูปที่ 5.16 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 0.2 เมตร



รูปที่ 5.17 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และก้านลูกตุ้มยาว 2.0 เมตร

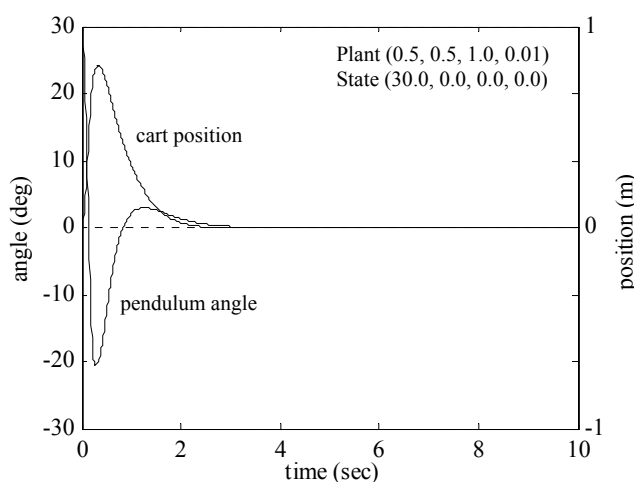
ผลการจำลองสถานการณ์เมื่อกำหนดให้มุมเริ่มต้นของลูกตุ้มผกผันเป็น 30 องศา โดยที่ตัวรถอยู่ห่างจากกึ่งกลางของรางเลื่อนไปทางด้านขวาเป็นระยะทาง 1.5 เมตร และก้านลูกตุ้มยาว 1.0 เมตร แสดงได้ด้วยรูปที่ 5.18 ซึ่งตัวควบคุมนิวโร-ตาบิว-ฟิชชี ใช้เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวทั้งสิ้น 3.36 วินาที โดยจะสังเกตเห็นว่า ในช่วงแรกตัวรถจะเคลื่อนที่ต่อไปทางด้านขวา ทำให้ลูกตุ้มหมุนทวนเข็มนาฬิกา ส่งผลให้มุมของลูกตุ้มเปลี่ยนจากค่าบวกเป็นลบ จากนั้นตัวรถจึงเคลื่อนที่กลับมายังกึ่งกลางของรางเลื่อน ส่งผลให้ก้านลูกตุ้มหมุนกลับในทิศตามเข็มนาฬิกา จนกระทั่งมุมของลูกตุ้มมีค่าเป็นบวกใกล้เคียงกับตำแหน่งสมดุล เมื่อพิจารณาผลการ



รูปที่ 5.18 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร

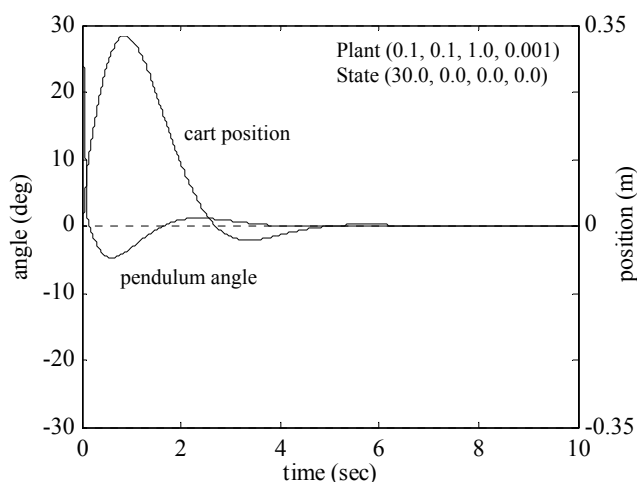
จำลองสถานการณ์ที่ปรากฏในรูปที่ 5.18 จะสังเกตเห็นว่า ในขณะที่ตัวรถจะเคลื่อนที่ห่างจากตำแหน่งกึ่งกลางของรางลื่นเพียงเล็กน้อยเท่านั้น ส่งผลให้ลูกตุ้มหมุนกลับในทิศทางเข็มนาฬิกา และสามารถกลับมาตั้งตรงที่ตำแหน่งสมดุลได้อีกครั้ง

จากนั้นได้ดำเนินการจำลองสถานการณ์ เมื่อปรับเปลี่ยนมวลของก้านลูกตุ้มเพิ่มขึ้นเป็น 0.5 กิโลกรัม โดยที่ก้านลูกตุ้มยังคงยาว 1.0 เมตร มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา และตำแหน่งเริ่มต้นของตัวรถอยู่ที่กึ่งกลางของรางลื่น ผลการจำลองสถานการณ์แสดงดังรูปที่ 5.19 ซึ่งตัวควบคุมนิวโร-ตาบิว-ฟิชชี สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในเวลาประมาณ 3.55 วินาที โดยจะพบว่า ผลการจำลองสถานการณ์ดังกล่าวใกล้เคียงผลการจำลองสถานการณ์ที่ปรากฏในรูปที่ 5.14 ซึ่งเป็นกรณีที่มีมวลของก้านลูกตุ้มมีค่าเป็น 0.1 กิโลกรัม ซึ่งในกรณีดังกล่าวตัวควบคุมนิวโร-ตาบิว-ฟิชชี ใช้เวลาในการสร้างเสถียรภาพให้ระบบประมาณ 3.58 วินาที ดังนั้น จะเห็นว่า เมื่อมวลของก้านลูกตุ้มมีค่าเพิ่มขึ้นจากเดิมห้าเท่า ตัวควบคุมนิวโร-ตาบิว-ฟิชชี ยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาที่ใกล้เคียงกับเวลาเดิม



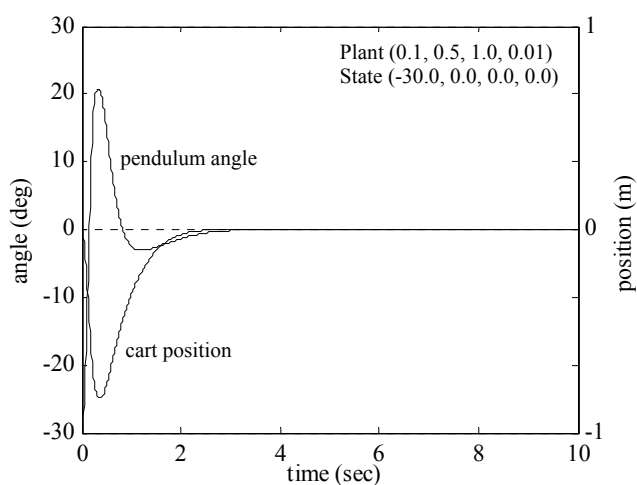
รูปที่ 5.19 ผลตอบสนองเมื่อปรับเพิ่มมวลของก้านลูกตุ้มเป็น 0.5 กิโลกรัม

ต่อมาเป็นการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อปรับลดช่วงเวลาแซมปลิงเป็น 0.001 วินาที โดยที่ก้านลูกตุ้มยาว 0.2 เมตร ผลการจำลองสถานการณ์ดังกล่าวแสดงได้ดังรูปที่ 5.20 ซึ่งเวลาที่ตัวควบคุมนิวโร-ตาบิว-ฟิชชีใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวคือ 4.877 วินาที เมื่อพิจารณาผลการจำลองสถานการณ์ที่ได้ พบว่าคล้ายคลึงกับผลการจำลองสถานการณ์ในรูปที่ 5.16 ซึ่งเป็นกรณีที่ช่วงเวลาแซมปลิงมีค่าเป็น 0.01 วินาที

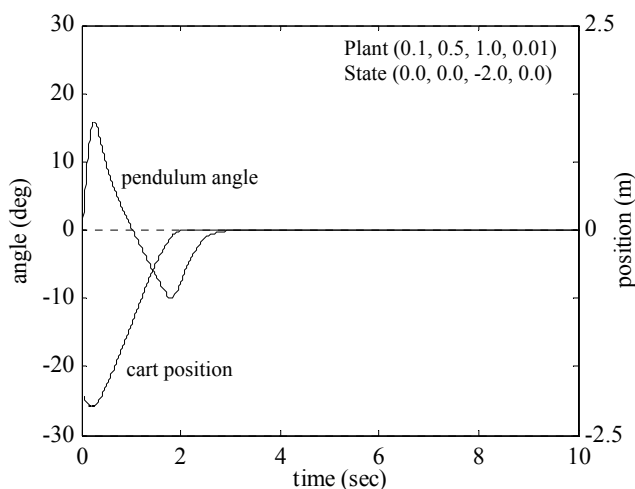


รูปที่ 5.20 ผลตอบสนองเมื่อปรับลดช่วงเวลาแซมปลิงเป็น 0.001 วินาที

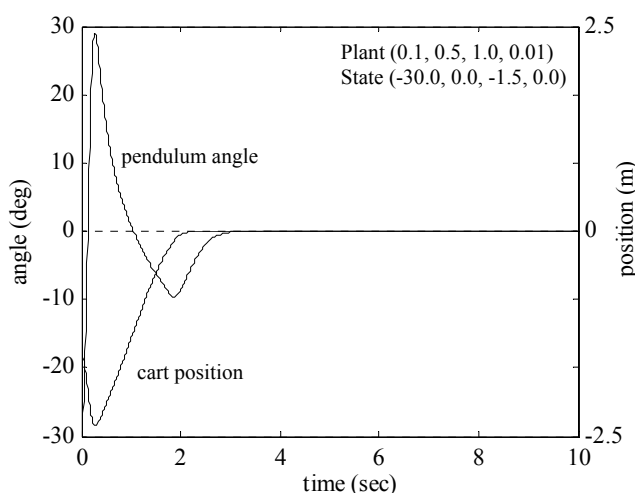
จากนั้นดำเนินการจำลองสถานการณ์ โดยกำหนดให้มุมเริ่มต้น และตำแหน่งเริ่มต้นของตัวรถ มีทิศทางข้ามกับที่ได้จำลองสถานการณ์ไว้แล้ว ทั้งนี้จะสร้างเสถียรภาพให้ระบบ 3 กรณี โดยกรณีแรกกำหนดให้มุมเริ่มต้นเป็น -30 องศา โดยที่ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร กรณีต่อมากำหนดให้ตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร โดยที่มุมเริ่มต้นของลูกตุ้มเป็นศูนย์องศา ส่วนกรณีสุดท้ายกำหนดให้มุมเริ่มต้นเป็น -30 องศา และตำแหน่งเริ่มต้นของตัวรถอยู่ที่ระยะ -1.5 เมตร ซึ่งทุกกรณีก้านลูกตุ้มยาว 1.0 เมตร ผลการจำลองสถานการณ์ในแต่ละกรณีแสดงได้ดังรูปที่ 5.21, 5.22 และ 5.23 ตามลำดับ



รูปที่ 5.21 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา



รูปที่ 5.22 ผลตอบสนองเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น -2.0 เมตร



รูปที่ 5.23 ผลตอบสนองเมื่อมุมเริ่มต้นเป็น -30 องศา และตำแหน่งเริ่มต้นของตัวรถเป็น -1.5 เมตร

เมื่อพิจารณาผลการจำลองสถานการณ์จากทั้ง 3 กรณี ดังปรากฏในรูปที่ 5.21, 5.22 และ 5.23 ตามลำดับ จะสังเกตเห็นว่า กระบวนการในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันยังคงมีลักษณะเช่นเดียวกับกรณีที่กำหนดให้มุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถมีทิศทางเป็นบวก เพียงแต่กระบวนการดังกล่าวเกิดขึ้นในทิศทางตรงข้ามกันเท่านั้น และเมื่อพิจารณาเวลาในการเข้าสู่เสถียรภาพของระบบผกผันแบบข้อต่อเดียวพบว่ายังคงมีค่าเท่ากับกรณีกำหนดให้มุมเริ่มต้นของลูกตุ้มและตำแหน่งเริ่มต้นของตัวรถมีค่าเป็นบวกด้วย โดยแต่ละกรณีใช้เวลาในการเข้าสู่เสถียรภาพทั้งสิ้น 3.58, 3.30 และ 3.36 วินาที ตามลำดับ

จากผลการจำลองสถานการณ์ที่ผ่านมาแสดงให้เห็นว่า ตัวควบคุมนิวโร-ตาบู-ฟิซซี่ สามารถรักษาเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น แม้ว่าสถานการณ์ในการควบคุมจะแปรเปลี่ยนไป และเมื่อพิจารณาค่าระดับความสำคัญพลวัตในแต่ละสถานการณ์ตามที่ปรากฏในตารางที่ 5.3 พบว่า เมื่อปรับเปลี่ยนสถานการณ์ในการควบคุม โครงข่ายประสาทเทียมจะทำการเรียนรู้ เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะกับสถานการณ์ที่เปลี่ยนแปลงไป

ตารางที่ 5.3 ค่าระดับความสำคัญพลวัตของตัวควบคุมนิวโร-ตาบู-ฟิซซี่ในสถานการณ์ต่างๆ

สถานการณ์	ค่าระดับความสำคัญพลวัต			
	DID-1	DID-2	DID-3	DID-4
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	5.9686	4.7109	6.2816	2.5000
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 0.2 เมตร	5.4942	4.1589	3.4246	1.6518
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 2.0 เมตร	5.5179	4.6784	2.6988	1.5249
มุมเริ่มต้น 0 องศา ตำแหน่งเริ่มต้น 2.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.2455	5.4648	4.0284	2.6881
มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 1.5 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.0271	5.3959	2.9930	2.4095
มุมเริ่มต้น -30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	5.9686	4.7109	6.2816	2.5000
มุมเริ่มต้น 0 องศา ตำแหน่งเริ่มต้น -2.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.2455	5.4648	4.0284	2.6881
มุมเริ่มต้น -30 องศา ตำแหน่งเริ่มต้น -1.5 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	6.0271	5.3959	2.9930	2.4095



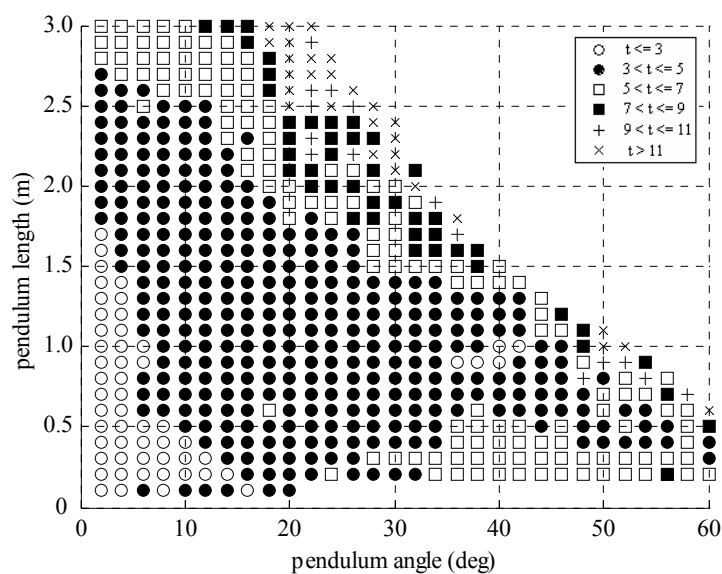
นอกจากนี้ เมื่อสถานการณ์ที่ควบคุมกำหนดขึ้นในทิศทางตรงข้ามกัน เช่น กรณีมุมเริ่มต้นของ ลูกตุ้มเป็น +30 และ -30 องศา โดยตำแหน่งเริ่มต้นของตัวรถอยู่ที่จุดเดียวกัน รวมทั้งก้านลูกตุ้มยาวเท่ากัน จะสังเกตเห็นได้ว่า ค่าระดับความสำคัญพลวัตแต่ละตัวมีค่าเท่ากัน ดังนั้นเวลาในการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียวจึงเท่ากันด้วย

เมื่อพิจารณาเวลาในการสร้างเสถียรภาพให้แก่ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ตาบู-ฟิชชี โดยดำเนินการปรับเปลี่ยนสถานการณ์ที่ควบคุม ตัวอย่างเช่น ปรับเปลี่ยนความยาวก้านลูกตุ้ม ตำแหน่งมุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถ เป็นต้น จะพบว่า ตัวควบคุมดังกล่าวยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น ดังแสดงในตารางที่ 5.4 ซึ่งเวลาในการเข้าสู่เสถียรภาพเหล่านี้พิจารณาได้จากผลการจำลองสถานการณ์ที่แสดงไว้ในรูปที่ 5.14, 5.15, 5.16, 5.17 และ 5.18 ตามลำดับ ทั้งนี้ สถานการณ์ต่างๆ ทั้ง 5 สถานการณ์ที่ปรากฏในตารางที่ 5.4 ได้กำหนดให้มุมเริ่มต้นของลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถมีทิศทางเป็นบวกเท่านั้น เนื่องจากการกำหนดค่าเริ่มต้นเหล่านี้ให้มีทิศทางเป็นลบ ตัวควบคุมนิวโร-ตาบู-ฟิชชี ยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ด้วยเวลาที่เท่ากัน

ตารางที่ 5.4 เวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมนิวโร-ตาบู-ฟิชชี

สถานการณ์	เวลาในการเข้าสู่เสถียรภาพ (วินาที)
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	3.58
มุมเริ่มต้นของลูกตุ้มเป็น 0.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	3.30
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 0.2 เมตร	4.48
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร ความยาวของก้านลูกตุ้มเป็น 2.0 เมตร	6.31
มุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา ตำแหน่งเริ่มต้นของตัวรถเป็น 1.5 เมตร ความยาวของก้านลูกตุ้มเป็น 1.0 เมตร	3.36

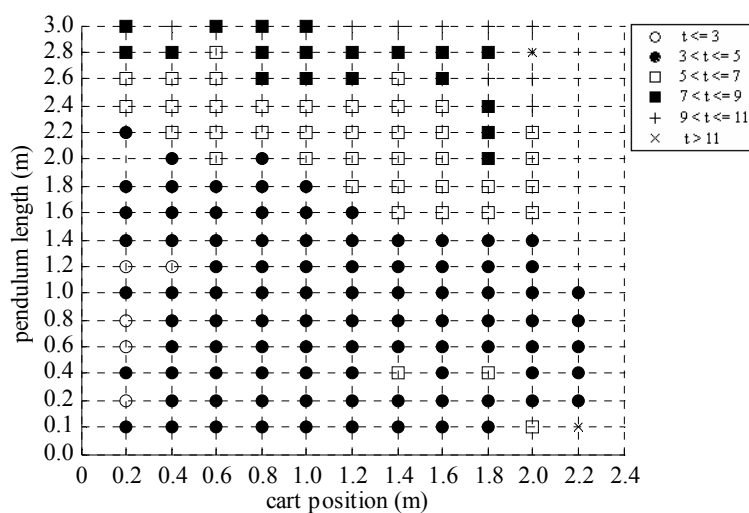
เพื่อทดสอบสมรรถนะของตัวควบคุมนิวโร-ตาบ-พีชชี ได้ดำเนินการปรับเปลี่ยนสถานการณ์ในการควบคุม โดยปรับเพิ่มมุมเริ่มต้นของลูกตุ้มครั้งละ 2 องศา โดยเริ่มจาก 2 องศา จนกระทั่งถึง 60 องศา และที่ทุกตำแหน่งมุมเริ่มต้นที่แปรเปลี่ยนไป ทำการปรับเพิ่มความยาวก้านลูกตุ้มครั้ง 0.1 เมตร โดยเริ่มต้นที่ก้านลูกตุ้มยาว 0.1 เมตร จนกระทั่งถึง 3.0 เมตร ทั้งนี้กำหนดให้ตำแหน่งเริ่มต้นของตัวรถอยู่ตรงกึ่งกลางของรางลื่น ผลการทดสอบแสดงได้ดังรูปที่ 5.24 โดยสัญลักษณ์ต่างๆ ที่ปรากฏในรูปแทนด้วยช่วงเวลาในตัวควบคุมนิวโร-ตาบ-พีชชี ใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ตัวอย่างเช่น กรณีมุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ก้านลูกตุ้มยาว 1.0 เมตร เวลาที่ระบบเข้าสู่เสถียรภาพคือ 3.58 วินาที ดังนั้นจึงแทนด้วยสัญลักษณ์ ● เช่นเดียวกับกรณีที่มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา และก้านลูกตุ้มยาว 0.2 เมตร ซึ่งใช้เวลาในการเข้าสู่เสถียรภาพ 4.48 วินาที ส่วนกรณีที่ก้านลูกตุ้มยาวเพิ่มขึ้นเป็น 2.0 เมตร ตัวควบคุมนิวโร-ตาบ-พีชชี สามารถสร้างเสถียรภาพให้ระบบได้ภายในเวลา 6.31 วินาที ดังนั้นจึงแทนด้วยสัญลักษณ์ □ เป็นต้น ส่วนบริเวณที่ไม่ปรากฏสัญลักษณ์ใดๆ หมายถึง ตัวควบคุมไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้



รูปที่ 5.24 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม

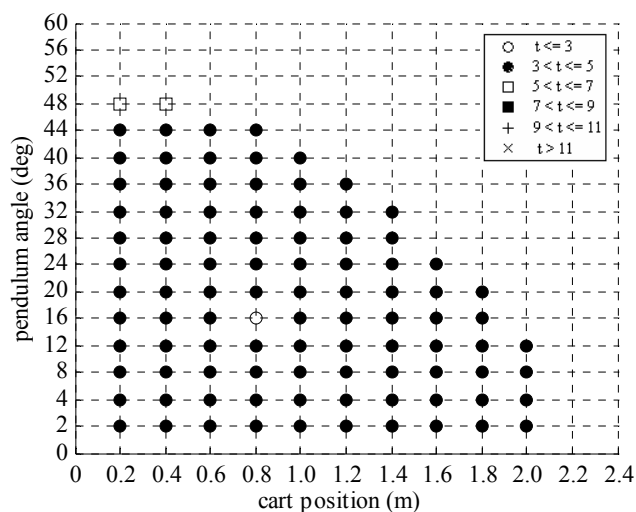
เมื่อพิจารณาผลการทดสอบในรูปที่ 5.24 ประกอบ จะสังเกตเห็นว่า ตัวควบคุมนิวโร-ตาบ-พีชชี สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น กล่าวคือ โดยส่วนใหญ่แล้วใช้เวลาน้อยกว่าหรือเท่ากับ 5 วินาที

รูปที่ 5.25 แสดงผลการทดสอบตัวควบคุมนิวโร-ตาบู-พีซซี เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับปรับเปลี่ยนความยาวก้านลูกตุ้ม โดยทำการเพิ่มตำแหน่งเริ่มต้นของตัวรถไปทางด้านขวาครั้งละ 0.2 เมตร เริ่มจากตำแหน่ง 0.2 เมตร จนกระทั่งถึงปลายสุดคือที่ตำแหน่ง 2.4 เมตร พร้อมทั้งดำเนินการปรับเพิ่มความยาวก้านลูกตุ้มครั้งละ 0.2 เมตร จนกระทั่งก้านลูกตุ้มยาว 3.0 เมตร ทั้งนี้กำหนดให้มุมเริ่มต้นของลูกตุ้มอยู่ที่ตำแหน่งศูนย์องศา จากผลการทดสอบพบว่า ตัวควบคุมที่กล่าวถึงยังคงสามารถสร้างเสถียรภาพให้ระบบได้ แม้ว่าก้านลูกตุ้มจะยาวมากถึง 3.0 เมตร โดยที่ตำแหน่งเริ่มต้นของตัวรถอยู่ภายในระยะ 2.0 เมตร แต่ถ้าตำแหน่งเริ่มต้นของตัวรถอยู่ที่ปลายสุดของรางเส้น ตัวควบคุมดังกล่าวจะไม่สามารถสร้างเสถียรภาพให้ระบบได้ เนื่องจากในกระบวนการสร้างเสถียรภาพให้แกระบบ จำเป็นต้องทำให้ตัวรถเคลื่อนที่ต่อไปทางด้านขวา เพื่อให้ลูกตุ้มเสถียรสมดุล ซึ่งจะส่งผลให้ตัวรถเคลื่อนที่เกินความยาวของรางเส้นที่กำหนดไว้



รูปที่ 5.25 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม

จากนั้นได้ดำเนินการทดสอบการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว โดยปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถ พร้อมทั้งปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม เมื่อก้านลูกตุ้มยาว 1.0 เมตร ผลการทดสอบดังกล่าวแสดงได้ดังรูปที่ 5.26 ซึ่งจะสังเกตเห็นว่า ตัวควบคุมนิวโร-ตาบู-พีซซี สามารถสร้างเสถียรภาพให้ระบบได้ภายในเวลา 5 วินาที โดยขอบเขตในการสร้างเสถียรภาพให้ระบบได้สำเร็จนั้นแคบลง เนื่องจากเมื่อตำแหน่งเริ่มต้นของตัวรถและมุมเริ่มต้นลูกตุ้มเพิ่มขึ้น ส่งผลให้ตัวรถต้องเคลื่อนที่ออกห่างจากรางเส้นมากขึ้น จนกระทั่งเกินความยาวของรางเส้นที่กำหนดไว้ 2.4 เมตร ซึ่งกรณีนี้จะพิจารณาว่าตัวควบคุมไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้



รูปที่ 5.26 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม

เมื่อพิจารณาเวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ตาบู-ฟัซซี่ โดยปรับเปลี่ยนมุมเริ่มต้นลูกตุ้มพร้อมกับความยาวก้านลูกตุ้ม ดังแสดงในรูปที่ 5.24 และปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับปรับเปลี่ยนความยาวก้านลูกตุ้ม ดังแสดงในรูปที่ 5.25 รวมทั้งได้ปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถและมุมเริ่มต้นของลูกตุ้ม ซึ่งผลการทดสอบดังกล่าวปรากฏในรูปที่ 5.26 จะสังเกตเห็นว่า ตัวควบคุมนิวโร-ตาบู-ฟัซซี่ ยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น รวมถึงสามารถสร้างเสถียรภาพให้ระบบได้ในขอบเขตที่กว้าง แม้จะดำเนินการแปรเปลี่ยนสถานการณ์ที่ควบคุม

## 5.7 การทดสอบตัวควบคุมนิวโร-ตาบู-ฟัซซี่

เพื่อศึกษาถึงผลกระทบของค่าพารามิเตอร์เริ่มต้นสำหรับการเรียนรู้ของโครงข่ายประสาทเทียม ที่ได้จากวิธีการค้นหาแบบตาบูเชิงปรับตัว จึงดำเนินการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ตาบู-ฟัซซี่ โดยทดสอบการสร้างเสถียรภาพให้ระบบ เมื่อปรับเปลี่ยนสถานการณ์การควบคุมดังปรากฏในตารางที่ 5.5 ทั้งนี้ทุกสถานการณ์จะใช้ค่าพารามิเตอร์เริ่มต้นชุดเดียวกัน ซึ่งได้จากวิธีการค้นหาแบบตาบูเชิงปรับตัว เมื่อกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร และก้านลูกตุ้มยาว 1.0 เมตร การทดสอบดังกล่าวจะดำเนินการค้นหาค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียมจำนวน 4000 ครั้ง ดังนั้นจึงได้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมจำนวน 4000 ชุด จากนั้นจึงนำค่าพารามิเตอร์เริ่มต้นเหล่านี้ไปใช้ในการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ในสถานการณ์ต่างๆ ตามที่ได้กำหนดไว้ ผลการทดสอบการสร้างเสถียรภาพ

ให้ระบบด้วยตัวควบคุมนิวโร-ตาบ-พีซีซึ่งแสดงได้ดังตารางที่ 5.5 โดยจะสังเกตเห็นว่า ในกรณีที่ 1 และ กรณีที่ 3 ซึ่งกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์ เมตร โดยก้านลูกตุ้มยาว 1.0 เมตร และ 0.2 เมตร ตามลำดับ ปริมาณการเข้าสู่เสถียรภาพเป็น 100% หมายความว่า ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากวิธีการค้นหาแบบตาบูเซิงปรับตัวทั้ง 4000 ชุด สามารถสร้างเสถียรภาพให้ระบบได้สำเร็จครบทั้ง 4000 ชุด นั่นคือ โครงข่ายประสาทเทียมสามารถเรียนรู้โดยใช้ค่าพารามิเตอร์เริ่มต้นเหล่านี้ในการปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับสถานการณ์ในกรณีที่ 1 และกรณีที่ 3 ในขณะที่ผลการทดสอบในกรณีที่ 2 และกรณีที่ 4 จะได้ว่า ปริมาณการเข้าสู่เสถียรภาพของระบบคิดเป็น 97.73% และ 94.68% ตามลำดับ ซึ่งแสดงว่า ค่าพารามิเตอร์เริ่มต้นบางชุดที่ได้จากวิธีการค้นหาแบบตาบูเซิงปรับตัวส่งผลให้โครงข่ายประสาทเทียมไม่สามารถเรียนรู้เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับสถานการณ์เหล่านั้นได้ ส่วนสถานการณ์สุดท้าย พบว่า ปริมาณการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียวคิดเป็น 74.75%

ตารางที่ 5.5 การทดสอบเมื่อใช้ค่าพารามิเตอร์เริ่มต้นที่ได้จากการค้นหาแบบตาบูเซิงปรับตัว

กรณีที่	สถานการณ์	ปริมาณการเข้าสู่เสถียรภาพ	
		4000 ครั้ง	10000 ครั้ง
1	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	100.0%	100.0%
2	มุมเริ่มต้น 0 องศา ตำแหน่งเริ่มต้น 2.0 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	97.73%	97.77%
3	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 0.2 เมตร	100.0%	100.0%
4	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 0.0 เมตร ก้านลูกตุ้มยาว 2.0 เมตร	94.68%	95.02%
5	มุมเริ่มต้น 30 องศา ตำแหน่งเริ่มต้น 1.5 เมตร ก้านลูกตุ้มยาว 1.0 เมตร	74.75%	75.26%

นอกจากนี้ได้ทำการค้นหาค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียม ด้วยวิธีการค้นหาแบบตาบอดเชิงปรับตัวเพิ่มขึ้นเป็น 10000 ครั้ง เพื่อให้ผลการทดสอบของตัวควบคุมนิวโร-ตาบอด-พีชชี มีความน่าเชื่อถือยิ่งขึ้น ซึ่งจะพิจารณาปริมาณการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียวเมื่อสถานการณ์การควบคุมเปลี่ยนไป โดยยังคงดำเนินการทดสอบค่าพารามิเตอร์เริ่มต้นเหล่านี้เช่นเดียวกันกับการทดสอบค่าพารามิเตอร์เริ่มต้นที่ได้จากวิธีการค้นหาแบบตาบอดเชิงปรับตัว ทั้ง 4000 ครั้ง ดังที่ได้อธิบายรายละเอียดไว้แล้ว ผลการทดสอบดังกล่าวแสดงได้ดังตารางที่ 5.5 ซึ่งพบว่า ปริมาณการเข้าสู่เสถียรภาพของระบบที่ได้จากการทดสอบทั้ง 10000 ครั้ง นั้นใกล้เคียงกับผลการทดสอบในตอนแรก ซึ่งทดสอบจากค่าพารามิเตอร์เริ่มต้นจำนวน 4000 ครั้ง โดยปริมาณการเข้าสู่เสถียรภาพของระบบในแต่ละสถานการณ์ค่อนข้างสูงเป็นที่พึงพอใจ ดังนั้นอาจกล่าวได้ว่าค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากวิธีการค้นหาแบบตาบอดเชิงปรับตัว ช่วยให้ตัวควบคุมนิวโร-ตาบอด-พีชชี สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ แม้สถานการณ์การควบคุมจะแปรเปลี่ยนไป

## 5.8 สรุป

ตัวควบคุมนิวโร-ตาบอด-พีชชี เป็นวิธีการควบคุมที่ปรับปรุงมาจากตัวควบคุมนิวโร-พีชชี ประกอบด้วยโครงข่ายประสาทเทียม พีชชีลอจิก และการค้นหาแบบตาบอดเชิงปรับตัว เพื่อลดผลกระทบจากค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสมของโครงข่ายประสาทเทียมที่ได้จากการสุ่มค่า ซึ่งค่าพารามิเตอร์เริ่มต้นเหล่านี้ได้มาจากการค้นหาแบบตาบอดเชิงปรับตัว โดยสถานการณ์ที่ใช้ในการค้นหาค่าพารามิเตอร์เริ่มต้นดังกล่าวกำหนดให้มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร และก้านลูกตุ้มยาว 1.0 เมตร จากนั้นเมื่อสถานการณ์ที่ควบคุมเปลี่ยนแปลงไป (มุมเริ่มต้นของลูกตุ้ม ความยาวก้านลูกตุ้ม และตำแหน่งเริ่มต้นของตัวรถเปลี่ยน) โครงข่ายประสาทเทียมจะเรียนรู้เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับแต่ละสถานการณ์ที่ต้องการควบคุม ทั้งนี้ยังคงใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมเพียงชุดเดียวที่ได้จากการค้นหาแบบตาบอดเชิงปรับตัวสำหรับทุกสถานการณ์ จากผลการทดสอบแสดงให้เห็นว่า ตัวควบคุมนิวโร-ตาบอด-พีชชี สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น นอกจากนี้เมื่อพิจารณาผลการทดสอบการปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม ตำแหน่งของตัวรถ รวมทั้งความยาวของก้านลูกตุ้ม พบว่า ตัวควบคุมที่กล่าวถึงนี้สามารถสร้างเสถียรภาพให้ระบบได้ขอบเขตที่กว้างดังอธิบายไว้แล้วข้างต้น ซึ่งกรณีส่วนใหญ่ตัวควบคุมนิวโร-ตาบอด-พีชชี สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในเวลา 5 วินาที และจากการพิจารณาผลทดสอบค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากวิธีการค้นหาแบบตาบอดเชิงปรับตัวจำนวน 4000 ชุด ด้วยการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบ

ในแต่ละสถานการณ์จำนวน 4000 ครั้ง โดยปรับเปลี่ยนสถานการณ์การควบคุมทั้งสิ้น 5 สถานการณ์ ซึ่งในแต่ละสถานการณ์จะใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมชุดเดียวกัน พบว่า ตัวควบคุมนิวโร-ตาบู-ฟิซซี่ สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้อย่างน่าพึงพอใจ และเพื่อให้ผลการทดสอบดังกล่าวน่าเชื่อถือยิ่งขึ้นจึงได้ทำการค้นหาค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียมด้วยวิธีการค้นหาแบบตาบูเชิงปรับตัวเพิ่มขึ้นเป็นจำนวน 10000 ชุด โดยยังคงดำเนินการทดสอบค่าพารามิเตอร์เริ่มต้นเหล่านี้เช่นเดียวกันกับกรณีที่ทำการค้นหาค่าพารามิเตอร์เริ่มต้นเป็นจำนวน 4000 ชุด ซึ่งจากผลการทดสอบพบว่าปริมาณการเข้าสู่เสถียรภาพของระบบยังคงเป็นไปในแนวทางเดียวกัน ดังนั้นอาจกล่าวได้ว่า ตัวควบคุมนิวโร-ตาบู-ฟิซซี่ที่ได้รับการปรับปรุงโดยนำวิธีค้นหาแบบตาบูเชิงปรับตัวเข้าช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้โครงข่ายประสาทเทียม สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น โดยค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากวิธีการค้นหาแบบตาบูมีส่วนช่วยให้โครงข่ายประสาทเทียมสามารถเรียนรู้เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับแต่ละสถานการณ์ จึงทำให้ตัวควบคุมนิวโร-ตาบู-ฟิซซี่ ยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ แม้ว่าสถานการณ์ที่ควบคุมจะเปลี่ยนแปลงไป

## บทที่ 6

### อภิปรายสมรรถนะของตัวควบคุม

#### 6.1 บทนำ

เนื้อหาวิทยานิพนธ์ในบทที่ผ่านมาได้กล่าวถึง หลักการทำงานของตัวควบคุมแบบต่างๆ ในการสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว ซึ่งได้แก่ ตัวควบคุมพีชชีแบบกฎอินพุตเดียว ตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-ตาบู-พีชชี รวมถึงผลการจำลองสถานการณ์ที่ใช้ตัวควบคุมแบบต่างๆ เมื่อดำเนินการปรับเปลี่ยนสถานการณ์ในการควบคุมให้แตกต่างกัน ซึ่งจากผลการจำลองสถานการณ์ที่ได้แสดงให้เห็นว่า ตัวควบคุมแต่ละแบบใช้เวลาไม่เท่ากันในการสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว และในบางสถานการณ์ตัวควบคุมบางตัวก็อาจไม่สามารถสร้างเสถียรภาพให้ระบบได้ เมื่อพิจารณาเวลาในการสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว พบว่า ตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-ตาบู-พีชชี ที่พัฒนาขึ้นสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาที่สั้นกว่าการใช้ตัวควบคุมพีชชีแบบกฎอินพุตเดียว อย่างไรก็ตาม ในการพิจารณาสมรรถนะของตัวควบคุมนั้น นอกจากจะพิจารณาระยะเวลาในการเข้าสู่เสถียรภาพของระบบแล้ว จำเป็นต้องศึกษาถึงผลกระทบจากปัจจัยอื่นที่อาจส่งผลกระทบต่อสมรรถนะของตัวควบคุมด้วยเช่นกัน ในงานวิจัยวิทยานิพนธ์นี้ได้ดำเนินการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ ที่กล่าวถึงด้วยการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบ เมื่อปรับเปลี่ยนสถานการณ์ในการควบคุมโดยมีการรบกวนระบบจากภายนอกพร้อมด้วย ทั้งนี้เพื่อพิจารณาว่าตัวควบคุมแบบต่างๆ เหล่านี้ยังคงสามารถรักษาเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียวได้หรือไม่

เนื้อหาในบทนี้กล่าวถึง การเปรียบเทียบผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมแบบต่างๆ จากนั้นได้กล่าวถึงผลการทดสอบสมรรถนะของตัวควบคุม เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบที่ใช้ในการจำลองสถานการณ์ รวมทั้งแสดงผลการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ ในการสร้างเสถียรภาพให้ระบบ เมื่อมีการรบกวนจากภายนอก นอกจากนี้ยังนำเสนอผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบ ด้วยตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-ตาบู-พีชชี เพื่อพิจารณาผลกระทบจากค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสมของโครงข่ายประสาทเทียม ที่ส่งผลต่อการสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว



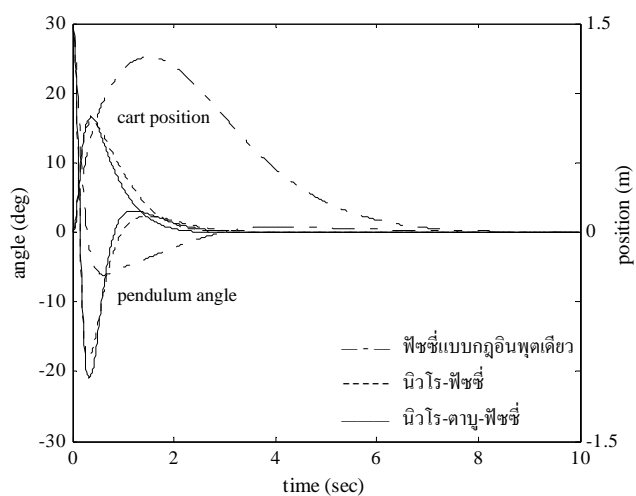
## 6.2 การสร้างเสถียรภาพให้ระบบลูกค้ำผกผันด้วยตัวควบคุมแบบต่างๆ

หลังจากที่ได้อธิบายรายละเอียดหลักการทำงานของตัวควบคุมพีซซีแบบกฎอินพุตเดียว ตัวควบคุมนิวโร-พีซซี และ ตัวควบคุมนิวโร-ดาบู-พีซซี พร้อมทั้งผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว ไว้ในบทที่ 3, 4 และ 5 ตามลำดับแล้วนั้น เพื่อพิจารณาเปรียบเทียบผลการจำลองสถานการณ์ในแต่ละสถานการณ์ที่ได้จากตัวควบคุมแบบต่างๆ จึงได้ดำเนินการสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนมุมเริ่มต้นของลูกค้ำผกผัน ตำแหน่งเริ่มต้นของตัวรถ และความยาวของก้านลูกค้ำ รวมทั้งสิ้น 5 สถานการณ์ ทดสอบ ดังปรากฏในตารางที่ 6.1 ทั้งนี้ตัวควบคุมนิวโร-พีซซียังคงใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการสุ่มค่าชุดเดียวกันสำหรับทุกสถานการณ์ เช่นเดียวกับตัวควบคุมนิวโร-ดาบู-พีซซี ซึ่งใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมเพียงชุดเดียวที่ได้จากวิธีการค้นหาแบบดาบูเชิงปรับตัว

ตารางที่ 6.1 สถานการณ์ที่จัดตั้งให้มีการสร้างเสถียรภาพแก่ระบบลูกค้ำผกผันแบบข้อต่อเดียวด้วยตัวควบคุมแบบต่างๆ

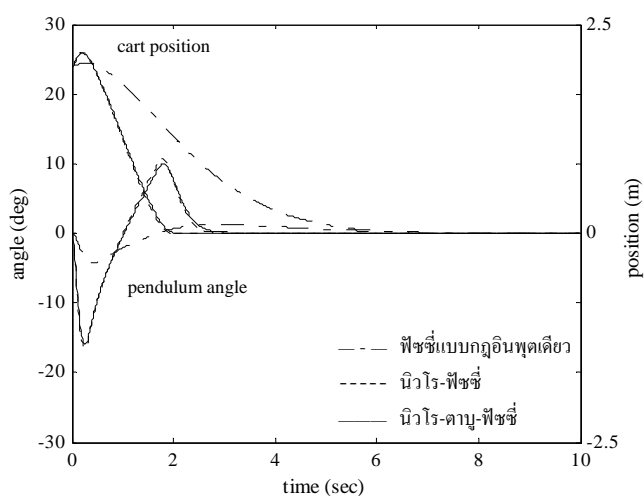
สถานการณ์ที่	มุมเริ่มต้น (องศา)	ตำแหน่งรถเริ่มต้น (เมตร)	ความยาวของก้านลูกค้ำ (เมตร)
1	30.0	0.0	1.0
2	0.00	2.0	1.0
3	30.0	0.0	0.2
4	30.0	0.0	2.0
5	30.0	1.5	1.0

ผลการจำลองสถานการณ์ในสถานการณ์แรกแสดงได้ดังรูปที่ 6.1 ซึ่งจะสังเกตเห็นว่า พลวัตของระบบลูกค้ำผกผันที่ได้จากตัวควบคุมแบบต่างๆ เป็นไปในแนวทางเดียวกัน กล่าวคือ ในลำดับแรกตัวรถจะเคลื่อนที่ไปทางด้านขวาของรางลื่น ทำให้ลูกค้ำหมุนในทิศทวนเข็มนาฬิกา จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ส่งผลให้ลูกค้ำหมุนในทิศตามเข็มนาฬิกา จนกระทั่งลูกค้ำสามารถตั้งตรงที่ตำแหน่งสมดุลศูนย์กลาง พร้อมกันนี้ตัวรถก็สามารถเคลื่อนที่กลับมายังกึ่งกลางของรางลื่นได้อีกครั้ง และเมื่อพิจารณาเวลาในการเข้าสู่เสถียรภาพของระบบ พบว่า ตัวควบคุมนิวโร-พีซซี และตัวควบคุมนิวโร-ดาบู-พีซซี สามารถสร้างเสถียรภาพให้ระบบในเวลาใกล้เคียงกัน นอกจากนี้ยังสามารถทำให้ระบบเข้าสู่เสถียรภาพได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมพีซซีแบบกฎอินพุตเดียวด้วย



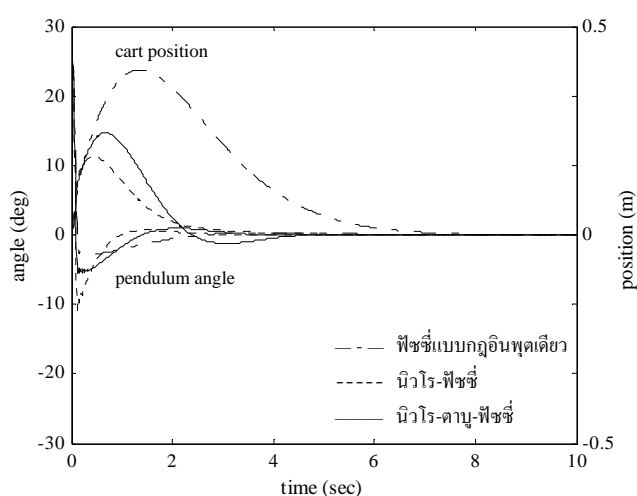
รูปที่ 6.1 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 1 ของตัวควบคุมแบบต่างๆ

ในสถานการณ์ต่อมา กำหนดให้ตำแหน่งเริ่มต้นของตัวรถอยู่ห่างจากกึ่งกลางของรางลื่นไปทางด้านขวาเป็นระยะทาง 2.0 เมตร โดยที่มุมเริ่มต้นของลูกตุ้มผกผันเป็นศูนย์องศา ผลการจำลองสถานการณ์แสดงได้ดังรูปที่ 6.2 ซึ่งจะพบว่าในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ตัวควบคุมแบบต่างๆ ที่กล่าวถึงจะควบคุมให้ตัวรถเคลื่อนที่ต่อไปทางด้านขวาเล็กน้อย เพื่อให้มุมของลูกตุ้มเสถียรสมดุลไม่เป็นศูนย์องศา จากนั้นจึงดำเนินการสร้างเสถียรภาพให้ระบบโดยควบคุมให้ตัวรถเคลื่อนที่กลับมายังด้านซ้ายของรางลื่น จนกระทั่งสามารถทำให้ก้านลูกตุ้มตั้งตรงและตัวรถเคลื่อนที่กลับมายังกึ่งกลางของรางลื่น

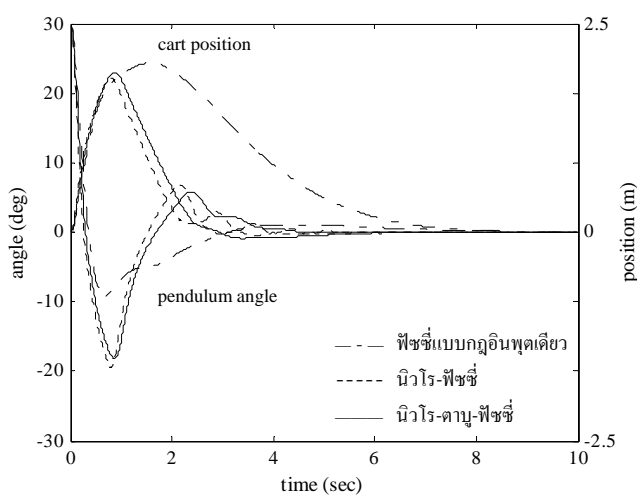


รูปที่ 6.2 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 2 ของตัวควบคุมแบบต่างๆ

รูปที่ 6.3 และ 6.4 เป็นผลการจำลองสถานการณ์ที่ 3 และสถานการณ์ที่ 4 ตามลำดับ ซึ่งได้ดำเนินการปรับเปลี่ยนความยาวของก้านลูกตุ้ม เมื่อพิจารณาผลการจำลองสถานการณ์ดังกล่าว จะสังเกตเห็นว่ากระบวนการในการสร้างเสถียรภาพให้ระบบเป็นไปในลักษณะเดียวกับผลการจำลองสถานการณ์ในรูปที่ 6.1 ในกรณีที่ก้านลูกตุ้มยาวไม่มากนัก ตัวรถจะเคลื่อนออกจากตำแหน่งกึ่งกลางไปเพียงระยะทางสั้นๆ จากนั้นจึงเคลื่อนที่กลับมาที่เดิม ดังแสดงในรูปที่ 6.3 ถ้าก้านลูกตุ้มยาวมากขึ้น ตัวรถจำเป็นต้องเคลื่อนที่ออกห่างจากรางเส้นเป็นระยะทางที่มากกว่า จึงจะสามารถสร้างเสถียรภาพให้ระบบได้ ดังแสดงในรูปที่ 6.4

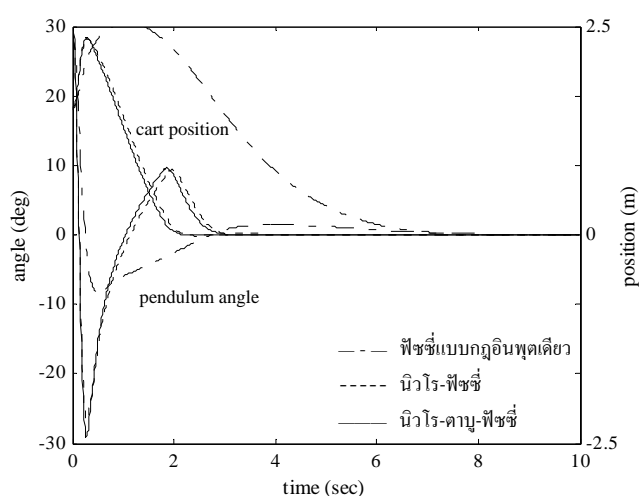


รูปที่ 6.3 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 3 ของตัวควบคุมแบบต่างๆ



รูปที่ 6.4 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 4 ของตัวควบคุมแบบต่างๆ

ผลการจำลองสถานการณ์ในสถานการณ์สุดท้าย แสดงได้ดังรูปที่ 6.5 ซึ่งจะพบว่าตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ เนื่องจากตัวรถเคลื่อนที่เกินความยาวของรางลื่นที่กำหนดได้คือ  $[-2.4, +2.4]$  เมตร และเมื่อพิจารณาตำแหน่งของตัวรถจะสังเกตเห็นว่า ในช่วงแรกตัวควบคุมแบบต่างๆ จะควบคุมให้ตัวรถเคลื่อนที่ต่อไปทางด้านขวาของรางลื่น เพื่อทำให้ลูกตุ้มหมุนในทิศทวนเข็มนาฬิกา จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ส่งผลให้ลูกตุ้มหมุนในทิศตามเข็มนาฬิกาจนกระทั่งสามารถตั้งตรง และตัวรถหยุดที่ตำแหน่งกึ่งกลางของรางลื่น



รูปที่ 6.5 การเปรียบเทียบผลการจำลองสถานการณ์ที่ 5 ของตัวควบคุมแบบต่างๆ

เมื่อพิจารณาผลการจำลองสถานการณ์ในแต่ละสถานการณ์จะสังเกตเห็นว่า ในช่วงแรกขณะที่มุมของลูกตุ้มเปลี่ยนจากมุมบวกเป็นมุมลบนั้น มุมของลูกตุ้มจากตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวจะมีค่าเป็นลบน้อยกว่าการควบคุมด้วยตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบู-ฟัซซี่ และเมื่อพิจารณาการเคลื่อนที่ของตัวรถ พบว่า ตัวรถของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว จะเคลื่อนที่ออกห่างจากกึ่งกลางของรางลื่นเป็นระยะทางที่มากกว่าการควบคุมด้วยตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบู-ฟัซซี่ ซึ่งต้องใช้เวลาค่อนข้างมาก เพื่อให้ตัวรถเคลื่อนที่กลับมายังตำแหน่งกึ่งกลางได้อีกครั้ง ดังนั้นตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบู-ฟัซซี่ จึงสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว นอกจากนี้พบว่าพลวัตของระบบในบางสถานการณ์ของตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบู-ฟัซซี่จะใกล้เคียงกัน ทั้งนี้เนื่องจากค่าระดับความสำคัญพลวัตที่ได้จากการเรียนรู้ของโครงข่ายประสาทเทียมของตัวควบคุมทั้งสองมีค่าใกล้เคียงกัน ดังแสดงในตารางที่ 6.2

ตารางที่ 6.2 ค่าระดับความสำคัญพลวัตของตัวควบคุมนิรโทษชี้ และตัวควบคุมนิรโทษชี้-ตามู-พีชชี้

สถานการณ์	ตัวควบคุมนิรโทษชี้				ตัวควบคุมนิรโทษชี้-ตามู-พีชชี้			
	DID-1	DID-2	DID-3	DID-4	DID-1	DID-2	DID-3	DID-4
1	6.5946	5.1130	5.8458	2.5734	5.9686	4.7109	6.2816	2.5000
2	6.4701	5.8249	4.3502	2.9421	6.2455	5.4648	4.0284	2.6881
3	5.6961	4.0936	2.4867	1.8806	5.4942	4.1589	3.4246	1.6518
4	7.1303	6.5145	3.9356	2.3777	5.5179	4.6784	2.6988	1.5249
5	6.0634	5.7232	3.1463	2.4189	6.0271	5.3959	2.9930	2.4095

เมื่อพิจารณาเวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมพีชชี้แบบกฎอินพุตเดียว ตัวควบคุมนิรโทษชี้ และตัวควบคุมนิรโทษชี้-ตามู-พีชชี้ เมื่อสถานการณ์ในการควบคุมเปลี่ยนแปลงไป แสดงได้ดังตารางที่ 6.3 โดยเวลาในการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมต่างๆ เหล่านี้ได้นำเสนอไว้ในตารางที่ 3.7 ตารางที่ 4.4 และตารางที่ 5.4 ตามลำดับ ซึ่งพบว่าตัวควบคุมนิรโทษชี้-ตามู-พีชชี้ สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมพีชชี้แบบกฎอินพุตเดียวสำหรับทุกสถานการณ์ และเมื่อพิจารณาปริมาณเวลาในการเข้าสู่เสถียรภาพที่สั้นกว่านี้ พบว่าอยู่ในช่วงประมาณ 30% ถึง 60% โดยสถานการณ์สุดท้ายคิดเป็น 100% เนื่องจากตัวควบคุมพีชชี้แบบกฎอินพุตเดียวไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ ตัวอย่างการคำนวณปริมาณเวลาเข้าสู่เสถียรภาพของตัวควบคุมนิรโทษชี้-ตามู-พีชชี้ที่สั้นกว่าตัวควบคุมพีชชี้แบบกฎอินพุตเดียวสำหรับสถานการณ์แรกสามารถแสดงได้ดังนี้

$$\frac{8.24 - 3.58}{8.24} \times 100 = 56.55\%$$

ตารางที่ 6.3 เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันด้วยตัวควบคุมแบบต่างๆ

สถานการณ์ที่	เวลา (วินาที)			ปริมาณเวลาที่สั้นกว่า	
	พีชชี้แบบกฎอินพุตเดียว	นิรโทษชี้	นิรโทษชี้-ตามู-พีชชี้	นิรโทษชี้	นิรโทษชี้-ตามู-พีชชี้
1	8.24	3.95	3.58	52.06%	56.55%
2	7.19	3.16	3.30	56.05%	54.10%
3	6.59	3.19	4.48	51.59%	32.02%
4	8.95	9.20	6.31	-2.79%	29.50%
5	-	4.06	3.36	100.0%	100.0%

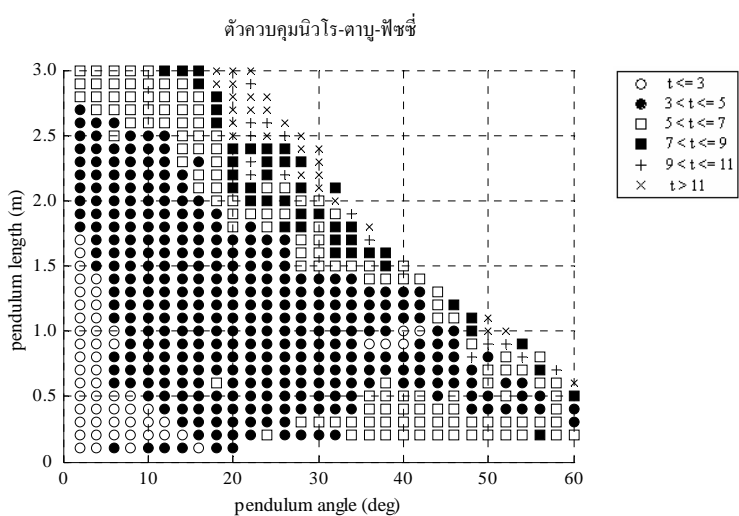
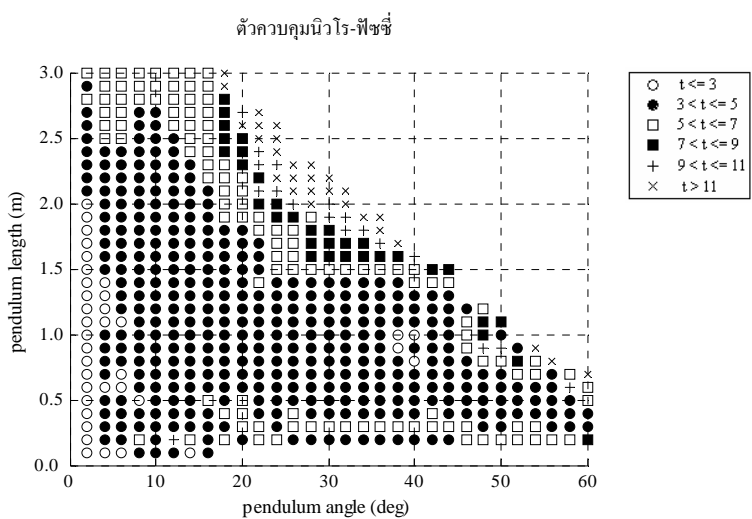
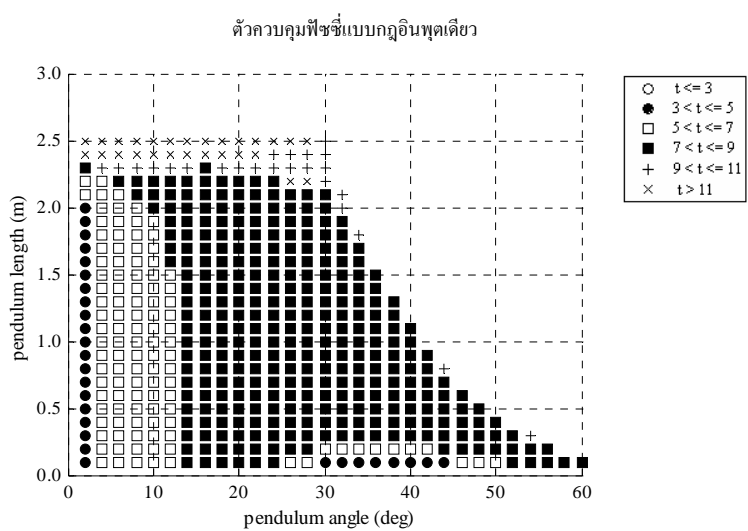
เมื่อพิจารณาเวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ฟัซซี่ พบว่า มีเพียงสถานการณ์เดียวที่ตัวควบคุมดังกล่าวใช้เวลาในการเข้าสู่เสถียรภาพมากกว่าตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว นั่นคือ ในสถานการณ์ที่ 4 เมื่อพิจารณาสถานการณ์อื่นๆ พบว่า เวลาที่ใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันด้วยตัวควบคุมนิวโร-ฟัซซี่ใกล้เคียงกับตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ ดังที่ได้ปรากฏในตารางที่ 6.3 ซึ่งปริมาณเวลาในการเข้าสู่เสถียรภาพที่สั้นกว่า เมื่อพิจารณาเปรียบเทียบกับตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวจะได้ประมาณ -3% ถึง 60% โดยในสถานการณ์สุดท้ายคิดเป็น 100% เนื่องจากตัวควบคุมนิวโร-ฟัซซี่ยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้

### 6.3 เวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันเมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบ

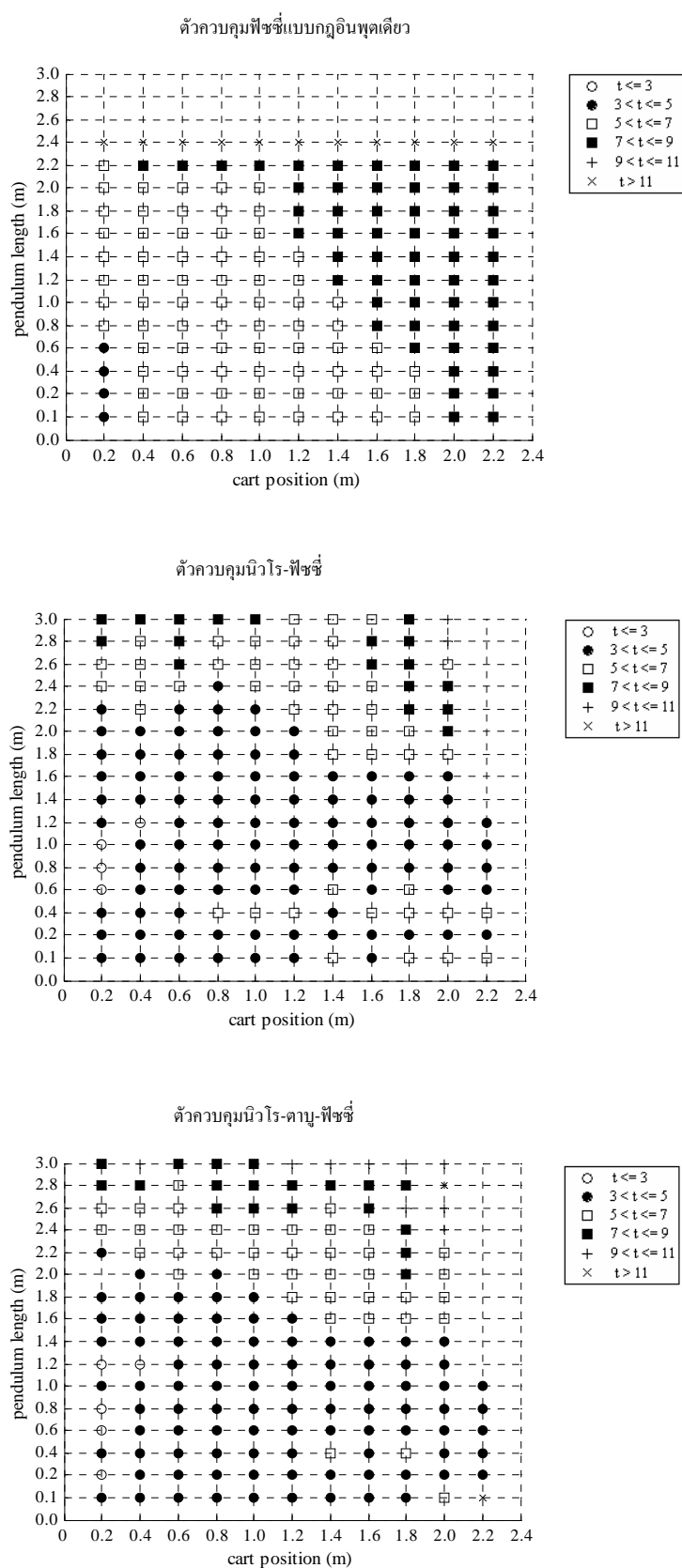
เพื่อพิจารณาสมรรถนะของตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว ตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ จึงได้ดำเนินการทดสอบการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวด้วยตัวควบคุมแบบต่างๆ เหล่านี้ เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบในการจำลองสถานการณ์ ซึ่งได้แก่ มุมเริ่มต้นของลูกตุ้ม ตำแหน่งเริ่มต้นของตัวรถ และความยาวก้านลูกตุ้ม ดังนั้นเพื่อพิจารณาเปรียบเทียบเวลาในการเข้าสู่เสถียรภาพของตัวควบคุมแบบต่างๆ จึงได้นำผลการทดสอบดังกล่าวมาแสดงรวมกันอีกครั้งในรูปที่ 6.6, 6.7 และ 6.8 ทั้งนี้สัญลักษณ์ต่างๆ ที่ปรากฏในรูปแบบแทนด้วยช่วงเวลาที่ตัวควบคุมใช้ในการสร้างเสถียรภาพให้ระบบเช่นเดียวกับที่ได้กล่าวไว้แล้ว

เมื่อพิจารณารูปที่ 6.6 ซึ่งเป็นกรณีที่ปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม พร้อมกับปรับเปลี่ยนความยาวก้านลูกตุ้ม โดยที่ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร จะสังเกตเห็นว่า แม้จะปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้มและความยาวของก้านลูกตุ้มเพิ่มขึ้น ตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ ยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว กล่าวคือ เวลาโดยส่วนใหญ่ที่ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันคือประมาณ 9 วินาที ในขณะที่ตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ จะใช้เวลาในการสร้างเสถียรภาพให้ระบบได้ภายในเวลา 5 วินาที

ผลการทดสอบที่ปรากฏในรูปที่ 6.7 เป็นกรณีที่ปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับความยาวก้านลูกตุ้ม โดยมุมเริ่มต้นของลูกตุ้มเป็นศูนย์องศา ส่วนรูปที่ 6.8 เป็นกรณีที่ดำเนินการปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับมุมเริ่มต้นของลูกตุ้ม โดยก้านลูกตุ้มยาว 1.0 เมตร จากผลการทดสอบดังกล่าว แสดงให้เห็นว่า ตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบได้รวดเร็วกว่าตัวควบคุมฟัซซี่แบบกฎอินพุตเดียวเช่นกัน

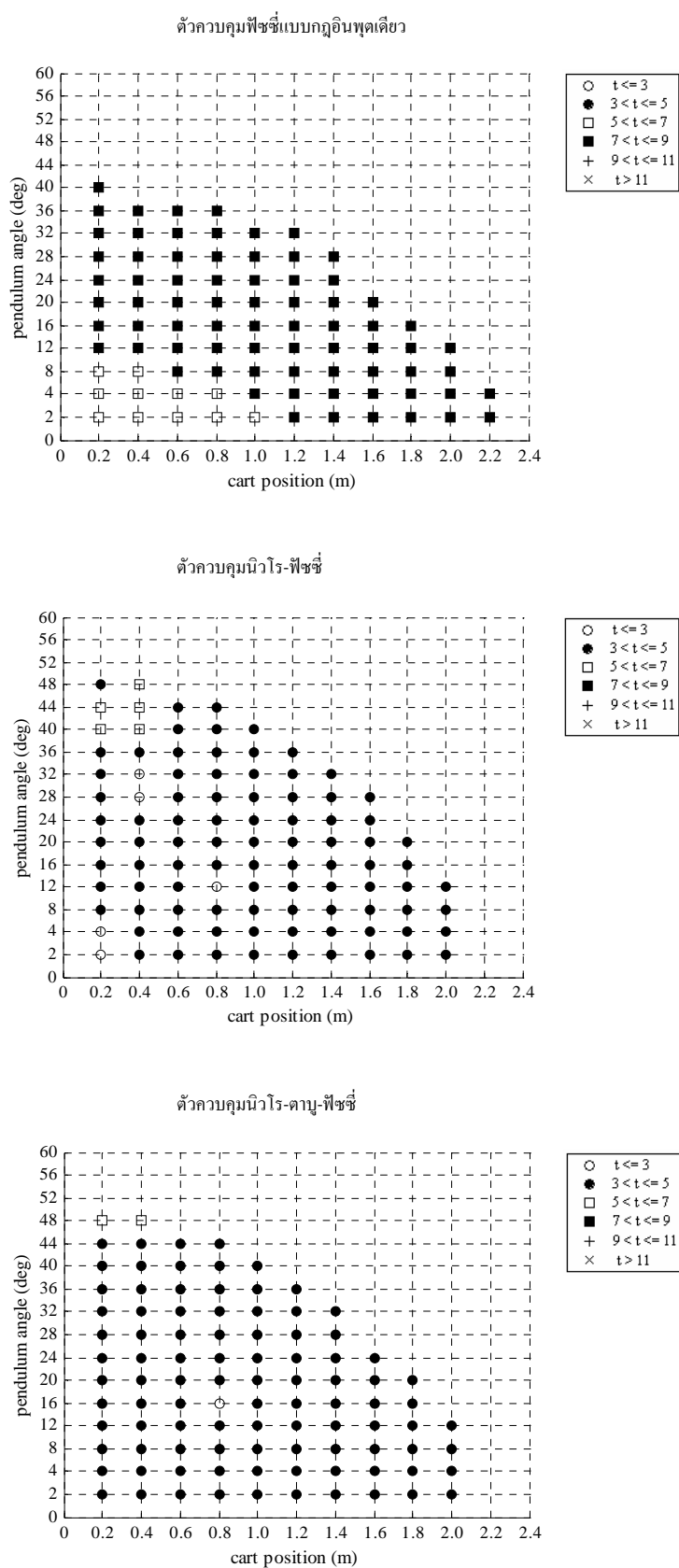


รูปที่ 6.6 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม



รูปที่ 6.7 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม



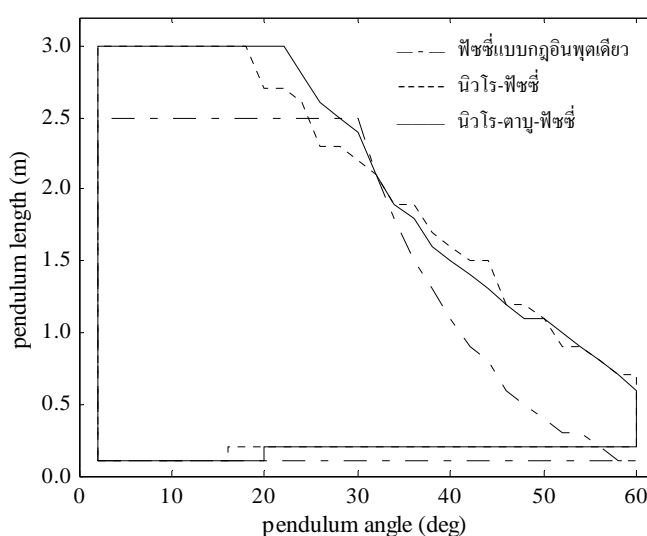


รูปที่ 6.8 เวลาเข้าสู่เสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม

นอกจากนี้เมื่อพิจารณาผลการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ ในภาคผนวก จ. ซึ่งเป็นกรณีที่ดำเนินการปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้มพร้อมกับความยาวก้านลูกตุ้ม โดยที่ตำแหน่งเริ่มต้นของตัวรถไม่ได้อยู่ที่กึ่งกลางของรางลื่น และกรณีที่ปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถกับความยาวก้านลูกตุ้ม โดยที่มุมเริ่มต้นลูกตุ้มไม่เป็นศูนย์องศา รวมทั้งได้ทำการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับปรับเปลี่ยนมุมเริ่มต้นลูกตุ้ม โดยที่ก้านลูกตุ้มยาว 0.2 เมตร และ 2.0 เมตร ซึ่งจากผลการทดสอบดังกล่าว แสดงให้เห็นว่า ตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-ตาบู-ฟิชชี ยังคงสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟิชชีแบบกฎอินพุตเดียว

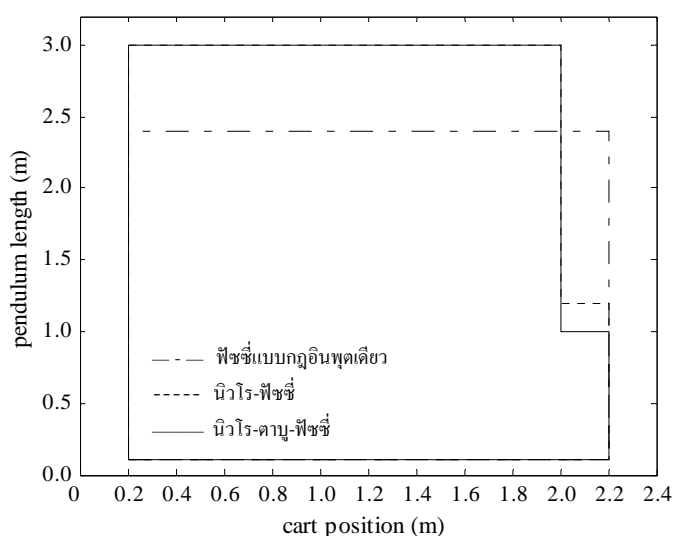
#### 6.4 ขอบเขตการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันของตัวควบคุมแบบต่างๆ

หลังจากที่ได้พิจารณาเปรียบเทียบเวลาที่ตัวควบคุมแบบต่างๆ ใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบในการจำลองสถานการณ์แล้ว ในลำดับต่อมาได้กล่าวถึง การพิจารณาเปรียบเทียบขอบเขตการแปรเปลี่ยนค่าเริ่มต้นของระบบที่ตัวควบคุมฟิชชีแบบกฎอินพุตเดียว ตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-ตาบู-ฟิชชี ยังคงสามารถสร้างเสถียรภาพให้ระบบได้ ทั้งนี้ยังคงใช้ผลการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ ดังที่ได้แสดงไว้ในรูปที่ 6.6, 6.7 และ 6.8 ดังนั้น เพื่อให้เห็นขอบเขตความสามารถของตัวควบคุมแบบต่างๆ เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบได้อย่างชัดเจนยิ่งขึ้น จึงได้แสดงขอบเขตดังกล่าวไว้ในรูปเดียวกัน ดังปรากฏในรูปที่ 6.9, 6.10 และ 6.11



รูปที่ 6.9 ขอบเขตการสร้างเสถียรภาพเมื่อปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านลูกตุ้ม

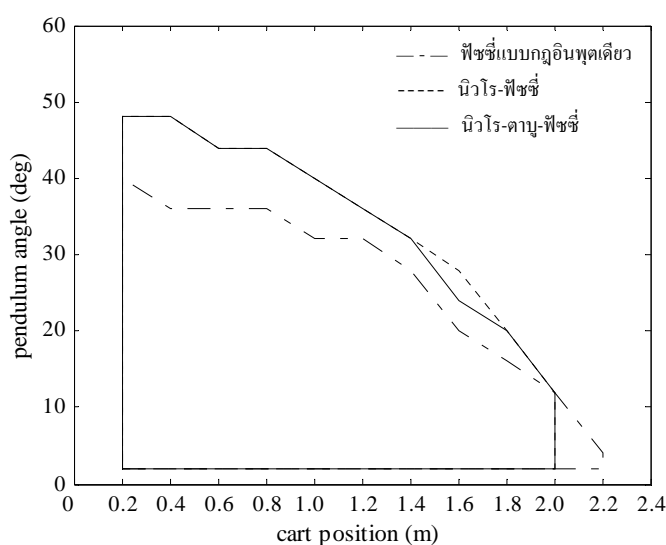
รูปที่ 6.9 แสดงขอบเขตการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ด้วยตัวควบคุมแบบต่างๆ เมื่อปรับเปลี่ยนมุมเริ่มต้นลูกตุ้มพร้อมกับปรับเปลี่ยนความยาวก้านลูกตุ้ม ซึ่งจะเห็นได้ว่า เมื่อมุมเริ่มต้นของลูกตุ้มและความยาวก้านลูกตุ้มมีค่าเพิ่มขึ้น ตัวควบคุมแบบต่างๆ จะสามารถสร้างเสถียรภาพให้ระบบได้ในขอบเขตที่แคบลง ในกรณีของตัวควบคุมฟิชชีแบบกฎอินพุตเดียวจะสามารถสร้างเสถียรภาพให้ระบบได้ก็ต่อเมื่อก้านลูกตุ้มมีความยาวไม่เกิน 2.5 เมตร ในขณะที่ตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-คาบิว-ฟิชชี สามารถสร้างเสถียรภาพให้ระบบได้จนกระทั่งก้านลูกตุ้มยาวถึง 3.0 เมตร แต่กรณีที่ก้านลูกตุ้มมีความยาวน้อยมากคือยาวเพียง 0.1 เมตร พบว่า เมื่อมุมเริ่มต้นของลูกตุ้มเพิ่มขึ้น ตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-คาบิว-ฟิชชี จะไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ ดังนั้นเมื่อพิจารณารูปที่ 6.9 อาจกล่าวได้ว่า ตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-คาบิว-ฟิชชี มีขอบเขตการสร้างเสถียรภาพให้ระบบที่กว้างกว่าตัวควบคุมฟิชชีแบบกฎอินพุตเดี่ยวก่อนข้างมาก และจากการพิจารณาขอบเขตการสร้างเสถียรภาพให้ระบบของตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-คาบิว-ฟิชชี พบว่าตัวควบคุมทั้งสองมีขอบเขตความสามารถในการสร้างเสถียรภาพให้ระบบที่ใกล้เคียงกันมาก



รูปที่ 6.10 ขอบเขตการสร้างเสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและความยาวก้านลูกตุ้ม

รูปที่ 6.10 แสดงขอบเขตของตัวควบคุมแบบต่างๆ เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับปรับเปลี่ยนความยาวก้านลูกตุ้ม ส่วนรูปที่ 6.11 แสดงขอบเขตการสร้างเสถียรภาพให้กับระบบเมื่อปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม ซึ่งจะสังเกตเห็นว่า ตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-คาบิว-ฟิชชี มีขอบเขตในการสร้าง

เสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้กว้างกว่าตัวควบคุมพีชชีแบบกฎอินพุตเดียวเช่นกัน โดยที่ตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-คาบิว-พีชชี มีขอบเขตที่ใกล้เคียงกัน และเมื่อพิจารณารูปที่ 6.10 จะพบว่าตัวควบคุมทั้งสองสามารถสร้างเสถียรภาพให้ระบบได้จนกระทั่งก้านลูกตุ้มยาวถึง 3.0 เมตร โดยที่ตำแหน่งเริ่มต้นของตัวรถอยู่ห่างจากรางลื่นเป็นระยะ 2.0 เมตร ในขณะที่ตัวควบคุมพีชชีแบบกฎอินพุตเดียวสามารถปรับเพิ่มความยาวก้านลูกตุ้มได้เพียง 2.4 เมตร ส่วนรูปที่ 6.11 แสดงผลว่าตัวควบคุมทั้งสองที่พัฒนาขึ้นนี้ สามารถสร้างเสถียรภาพให้ระบบได้จนกระทั่งมุมเริ่มต้นของลูกตุ้มสูงถึง 48 องศา



รูปที่ 6.11 ขอบเขตการสร้างเสถียรภาพเมื่อปรับเปลี่ยนตำแหน่งรถเริ่มต้นและมุมเริ่มต้นของลูกตุ้ม

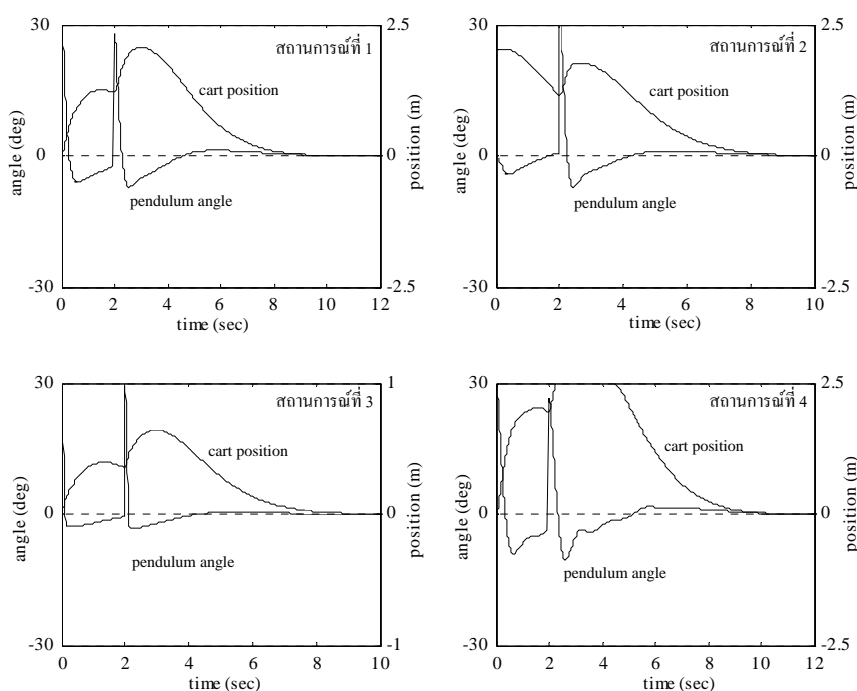
## 6.5 การทดสอบสมรรถนะของตัวควบคุมเมื่อมีการรบกวนจากภายนอก

การทดสอบสมรรถนะของตัวควบคุมพีชชีแบบกฎอินพุตเดียว ตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-คาบิว-พีชชี เพื่อสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อมีการรบกวนระบบจากภายนอก ได้ดำเนินการเป็นสองกรณี ได้แก่ กรณีการรบกวนเกิดในรูปของมุมลูกตุ้มผกผันขณะที่ตัวควบคุมกำลังทำงานสร้างเสถียรภาพ และอีกกรณีหนึ่ง คือมีการรบกวนเกิดขึ้นในรูปของมุมลูกตุ้มผกผันเช่นกัน โดยเกิดหลังจากที่ตัวควบคุมสร้างเสถียรภาพให้ระบบเสร็จสมบูรณ์แล้ว การรบกวนถูกกำหนดให้เกิดขึ้นที่เวลา 2 และ 10 วินาที ตามลำดับ ด้วยขนาดของมุมเป็น 30 องศา โดยในการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อมีการรบกวนระบบจากภายนอก กำหนดให้ใช้สถานการณ์ที่ 1 จนถึงสถานการณ์ที่ 4 เป็นสถานการณ์ทดสอบ ดังได้แสดงไว้ในตารางที่ 6.1

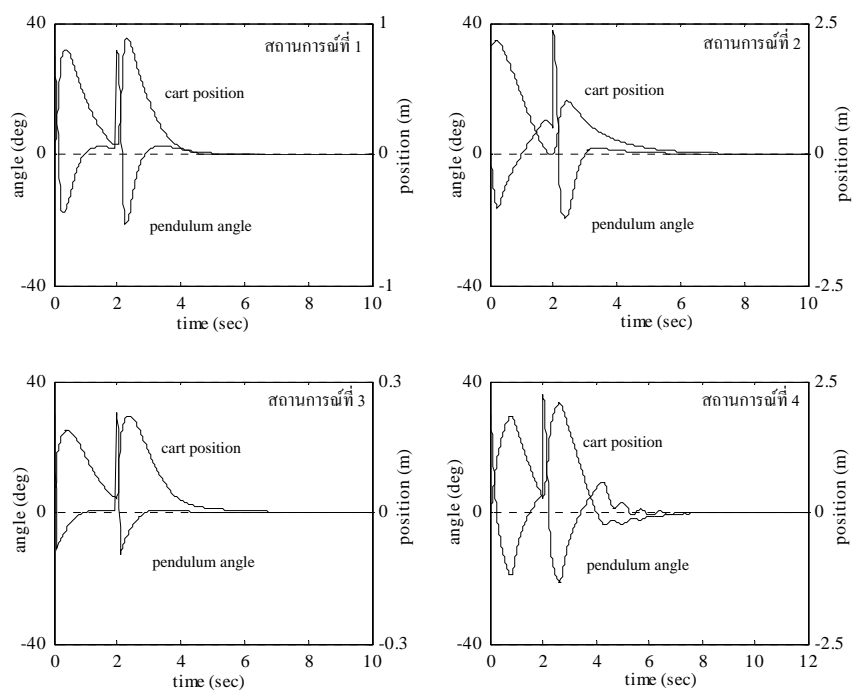
### 6.5.1 การรบกวนระบบที่เวลา 2 วินาที

ผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวด้วยตัวควบคุมพีชชีแบบกฎอินพุตเดียว ตัวควบคุมนิโร-พีชชี และตัวควบคุมนิโร-ตามู-พีชชี เมื่อมีการรบกวนด้วยมุม 30 องศา ที่เวลา 2 วินาที แสดงได้ดังรูปที่ 6.12, 6.13 และ 6.14 ตามลำดับ

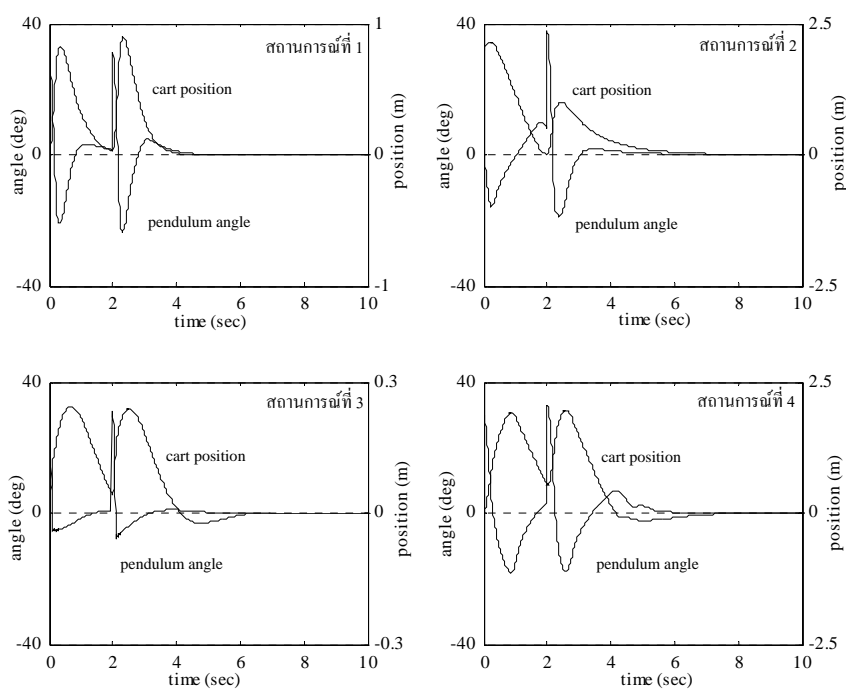
เมื่อพิจารณาสถานการณ์แรกในรูปที่ 6.12 จะพบว่าในขณะที่ตัวควบคุมพีชชีแบบกฎอินพุตเดียวกำลังสร้างเสถียรภาพให้ระบบอยู่นั้น ได้มีมุมรบกวนจากภายนอกมากระทำต่อระบบ ทำให้ที่เวลา 2 วินาทีมุมของลูกตุ้มมีค่าเพิ่มขึ้นจากเดิม 30 องศา ส่งผลให้มุมของลูกตุ้มมีค่าเป็นบวก ซึ่งในการสร้างเสถียรภาพให้ระบบ จำเป็นต้องควบคุมให้ตัวรถเคลื่อนที่ไปทางด้านขวา เพื่อให้ลูกตุ้มหมุนในทิศทวนเข็มนาฬิกา จนกระทั่งมุมของลูกตุ้มมีค่าเป็นลบ จากนั้นตัวรถจึงเคลื่อนที่กลับมาทางด้านซ้าย ส่งผลให้ลูกตุ้มหมุนกลับในทิศตามเข็มนาฬิกาจนสามารถตั้งตรงที่ตำแหน่งศูนย์องศา พร้อมกันนี้ตัวรถก็สามารถเคลื่อนที่กลับมายังกึ่งกลางของรางเดินได้อีกครั้ง และเมื่อพิจารณาผลการจำลองสถานการณ์อื่นๆ พบว่า กระบวนการในการสร้างเสถียรภาพให้ระบบเมื่อมีมุมภายนอกมารบกวนระบบยังคงมีลักษณะเช่นเดียวกันนี้ ซึ่งในสถานการณ์สุดท้ายตัวควบคุมพีชชีแบบกฎอินพุตเดียวไม่สามารถรักษาเสถียรภาพของระบบไว้ได้ เนื่องจากตัวรถเคลื่อนที่เกินขอบเขตความยาวของรางเดินกำหนดไว้



รูปที่ 6.12 ผลทดสอบสมรรถนะของตัวควบคุมพีชชีแบบกฎอินพุตเดียวที่เวลา 2 วินาที



รูปที่ 6.13 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-ฟัซซี่ที่เวลา 2 วินาที



รูปที่ 6.14 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-ตาบู-ฟัซซี่ที่เวลา 2 วินาที

รูปที่ 6.13 และ 6.14 แสดงการจำลองสถานการณ์เพื่อทดสอบการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน ด้วยตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-คาบิว-ฟัซซี่ ตามลำดับ ซึ่งจากผลการจำลองสถานการณ์ดังกล่าวแสดงให้เห็นว่า ตัวควบคุมทั้งสองสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ครบทั้ง 4 สถานการณ์ รวมทั้งสามารถทำให้ระบบเข้าสู่เสถียรภาพได้ภายในระยะเวลาอันสั้น ดังนั้นอาจกล่าวได้ว่า ค่าระดับความสำคัญพลวัตที่ได้จากการเรียนรู้ของโครงข่ายประสาทเทียมในแต่ละสถานการณ์ ขณะที่ไม่มีกรรบกวนระบบด้วยมุมภายนอก ยังคงสามารถนำมาใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อเดียวได้ แม้ว่าจะมีการรบกวนระบบด้วยมุมภายนอกก็ตาม นอกจากนี้ เมื่อพิจารณามุมของลูกตุ้มและตำแหน่งของตัวรถที่เวลา 2 วินาที พบว่าอยู่ห่างจากตำแหน่งสมดุลไม่มากนัก ดังนั้นเมื่อมีมูรบกวนจากภายนอกขนาด 30 องศา มากระทำต่อระบบ ทำให้กระบวนการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน มีลักษณะที่คล้ายคลึงกับกรณีที่ไม่มีกรรบกวนระบบด้วยมุมภายนอก

เมื่อพิจารณาเวลาที่ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว ตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-คาบิว-ฟัซซี่ ใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อมีการรบกวนระบบด้วยมุม 30 องศา ที่เวลา 2 วินาที แสดงได้ดังตารางที่ 6.4 ซึ่งจะสังเกตเห็นว่า ตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-คาบิว-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว รวมทั้งสามารถสร้างเสถียรภาพให้ระบบได้ทุกสถานการณ์ โดยตัวควบคุมนิวโร-ฟัซซี่ และตัวควบคุมนิวโร-คาบิว-ฟัซซี่ ใช้เวลาในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันในแต่ละสถานการณ์ที่ใกล้เคียงกัน นอกจากนี้ตัวควบคุมนิวโร-คาบิว-ฟัซซี่ สามารถรักษาเสถียรภาพให้ระบบทั้ง 4 สถานการณ์ได้ภายในเวลา 8.5 วินาที

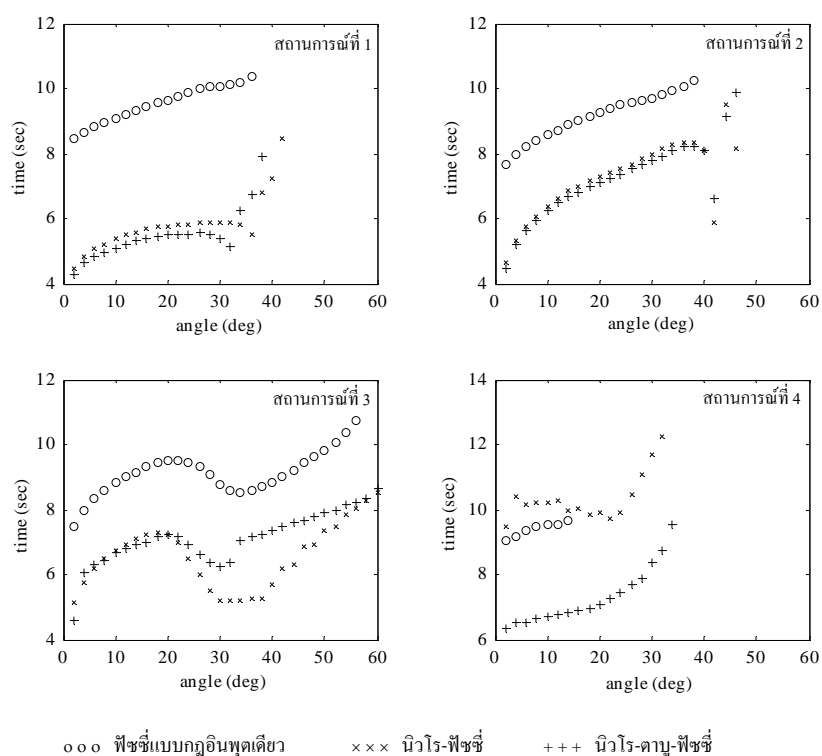
ตารางที่ 6.4 เวลาเข้าสู่เสถียรภาพของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 2 วินาที

สถานการณ์	เวลา (วินาที)		
	ฟัซซี่แบบกฎอินพุตเดียว	นิวโร-ฟัซซี่	นิวโร-คาบิว-ฟัซซี่
1	10.09	5.88	5.39
2	9.69	7.98	7.79
3	8.75	5.17	6.27
4	-	11.68	8.40

นอกจากนี้ได้ดำเนินการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ ในแต่ละสถานการณ์ เมื่อปรับเพิ่มมูรบกวนระบบจากภายนอกที่เวลา 2 วินาที ครั้งละ 2 องศา โดยเริ่มจาก 2 องศา จนกระทั่งถึง 60 องศา ผลการทดสอบดังกล่าวแสดงได้ดังรูปที่ 6.15 ซึ่งสามารถพิจารณาได้

ทั้งเวลาที่ตัวควบคุมแบบต่างๆ ใช้ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว รวมทั้งขอบเขตของมุมรบกวนภายนอกที่ยังคงทำให้ตัวควบคุมเหล่านี้สามารถสร้างเสถียรภาพให้ระบบได้ โดยสัญลักษณ์ที่ปรากฏในแต่ละสถานการณ์แทนด้วยตัวควบคุมแบบต่างๆ ดังที่ได้อธิบายแทรกไว้เป็นส่วนหนึ่งของรูปที่ 6.15

เมื่อพิจารณาผลทดสอบในสถานการณ์แรก พบว่า ตัวควบคุมพีชชีแบบกฎอินพุตเดียวใช้เวลาในการเข้าสู่เสถียรภาพมากที่สุด ในขณะที่ตัวควบคุมนิวโร-ตาบู-พีชชี ใช้เวลาในการสร้างเสถียรภาพให้ระบบน้อยที่สุด ซึ่งตัวควบคุมดังกล่าวใช้เวลาใกล้เคียงกับตัวควบคุมนิวโร-พีชชี นอกจากนี้จะเห็นได้ว่า ตัวควบคุมทั้งสองสามารถทนต่อมุมรบกวนภายนอกได้มากกว่าตัวควบคุมพีชชีแบบกฎอินพุตเดียว เมื่อพิจารณาสถานการณ์ที่ 2 และสถานการณ์ที่ 3 พบว่าผลการทดสอบที่ได้ยังมีแนวโน้มเช่นเดียวกับสถานการณ์แรก ส่วนสถานการณ์สุดท้ายจะเห็นได้ว่า ตัวควบคุมพีชชีแบบกฎอินพุตเดียวสามารถรักษาเสถียรภาพให้ระบบได้เมื่อมุมรบกวนสูงสุดเป็น 14 องศา ในขณะที่ตัวควบคุมนิวโร-ตาบู-พีชชีสามารถทนมุมรบกวนสูงสุดได้ถึง 34 องศา รวมทั้งสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมนิวโร-พีชชี และตัวควบคุมพีชชีแบบกฎอินพุตเดียวด้วยเช่นกัน



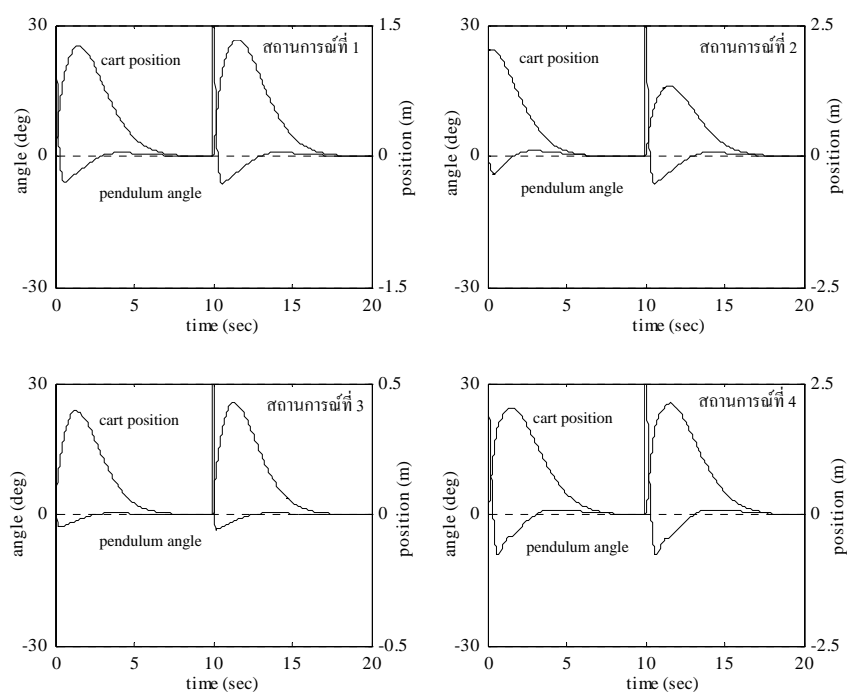
รูปที่ 6.15 ผลทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 2 วินาที



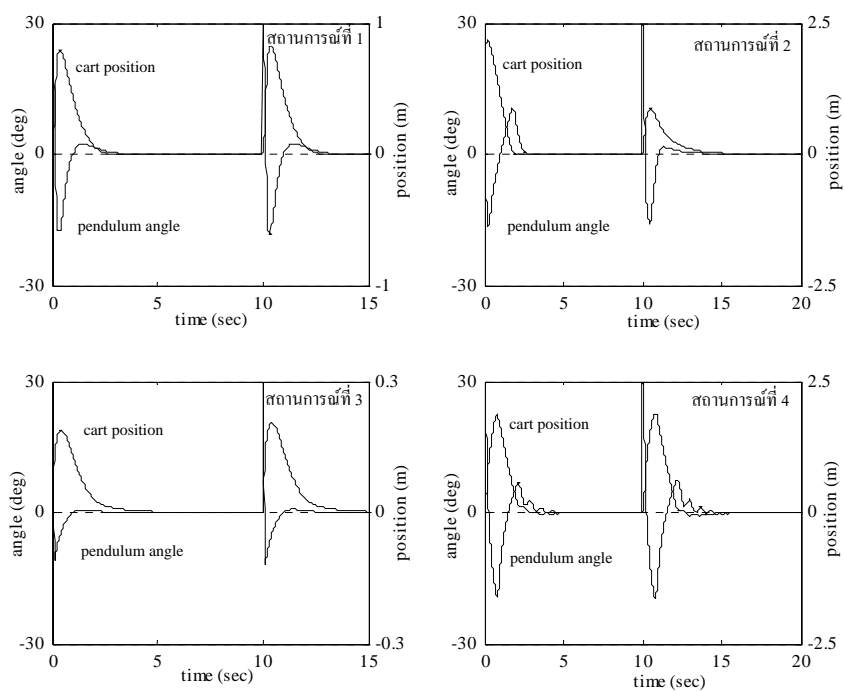
## 6.5.2 การรบกวนระบบที่เวลา 10 วินาที

หลังจากที่ได้ทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อมีมุมภายนอกมากระทำต่อระบบในขณะที่ตัวควบคุมกำลังสร้างเสถียรภาพให้ระบบลูกค้อนผกผันแบบข้อต่อเดียวแล้ว ในลำดับต่อมาได้ดำเนินการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อมีการรบกวนระบบหลังจากที่ระบบลูกค้อนผกผันเข้าสู่เสถียรภาพเสร็จสมบูรณ์แล้ว ทั้งนี้กำหนดให้มุมภายนอกสำหรับใช้ทดสอบสมรรถนะของตัวควบคุมมีขนาด 30 องศา โดยจะทำการรบกวนระบบที่เวลา 10 วินาที ผลการทดสอบตัวควบคุมพีชชีแบบกฎอินพุตเดียว ตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-ตาบู่-พีชชี แสดงได้ดังรูปที่ 6.16, 6.17 และ 6.18 ตามลำดับ

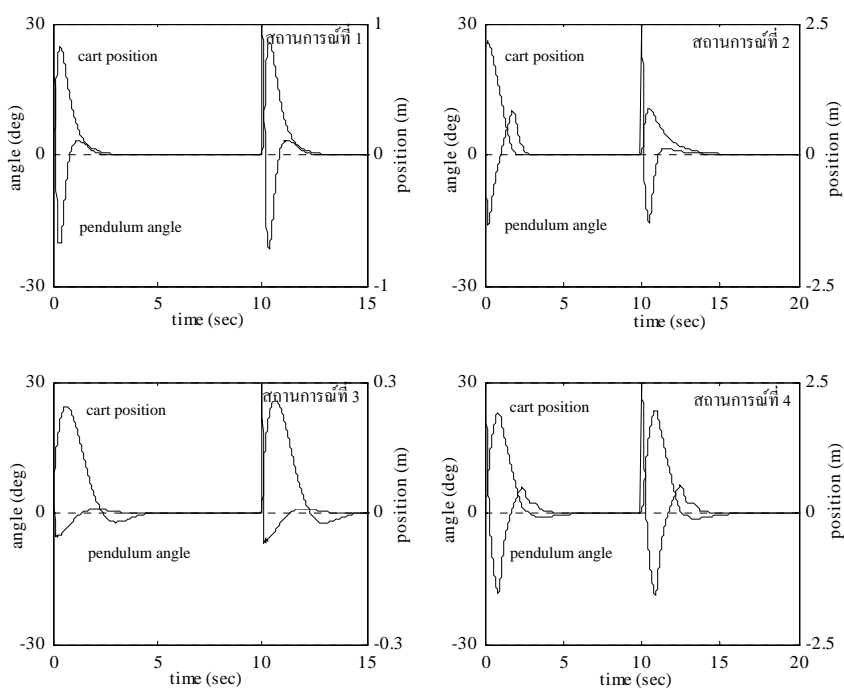
เมื่อพิจารณาสถานการณ์แรกของรูปที่ 6.16 พบว่า ผลการจำลองสถานการณ์หลังจากที่มีมุมภายนอกมากระทำต่อระบบจะคล้ายคลึงกับผลการจำลองสถานการณ์ในช่วง 10 วินาทีแรก ซึ่งยังไม่มีกรรบกวนระบบจากภายนอก ส่วนสถานการณ์ที่สองแสดงให้เห็นว่า พลวัตของระบบลูกค้อนผกผันหลังจากที่มีการรบกวนระบบมีลักษณะที่คล้ายคลึงกับสถานการณ์แรก ทั้งนี้เนื่องจากที่เวลา 10 วินาที จะเสมือนว่ามุมเริ่มต้นของระบบเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร ส่งผลให้กระบวนการในการสร้างเสถียรภาพให้ระบบลูกค้อนผกผันมีลักษณะเช่นเดียวกัน



รูปที่ 6.16 ผลทดสอบสมรรถนะของตัวควบคุมพีชชีแบบกฎอินพุตเดียวที่เวลา 10 วินาที



รูปที่ 6.17 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-ฟัซซี่ที่เวลา 10 วินาที

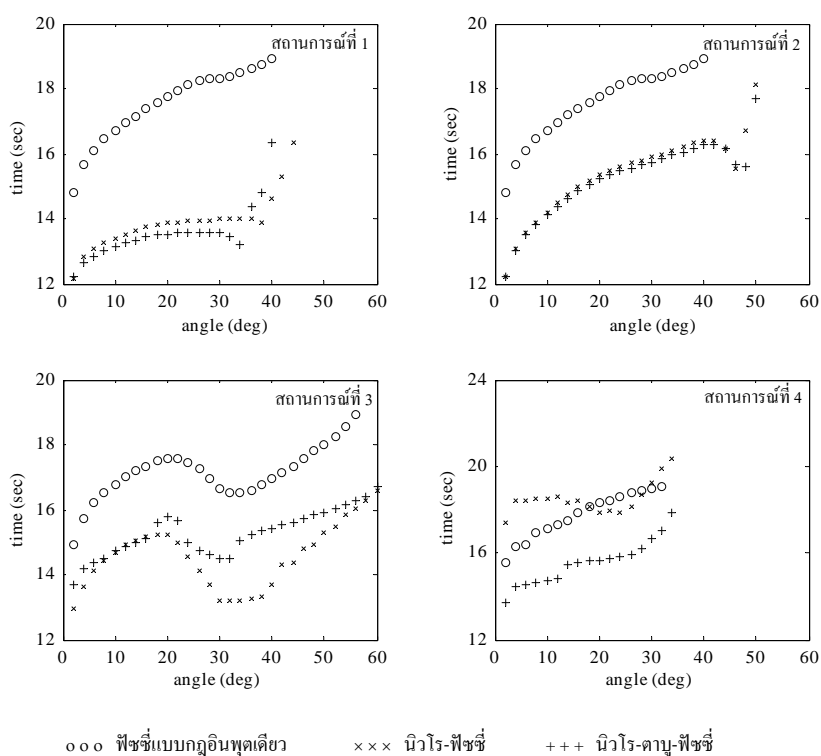


รูปที่ 6.18 ผลทดสอบสมรรถนะของตัวควบคุมนิวโร-ฟัซซี่ที่เวลา 10 วินาที

เมื่อพิจารณาผลการทดสอบสมรรถนะของตัวควบคุมนิวโร-ฟิชซี และตัวควบคุมนิวโร-ตาบูลู-ฟิชซี ที่ปรากฏในรูปที่ 6.17 และ 6.18 ตามลำดับนั้น อาจกล่าวได้ว่า ตัวควบคุมทั้งสองสามารถสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟิชซีแบบกฎอินพุตเดียว ทั้งนี้เวลาในการสร้างเสถียรภาพให้แก่ระบบของตัวควบคุมแบบต่างๆ แสดงได้ดังตารางที่ 6.5 ซึ่งพบว่าตัวควบคุมทั้งสามสามารถสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบข้อต่อเดียว ได้ครบทั้ง 4 สถานการณ์ภายในเวลา 20 วินาที

ตารางที่ 6.5 เวลาเข้าสู่เสถียรภาพของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 10 วินาที

สถานการณ์	เวลา (วินาที)		
	ฟิชซีแบบกฎอินพุตเดียว	นิวโร-ฟิชซี	นิวโร-ตาบูลู-ฟิชซี
1	18.31	13.97	13.58
2	18.31	15.89	15.72
3	16.66	13.21	14.47
4	18.99	19.29	16.62



รูปที่ 6.19 ผลทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อรบกวนระบบที่เวลา 10 วินาที

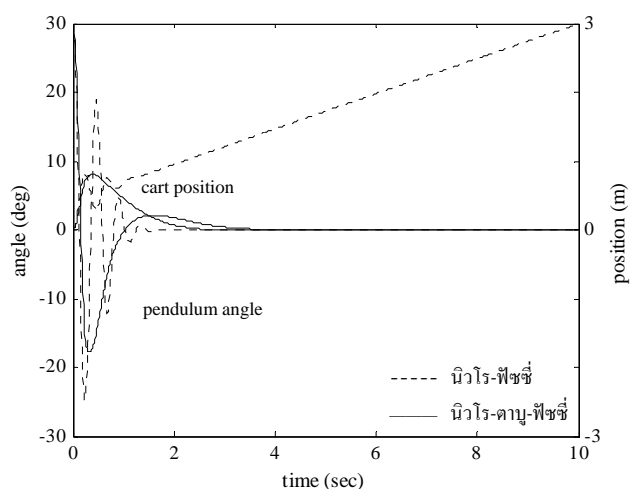
ผลการทดสอบตัวควบคุมแบบต่างๆ เมื่อมีมูรบกวนจากภายนอกมากระทำต่อระบบที่เวลา 10 วินาที แสดงได้ดังรูปที่ 6.19 ซึ่งได้ดำเนินการเพิ่มมูรบกวนครั้งละ 2 องศา โดยเริ่มจาก 2 องศา จนกระทั่งถึง 60 องศา ผลการทดสอบดังกล่าวแสดงให้เห็นว่า ตัวควบคุมนิวโร-ฟิชซี และตัวควบคุมนิวโร-ดาบู-ฟิชซี สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟิชซีแบบกฎอินพุตเดียว รวมทั้งสามารถรักษาเสถียรภาพของระบบได้ในขอบเขตของมูรบกวนที่กว้างกว่าตัวควบคุมฟิชซีแบบกฎอินพุตเดียวด้วยเช่นกัน ทั้งนี้เมื่อพิจารณาผลการทดสอบในสถานการณ์สุดท้าย จะพบว่าตัวควบคุมนิวโร-ดาบู-ฟิชซีใช้เวลาสร้างเสถียรภาพให้ระบบน้อยกว่าตัวควบคุมฟิชซีแบบกฎอินพุตเดียวและตัวควบคุมนิวโร-ฟิชซีค่อนข้างมาก

## 6.6 ผลทดสอบค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสมของโครงข่ายประสาทเทียม

ผลการทดสอบตัวควบคุมนิวโร-ฟิชซี และตัวควบคุมนิวโร-ดาบู-ฟิชซี ที่ผ่านมาเป็นการจำลองสถานการณ์โดยเลือกใช้ค่าพารามิเตอร์เริ่มต้นที่เหมาะสมของโครงข่ายประสาทเทียม ซึ่งตัวควบคุมนิวโร-ฟิชซีได้ค่าพารามิเตอร์เริ่มต้นเหล่านี้มาจากการสุ่มค่า ในขณะที่ตัวควบคุมนิวโร-ดาบู-ฟิชซี ใช้วิธีการค้นหาแบบดาบูเชิงปรับตัวเข้าช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสม ในการทดสอบสมรรถนะของตัวควบคุมเหล่านี้จะใช้ค่าพารามิเตอร์เริ่มต้นเพียงชุดเดียวตลอดการจำลองสถานการณ์ ซึ่งผลการจำลองสถานการณ์ที่ผ่านมาแสดงให้เห็นว่า ตัวควบคุมทั้งสองสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น

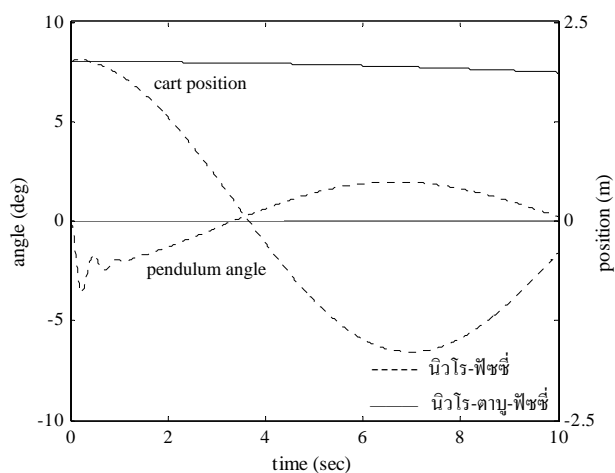
ในลำดับต่อมาเป็นการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน เมื่อเลือกใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ไม่เหมาะสม เพื่ออภิปรายถึงผลกระทบของค่าพารามิเตอร์เริ่มต้นเหล่านี้ที่มีต่อการสร้างเสถียรภาพให้ระบบ กล่าวคือ ในส่วนของตัวควบคุมนิวโร-ฟิชซี จะเลือกค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมจากการสุ่มค่าที่ทำให้ผลการทดสอบไม่ดีนัก โดยดำเนินการจำลองสถานการณ์ทั้ง 5 สถานการณ์ทดสอบดังที่ปรากฏในตารางที่ 6.1 ในขณะที่ตัวควบคุมนิวโร-ดาบู-ฟิชซี จะนำค่าพารามิเตอร์เริ่มต้นที่ได้จากวิธีการค้นหาแบบดาบูเชิงปรับตัว ซึ่งไม่สามารถสร้างเสถียรภาพให้ระบบได้ครบทั้ง 5 สถานการณ์มาดำเนินการจำลองสถานการณ์ดังกล่าว ทั้งนี้ในแต่ละสถานการณ์ยังคงใช้ค่าพารามิเตอร์เริ่มต้นชุดเดียวกัน

รูปที่ 6.20 เป็นผลการจำลองสถานการณ์แรก ซึ่งจะเห็นได้ว่า ตัวควบคุมนิวโร-ดาบู-ฟิชซี ยังคงสามารถสร้างเสถียรภาพให้ระบบได้โดยใช้เวลาเข้าสู่เสถียรภาพทั้งสิ้น 4.06 วินาที ในขณะที่ตัวควบคุมนิวโร-ฟิชซี ไม่สามารถสร้างเสถียรภาพให้ระบบได้ เนื่องจากตัวรถเคลื่อนที่เกินความยาวของรางลื่นที่กำหนดไว้ รวมทั้งไม่สามารถเคลื่อนที่กลับมายังตำแหน่งกึ่งกลางของรางลื่นได้ แม้ว่าลูกตุ้มจะตั้งตรงที่ตำแหน่งสมดุลศูนย์กลางแล้วก็ตาม



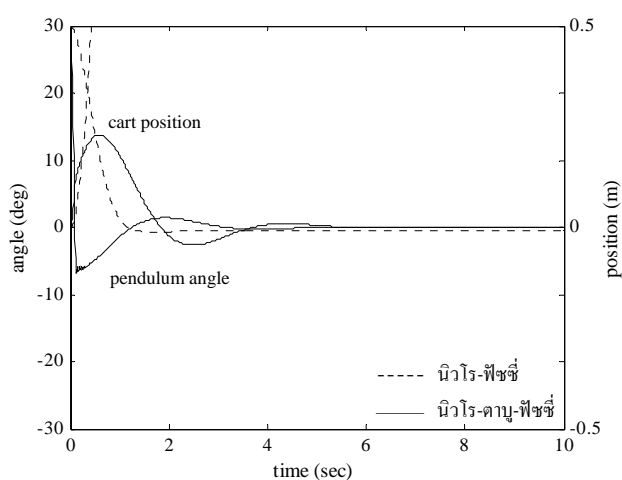
รูปที่ 6.20 ผลการจำลองสถานการณ์ที่ 1 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม

ผลการจำลองสถานการณ์ที่ 2 แสดงได้ดังรูปที่ 6.21 ซึ่งเป็นกรณีที่ตำแหน่งเริ่มต้นของตัวรถเป็น 2.0 เมตร และมุมเริ่มต้นของลูกตุ้มเป็นศูนย์องศา โดยที่ก้านลูกตุ้มยาว 1.0 เมตร จะสังเกตเห็นว่า ตัวควบคุมทั้งสองไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ เมื่อพิจารณาพลวัตของระบบจากตัวควบคุมนิวโร-คาบรู-ฟิชชี จะเห็นได้ว่าตัวรถจะเคลื่อนที่อย่างช้าๆ จึงทำให้มุมของลูกตุ้มเปลี่ยนแปลงเพียงเล็กน้อยเท่านั้น ทำให้ไม่สามารถสร้างเสถียรภาพให้ระบบได้ในส่วนของตัวควบคุมนิวโร-ฟิชชี พบว่า มีแนวโน้มที่จะลู่เข้าสู่เสถียรภาพได้ แต่อาจต้องใช้ระยะเวลาหลายๆ เนื่องจากพลวัตของระบบเปลี่ยนแปลงไปอย่างช้าๆ เช่นเดียวกัน

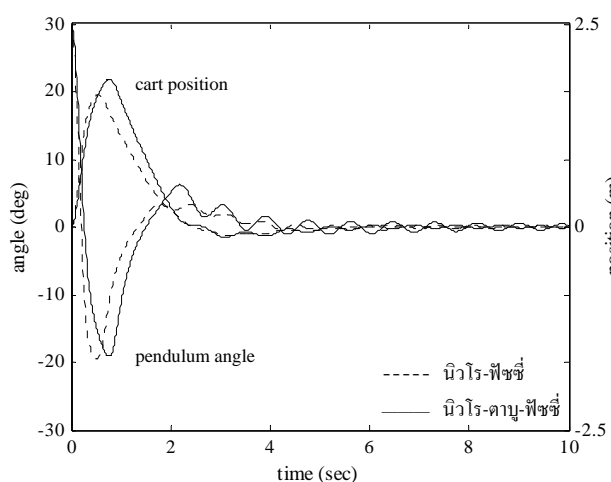


รูปที่ 6.21 ผลการจำลองสถานการณ์ที่ 2 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม

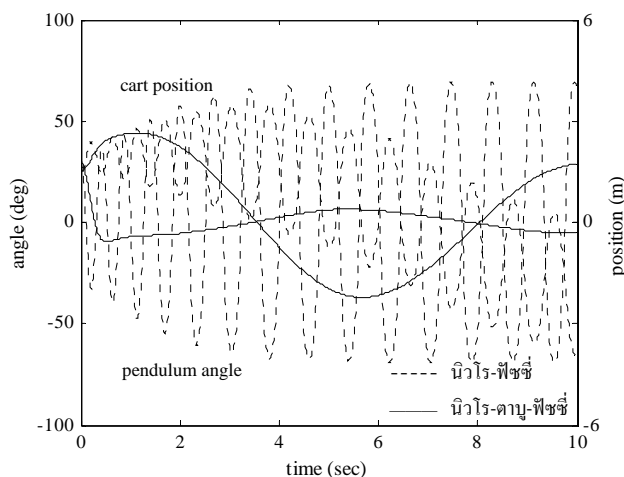
รูปที่ 6.22 และ 6.23 แสดงผลการจำลองสถานการณ์ที่ 3 และสถานการณ์ที่ 4 ตามลำดับ ซึ่งทำการปรับเปลี่ยนความยาวของก้านลูกตุ้ม เมื่อพิจารณาผลการจำลองสถานการณ์ในรูปที่ 6.22 พบว่า ตัวควบคุมนิวโร-ตาบ-พีชชี สามารถสร้างเสถียรภาพให้ระบบได้ภายในเวลา 5.32 วินาที ในขณะที่ตัวควบคุมนิวโร-พีชชี ไม่สามารถสร้างเสถียรภาพให้ระบบได้ ส่วนผลการจำลองสถานการณ์ในรูปที่ 6.23 แสดงให้เห็นว่า ตัวควบคุมทั้งสองไม่สามารถสร้างเสถียรภาพให้ระบบได้เช่นกัน เนื่องจากตัวรถไม่หยุดนิ่งที่กึ่งกลางของรางเดิน ยังคงมีการเคลื่อนที่กลับไปกลับมาในระยะสั้นๆ ส่งผลให้มุมลูกตุ้มมีการแกว่งกวัดเกิดขึ้น ทำให้ไม่สามารถตั้งตรงที่ตำแหน่งศูนย์องศาได้



รูปที่ 6.22 ผลการจำลองสถานการณ์ที่ 3 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม



รูปที่ 6.23 ผลการจำลองสถานการณ์ที่ 4 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม



รูปที่ 6.24 ผลการจำลองสถานการณ์ที่ 5 โดยใช้ค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสม

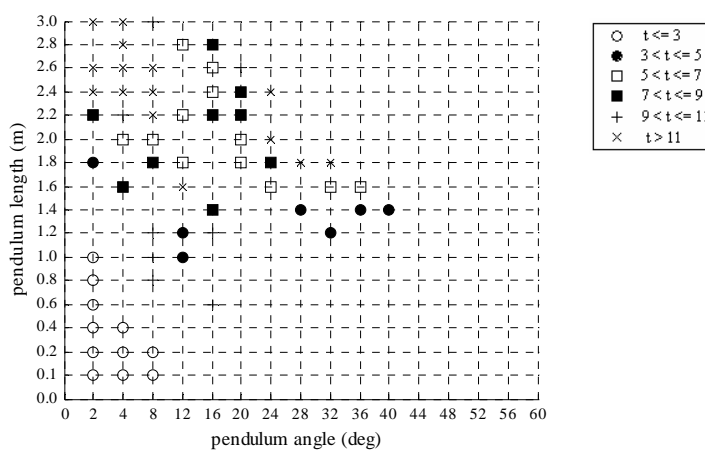
ผลการจำลองสถานการณ์สุดท้ายปรากฏดังรูปที่ 6.24 ซึ่งจะเห็นได้ว่า ตัวควบคุมทั้งสองไม่สามารถสร้างเสถียรภาพให้ระบบลูกค้อนผกผันแบบข้อต่อเดียวได้ เมื่อพิจารณาผลการสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมนิวโร-พีซซี พบว่า ตัวรถจะเคลื่อนที่ออกจากตำแหน่งเริ่มต้นในลักษณะกลับไปกลับมา ส่งผลให้ลูกค้อนหมุนในทิศทางเข็มนาฬิกาสลับกับการหมุนในทิศตามเข็มนาฬิกาตลอดเวลา เมื่อพิจารณาตัวควบคุมนิวโร-คาบ-พีซซี พบว่าไม่สามารถสร้างเสถียรภาพให้ระบบได้เช่นกัน ทั้งนี้เนื่องจากตัวรถเคลื่อนที่เกินความยาวของรางลื่นที่กำหนดไว้

ค่าระดับความสำคัญพลวัตในแต่ละสถานการณ์ที่ได้จากการเรียนรู้ของโครงข่ายประสาทเทียมของตัวควบคุมนิวโร-พีซซี และตัวควบคุมนิวโร-คาบ-พีซซี แสดงได้ดังตารางที่ 6.6 ซึ่งจะสังเกตเห็นว่าค่าระดับความสำคัญพลวัตในบางสถานการณ์มีค่าน้อยมาก จึงทำให้ไม่สามารถสร้างเสถียรภาพให้ระบบลูกค้อนผกผันแบบข้อต่อเดียวได้

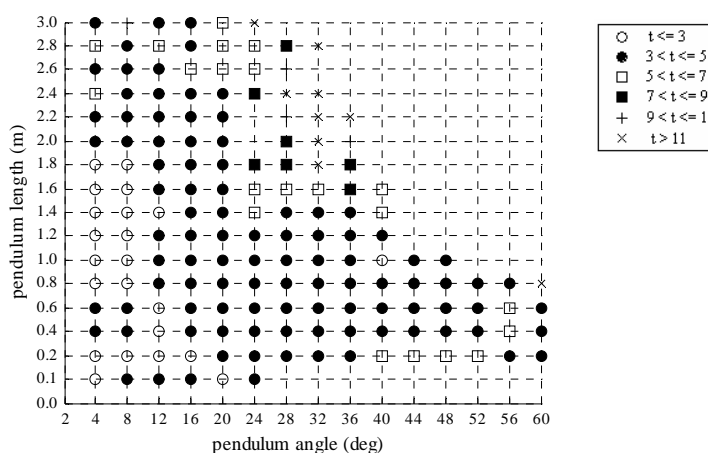
ตารางที่ 6.6 ค่าระดับความสำคัญพลวัตจากการเรียนรู้ของโครงข่ายประสาทเทียม

สถานการณ์	ตัวควบคุมนิวโร-พีซซี				ตัวควบคุมนิวโร-คาบ-พีซซี			
	DID-1	DID-2	DID-3	DID-4	DID-1	DID-2	DID-3	DID-4
1	7.1259	1.15712	0.0002	0.0041	5.8375	4.3237	4.8554	2.1517
2	6.2994	0.9161	0.5469	0.0799	2.8421	2.4770	0.0018	0.0193
3	0.6350	0.1589	0.00003	0.0024	5.2032	3.9218	4.3549	1.6259
4	7.1753	4.8973	4.3314	1.8478	5.0498	3.8169	2.6522	1.3765
5	9.3442	0.7056	0.0077	0.0180	2.4516	1.2584	0.4222	0.8946

เพื่อทดสอบสมรรถนะของตัวควบคุม เมื่อใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ไม่เหมาะสม ได้ดำเนินการปรับเปลี่ยนค่าเริ่มต้นของระบบที่ใช้ในการจำลองสถานการณ์ โดยที่ตัวควบคุมเหล่านี้ยังคงใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมชุดเดิม รูปที่ 6.25 แสดงผลการทดสอบตัวควบคุมนิโร-พีชชี และตัวควบคุมนิโร-คาบู-พีชชี เมื่อปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้มพร้อมกับความยาวก้านลูกตุ้ม ซึ่งจะสังเกตเห็นว่า ตัวควบคุมนิโร-พีชชีสามารถสร้างเสถียรภาพให้ระบบได้บางสถานการณ์เท่านั้น ในขณะที่ตัวควบคุมนิโร-คาบู-พีชชี สามารถทำให้ระบบเข้าสู่เสถียรภาพภายในระยะเวลาอันสั้น รวมทั้งสามารถทำงานสร้างเสถียรภาพให้ระบบ เมื่อความยาวของก้านลูกตุ้มและมุมเริ่มต้นของลูกตุ้มมีค่าแปรไปในขอบเขตที่กว้าง



ก)



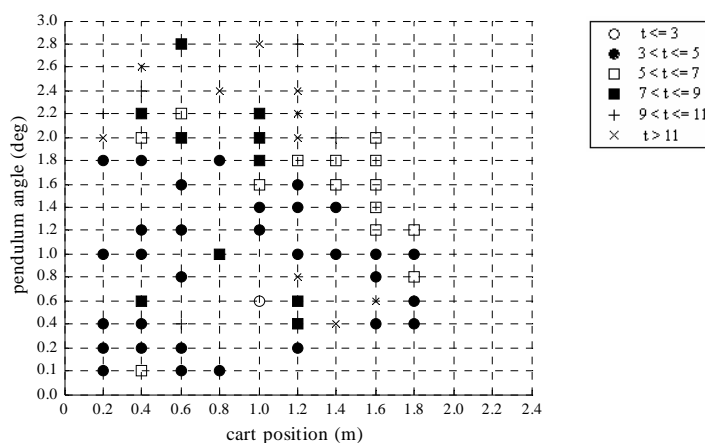
ข)

รูปที่ 6.25 ก) เวลาสร้างเสถียรภาพของตัวควบคุมนิโร-พีชชีเมื่อปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้ม และความยาวก้านลูกตุ้ม

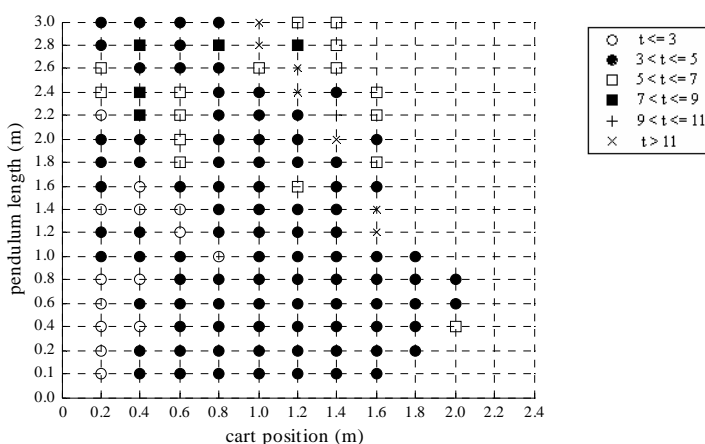
ข) เวลาสร้างเสถียรภาพของตัวควบคุมนิโร-คาบู-พีชชีเมื่อทดสอบในลักษณะเดียวกัน



ผลการทดสอบการปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถพร้อมกับปรับเปลี่ยนความยาวก้านลูกตุ้ม โดยที่มุมเริ่มต้นลูกตุ้มเป็นศูนย์องศา แสดงได้ดังรูปที่ 6.26 ซึ่งเมื่อพิจารณาตัวควบคุมนิวโร-ฟัซซี่ พบว่า ยังคงสร้างเสถียรภาพให้ระบบได้เพียงบางสถานการณ์เท่านั้น นั่นคือ โครงข่ายประสาทเทียมไม่สามารถเรียนรู้เพื่อปรับค่าระดับความสำคัญให้เหมาะกับสถานการณ์ที่เปลี่ยนแปลงไปได้ เมื่อพิจารณาผลการทดสอบของตัวควบคุมนิวโร-คาบิว-ฟัซซี่ จะได้ว่า ตัวควบคุมดังกล่าวสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาอันสั้น นอกจากนี้จะสังเกตเห็นว่า ขณะที่ตำแหน่งเริ่มต้นของตัวรถไม่เกิน 1.4 เมตร และความยาวของก้านลูกตุ้มอยู่ในช่วง 3.0 เมตร ตัวควบคุมนิวโร-คาบิว-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ทุกสถานการณ์



ก)

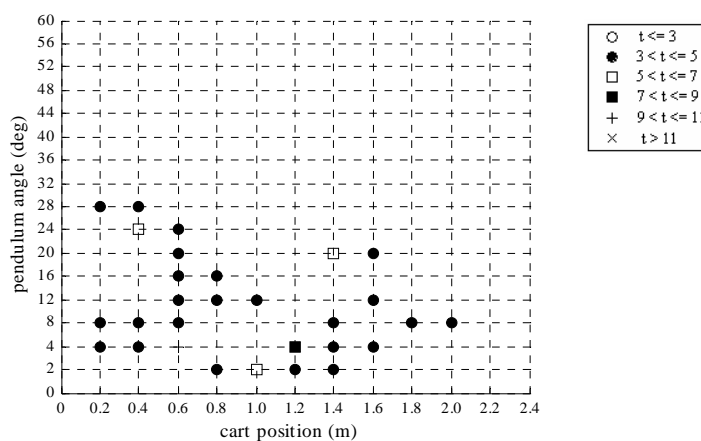


ข)

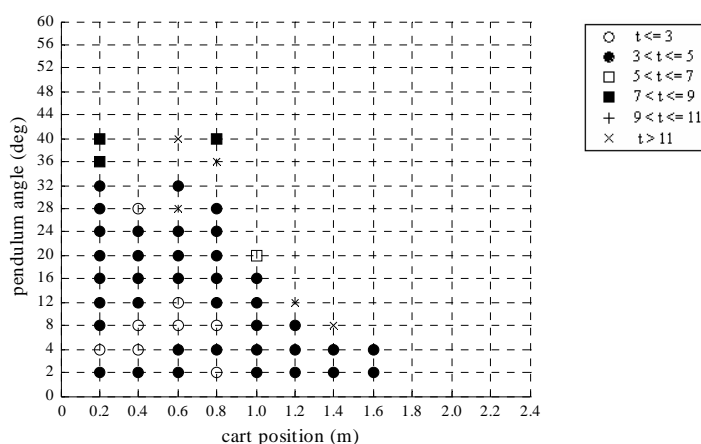
รูปที่ 6.26 ก) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-ฟัซซี่เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถและความยาวก้านลูกตุ้ม

ข) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-คาบิว-ฟัซซี่เมื่อทดสอบในลักษณะเดียวกัน

รูปที่ 6.27 เป็นการทดสอบสมรรถนะของตัวควบคุม เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถ พร้อมกับตำแหน่งเริ่มต้นของลูกตุ้ม โดยที่ก้านลูกตุ้มยาว 1.0 เมตร ผลการทดสอบดังกล่าวมีแนวโน้มเช่นเดียวกันกับผลการทดสอบที่แสดงในรูปที่ 6.25 และ 6.26 กล่าวคือ ตัวควบคุมนิวโร-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบได้บางสถานการณ์ในขอบเขตแคบๆ และมีลักษณะกระจายไม่เกาะกลุ่มเหมือนดังเช่นสถานการณ์ที่ควบคุมด้วยตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ เมื่อพิจารณาผลการทดสอบของตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่ พบว่า สามารถสร้างเสถียรภาพให้ระบบได้หลายสถานการณ์กว่าตัวควบคุมนิวโร-ฟัซซี่ ซึ่งเวลาโดยส่วนใหญ่ในการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียวอยู่ในช่วงประมาณ 5 วินาที



ก)



ข)

รูปที่ 6.27 ก) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-ฟัซซี่เมื่อปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถและมุมเริ่มต้นของลูกตุ้ม

ข) เวลาสร้างเสถียรภาพของตัวควบคุมนิวโร-ตาบูลู-ฟัซซี่เมื่อทดสอบในลักษณะเดียวกัน

จากการพิจารณาผลการจำลองสถานการณ์และผลการทดสอบสมรรถนะของตัวควบคุมนิวโร-ฟิซซี และตัวควบคุมนิวโร-ตาบู-ฟิซซี ในการสร้างเสถียรภาพให้ระบบลูกคัมผผันแบบข้อต่อเดี่ยว อาจกล่าวได้ว่า ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมมีผลกระทบต่อการสร้างเสถียรภาพให้ระบบ เมื่อพิจารณาผลการทดสอบการสร้างเสถียรภาพให้ระบบทั้ง 5 สถานการณ์ โดยใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการสุ่มค่าจำนวน 4000 ครั้ง ของตัวควบคุมนิวโร-ฟิซซี ดังแสดงในตารางที่ 4.5 เปรียบเทียบกับผลการทดสอบของตัวควบคุมนิวโร-ตาบู-ฟิซซีในตารางที่ 5.5 ซึ่งค่าพารามิเตอร์เริ่มต้นดังกล่าวได้มาจากวิธีการค้นหาแบบตามูเชิงปรับตัว ปรากฏในตารางที่ 6.7

ตารางที่ 6.7 ปริมาณการเข้าสู่เสถียรภาพของตัวควบคุมนิวโร-ฟิซซี และตัวควบคุมนิวโร-ตาบู-ฟิซซี

ตัวควบคุม	สถานการณ์ที่				
	1	2	3	4	5
นิวโร-ฟิซซี (4000 ครั้ง)	67.830 %	59.525%	32.825%	36.825%	17.925%
นิวโร-ตาบู-ฟิซซี (4000 ครั้ง)	100.0%	97.725%	100.0%	94.675%	74.75%
จำนวนเท่ากับนิวโร-ฟิซซี	1.47	1.64	3.03	2.57	4.17
นิวโร-ตาบู-ฟิซซี (10000 ครั้ง)	100.0%	97.77%	100.0%	95.02%	75.26%
จำนวนเท่ากับนิวโร-ฟิซซี	1.47	1.64	3.03	2.58	4.20

เมื่อพิจารณาตารางที่ 6.7 จะสังเกตเห็นว่า ปริมาณการเข้าสู่เสถียรภาพ ณ ตำแหน่งที่ต้องการของตัวควบคุมนิวโร-ฟิซซี ในแต่ละสถานการณ์มีปริมาณที่ไม่มากนัก โดยเฉพาะอย่างยิ่งในสถานการณ์สุดท้ายซึ่งระบบสามารถเข้าสู่เสถียรภาพได้เพียง 17.925% เท่านั้น ในขณะที่ตัวควบคุมนิวโร-ตาบู-ฟิซซี สามารถสร้างเสถียรภาพให้ระบบลูกคัมผผันแต่ละสถานการณ์ในปริมาณที่มากกว่า กล่าวคือ ในสถานการณ์ที่ 1 และสถานการณ์ที่ 3 จะพบว่าตัวควบคุมดังกล่าวสามารถสร้างเสถียรภาพให้ระบบได้ถึง 100% ส่วนสถานการณ์สุดท้ายคิดเป็น 74.75% ทั้งนี้เมื่อพิจารณาจำนวนเท่าของปริมาณการเข้าสู่เสถียรภาพของตัวควบคุมนิวโร-ตาบู-ฟิซซีเปรียบเทียบกับตัวควบคุมนิวโร-ฟิซซี พบว่ามากกว่าประมาณ 1.47-4.17 เท่า

นอกจากนี้เพื่อให้ผลการทดสอบตัวควบคุมนิวโร-ตาบู-ฟิซซีน่าเชื่อถือมากยิ่งขึ้น จึงได้ดำเนินการค้นหาค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียมเพิ่มขึ้นเป็น 10000 ครั้ง ซึ่งพบว่า ตัวควบคุมดังกล่าวยังคงสามารถสร้างเสถียรภาพให้ระบบในแต่ละสถานการณ์ได้ในปริมาณมาก ดังแสดงในตารางที่ 6.7 ทั้งนี้จะสังเกตเห็นว่า ปริมาณการเข้าสู่เสถียรภาพที่ได้จากการทดสอบจำนวน 10000 ครั้ง นั้นใกล้เคียงกับผลการทดสอบจำนวน 4000 ครั้ง

## 6.7 สรุป

การเปรียบเทียบผลการจำลองสถานการณ์ของตัวควบคุมแบบต่างๆ จากสถานการณ์ที่กำหนดขึ้นทั้ง 5 สถานการณ์ แสดงให้เห็นว่า ตัวควบคุมนิวโร-ฟิซซี และตัวควบคุมนิวโร-ตาบู-ฟิซซีสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟิซซีแบบกฎอินพุตเดียว รวมทั้งสามารถสร้างเสถียรภาพให้ระบบได้ครบทั้ง 5 สถานการณ์ทดสอบ ในขณะที่ตัวควบคุมฟิซซีแบบกฎอินพุตเดียวไม่สามารถสร้างเสถียรภาพให้ระบบในสถานการณ์สุดท้ายได้ เนื่องจากตัวรถเคลื่อนที่เกินความยาวของรางลื่นที่กำหนดไว้ และจากการพิจารณาขอบเขตการสร้างเสถียรภาพให้ระบบ เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบในการจำลองสถานการณ์ จะได้ว่าตัวควบคุมนิวโร-ตาบู-ฟิซซีมีขอบเขตการสร้างเสถียรภาพให้กับระบบที่ใกล้เคียงกับตัวควบคุมนิวโร-ฟิซซี รวมทั้งมีขอบเขตเสถียรภาพที่กว้างกว่าตัวควบคุมฟิซซีแบบกฎอินพุตเดียว นอกจากนี้เมื่อพิจารณาเวลาในการเข้าสู่เสถียรภาพของระบบที่สถานการณ์ต่างๆ พบว่า ตัวควบคุมนิวโร-ฟิซซี และตัวควบคุมนิวโร-ตาบู-ฟิซซี ยังคงสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟิซซีแบบกฎอินพุตเดียว

ผลการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ ที่กล่าวถึงเมื่อจัดให้มีการรบกวนระบบจากภายนอกในรูปของมุมลูกตุ้มที่เวลา 2 และ 10 วินาที แสดงให้เห็นว่าตัวควบคุมนิวโร-ตาบู-ฟิซซีสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาอันสั้น เมื่อพิจารณาสมรรถนะของตัวควบคุมนิวโร-ฟิซซี จะพบว่ามีผลการทดสอบที่ใกล้เคียงกับตัวควบคุมนิวโร-ตาบู-ฟิซซี ยกเว้นในสถานการณ์ที่ 4 ซึ่งตัวควบคุมนิวโร-ตาบู-ฟิซซี สามารถสร้างเสถียรภาพให้ระบบได้รวดเร็วกว่าตัวควบคุมนิวโร-ฟิซซีค่อนข้างมาก ส่วนตัวควบคุมฟิซซีแบบกฎอินพุตเดียว พบว่า ต้องใช้ระยะเวลาที่มากกว่าตัวควบคุมทั้งสองข้างต้น จึงจะสามารถทำให้ก้านลูกตุ้มตั้งตรงที่ตำแหน่งสมดุล และทำให้ตัวรถเคลื่อนที่กลับมายังกึ่งกลางของรางลื่นได้อีกครั้ง รวมทั้งมีขอบเขตความสามารถในการแก้ปัญหาของมุมรบกวนที่น้อยกว่าตัวควบคุมทั้งสองด้วยเช่นกัน นอกจากนี้บางสถานการณ์ทดสอบเมื่อมีมุมรบกวนจากภายนอกมากกระทำต่อระบบ ตัวควบคุมดังกล่าวไม่สามารถรักษาเสถียรภาพให้ระบบได้

เมื่อพิจารณาผลการทดสอบค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสมของโครงข่ายประสาทเทียมจากตัวควบคุมนิวโร-ฟิซซี และตัวควบคุมนิวโร-ตาบู-ฟิซซี เพื่ออภิปรายผลกระทบต่อการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว พบว่า ตัวควบคุมนิวโร-ตาบู-ฟิซซี มีปริมาณการเข้าสู่เสถียรภาพที่สูงกว่าตัวควบคุมนิวโร-ฟิซซีทุกสถานการณ์ทดสอบ ดังนั้นอาจกล่าวได้ว่าค่าพารามิเตอร์เริ่มต้นที่ได้จากการสุ่มค่าของตัวควบคุมนิวโร-ฟิซซี มีโอกาสที่ไม่สามารถสร้างเสถียรภาพให้ระบบได้ค่อนข้างมาก ในขณะที่ค่าพารามิเตอร์เริ่มต้นที่ได้จากวิธีการค้นหาแบบตาบูเชิงปรับตัวของตัวควบคุมนิวโร-ตาบู-ฟิซซี มีเพียงบางสถานการณ์เท่านั้นที่ไม่สามารถสร้างเสถียรภาพให้ระบบดังกล่าวได้

จากผลการทดสอบที่ผ่านมา อาจกล่าวได้ว่า ค่าพารามิเตอร์เริ่มต้นในกระบวนการเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม เป็นปัจจัยสำคัญอันส่งผลต่อการสร้างเสถียรภาพให้ระบบ แม้ว่าตัวควบคุมนิวโร-ฟิชชี จะสามารถสร้างเสถียรภาพให้ระบบได้ภายในเวลาที่สั้นกว่าตัวควบคุมแบบกฎอินพุตเดียว อย่างไรก็ตาม ไม่เป็นการง่ายนักที่จะหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้แก่โครงข่ายประสาทเทียม โดยเฉพาะอย่างยิ่งกรณีระบบที่ต้องการควบคุมมีความซับซ้อนมากขึ้น เช่น ระบบลูกตุ้มผกผันแบบสองข้อต่อ ซึ่งมีจำนวนตัวแปรอินพุตที่พิจารณาเพิ่มขึ้น ทำให้ค่าระดับความสำคัญพลวัตที่โครงข่ายประสาทเทียมต้องเรียนรู้เพื่อปรับค่าให้ได้เหมาะสมเพิ่มขึ้นด้วยเช่นกัน ดังนั้นค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ตัวควบคุมนิวโร-ฟิชชี ต้องดำเนินการหาค่าย่อมมีจำนวนเพิ่มมากขึ้นด้วย จากผลการทดสอบพบว่า ตัวควบคุมนิวโร-ฟิชชี ไม่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อได้ เนื่องจากไม่สามารถหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม ในขณะที่ตัวควบคุมนิวโร-ตาบู-ฟิชชี สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อได้ รวมทั้งสามารถสร้างเสถียรภาพให้ระบบดังกล่าวได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟิชชีแบบกฎอินพุตเดียวด้วย ซึ่งรายละเอียดในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อ ด้วยตัวควบคุมนิวโร-ตาบู-ฟิชชี รวมทั้งผลการจำลองสถานการณ์ได้อธิบายไว้ในภาคผนวก ก.

## บทที่ 7

### บทสรุปและข้อเสนอแนะ

#### 7.1 สรุป

งานวิจัยวิทยานิพนธ์นี้ดำเนินการศึกษา และพัฒนาโครงสร้างการควบคุมแบบใหม่ เพื่อสร้างเสถียรภาพให้ระบบที่มีความไม่เป็นเชิงเส้น โดยตัวควบคุมที่พัฒนาขึ้นใหม่นี้ ได้แก่ ตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-คาบูกู-ฟิชชี ซึ่งโครงสร้างของตัวควบคุมทั้งสองเป็นการผสมผสานระหว่างฟิชชีลอจิกและโครงข่ายประสาทเทียม โดยการกำหนดกฎฟิชชีแต่ละกฎจะประกอบด้วยตัวแปรอินพุตเพียงตัวเดียวเท่านั้น เรียกกฎฟิชชีในลักษณะนี้ว่า กฎฟิชชีแบบกฎอินพุตเดียว ซึ่งช่วยลดจำนวนกฎฟิชชีและง่ายต่อการกำหนดกฎฟิชชีให้ได้อย่างเหมาะสม โดยมีค่าระดับความสำคัญพลวัตเป็นตัวบ่งชี้ความสำคัญของอินพุตแต่ละตัว ในโครงสร้างของตัวควบคุมที่นำเสนอในวิทยานิพนธ์นี้ ค่าระดับความสำคัญของอินพุตแต่ละตัวได้มาจากการเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม โดยค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมสำหรับโครงสร้างตัวควบคุมนิวโร-ฟิชชีได้มาจากการสุ่มค่า ในขณะที่โครงสร้างของตัวควบคุมนิวโร-คาบูกู-ฟิชชี ได้นำวิธีการค้นหาแบบคาบูกูเชิงปรับตัวเข้าช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม ค่าพารามิเตอร์ที่เหมาะสมนี้จะช่วยให้กระบวนการเรียนรู้มีประสิทธิภาพยิ่งขึ้น ซึ่งผลการจำลองสถานการณ์แสดงให้เห็นว่า โครงสร้างของตัวควบคุมทั้งสองที่พัฒนาขึ้น สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟิชชีแบบกฎอินพุตเดียว นอกจากนี้ โครงสร้างของตัวควบคุมนิวโร-คาบูกู-ฟิชชีที่พัฒนาขึ้น ยังสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อได้ด้วยเช่นกัน

ในบทที่ 1 ของวิทยานิพนธ์ได้กล่าวถึงความสำคัญของปัญหา คือ วิธีการควบคุมแบบดั้งเดิมยังมีข้อจำกัดในการควบคุมระบบที่มีความซับซ้อน ความไม่แน่นอน ความไม่เป็นเชิงเส้น และไม่เสถียร ดังนั้นต้องอาศัยวิธีการควบคุมแบบชาญฉลาดเข้าช่วยเพื่อแก้ปัญหาต่างๆ เหล่านี้ โดยได้กล่าวถึงวิธีการควบคุมฟิชชีแบบกฎอินพุตเดียวตามแนวทางของ Yi และ Yubazaki (2000) ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว ซึ่งเป็นตัวอย่างระบบทดสอบที่มีความไม่เป็นเชิงเส้น ขาดเสถียรภาพ และควบคุมได้ยาก โดยพบว่าวิธีการควบคุมดังกล่าวยังคงใช้เวลาค่อนข้างมากจึงจะสามารถสร้างเสถียรภาพให้ระบบได้ ดังนั้นในงานวิจัยวิทยานิพนธ์นี้จึงดำเนินการพัฒนาวิธีการควบคุมแบบใหม่ขึ้น เพื่อให้สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผัน

แบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้น ส่วนในบทที่ 2 ของวิทยานิพนธ์ได้กล่าวถึงระบบควบคุมแบบฟuzzy เพื่อเป็นความรู้พื้นฐานในการพัฒนาวิธีการควบคุมต่อไป ในบทที่ 3 ของวิทยานิพนธ์ได้อธิบายหลักการการทำงานของตัวควบคุมฟuzzyแบบกฎอินพุตเดียว รวมทั้งพัฒนาโปรแกรม MATLAB ขึ้นสำหรับการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบด้วยตัวควบคุมดังกล่าว เมื่อสถานการณ์ที่ควบคุมเปลี่ยนแปลงไป โดยตรวจสอบความถูกต้องของโปรแกรมที่พัฒนาขึ้นกับผลการจำลองสถานการณ์ที่นำเสนอโดย Yi และ Yubazaki (2000) ส่วนวิทยานิพนธ์ในบทที่ 4 ได้กล่าวถึงโครงสร้างของตัวควบคุมนิวโร-ฟuzzyที่พัฒนาขึ้น หลักการทำงานของตัวควบคุมดังกล่าวในการสร้างเสถียรภาพให้ระบบ รวมทั้งผลการจำลองสถานการณ์เมื่อปรับเปลี่ยนสถานการณ์ในการควบคุม โดยใช้ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการสุ่มค่าเพียงชุดเดียวตลอดไป แม้จะปรับเปลี่ยนสถานการณ์ที่ควบคุม ซึ่งพบว่าสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาอันสั้น รวมทั้งได้ศึกษาผลของค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมจากการสุ่มค่าที่ส่งผลกระทบต่อการสร้างเสถียรภาพให้ระบบ พบว่า ถ้าค่าพารามิเตอร์เริ่มต้นไม่เหมาะสม โครงข่ายประสาทเทียมจะไม่สามารถเรียนรู้เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับสถานการณ์ที่เปลี่ยนแปลงไปได้ ในบทที่ 5 ของวิทยานิพนธ์ ได้นำเสนอโครงสร้างของตัวควบคุมนิวโร-ตาบู-ฟuzzy โดยนำวิธีการค้นหาแบบตามูเชิงปรับตัวเข้าช่วยในการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม เพื่อลดผลกระทบจากค่าพารามิเตอร์เริ่มต้นที่ไม่เหมาะสมจากการสุ่มค่า โดยยังคงใช้ค่าพารามิเตอร์เริ่มต้นที่ได้จากวิธีการค้นหาแบบตามูเชิงปรับตัวเพียงชุดเดียวตลอดไป ซึ่งจากผลการจำลองสถานการณ์แสดงให้เห็นว่าโครงสร้างของตัวควบคุมที่นำเสนอนี้ สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้ภายในระยะเวลาอันสั้นเช่นกัน

การทดสอบสมรรถนะของตัวควบคุมฟuzzyแบบกฎอินพุตเดียว ตัวควบคุมนิวโร-ฟuzzy และตัวควบคุมนิวโร-ตาบู-ฟuzzy ต่อการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวได้นำเสนอในบทที่ 6 ของวิทยานิพนธ์ โดยได้เปรียบเทียบการจำลองสถานการณ์ทั้ง 5 สถานการณ์ ทดสอบด้วยตัวควบคุมแบบต่างๆ พบว่า ตัวควบคุมนิวโร-ฟuzzy และตัวควบคุมนิวโร-ตาบู-ฟuzzy สามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟuzzyแบบกฎอินพุตเดียว รวมทั้งสามารถสร้างเสถียรภาพให้ระบบได้ทั้ง 5 สถานการณ์ ในขณะที่ตัวควบคุมฟuzzyแบบกฎอินพุตเดียวไม่สามารถสร้างเสถียรภาพให้ระบบได้ในบางสถานการณ์ จากนั้นได้ดำเนินการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบ ได้แก่ มุมเริ่มต้นของลูกตุ้ม ตำแหน่งเริ่มต้นของตัวรถ และความยาวก้านลูกตุ้ม ซึ่งจากผลการทดสอบพบว่า ตัวควบคุมนิวโร-ฟuzzy และตัวควบคุมนิวโร-ตาบู-ฟuzzy ยังคงสามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมฟuzzyแบบกฎอินพุตเดียว รวมทั้งมีขอบเขตการสร้างเสถียรภาพ

ที่กว้างกว่าตัวควบคุมพีชชีแบบกฏอินพุตเดียวด้วยเช่นกัน ต่อมาได้ดำเนินการทดสอบสมรรถนะของตัวควบคุมแบบต่างๆ เมื่อมีการรบกวนภายนอกมากระทำต่อระบบขณะที่ตัวควบคุมกำลังสร้างเสถียรภาพให้ระบบ โดยพิจารณาที่เวลา 2 วินาที และกรณีที่การรบกวนภายนอกมากระทำต่อระบบหลังจากที่ตัวควบคุมแบบต่างๆ สร้างเสถียรภาพให้ระบบเสร็จสมบูรณ์แล้ว นั่นคือ มีการรบกวนระบบที่เวลา 10 วินาที การรบกวนกระทำต่อระบบในรูปของมุมลูกตุ้มที่ถูกรบกวน เมื่อพิจารณาผลการทดสอบอาจกล่าวได้ว่า ตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-ตาบูลู-พีชชี ยังคงสามารถรักษาเสถียรภาพให้ระบบไว้ได้ แม้ขนาดของมุมที่มารบกวนจะเพิ่มขึ้นก็ตาม รวมทั้งสามารถทำให้เสถียรภาพของระบบกลับคืนมาดังเดิมในช่วงเวลาอันสั้นกว่าตัวควบคุมพีชชีแบบกฏอินพุตเดียวด้วย และเมื่อพิจารณาปริมาณการเข้าสู่เสถียรภาพของตัวควบคุมนิวโร-พีชชี และตัวควบคุมนิวโร-ตาบูลู-พีชชี ที่ได้จากการทดสอบ พบว่า ตัวควบคุมนิวโร-ตาบูลู-พีชชี มีปริมาณการเข้าสู่เสถียรภาพที่มากกว่าตัวควบคุมนิวโร-พีชชี ดังนั้นอาจกล่าวได้ว่า ค่าพารามิเตอร์เริ่มต้นที่เหมาะสมจากวิธีการค้นหาแบบตาบูลูเชิงปรับตัว ช่วยให้การเรียนรู้อันชาญฉลาดของโครงข่ายประสาทเทียมสามารถปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับสถานการณ์ที่แปรเปลี่ยนไปได้อย่างมีประสิทธิภาพยิ่งขึ้น นอกจากนี้โครงสร้างตัวควบคุมนิวโร-ตาบูลู-พีชชีที่พัฒนาขึ้น ยังสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อได้ภายในระยะเวลาอันสั้นด้วยเช่นกัน

## 7.2 ข้อเสนอแนะ

1. ค่าอัตราการเรียนรู้ ( $\eta$ ) และค่าคงที่โมเมนตัม ( $\alpha$ ) ในกระบวนการเรียนรู้อันชาญฉลาดของโครงข่ายประสาทเทียม ควรได้รับการปรับปรุงให้สามารถปรับตัวเองได้ (adaptive) เพื่อให้กระบวนการเรียนรู้อันชาญฉลาดของโครงข่ายประสาทเทียมสามารถปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับแต่ละสถานการณ์ที่ควบคุมได้อย่างมีประสิทธิภาพยิ่งขึ้น รวมทั้งช่วยให้โครงข่ายประสาทเทียมสามารถเรียนรู้ได้อย่างรวดเร็วขึ้นด้วย

2. ควรค้นหาชุดพารามิเตอร์เริ่มต้นที่เหมาะสมของโครงข่ายประสาทเทียมเพิ่มขึ้น เพื่อให้สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันได้ในขอบเขตที่กว้างขึ้น กล่าวคือ ผลการจำลองสถานการณ์ที่ผ่านมา ใช้ค่าพารามิเตอร์เริ่มต้นที่เหมาะสมจากวิธีการค้นหาแบบตาบูลูเชิงปรับตัว ที่จุดปฏิบัติการเดียว นั่นคือ มุมเริ่มต้นของลูกตุ้มเป็น 30 องศา ตำแหน่งเริ่มต้นของตัวรถเป็นศูนย์เมตร และก้านลูกตุ้มยาว 1.0 เมตร โดยนำค่าพารามิเตอร์เริ่มต้นที่ค้นหาได้นี้ไปใช้เป็นค่าพารามิเตอร์เริ่มต้นให้กับกระบวนการเรียนรู้อันชาญฉลาดของโครงข่ายประสาทเทียมตลอดไป ซึ่งเมื่อสถานการณ์ที่ควบคุมเปลี่ยนไปมาจากสถานการณ์ที่ใช้ค้นหาพารามิเตอร์เริ่มต้นเหล่านี้ ย่อมทำให้โครงข่ายประสาทเทียมไม่สามารถเรียนรู้ เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับทุกสถานการณ์ได้ ดังนั้น



ควรเพิ่มสถานการณ์ในการค้นหาค่าพารามิเตอร์เริ่มต้นเหล่านี้ เพื่อให้มีค่าพารามิเตอร์เริ่มต้นที่เหมาะสมหลายชุด

3. ควรศึกษาการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว เมื่อต้องควบคุมแรงดันอินพุตที่ป้อนให้กับมอเตอร์ เนื่องจากในทางปฏิบัติตัวรถอาจวางอยู่บนสายพาน ที่เชื่อมต่อกับพื้นเฟืองและมอเตอร์ ดังนั้นเพื่อให้ก้านลูกตุ้มตั้งตรง และตัวรถสามารถหยุดตรงตำแหน่งที่ต้องการได้ จำเป็นต้องควบคุมแรงดันอินพุตของมอเตอร์ให้เหมาะสม เพื่อกำหนดแรงกระทำต่อตัวรถที่สามารถสร้างเสถียรภาพให้ระบบได้ นอกจากนี้ ควรพิจารณาผลจากการสั่นของสายพานขณะที่ตัวรถเคลื่อนที่ด้วย

4. การนำโครงสร้างตัวควบคุมนิวโร-ตาบู-ฟิชชี ไปใช้สร้างเสถียรภาพให้ระบบอื่นๆ ควรกำหนดตัวปรับค่าสำหรับตัวแปรอินพุตแต่ละตัวให้เหมาะสม รวมทั้งตัวปรับค่าทางด้านเอาต์พุตของระบบด้วย เพื่อให้ตัวควบคุมนิวโร-ตาบู-ฟิชชี ยังคงสามารถสร้างเสถียรภาพให้ระบบได้ในขอบเขตที่กว้าง

5. ควรปรับปรุงวิธีการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่งเดิมใช้วิธีการเรียนรู้แบบแพร่กลับค่าความคลาดเคลื่อน เพื่อให้โครงข่ายประสาทเทียมสามารถเรียนรู้และปรับค่าระดับความสำคัญพลวัตให้เหมาะกับสถานการณ์การควบคุมที่แปรเปลี่ยนไปได้อย่างรวดเร็วยิ่งขึ้น

6. ควรศึกษาวิธีการตรวจสอบเสถียรภาพของตัวควบคุมนิวโร-ฟิชชี และตัวควบคุมนิวโร-ตาบู-ฟิชชี ที่พัฒนาขึ้นในการสร้างเสถียรภาพให้กับระบบที่มีความไม่เป็นเชิงเส้น

## รายการอ้างอิง

- สรารุฒิ สุจิตจร. (2538). ฟัซซี่ลอจิกกับระบบควบคุม. วารสารเทคโนโลยีสุรนารี 2(3): 171-196.
- Cavalcante, C., and Hirota, K. (1998). SIRMs connected fuzzy inference model tuning using genetic algorithm. **Proc. Fuzzy Systems IEEE Int. Conf. World Congress on Computational Intelligence**. 2: 1277-1280.
- Cui, X., and Shin, K. G. (1993). Direct control and coordination using neural networks. **IEEE Trans. Systems, Man, and Cybernetics**. 23(3): 686-697.
- Di, L., and Katsunori, I. (1997). Neural network based self-organized fuzzy logic control in arc welding process. **Proc. KES'97 Int. Conf. Knowledge-Based Intelligent Electronic Systems**. 2: 684-689.
- Filippidis, A., Jain, L. C., and Silva, C. W. de. (1999). Intelligent control techniques. In L. C. Jain and C. W. de Silva (eds). **Intelligent adaptive control industrial applications** (pp. 1-24). Boca Raton: CRC Press LLC.
- Hagras, H., and Sobh, T. (2002). Intelligent learning and control of autonomous robotic agents operating in unstructured environments. **Information Sciences**. 145(1-2): 1-12.
- Haykin, S. (1994). **Neural Networks a comprehensive foundation**. New York: Macmillan.
- Iokibe, T., Tobi, T., and Araki, S. (1999). Intelligent control of air conditioning systems. In L. C. Jain and C. W. de Silva (eds). **Intelligent adaptive control industrial applications** (pp. 219-246). Boca Raton: CRC Press LLC.

- Jin, W., Li, Z. J., Wei, L. S., and Zhen, H. (2000). The improvements of BP neural network learning algorithm. **Proc. WCCC-ICSP Int. Conf. Signal Processing**. 3: 1647-1649.
- Kandadai, R. M., and Tien, J. M. (1996). On a fuzzy-neural hierarchical controller with a self-generating knowledge base. **IEEE Int. Conf. Systems, Man, and Cybernetics**. 4: 2625-2630.
- King, R. E. (1999). **Computational intelligence in control engineering**. New York: Marcel Dekker.
- Kyung, K. H., and Lee, B. H. (1993). Fuzzy rule base derivation using neural network-based fuzzy logic controller by self-learning. **Proc. IECON'93 Int. Conf. Industrial Electronics, Control, and Instrumentation**. 1: 435-440.
- Meireles, M. R. G., Almeida, P. E. M., and Simoes, M. G. (2003). A comprehensive review for industrial applicability of artificial neural network. **IEEE Trans. Industrial Electronics**. 50(3): 585-601.
- Mizumoto, M. (1996). Product-sum-gravity method = fuzzy singleton-type reasoning method = simplified fuzzy reasoning method. **IEEE Proc. Int. Conf. Fuzzy Systems**. 3: 2098-2102.
- Omatu, S., Fujinaka, T., and Yoshioka, M. (2000). Neuro-PID control for inverted single and double pendulums. **IEEE Int. Conf. Systems, Man, and Cybernetics**. 4: 2685-2690.
- Puangdownreong, D., Areerak, K-N., Srikaew, A., Sujitjorn, S., and Totarong, P. (2002). System identification via adaptive tabu search. **IEEE Int. Conf. Industrial Technology ICIT'02**. 2: 915-920.

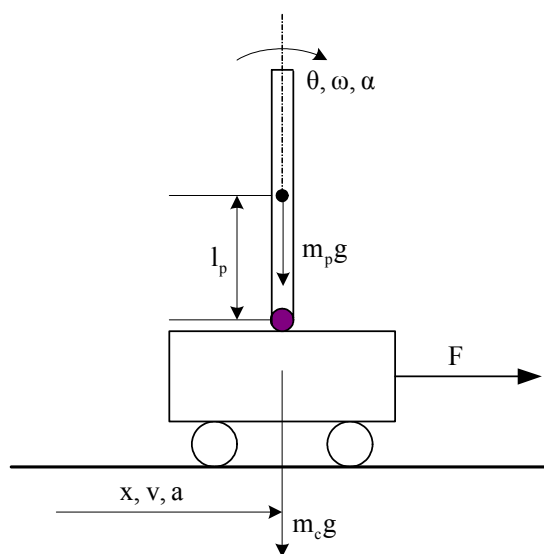
- Puangdownreong, D., Kulworawanichpong, T., and Sujitjorn, S. (2004). Finite convergence and performance evaluation of adaptive tabu search. **Lecture Notes in Computer Science**. Springer-Verlag Heidelberg. (online)
- Sakai, S., and Takahama, T. (2001). Learning fuzzy control rules by vector simplex method. **Int. Conf. IFSA World Congress and NAFIPS**. 5: 2541-2546.
- Sujitjorn, S., Mellitt, B., and Rabukwella, N. B. (1987). Energy minimisation in transit systems using a train based expert system for dynamic coast control. **Computer in railway operations**. Springer-Verlag: 303-316.
- Sun, X-M., Ren, C-M., Wu, Y-W., Ning, L-Q., and Wang, J-R. (2003). The design and application of neural network controller based on genetic algorithms. **Int. Conf. Machine Learning and Cybernetics**. 3: 1318-1321.
- Thapa, B. K., Jones, B., and Zhu, Q. M. (2000). Non-linear control with neural networks. **Proc. Int. Conf. Knowledge-Based Intelligent Engineering Systems and Allied Technologies**. 2: 868-873.
- Toh, K-A. (2003). Deterministic global optimization for FNN training. **IEEE Trans. Systems, Man, and Cybernetics Part B**. 33(6): 977-983.
- Yi, J., and Yubazaki, N. (2000). Stabilization fuzzy control of inverted pendulum systems. **Artificial Intelligent in Engineering**. 14(2): 153-163.
- Yi, J., Yubazaki, N., and Hirota, K. (2001). Stabilization control of series-type double inverted pendulum systems using the SIRMs dynamically connected fuzzy inference model. **Artificial Intelligence in Engineering**. 15(3): 297-308.

## ภาคผนวก ก

แบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบข้อต่อเดียว

### แบบจำลองทางคณิตศาสตร์ของลูกตุ้มผกผันแบบข้อต่อเดียว

ระบบลูกตุ้มผกผันแบบข้อต่อเดียวเป็นตัวอย่างระบบที่นิยมนำมาทดสอบสมรรถนะและประสิทธิภาพของตัวควบคุมที่ได้พัฒนาขึ้นมาอย่างแพร่หลาย เนื่องจากเป็นระบบที่มีความไม่เป็นเชิงเส้น ขาดเสถียรภาพ และควบคุมได้ยาก ดังนั้นในการออกแบบตัวควบคุมเพื่อสร้างเสถียรภาพให้ระบบดังกล่าว จึงจำเป็นอย่างยิ่งที่ต้องวิเคราะห์แบบจำลองทางคณิตศาสตร์ของระบบอย่างถูกต้อง เพื่อให้ได้ผลการจำลองสถานการณ์ที่น่าเชื่อถือ ระบบลูกตุ้มผกผันแบบข้อต่อเดียวที่พิจารณาแสดงได้ด้วยแผนภาพดังรูปที่ ก.1 ระบบดังกล่าวประกอบด้วยรางเลื่อน ตัวรถ ก้านลูกตุ้มผกผัน และแรงขับเคลื่อนที่กระทำต่อตัวรถในแนวระดับ โดยที่ตัวรถสามารถเคลื่อนที่ไปทางด้านซ้ายหรือด้านขวาบนรางเลื่อนอย่างอิสระตามทิศทางแรงกระทำ ก้านลูกตุ้มถูกแขวนติดตรงส่วนกลางของตัวรถ สามารถหมุนได้อย่างอิสระตามทิศทางการเคลื่อนที่ของรถ ทั้งนี้กำหนดให้ไม่มีแรงเสียดทานเกิดขึ้นกับระบบในขณะที่ทำการควบคุม



รูปที่ ก.1 ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

โดยที่ ค่าพารามิเตอร์ของระบบลูกตุ้มผกผันแบบข้อต่อเดียว กำหนดดังนี้คือ

$F$  คือ แรงขับเคลื่อนที่กระทำต่อตัวรถในแนวระดับ (N)

$m_c$  คือ มวลของตัวรถ (kg)

$m_p$  คือ มวลของก้านลูกตุ้มผกผัน (kg)

$g$  คือ ค่าแรงโน้มถ่วงของโลกเท่ากับ  $9.8 \text{ (m/sec}^2\text{)}$

$l_p$  คือ ความยาวของก้านลูกตุ้มจากตรงกลางถึงจุดต่อของตัวรถหรือความยาวครึ่งหนึ่งของความยาวก้านทั้งหมด (m)

$\theta$  คือ มุมของลูกตุ้มผกผันพิจารณาจากตำแหน่งสมดุล (rad)

$\omega$  คือ ความเร็วเชิงมุมของลูกตุ้มผกผัน (rad/sec)

$\alpha$  คือ ความเร่งเชิงมุมของลูกตุ้มผกผัน (rad/sec<sup>2</sup>)

$x$  คือ ตำแหน่งตัวรถพิจารณาจากจุดกึ่งกลางบนรางเลื่อน (m)

$v$  คือ ความเร็วของตัวรถ (m/sec)

$a$  คือ ความเร่งของตัวรถ (m/sec<sup>2</sup>)

การวิเคราะห์แบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบข้อต่อเดียว สามารถพิจารณาได้หลายวิธี เช่น ใช้กฎการเคลื่อนที่ข้อที่สองของนิวตัน โดยการพิจารณาแนวแรงที่เกิดขึ้นกับระบบ และวิธีของลากรานจ์ (Lagrange's method) ซึ่งแสดงความสัมพันธ์ของระบบในรูปแบบสมการเชิงอนุพันธ์ที่เรียกว่าสมการของลากรานจ์ (Lagrange's equation) โดยในวิชานี้จะประยุกต์ใช้วิธีการของลากรานจ์เข้าช่วยวิเคราะห์หาแบบจำลองทางคณิตศาสตร์ จนกระทั่งได้สมการพลวัต (dynamic equation) ของระบบ เพื่อใช้สำหรับการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว

เมื่อพิจารณาสมการของลากรานจ์ จะได้ดังนี้

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}_i} \right] - \frac{\partial L}{\partial x_i} + \frac{\partial D}{\partial x_i} = Q_i \quad (\text{ก-1})$$

เมื่อ  $x_i$  คือ ตัวแปรสถานะที่พิจารณา

$Q_i$  คือ แรงภายนอกที่กระทำกับระบบ

$D$  คือ พลังงานซึมซาบ (dissipation energy)

$L$  คือ ฟังก์ชันของลากรานจ์ ซึ่งสามารถเขียนให้อยู่ในรูป  $L = K - P$

โดยที่  $K$  คือ ผลรวมของพลังงานจลน์ (kinetic energy)

$P$  คือ ผลรวมของพลังงานศักย์ (potential energy)

$x_1$  คือ มุมของลูกตุ้มผกผันแทนด้วย  $\theta$

$x_2$  คือ ความเร็วเชิงมุมของลูกตุ้มแทนด้วย  $\omega$

$x_3$  คือ ตำแหน่งของตัวรถแทนด้วย  $x$

$x_4$  คือ ความเร็วในการเคลื่อนที่ของตัวรถแทนด้วย  $v$

เนื่องจากตัวแปรสถานะที่พิจารณาคือมุมของลูกตุ้มและตำแหน่งของตัวรถ ดังนั้นจะได้ว่า

$$x_i = \begin{bmatrix} \theta \\ x \end{bmatrix} \text{ และ } Q_i = \begin{bmatrix} F \\ 0 \end{bmatrix} \text{ เมื่อ } F \text{ คือ แรงที่ทำให้ตัวรถเคลื่อนที่}$$

กำหนดให้  $K = \frac{1}{2} m \dot{x}^2$  คือ พลังงานจลน์สำหรับการเคลื่อนที่เชิงเส้นของตัวรถ

$K = \frac{1}{2} J \dot{\theta}^2$  คือ พลังงานจลน์สำหรับการเคลื่อนที่เชิงมุมของลูกตุ้ม

$P = mgh$  คือ พลังงานศักย์สำหรับการเคลื่อนที่ในแนวตั้ง

$D = \frac{1}{2} b \dot{x}$  คือ พลังงานซึมซาบสำหรับการเคลื่อนที่เชิงเส้นของตัวรถ

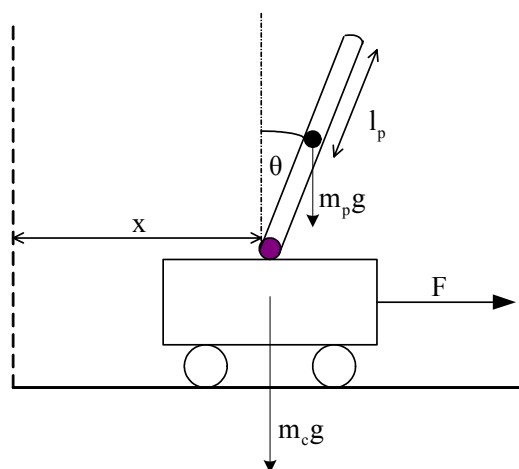
$D = \frac{1}{2} b \dot{\theta}$  คือ พลังงานซึมซาบสำหรับการเคลื่อนที่เชิงมุมของลูกตุ้ม

โดยที่  $m$  คือ มวลของวัตถุที่พิจารณา (kg)

$h$  คือ ความสูงจากจุดที่พิจารณา (m)

$b$  คือ สัมประสิทธิ์แรงเสียดทาน

$J$  คือ ค่าโมเมนต์ความเฉื่อย (moment of inertia)



รูปที่ ก.2 แผนภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว

พลังงานจลน์รวมของระบบลูกตุ้มผกผันแบบข้อต่อเดียวคือ

$$K = K_{\text{cart}} + K_{\text{pendulum}} \quad (\text{ก-2})$$

โดยที่  $K_{\text{cart}} = \frac{1}{2} m_c \dot{x}^2$



$$K_{\text{pendulum}} = \frac{1}{2} m_p \dot{s}^2 + \frac{1}{2} J \dot{\theta}^2$$

ซึ่ง  $s = s_x + s_y$

จากรูปที่ ก.2 จะได้ว่า

$$s_x = x + l_p \sin \theta$$

$$s_y = l_p \cos \theta$$

ดังนั้น  $\dot{s}^2 = \dot{s}_x^2 + \dot{s}_y^2$

$$\begin{aligned} \dot{s}^2 &= \left\{ \frac{d}{dt} [x + l_p \sin \theta] \right\}^2 + \left\{ \frac{d}{dt} [l_p \cos \theta] \right\}^2 \\ &= [\dot{x} + l_p \dot{\theta} \cos \theta]^2 + [l_p \dot{\theta} \sin \theta]^2 \\ &= \dot{x}^2 + 2\dot{x} \dot{\theta} l_p \cos \theta + l_p^2 \dot{\theta}^2 \cos^2 \theta + l_p^2 \dot{\theta}^2 \sin^2 \theta \\ &= \dot{x}^2 + 2\dot{x} \dot{\theta} l_p \cos \theta + l_p^2 \dot{\theta}^2 \end{aligned}$$

ดังนั้น  $K_{\text{pendulum}} = \frac{1}{2} m_p (\dot{x}^2 + 2\dot{x} \dot{\theta} l_p \cos \theta + l_p^2 \dot{\theta}^2) + \frac{1}{2} J \dot{\theta}^2$

เนื่องจากพิจารณาว่าก้านลูกตุ้มแข็งเกร็งจะได้  $J = \frac{1}{3} m_p l_p^2$

$$K_{\text{pendulum}} = \frac{1}{2} m_p (\dot{x}^2 + 2\dot{x} \dot{\theta} l_p \cos \theta + l_p^2 \dot{\theta}^2) + \frac{1}{2} \left( \frac{1}{3} m_p l_p^2 \right) \dot{\theta}^2 \quad (\text{ก-3})$$

ดังนั้นจากสมการที่ (ก-2) จะได้พลังงานจลน์รวมคือ

$$K = \frac{1}{2} m_c \dot{x}^2 + \frac{1}{2} m_p (\dot{x}^2 + 2\dot{x}\dot{\theta}l_p \cos\theta + l_p^2 \dot{\theta}^2) + \frac{1}{6} m_p l_p^2 \dot{\theta}^2 \quad (\text{ก-4})$$

พลังงานศักย์รวมของระบบลูกตุ้มผกผันคือ

$$P = P_{\text{cart}} + p_{\text{pendulum}} \quad (\text{ก-5})$$

จากรูปที่ ก.2 จะได้ว่า

$$P_{\text{cart}} = 0$$

$$p_{\text{pendulum}} = m_p g l_p \cos\theta \quad \text{โดยที่ } h = l_p \cos\theta$$

ดังนั้นจากสมการที่ (ก-5) จะได้พลังงานศักย์รวมคือ

$$P = m_p g l_p \cos\theta \quad (\text{ก-6})$$

เนื่องจาก  $L = K - P$  ดังนั้นจะได้

$$L = \frac{1}{2} m_c \dot{x}^2 + \frac{1}{2} m_p (\dot{x}^2 + 2\dot{x}\dot{\theta}l_p \cos\theta + l_p^2 \dot{\theta}^2) + \frac{1}{6} m_p l_p^2 \dot{\theta}^2 - m_p g l_p \cos\theta \quad (\text{ก-7})$$

จากสมการที่ (ก-7) จะได้

$$L = \frac{1}{2} m_c \dot{x}^2 + \frac{1}{2} m_p \dot{x}^2 + m_p \dot{x}\dot{\theta}l_p \cos\theta + \frac{2}{3} m_p l_p^2 \dot{\theta}^2 - m_p g l_p \cos\theta$$

จากสมการที่ (ก-1) เมื่อไม่คิดผลของพลังงานซึมซาบจะได้ว่า

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}_i} \right] - \frac{\partial L}{\partial x_i} = Q_i \quad (\text{ก-8})$$

พิจารณาตัวแปรสถานะ เมื่อ  $x_i = x$  และ  $Q_i = F$

$$\frac{\partial L}{\partial \dot{x}} = (m_c + m_p) \dot{x} + m_p \dot{\theta} l_p \cos \theta$$

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}} \right] = (m_c + m_p) \ddot{x} - m_p \dot{\theta}^2 l_p \sin \theta + m_p \ddot{\theta} l_p \cos \theta$$

$$\frac{\partial L}{\partial x} = 0$$

จากสมการที่ (ก-8) จะได้

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}} \right] - \frac{\partial L}{\partial x} = F$$

$$(m_c + m_p) \ddot{x} - m_p l_p \dot{\theta}^2 \sin \theta + m_p l_p \cos \theta \ddot{\theta} = F \quad (\text{ก-9})$$

พิจารณาตัวแปรสถานะ เมื่อ  $x_i = \theta$  และ  $Q_i = 0$

$$\frac{\partial L}{\partial \dot{\theta}} = m_p \dot{x} l_p \cos \theta + \frac{4}{3} m_p l_p^2 \dot{\theta}$$

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{\theta}} \right] = m_p \ddot{x} l_p \cos \theta - m_p \dot{x} \dot{\theta} l_p \sin \theta + \frac{4}{3} m_p l_p^2 \ddot{\theta}$$

$$\frac{\partial L}{\partial \theta} = -m_p \dot{x} \dot{\theta} l_p \sin \theta + m_p g l_p \sin \theta$$

ดังนั้นจากสมการที่ (ก-8) จะได้

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{\theta}} \right] - \frac{\partial L}{\partial \theta} = 0$$

$$m_p \ddot{x} l_p \cos \theta - m_p \dot{x} \dot{\theta} l_p \sin \theta + \frac{4}{3} m_p l_p^2 \ddot{\theta} + m_p \dot{x} \dot{\theta} l_p \sin \theta - m_p g l_p \sin \theta = 0$$

$$m_p l_p \cos \theta \ddot{x} + \frac{4}{3} m_p l_p^2 \ddot{\theta} - m_p g l_p \sin \theta = 0 \quad (\text{ก-10})$$

ดังนั้นจากสมการที่ (ก-9) และ (ก-10) จะได้สมการพลวัตของระบบลูกตุ้มผกผันแบบข้อต่อเดียวคือ

$$(m_c + m_p)\ddot{x} - m_p l_p \dot{\theta}^2 \sin\theta + m_p l_p \cos\theta \ddot{\theta} = F$$

$$m_p l_p \cos\theta \ddot{x} + \frac{4}{3} m_p l_p^2 \ddot{\theta} - m_p g l_p \sin\theta = 0$$

นั่นคือ

$$\ddot{\theta} = \frac{(m_c + m_p)g \sin\theta - \{F + m_p l_p \dot{\theta}^2 \sin\theta\} \cos\theta}{\left\{ \frac{4}{3}(m_c + m_p) - m_p \cos^2\theta \right\} l_p} \quad (\text{ก-11})$$

$$\ddot{x} = \frac{\frac{4}{3} \{F + m_p l_p \dot{\theta}^2 \sin\theta\} - m_p g \sin\theta \cos\theta}{\left\{ \frac{4}{3}(m_c + m_p) - m_p \cos^2\theta \right\}} \quad (\text{ก-12})$$

หรือ

$$\alpha = \frac{(m_c + m_p)g \sin\theta - \{F + m_p l_p \omega^2 \sin\theta\} \cos\theta}{\left\{ \frac{4}{3}(m_c + m_p) - m_p (\cos\theta)^2 \right\} l_p} \quad (\text{ก-13})$$

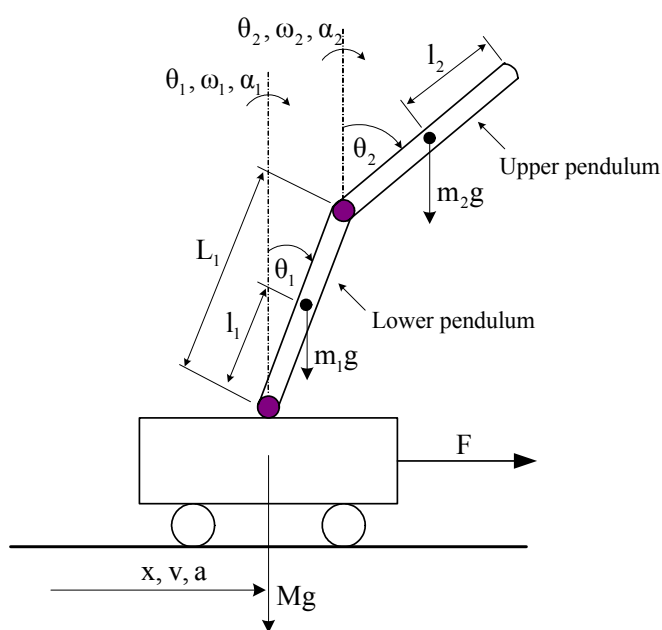
$$a = \frac{\frac{4}{3} \{F + m_p l_p \omega^2 \sin\theta\} - m_p g \sin\theta \cos\theta}{\left\{ \frac{4}{3}(m_c + m_p) - m_p (\cos\theta)^2 \right\}} \quad (\text{ก-14})$$

ภาคผนวก ข

แบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบสองข้อต่อ

### แบบจำลองทางคณิตศาสตร์ของลูกตุ้มผกผันแบบสองข้อต่อ

ระบบลูกตุ้มผกผันแบบสองข้อต่อที่พิจารณาแสดงได้ดังรูปที่ ข.1 ระบบดังกล่าวประกอบด้วยรางเลื่อนซึ่งมีลักษณะตรง ตัวรถที่สามารถเคลื่อนที่ไปมาบนรางเลื่อนได้อย่างอิสระ ก้านลูกตุ้มอันล่างซึ่งแขวนติดอยู่กับตัวรถ ก้านลูกตุ้มอันบนที่เชื่อมต่อกับส่วนปลายของก้านลูกตุ้มอันล่าง และแรงที่ใช้ขับเคลื่อนให้รถเคลื่อนที่ โดยที่ก้านลูกตุ้มทั้งสองสามารถหมุนในทิศทางเข็มนาฬิกาและตามเข็มนาฬิกาตามทิศทางของแนวแรงที่กระทำกับตัวรถ กำหนดให้ไม่มีแรงเสียดทานเกิดขึ้นกับระบบในขณะที่ทำการควบคุม



รูปที่ ข.1 ระบบลูกตุ้มผกผันแบบสองข้อต่อ

โดยที่ ค่าพารามิเตอร์ของระบบลูกตุ้มผกผันแบบสองข้อต่อ กำหนดดังนี้คือ

$M$  คือ มวลของตัวรถ (kg)

$m_1$  คือ มวลของก้านลูกตุ้มอันล่าง (kg)

$m_2$  คือ มวลของก้านลูกตุ้มอันบน (kg)

$g$  คือ ค่าแรงโน้มถ่วงของโลกเท่ากับ  $9.8 \text{ (m/sec}^2\text{)}$

$L_1$  คือ ความยาวของก้านลูกตุ้มอันล่าง (m)

$l_1$  คือ ความยาวครึ่งหนึ่งจากความยาวทั้งหมดของก้านลูกตุ้มอันล่าง (m)

$l_2$  คือ ความยาวครึ่งหนึ่งจากความยาวทั้งหมดของก้านลูกตุ้มอันบน (m)

- F คือ แรงขับเคลื่อนที่กระทำกับตัวรถในแนวระดับ (N)
- $\theta_1$  คือ มุมของลูกตุ้มผกผันอันล่างเมื่อพิจารณาจากตำแหน่งสมมูล (rad)
- $\theta_2$  คือ มุมของลูกตุ้มผกผันอันบนเมื่อพิจารณาจากตำแหน่งสมมูล (rad)
- $\omega_1$  คือ ความเร็วเชิงมุมของลูกตุ้มอันล่าง (rad/sec)
- $\omega_2$  คือ ความเร็วเชิงมุมของลูกตุ้มอันบน (rad/sec)
- $\alpha_1$  คือ ความเร่งเชิงมุมของลูกตุ้มอันล่าง (rad/sec<sup>2</sup>)
- $\alpha_2$  คือ ความเร่งเชิงมุมของลูกตุ้มอันบน (rad/sec<sup>2</sup>)
- x คือ ตำแหน่งตัวรถเมื่อพิจารณาจากจุดกึ่งกลางบนรางเลื่อน (m)
- v คือ ความเร็วของตัวรถ (m/sec)
- a คือ ความเร่งของตัวรถ (m/sec<sup>2</sup>)

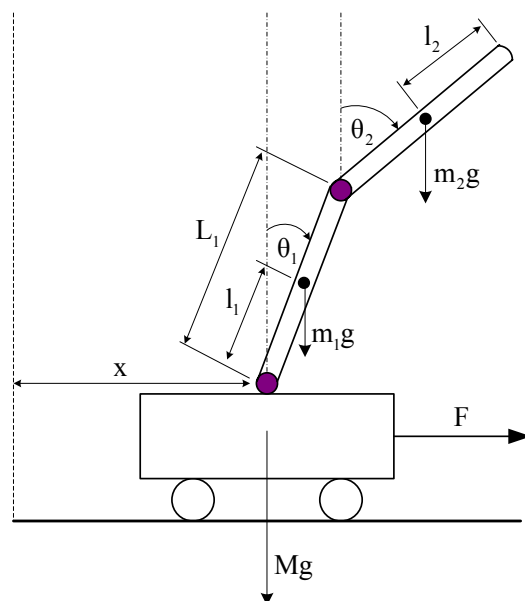
กำหนดให้ตำแหน่งของตัวรถที่พิจารณาจากจุดกึ่งกลางของรางเลื่อนมีค่าเป็นบวก เมื่อรถเคลื่อนที่ไปทางด้านขวาของรางเลื่อน มุมของลูกตุ้มอันบนและมุมของลูกตุ้มอันล่างจะมีทิศทางบวก เมื่อหมุนตามเข็มนาฬิกา เช่นเดียวกับแรงขับเคลื่อนที่กระทำกับตัวรถจะมีค่าเป็นบวก ก็ต่อเมื่อผลักดันให้ตัวรถเคลื่อนที่ไปทางด้านขวาของรางเลื่อน ในการวิเคราะห์แบบจำลองทางคณิตศาสตร์ของระบบ ลูกตุ้มผกผันแบบสองข้อต่อ สามารถพิจารณาด้วยวิธีของลากรานจ์ (Lagrange's method) ซึ่งแสดงความสัมพันธ์ตามสมการที่ (ข-1)

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}_i} \right] - \frac{\partial L}{\partial x_i} + \frac{\partial D}{\partial x_i} = Q_i \quad (\text{ข-1})$$

- เมื่อ  $x_i$  คือ ตัวแปรสถานะที่พิจารณา
- $Q_i$  คือ แรงภายนอกที่กระทำกับระบบ
- D คือ พลังงานซึมซาบ (dissipation energy)
- L คือ ฟังก์ชันของลากรานจ์ ซึ่งสามารถเขียนให้อยู่ในรูป  $L = K - P$
- โดยที่ K คือ ผลรวมของพลังงานจลน์ (kinetic energy)
- P คือ ผลรวมของพลังงานศักย์ (potential energy)
- $x_1$  คือ ตำแหน่งของตัวรถแทนด้วย x
- $x_2$  คือ มุมของลูกตุ้มอันล่างแทนด้วย  $\theta_1$
- $x_3$  คือ มุมของลูกตุ้มอันบนแทนด้วย  $\theta_2$

เนื่องจากตัวแปรสถานะที่พิจารณา คือ มุมของลูกตุ้มอันล่าง มุมของลูกตุ้มอันบน และ ตำแหน่งของตัวรถ และ  $F$  คือ แรงกระทำทำให้ตัวรถเคลื่อนที่ ดังนั้นจะได้ว่า

$$x_i = \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad \text{และ} \quad Q_i = \begin{bmatrix} F \\ 0 \\ 0 \end{bmatrix}$$



รูปที่ ข.2 แผนภาพของระบบลูกตุ้มผกผันแบบสองข้อต่อ

พลังงานจลน์รวมของระบบลูกตุ้มผกผันแบบสองข้อต่อ คือ

$$K = K_{\text{cart}} + K_{\text{lower pendulum}} + K_{\text{upper pendulum}} \quad (\text{ข-2})$$

หรือแทนด้วย

$$K = K_1 + K_2 + K_3$$

โดยที่ 
$$K_1 = \frac{1}{2} M \dot{x}^2$$



$$K_2 = \frac{1}{2}m_1\dot{s}_1^2 + \frac{1}{2}J_1\dot{\theta}_1^2$$

$$K_3 = \frac{1}{2}m_2\dot{s}_2^2 + \frac{1}{2}J_2\dot{\theta}_2^2$$

ซึ่ง  $J_1$  และ  $J_2$  คือ โมเมนต์ความเฉื่อยสำหรับก้านลูกตุ้มอันล่าง และก้านลูกตุ้มอันบน ตามลำดับ  
พิจารณา ก้านลูกตุ้มอันล่างจากรูปที่ ข.2 จะได้ว่า

$$s_{1x} = x + l_1 \sin \theta_1$$

$$s_{1y} = l_1 \cos \theta_1$$

$$\dot{s}_{1x} = \dot{x} + l_1 \dot{\theta}_1 \cos \theta_1$$

$$\dot{s}_{1y} = -l_1 \dot{\theta}_1 \sin \theta_1$$

ดังนั้น

$$\dot{s}_1^2 = \dot{s}_{1x}^2 + \dot{s}_{1y}^2$$

$$\begin{aligned} \dot{s}_1^2 &= [\dot{x} + l_1 \dot{\theta}_1 \cos \theta_1]^2 + [-l_1 \dot{\theta}_1 \sin \theta_1]^2 \\ &= \dot{x}^2 + 2\dot{x}\dot{\theta}_1 l_1 \cos \theta_1 + l_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 + l_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 \\ &= \dot{x}^2 + 2\dot{x}\dot{\theta}_1 l_1 \cos \theta_1 + l_1^2 \dot{\theta}_1^2 \end{aligned}$$

$$J_1 = \frac{1}{3}m_1 l_1^2$$

ดังนั้น

$$K_2 = \frac{1}{2}m_1 (\dot{x}^2 + 2\dot{x}\dot{\theta}_1 l_1 \cos \theta_1 + l_1^2 \dot{\theta}_1^2) + \frac{1}{6}m_1 l_1^2 \dot{\theta}_1^2 \quad (\text{ข-3})$$

พิจารณากำหนดลูกตุ้มอันบนจากรูปที่ ข.2 จะได้ว่า

$$s_{2x} = x + L_1 \sin \theta_1 + l_2 \sin \theta_2$$

$$s_{2y} = L_1 \cos \theta_1 + l_2 \cos \theta_2$$

$$\dot{s}_{2x} = \dot{x} + L_1 \dot{\theta}_1 \cos \theta_1 + l_2 \dot{\theta}_2 \cos \theta_2$$

$$\dot{s}_{2y} = -L_1 \dot{\theta}_1 \sin \theta_1 - l_2 \dot{\theta}_2 \sin \theta_2 = -(L_1 \dot{\theta}_1 \sin \theta_1 + l_2 \dot{\theta}_2 \sin \theta_2)$$

ดังนั้น

$$\dot{s}_2^2 = \dot{s}_{2x}^2 + \dot{s}_{2y}^2$$

$$\begin{aligned} \dot{s}_2^2 &= \left[ \dot{x} + L_1 \dot{\theta}_1 \cos \theta_1 + l_2 \dot{\theta}_2 \cos \theta_2 \right]^2 + \left[ -(L_1 \dot{\theta}_1 \sin \theta_1 + l_2 \dot{\theta}_2 \sin \theta_2) \right]^2 \\ &= \dot{x}^2 + 2\dot{x}\dot{\theta}_1 L_1 \cos \theta_1 + 2\dot{x}\dot{\theta}_2 l_2 \cos \theta_2 + 2\dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos \theta_1 \cos \theta_2 \\ &\quad + L_1^2 \dot{\theta}_1^2 (\cos^2 \theta_1 + \sin^2 \theta_1) + l_2^2 \dot{\theta}_2^2 (\cos^2 \theta_2 + \sin^2 \theta_2) \\ &\quad + 2\dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \sin \theta_1 \sin \theta_2 \\ &= \dot{x}^2 + 2\dot{x}\dot{\theta}_1 L_1 \cos \theta_1 + 2\dot{x}\dot{\theta}_2 l_2 \cos \theta_2 + 2\dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) + L_1^2 \dot{\theta}_1^2 \\ &\quad + l_2^2 \dot{\theta}_2^2 \end{aligned}$$

$$J_2 = \frac{1}{3} m_2 l_2^2$$

ดังนั้น

$$\begin{aligned} K_3 &= \frac{1}{2} m_2 [\dot{x}^2 + 2\dot{x}\dot{\theta}_1 L_1 \cos \theta_1 + 2\dot{x}\dot{\theta}_2 l_2 \cos \theta_2 + 2\dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) \\ &\quad + L_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2] + \frac{1}{6} m_2 l_2^2 \dot{\theta}_2^2 \end{aligned} \quad (\text{ข-4})$$

ดังนั้นจากสมการที่ (ข-2) จะได้พลังงานจลน์รวมคือ

$$\begin{aligned}
 K &= \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m_1 (\dot{x}^2 + 2\dot{x}\dot{\theta}_1 l_1 \cos\theta_1 + l_1^2 \dot{\theta}_1^2) + \frac{1}{6} m_1 l_1^2 \dot{\theta}_1^2 \\
 &+ \frac{1}{2} m_2 [\dot{x}^2 + 2\dot{x}\dot{\theta}_1 L_1 \cos\theta_1 + 2\dot{x}\dot{\theta}_2 l_2 \cos\theta_2 + 2\dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) \\
 &+ L_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2] + \frac{1}{6} m_2 l_2^2 \dot{\theta}_2^2 \\
 &= \frac{1}{2} (M + m_1 + m_2) \dot{x}^2 + \frac{2}{3} m_1 l_1^2 \dot{\theta}_1^2 + \frac{2}{3} m_2 l_2^2 \dot{\theta}_2^2 + (m_1 l_1 + m_2 L_1) \dot{x} \dot{\theta}_1 \cos\theta_1 \\
 &+ m_2 \dot{x} \dot{\theta}_2 l_2 \cos\theta_2 + m_2 \dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) + \frac{1}{2} m_2 L_1^2 \dot{\theta}_1^2
 \end{aligned}$$

พลังงานศักย์รวมของระบบลูกตุ้มผกผันคือ (ข-5)

$$P = P_{\text{cart}} + P_{\text{lower pendulum}} + P_{\text{upper pendulum}} \quad (\text{ข-6})$$

จากรูปที่ ข.2 จะได้ว่า

$$P_{\text{cart}} = 0$$

$$P_{\text{lower pendulum}} = m_1 g l_1 \cos\theta_1$$

$$P_{\text{upper pendulum}} = m_2 g (L_1 \cos\theta_1 + l_2 \cos\theta_2)$$

ดังนั้นจากสมการที่ (ข-6) จะได้พลังงานศักย์รวมคือ

$$P = m_1 g l_1 \cos\theta_1 + m_2 g (L_1 \cos\theta_1 + l_2 \cos\theta_2) \quad (\text{ข-7})$$

เนื่องจาก  $L = K - P$  จะได้

$$\begin{aligned}
 L = & \frac{1}{2}(M + m_1 + m_2)\dot{x}^2 + \frac{2}{3}m_1l_1^2\dot{\theta}_1^2 + \frac{2}{3}m_2l_2^2\dot{\theta}_2^2 + (m_1l_1 + m_2L_1)\dot{x}\dot{\theta}_1\cos\theta_1 \\
 & + m_2\dot{x}\dot{\theta}_2l_2\cos\theta_2 + m_2\dot{\theta}_1\dot{\theta}_2L_1l_2\cos(\theta_1 - \theta_2) + \frac{1}{2}m_2L_1^2\dot{\theta}_1^2 - m_1gl_1\cos\theta_1 \\
 & - m_2g(L_1\cos\theta_1 + l_2\cos\theta_2)
 \end{aligned} \tag{ข-8}$$

จากสมการที่ (ข-1) เมื่อไม่คิดผลของพลังงานซึมซาบจะได้ว่า

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}_i} \right] - \frac{\partial L}{\partial x_i} = Q_i \tag{ข-9}$$

พิจารณาตัวแปรสถานะ เมื่อ  $x_i = x_1 = x$  และ  $Q_1 = F$

$$\begin{aligned}
 \frac{\partial L}{\partial \dot{x}} &= (M + m_1 + m_2)\dot{x} + (m_1l_1 + m_2L_1)\dot{\theta}_1\cos\theta_1 + m_2l_2\dot{\theta}_2\cos\theta_2 \\
 \frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}} \right] &= (M + m_1 + m_2)\ddot{x} + (m_1l_1 + m_2L_1)\ddot{\theta}_1\cos\theta_1 - (m_1l_1 + m_2L_1)\dot{\theta}_1^2\sin\theta_1 \\
 & \quad + m_2l_2\ddot{\theta}_2\cos\theta_2 - m_2l_2\dot{\theta}_2^2\sin\theta_2 \\
 \frac{\partial L}{\partial x} &= 0
 \end{aligned}$$

จากสมการที่ (ข-9) จะได้

$$\begin{aligned}
 \frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{x}} \right] - \frac{\partial L}{\partial x} &= F \\
 (M + m_1 + m_2)\ddot{x} + (m_1l_1 + m_2L_1)\ddot{\theta}_1\cos\theta_1 - (m_1l_1 + m_2L_1)\dot{\theta}_1^2\sin\theta_1 \\
 + m_2l_2\ddot{\theta}_2\cos\theta_2 - m_2l_2\dot{\theta}_2^2\sin\theta_2 &= F
 \end{aligned}$$

นั่นคือ

$$\begin{aligned} (M + m_1 + m_2) \ddot{x} + (m_1 l_1 + m_2 L_1) \cos \theta_1 \ddot{\theta}_1 + m_2 l_2 \cos \theta_2 \ddot{\theta}_2 \\ = F + (m_1 l_1 + m_2 L_1) \dot{\theta}_1^2 \sin \theta_1 + m_2 l_2 \dot{\theta}_2^2 \sin \theta_2 \end{aligned} \quad (\text{ข-10})$$

พิจารณาตัวแปรสถานะ เมื่อ  $x_1 = x_2 = \theta_1$  และ  $Q_2 = 0$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 l_1 + m_2 L_1) \dot{x} \cos \theta_1 + \frac{4}{3} m_1 l_1^2 \dot{\theta}_1 + m_2 \dot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) + m_2 L_1^2 \dot{\theta}_1$$

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{\theta}_1} \right] = (m_1 l_1 + m_2 L_1) \ddot{x} \cos \theta_1 + \frac{4}{3} m_1 l_1^2 \ddot{\theta}_1 - (m_1 l_1 + m_2 L_1) \dot{x} \dot{\theta}_1 \sin \theta_1$$

$$+ m_2 \ddot{\theta}_2 L_1 l_2 \cos(\theta_1 - \theta_2) - m_2 \dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \sin(\theta_1 - \theta_2)$$

$$+ m_2 \dot{\theta}_2^2 L_1 l_2 \sin(\theta_1 - \theta_2) + m_2 L_1^2 \ddot{\theta}_1$$

$$\frac{\partial L}{\partial \theta_1} = -(m_1 l_1 + m_2 L_1) \dot{x} \dot{\theta}_1 \sin \theta_1 - m_2 L_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_1 l_1 g \sin \theta_1$$

$$+ m_2 L_1 g \sin \theta_1$$

ดังนั้นจากสมการที่ (ข-9) จะได้

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{\theta}_1} \right] - \frac{\partial L}{\partial \theta_1} = 0$$

$$(m_1 l_1 + m_2 L_1) \cos \theta_1 \ddot{x} + \left( \frac{4}{3} m_1 l_1^2 + m_2 L_1^2 \right) \ddot{\theta}_1 + m_2 L_1 l_2 \cos(\theta_1 - \theta_2) \ddot{\theta}_2$$

$$+ m_2 \dot{\theta}_2^2 L_1 l_2 \sin(\theta_1 - \theta_2) - (m_1 l_1 + m_2 L_1) g \sin \theta_1 = 0 \quad (\text{ข-11})$$

$$\begin{aligned}
& (m_1 l_1 + m_2 L_1) \cos \theta_1 \ddot{x} + \left( \frac{4}{3} m_1 l_1^2 + m_2 L_1^2 \right) \ddot{\theta}_1 + m_2 L_1 l_2 \cos(\theta_1 - \theta_2) \ddot{\theta}_2 \\
& = (m_1 l_1 + m_2 L_1) g \sin \theta_1 - m_2 \dot{\theta}_2^2 L_1 l_2 \sin(\theta_1 - \theta_2)
\end{aligned}$$

พิจารณาตัวแปรสถานะ เมื่อ  $x_1 = x_3 = \theta_2$  และ  $Q_3 = 0$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_2 \dot{x} \cos \theta_2 + \frac{4}{3} m_2 l_2^2 \dot{\theta}_2 + m_2 \dot{\theta}_1 L_1 l_2 \cos(\theta_1 - \theta_2)$$

$$\begin{aligned}
\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{\theta}_2} \right] &= m_2 l_2 \ddot{x} \cos \theta_2 + \frac{4}{3} m_2 l_2^2 \ddot{\theta}_2 - m_2 l_2 \dot{x} \dot{\theta}_2 \sin \theta_2 + m_2 \ddot{\theta}_1 L_1 l_2 \cos(\theta_1 - \theta_2) \\
&+ m_2 \dot{\theta}_1 \dot{\theta}_2 L_1 l_2 \sin(\theta_1 - \theta_2) - m_2 \dot{\theta}_1^2 L_1 l_2 \sin(\theta_1 - \theta_2)
\end{aligned}$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_2 \dot{x} \sin \theta_2 + m_2 L_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_2 l_2 g \sin \theta_2$$

ดังนั้นจากสมการที่ (ข-9) จะได้

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{\theta}_2} \right] - \frac{\partial L}{\partial \theta_2} = 0$$

$$m_2 l_2 \cos \theta_2 \ddot{x} + m_2 L_1 l_2 \cos(\theta_1 - \theta_2) \ddot{\theta}_1 + \frac{4}{3} m_2 l_2^2 \ddot{\theta}_2$$

$$- m_2 \dot{\theta}_1^2 L_1 l_2 \sin(\theta_1 - \theta_2) - m_2 l_2 g \sin \theta_2 = 0$$

$$m_2 l_2 \cos \theta_2 \ddot{x} + m_2 L_1 l_2 \cos(\theta_1 - \theta_2) \ddot{\theta}_1 + \frac{4}{3} m_2 l_2^2 \ddot{\theta}_2$$

$$= m_2 \dot{\theta}_1^2 L_1 l_2 \sin(\theta_1 - \theta_2) + m_2 l_2 g \sin \theta_2 \quad (\text{ข-12})$$

ดังนั้นจากสมการที่ (ข-10) ถึง (ข-12) จะได้สมการพลวัตของระบบลูกตุ้มผกผันแบบสองข้อต่อคือ

$$a_{11}\ddot{x} + a_{12}\ddot{\theta}_1 + a_{13}\ddot{\theta}_2 = b_1$$

$$a_{21}\ddot{x} + a_{22}\ddot{\theta}_1 + a_{23}\ddot{\theta}_2 = b_2$$

$$a_{31}\ddot{x} + a_{32}\ddot{\theta}_1 + a_{33}\ddot{\theta}_2 = b_3$$

โดยที่

$$a_{11} = (M + m_1 + m_2)$$

$$a_{22} = \frac{4}{3}m_1l_1^2 + m_2L_1^2$$

$$a_{33} = \frac{4}{3}m_2l_2^2$$

$$a_{12} = a_{21} = (m_1l_1 + m_2L_1)\cos\theta_1$$

$$a_{13} = a_{31} = m_2l_2\cos\theta_2$$

$$a_{23} = a_{32} = m_2L_1l_2\cos(\theta_1 - \theta_2)$$

และ

$$b_1 = F + (m_1l_1 + m_2L_1)\dot{\theta}_1^2\sin\theta_1 + m_2l_2\dot{\theta}_2^2\sin\theta_2$$

$$b_2 = (m_1l_1 + m_2L_1)g\sin\theta_1 - m_2L_1l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2)$$

$$b_3 = m_2l_2g\sin\theta_2 + m_2L_1l_2\dot{\theta}_1^2\sin(\theta_1 - \theta_2)$$

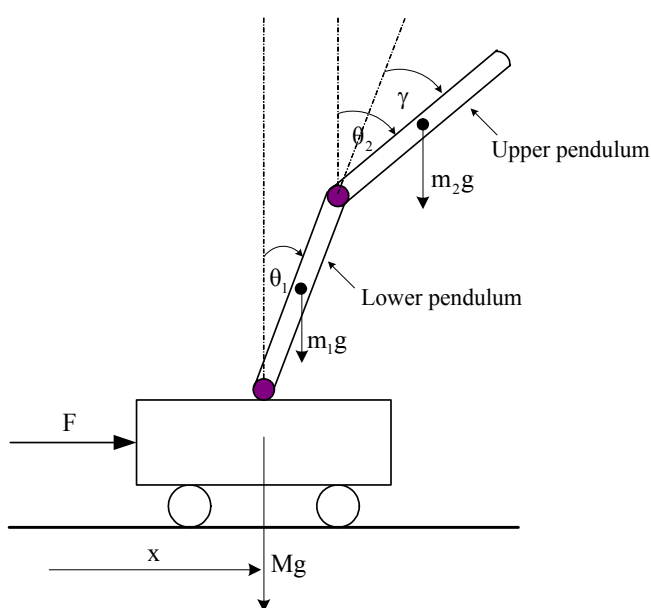
ภาคผนวก ค

การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อ  
ด้วยตัวควบคุมนิวโร-ตาบู-พีซซี



### ค.1 ระบบลูกตุ้มผกผันแบบสองข้อต่อ

ระบบลูกตุ้มผกผันแบบสองข้อต่อเป็นระบบที่มีความไม่เป็นเชิงเส้น ขาดเสถียรภาพ และควบคุมได้ยากกว่าระบบลูกตุ้มผกผันแบบข้อต่อเดียว ดังนั้นจึงเป็นระบบที่นำมาใช้ทดสอบความสามารถของตัวควบคุมนิวโร-คาบู่-ฟัซซี่ ที่พัฒนาขึ้น ระบบลูกตุ้มผกผันแบบสองข้อต่อที่ใช้ในการจำลองสถานการณ์มีโครงสร้างดังรูปที่ ค.1 ซึ่งประกอบด้วยรางเส้นตรง โดยตัวรถสามารถเคลื่อนที่ไปมาบนรางเส้นได้อย่างอิสระ และก้านลูกตุ้มทั้งสองที่เชื่อมต่อกับตัวรถ ทำให้สามารถหมุนได้ทั้งทิศทางเข็มนาฬิกาและตามเข็มนาฬิกาตามทิศทางของแรงกระทำ



รูปที่ ค.1 ระบบลูกตุ้มผกผันแบบสองข้อต่อ

เมื่อค่าพารามิเตอร์ของระบบเป็นดังนี้

$m_1$  คือ มวลของก้านลูกตุ้มอันล่าง (kg)

$m_2$  คือ มวลของก้านลูกตุ้มอันบน (kg)

$M$  คือ มวลของตัวรถ (kg)

$x$  คือ ตำแหน่งของตัวรถ (m)

$\theta_1$  คือ มุมของลูกตุ้มอันล่าง (rad)

$\theta_2$  คือ มุมของลูกตุ้มอันบน (rad)

$\gamma$  คือ ผลต่างเชิงมุมระหว่างลูกตุ้มอันบนและลูกตุ้มอันล่าง นั่นคือ  $\gamma = \theta_2 - \theta_1$

กำหนดให้ไม่มีแรงเสียดทานเกิดขึ้นกับระบบ ดังนั้นสมการพลวัตของระบบลูกตุ้มผกผันแบบสองข้อต่อจะได้ดังสมการที่ (ค-1) ซึ่งพิจารณาจากสมการของลากรางจ์ ดังได้อธิบายรายละเอียดในภาคผนวก ข.

$$a_{11}\ddot{x} + a_{12}\ddot{\theta}_1 + a_{13}\ddot{\theta}_2 = b_1$$

$$a_{21}\ddot{x} + a_{22}\ddot{\theta}_1 + a_{23}\ddot{\theta}_2 = b_2 \quad (\text{ค-1})$$

$$a_{31}\ddot{x} + a_{32}\ddot{\theta}_1 + a_{33}\ddot{\theta}_2 = b_3$$

โดยที่ค่าสัมประสิทธิ์ของสมการกำหนดดังนี้คือ

$$a_{11} = (M + m_1 + m_2)$$

$$a_{22} = \frac{4}{3}m_1l_1^2 + m_2L_1^2$$

$$a_{33} = \frac{4}{3}m_2l_2^2$$

$$a_{12} = a_{21} = (m_1l_1 + m_2L_1)\cos\theta_1$$

$$a_{13} = a_{31} = m_2l_2\cos\theta_2$$

$$a_{23} = a_{32} = m_2L_1l_2\cos(\theta_1 - \theta_2)$$

และ  $b_1 = F + (m_1l_1 + m_2L_1)\dot{\theta}_1^2\sin\theta_1 + m_2l_2\dot{\theta}_2^2\sin\theta_2$

$$b_2 = (m_1l_1 + m_2L_1)g\sin\theta_1 - m_2L_1l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2)$$

$$b_3 = m_2l_2g\sin\theta_2 + m_2L_1l_2\dot{\theta}_1^2\sin(\theta_1 - \theta_2)$$

ในการสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อ ต้องการให้ตัวรถเคลื่อนที่ กลับมายังตำแหน่งกึ่งกลางของรางลื่น และก้านลูกตุ้มทั้งสองอันสามารถตั้งตรงที่ตำแหน่งสมดุล ศูนย์องศา ซึ่งถ้าแรงกระทำมีค่าเป็นบวกจะส่งผลให้ตัวรถเคลื่อนที่ไปทางด้านขวาของรางลื่น แต่ถ้าแรงกระทำเป็นลบจะทำให้ตัวรถเคลื่อนที่กลับ ไปทางด้านซ้าย ทั้งนี้ทิศทางการเคลื่อนที่ของตัวรถจะ ส่งผลต่อทิศทางการหมุนของก้านลูกตุ้มทั้งสองด้วยเช่นกัน ดังนั้นจึงจำเป็นต้องควบคุมแรงกระทำ ต่อตัวรถให้ได้อย่างเหมาะสม จึงจะสามารถสร้างเสถียรภาพให้ระบบได้ ผลงานวิจัยที่เกี่ยวข้องกับ การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อ ได้แก่ การปรับค่าอัตราขยายของตัว ควบคุมพีโอดีด้วยตัวควบคุมนิวโร (Omatu, Fujinaka, and Yoshioka, 2000) รวมทั้งการใช้ ตัวควบคุมฟัซซี่แบบกฎอินพุตเดียว (Yi, Yubazaki, and Hirota, 2001) เป็นต้น โดยในงานวิจัย วิทยานิพนธ์นี้ได้เสนอตัวควบคุมนิวโร-ตามู-ฟัซซี่ เพื่อจำลองสถานการณ์การสร้างเสถียรภาพให้ ระบบลูกตุ้มผกผันแบบสองข้อต่อ

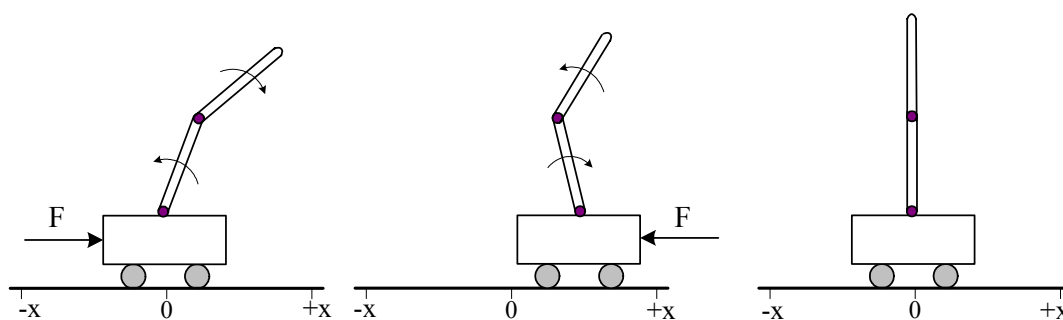
## ค.2 การออกแบบตัวควบคุมนิวโร-ตามู-ฟัซซี่

การออกแบบตัวควบคุมนิวโร-ตามู-ฟัซซี่ จะพิจารณาจากพฤติกรรมธรรมชาติของระบบ กล่าวคือ อันดับแรกควรควบคุมให้ก้านลูกตุ้มทั้งสองวางตัวในแนวเดียวกัน แล้วจึงพยายามทำให้ตั้ง ตรงที่ตำแหน่งสมดุล จากนั้นจึงทำให้รถเคลื่อนที่กลับมายังกึ่งกลางของรางลื่น ดังนั้นถ้า  $\gamma = 0$  หมายความว่า ก้านลูกตุ้มทั้งสองมีมุมเท่ากัน ในการออกแบบตัวควบคุมนิวโร-ตามู-ฟัซซี่ จะ พิจารณาตัวแปรอินพุตทั้งสิ้น 6 ค่า ได้แก่ ผลต่างเชิงมุมระหว่างลูกตุ้มทั้งสอง ( $\gamma$ ) ผลต่างความเร็ว เชิงมุม ( $\dot{\gamma}$ ) มุมของลูกตุ้มอันล่าง ( $\theta_1$ ) ความเร็วเชิงมุมของลูกตุ้มอันล่าง ( $\omega_1$ ) ตำแหน่งของตัวรถ ( $x$ ) และความเร็วของตัวรถ ( $v$ ) ซึ่งในการสร้างเสถียรภาพให้ระบบต้องการให้ตัวแปรทุกตัวมีค่าเข้าสู่ ศูนย์ หลังจากในตัวแปรอินพุตเหล่านี้ผ่านการนอร์มัลไลซ์ด้วยตัวปรับค่าที่กำหนดไว้จะได้ตัวแปร  $x_i$  ( $i = 1, 2, \dots, 6$ ) เป็นอินพุตให้กับการกำหนดกฎฟัซซี่แบบกฎอินพุตเดียว โดยที่ค่าระดับ ความสำคัญพลวัตสำหรับตัวแปรอินพุตแต่ละตัวได้มาจากการเรียนรู้ของโครงข่ายประสาทเทียม ซึ่ง ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมได้มาจากวิธีการค้นหาแบบตามูเชิงปรับตัว จนกระทั่งได้แรงกระทำต่อตัวรถที่สามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อได้

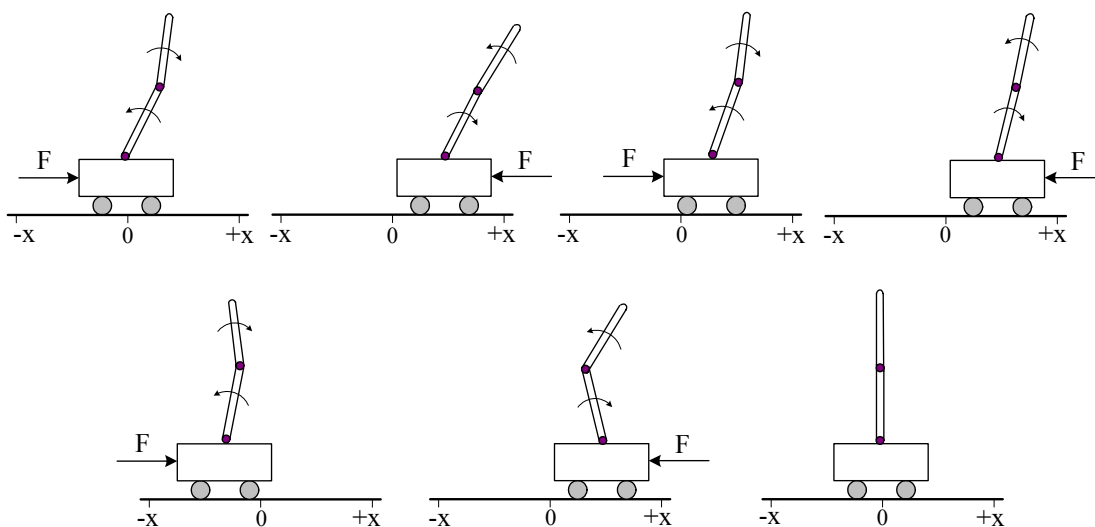
## ค.3 การกำหนดค่า SIRM

เมื่อพิจารณาพฤติกรรมธรรมชาติของระบบลูกตุ้มผกผันแบบสองข้อต่อ จะได้ว่าถ้าตัวรถ เคลื่อนที่ไปทางด้านขวา จะทำให้ลูกตุ้มอันล่างหมุนในทิศทวนเข็มนาฬิกา ในขณะที่เดียวกันจะทำให้ ลูกตุ้มอันบนหมุนตามเข็มนาฬิกา ซึ่งกรณีที่ผลต่างเชิงมุมมีค่าเป็นบวก นั่นคือ มุมของลูกตุ้มอันบน มากกว่ามุมของลูกตุ้มอันล่าง ถ้ามีแรงในทิศทางบวกกระทำกับตัวรถ จะทำให้ลูกตุ้มอันล่างหมุน

ทวนเข็มนาฬิกา และลูกตุ้มอันบนหมุนตามเข็มนาฬิกา ส่งผลให้ผลต่างเชิงมุมเพิ่มขึ้น ดังนั้นจึงจำเป็นต้องให้แรงกระทำในทิศทางลบกับทัวร์ถ เพื่อให้ลูกตุ้มอันล่างหมุนตามเข็มนาฬิกา และลูกตุ้มอันบนทวนเข็มนาฬิกา จนกระทั่งก้านลูกตุ้มทั้งสองตั้งตรง พร้อมกันนี้ตัวรถก็เคลื่อนที่กลับมายังกึ่งกลางรางคืนได้อีกครั้ง ซึ่งแสดงได้ดังรูปที่ ค.2



รูปที่ ค.2 การสร้างเสถียรภาพให้ระบบเมื่อผลต่างเชิงมุมมีค่าเป็นบวก



รูปที่ ค.3 การสร้างเสถียรภาพให้ระบบเมื่อผลต่างเชิงมุมมีค่าเป็นลบ

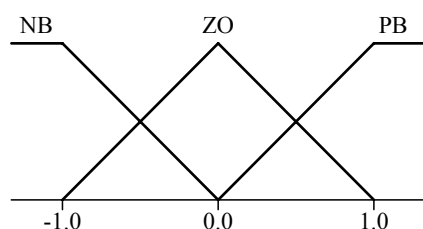
รูปที่ ค.3 เป็นกรณีที่ผลต่างเชิงมุมมีค่าเป็นลบ ถ้าแรงกระทำเป็นบวกจะทำให้ลูกตุ้มอันล่างหมุนทวนเข็มนาฬิกา และลูกตุ้มอันบนหมุนตามเข็มนาฬิกา จนกระทั่งมุมของลูกตุ้มทั้งสองเท่ากัน แต่ถ้ายังคงให้แรงกระทำในทิศทางบวกต่อไป จะทำให้ตัวรถเคลื่อนที่ไปทางด้านขวาห่างจากกึ่งกลางรางมากขึ้น ซึ่งส่งผลให้การสร้างเสถียรภาพให้ระบบยากยิ่งขึ้นด้วย ดังนั้นควรให้แรงกระทำที่เป็นลบ เพื่อให้ตัวรถเคลื่อนที่กลับมาทางด้านซ้าย และทำให้ลูกตุ้มอันล่างหมุนตามเข็มนาฬิกา ส่วน

ลูกค้อนอันบนจะหมุนทวนเข็มนาฬิกา จากนั้นจึงให้แรงกระทำที่มีทิศทางเป็นบวกและลบสลับกันไป เพื่อให้ตัวรถเคลื่อนที่ไปทางด้านซ้ายและด้านขวาของรางลื่น จนกระทั่งก้านลูกค้อนทั้งสองสามารถตั้งตรง และตัวรถเคลื่อนที่กลับมายังกึ่งกลางรางลื่นอีกครั้ง

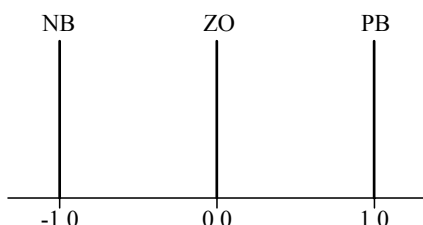
จากการวิเคราะห์ธรรมชาติของระบบลูกค้อนผกผันแบบสองข้อต่อ สามารถกำหนดกฎฟัซซี่ของตัวแปรแต่ละตัวสำหรับ SIRM ได้ดังตารางที่ ค.1 ซึ่งฟังก์ชันความเป็นสมาชิกสำหรับตัวแปรอินพุตกำหนดให้มีรูปแบบเหมือนกันคือ ประกอบด้วยรูปสามเหลี่ยมและรูปสี่เหลี่ยมคางหมู ดังแสดงในรูปที่ ค.4 ส่วนฟังก์ชันความเป็นสมาชิกของเอาต์พุตกำหนดให้เป็นจำนวนจริงเดี่ยว ซึ่งปรากฏดังรูปที่ ค.5

ตารางที่ ค.1 การกำหนดกฎฟัซซี่ใน SIRM

ตัวแปรในส่วนเงื่อนไข $x_i (i = 1, 2, \dots, 6)$	ตัวแปรในส่วนผลลัพธ์ $f_i (i = 1, 2, \dots, 6)$
NB	-1.0
ZO	0.0
PB	1.0



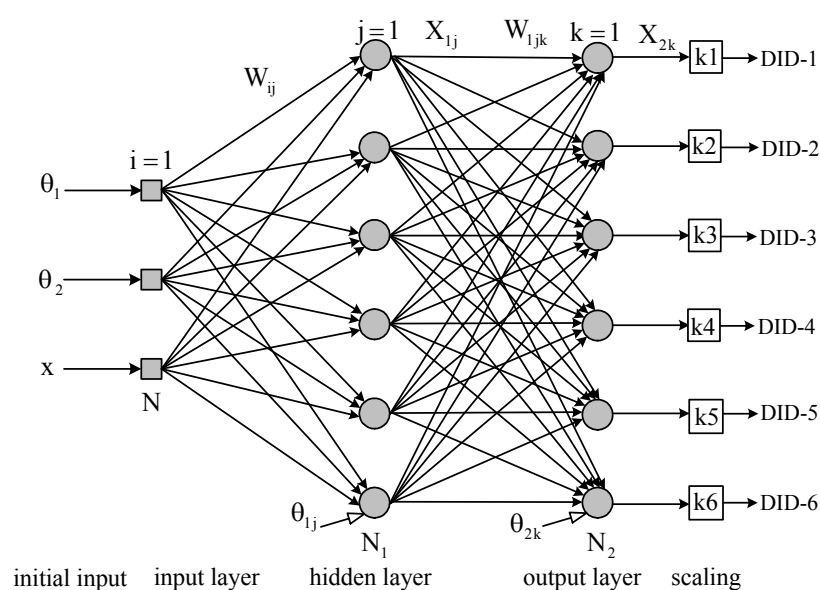
รูปที่ ค.4 ฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัวใน SIRM



รูปที่ ค.5 ฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตใน SIRM

#### ค.4 การกำหนดค่าระดับความสำคัญพลวัต

ในการสร้างเสถียรภาพให้กับระบบลูกค้อนผกผันแบบสองข้อต่อ ด้วยตัวควบคุมนิวโร-ตาบูปัสซี จะประกอบด้วยค่าระดับความสำคัญพลวัตทั้งหมด 6 ค่า สำหรับตัวแปรอินพุตแต่ละตัว ซึ่งค่าระดับความสำคัญพลวัตเหล่านี้ได้มาจากการเรียนรู้ของโครงข่ายประสาทเทียม ดังนั้นในชั้นเอาต์พุตของโครงข่ายประสาทเทียมจะมี 6 เซลล์ประสาทเทียม ส่วนชั้นอินพุตจะมี 3 เซลล์ประสาทเทียมสำหรับ มุมเริ่มต้นของลูกค้อนทั้งสองและตำแหน่งเริ่มต้นของตัวรถ ทั้งนี้โครงข่ายประสาทเทียมจะประกอบด้วยชั้นซ่อนเร้นหนึ่งชั้นซึ่งมี 6 เซลล์ประสาทเทียม ดังแสดงในรูปที่ ค.6



รูปที่ ค.6 ค่าระดับความสำคัญพลวัตที่ได้จากโครงข่ายประสาทเทียม

เมื่อพิจารณารูปที่ ค.6 จะสังเกตเห็นว่าประกอบด้วยค่าน้ำหนักและค่าไบแอสรวมทั้งสิ้น 66 ค่า ดังนั้นค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ต้องดำเนินการค้นหาด้วยวิธีการค้นหาแบบตาบูปัสซีปรับตัวจึงมีทั้งหมด 66 ค่า ซึ่งกำหนดให้ค่าพารามิเตอร์เริ่มต้นเหล่านี้มีขอบเขตอยู่ในช่วง  $[-1, +1]$  เนื่องจากในกระบวนการเรียนรู้ใช้ฟังก์ชันกระตุ้นแบบซิกมอยด์ ทำให้เอาต์พุตจากโครงข่ายประสาทเทียมมีค่าอยู่ในช่วง  $[0, +1]$  จึงจำเป็นต้องยกระดับเอาต์พุตที่ได้ด้วยตัวปรับค่าเพื่อให้ได้ค่าระดับความสำคัญพลวัตที่สามารถสร้างเสถียรภาพให้ระบบได้ กระบวนการเรียนรู้ของโครงข่ายประสาทเทียม เพื่อปรับค่าระดับความสำคัญพลวัตประกอบด้วย 4 ขั้นตอนดังนี้ คือ

ขั้นตอนที่ 1 ค้นหาพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ได้แก่ ค่าน้ำหนักและค่าไบแอส ด้วยวิธีการค้นหาแบบตาบูปัสซีปรับตัว โดยกำหนดให้มีค่าอยู่ในช่วง  $[-1, +1]$

ขั้นตอนที่ 2 คำนวณเอาต์พุตจากชั้นซ่อนเร้นและชั้นเอาต์พุต

- เอาต์พุตของชั้นซ่อนเร้น;  $X_{1j}$

$$X_{1j}(t) = \frac{1}{1 + \exp(-O_{1j} - \theta_{1j})}$$

โดยที่  $O_{1j} = \sum_{i=1}^3 W_{ij} X_i$  ,  $j=1, 2, \dots, 6$

- เอาต์พุตของชั้นเอาต์พุต;  $X_{2k}$

$$X_{2k}(t) = \frac{1}{1 + \exp(-O_{2k} - \theta_{2k})}$$

โดยที่  $O_{2k} = \sum_{j=1}^6 W_{jk} X_{1j}(t)$  ,  $k=1, 2, \dots, 6$

ขั้นตอนที่ 3 ปรับค่าน้ำหนักและค่าไบแอส

- ค่าน้ำหนักจากชั้นซ่อนเร้นไปยังชั้นเอาต์พุต;  $W_{1jk}$

$$W_{1jk}(t + \Delta t) = W_{1jk}(t) + \Delta W_{1jk}(t)$$

โดยที่  $\Delta W_{1jk}(t) = \eta \delta_{1k} X_{1j}(t) + \alpha \cdot \Delta W_{1jk}(t - \Delta t)$  และ

$$\delta_{1k} = (y_{kd}(t) - y_k(t)) \cdot \text{sign} \left( \frac{\partial y_k(t)}{\partial X_{2k}(t)} \right) \cdot X_{2k}(t) (1 - X_{2k}(t)) \quad (\text{ก-2})$$

- ค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้น;  $W_{ij}$

$$W_{ij}(t + \Delta t) = W_{ij}(t) + \Delta W_{ij}(t)$$

โดยที่  $\Delta W_{ij}(t) = \eta \delta_j X_i(t) + \alpha \cdot \Delta W_{ij}(t - \Delta t)$  และ

$$\delta_j = \left[ \sum_{k=1}^{N_2} \delta_{1k} W_{1jk}(t + \Delta t) \right] X_{1j}(t)(1 - X_{1j}(t))$$

- ค่าไบแอสที่ชั้นเอาต์พุต;  $\theta_{2k}$  และค่าไบแอสที่ชั้นซ่อนเร้น;  $\theta_{1j}$

$$\theta_{2k}(t + \Delta t) = \theta_{2k}(t) + \eta_{10} \delta_{1k}$$

$$\theta_{1j}(t + \Delta t) = \theta_{1j}(t) + \eta_0 \delta_j$$

ขั้นตอนที่ 4 ตรวจสอบเงื่อนไขเพื่อยุติการเรียนรู้ ถ้าไม่ตรงตามเงื่อนไขให้ปรับเพิ่มจำนวนรอบการเรียนรู้ และกลับไปยังขั้นตอนที่ 2

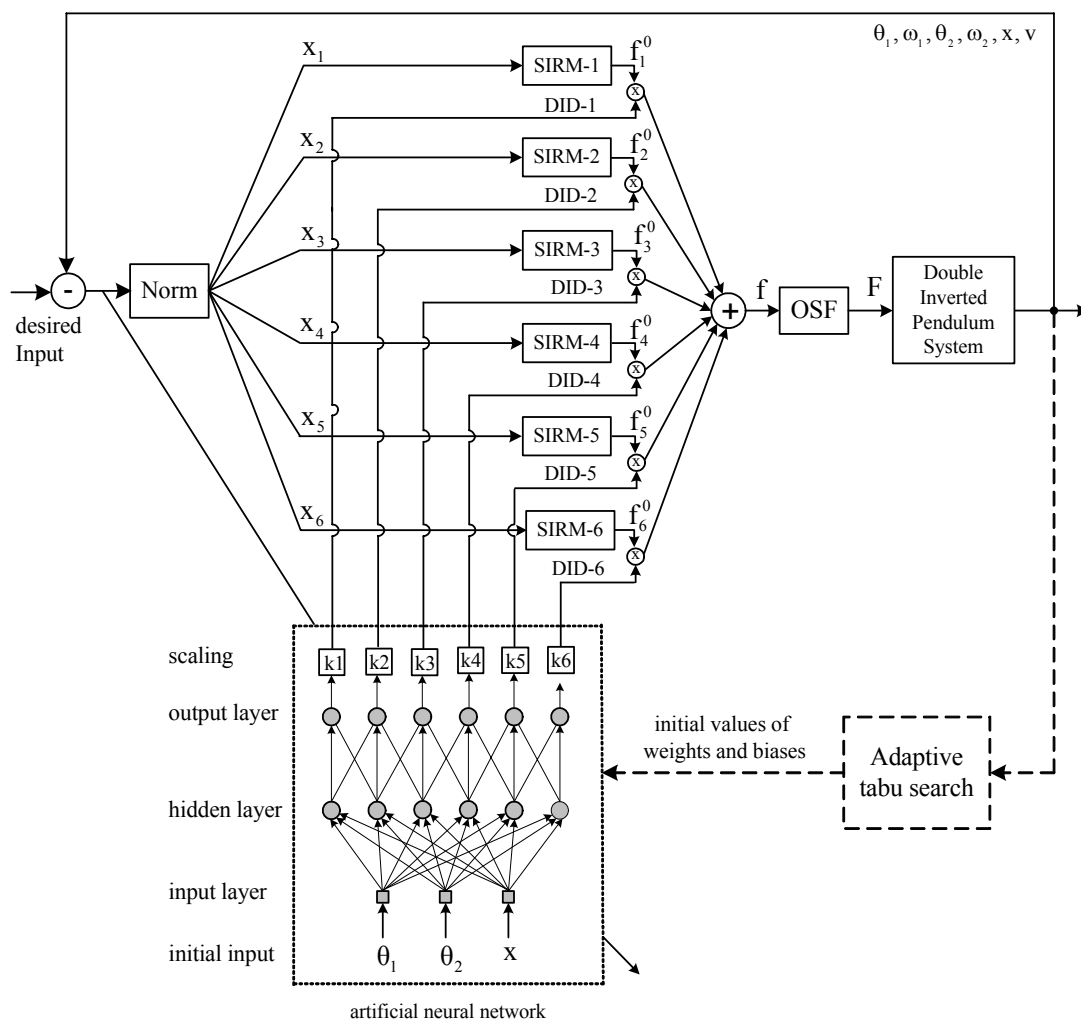
โดยค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัมที่ใช้ในกระบวนการเรียนรู้มีค่าเป็น 0.05 และ 0.25 ตามลำดับ อย่างไรก็ตาม ในการสร้างเสถียรภาพให้ระบบลูกคู้มผกผันแบบสองข้อต่อไม่สามารถระบุค่าระดับความสำคัญพลวัตที่ต้องการได้ ดังนั้นจะใช้เอาต์พุตของระบบคำนวณค่าความคลาดเคลื่อนสำหรับปรับค่าน้ำหนักและค่าไบแอส แทนเอาต์พุตที่ได้จากโครงข่ายประสาทเทียม ซึ่งกลไกดังกล่าวสามารถพิจารณาได้จากสมการที่ (ค-2)

#### ค.5 การสร้างเสถียรภาพให้ระบบลูกคู้มผกผันแบบสองข้อต่อ

โครงสร้างของตัวควบคุมนิวโร-ตาบิว-ฟิชชี ที่ใช้ในการสร้างเสถียรภาพให้ระบบลูกคู้มผกผันแบบสองข้อต่อแสดงดังรูปที่ ค.7 โดยตัวแปรอินพุต  $x_1$  ถึง  $x_6$  ที่ปรากฏในรูป หมายถึงผลต่างเชิงมุมของลูกคู้มทั้งสอง ( $x_1$ ) ผลต่างความเร็วเชิงมุมของลูกคู้มทั้งสอง ( $x_2$ ) มุมของลูกคู้มอันล่าง ( $x_3$ ) ความเร็วเชิงมุมของลูกคู้มอันล่าง ( $x_4$ ) ตำแหน่งของตัวรถ ( $x_5$ ) และความเร็วของตัวรถ ( $x_6$ ) จากนั้นตัวแปรอินพุตเหล่านี้จะถูกนอร์มัลไลซ์ให้อยู่ในช่วงเอกภพสัมพัทธ์ที่กำหนดด้วยตัวปรับค่าสำหรับตัวแปรอินพุตแต่ละตัวดังนี้ คือ 15 องศา 100 องศาต่อวินาที 15 องศา 100 องศาต่อวินาที 2.4 เมตร และ 1 เมตรต่อวินาที ตามลำดับ โดยตัวปรับค่าสำหรับแรงเอาต์พุต (OSF) กำหนดให้มีค่าเป็นสปีทเท่างของผลรวมระหว่างมวลของตัวรถกับมวลของก้านลูกคู้มทั้งสองอัน โดยพารามิเตอร์ของระบบลูกคู้มผกผันแบบสองข้อต่อที่ใช้ในการจำลองสถานการณ์แสดงดังตารางที่ ค.2 อย่างไรก็ตาม เมื่อพิจารณาค่าระดับความสำคัญพลวัตสำหรับตัวแปรอินพุตแต่ละตัวจากการเรียนรู้ของโครงข่ายประสาทเทียม พบว่าฟังก์ชันกระตุ้นที่ใช้เป็นแบบซิกมอยด์ ซึ่งทำให้เอาต์พุตที่ออกจากโครงข่ายประสาทเทียมมีค่าอยู่ในช่วง  $[0, 1]$  ดังนั้น จำเป็นต้องคูณค่าเหล่านี้ด้วยตัวปรับค่า เพื่อให้ได้ค่าระดับความสำคัญพลวัตที่มากพอสำหรับการสร้างเสถียรภาพให้ระบบ



ลูกตุ้มผกผันแบบสองข้อต่อ ซึ่งกำหนดให้ตัวปรับค่าสำหรับค่าระดับความสำคัญของตัวแปรอินพุตทุกตัวมีค่าเป็น 10



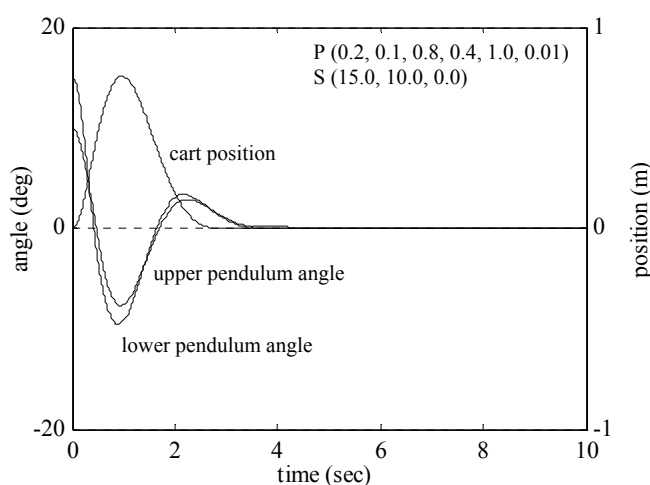
รูปที่ ค.7 โครงสร้างตัวควบคุมนิวโร-ตาบู-พีซสำหรับระบบลูกตุ้มผกผันแบบสองข้อต่อ

ตารางที่ ค.2 พารามิเตอร์ของระบบลูกตุ้มผกผันแบบสองข้อต่อ

พารามิเตอร์	ขนาด
มวลของตัวรถ	1.0 กิโลกรัม
มวลของก้านลูกตุ้มอันล่าง	0.2 กิโลกรัม
มวลของก้านลูกตุ้มอันบน	0.1 กิโลกรัม
ความยาวของก้านลูกตุ้มอันล่าง	0.8 เมตร
ความยาวของก้านลูกตุ้มอันบน	0.4 เมตร

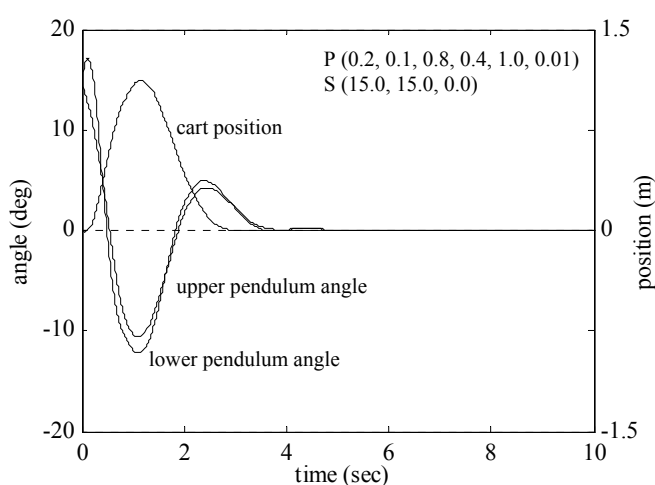
เพื่อทดสอบประสิทธิภาพของตัวควบคุมนิวโร-ตาบู-ฟิชซี ได้ดำเนินการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อ โดยปรับเปลี่ยนสถานการณ์ในการควบคุมให้แตกต่างกัน ตัวอย่างเช่น ทำการปรับเปลี่ยนมุมเริ่มต้นและความยาวก้านให้กับลูกตุ้มทั้งสองอัน เป็นต้น ซึ่งระบบดังกล่าวจะได้รับการพิจารณาว่าเข้าสู่เสถียรภาพแล้วก็ต่อเมื่อ มุมของลูกตุ้มทั้งสองคู่เข้าสู่  $\pm 0.1$  องศา ความเร็วเชิงมุมของลูกตุ้มทั้งสองคู่เข้าสู่  $\pm 0.1$  องศาต่อวินาที ตำแหน่งของตัวรถคู่เข้าสู่  $\pm 0.01$  เมตร และความเร็วของรถคู่เข้าสู่  $\pm 0.01$  เมตรต่อวินาที โดยตัวเลขที่ปรากฏใน  $P(0.2, 0.1, 0.8, 0.4, 1.0, 0.01)$  ซึ่งแทรกเป็นส่วนหนึ่งของรูปที่แสดงผลการจำลองสถานการณ์ หมายถึง มวลของก้านลูกตุ้มอันล่าง มวลของก้านลูกตุ้มอันบน ความยาวก้านลูกตุ้มอันล่าง ความยาวก้านลูกตุ้มอันบน มวลของตัวรถ และช่วงเวลาแซมปลิ่ง ตามลำดับ และตัวเลขที่ปรากฏใน  $S(15.0, 10.0, 0.0)$  หมายถึง มุมเริ่มต้นของลูกตุ้มอันล่าง มุมเริ่มต้นของลูกตุ้มอันบน และตำแหน่งของตัวรถ ตามลำดับ

รูปที่ ค.8 เป็นการจำลองสถานการณ์เมื่อมุมเริ่มต้นของลูกตุ้มอันล่างเป็น 15 องศา และมุมเริ่มต้นของลูกตุ้มอันบนเป็น 10 องศา โดยที่ตัวรถอยู่ตรงกึ่งกลางของรางลื่น ทำให้ผลต่างเชิงมุมมีค่าเป็น -5 องศา ซึ่งจะเห็นว่า เมื่อตัวรถเคลื่อนที่ไปทางด้านขวา จะทำให้ลูกตุ้มอันล่างหมุนทวนเข็มนาฬิกาและลูกตุ้มอันบนหมุนตามเข็มนาฬิกา ส่งผลให้ก้านลูกตุ้มทั้งสองเกือบวางตัวในแนวเดียวกัน จากนั้นจะให้แรงกระทำในทิศทางลบเพื่อทำให้ตัวรถเคลื่อนที่กลับมาทางด้านซ้าย ส่งผลให้ลูกตุ้มอันล่างหมุนตามเข็มนาฬิกาและลูกตุ้มอันบนหมุนทวนเข็มนาฬิกา จนกระทั่งก้านลูกตุ้มทั้งสองสามารถตั้งตรงที่ตำแหน่งสมดุล และตัวรถเคลื่อนที่กลับมายังกึ่งกลางของรางลื่นได้อีกครั้ง ซึ่งตัวควบคุมนิวโร-ตาบู-ฟิชซีสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อทั้งสิ้น 4.47 วินาที

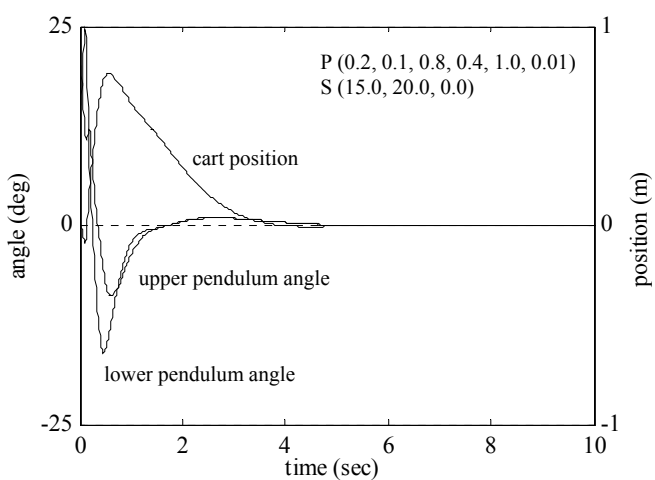


รูปที่ ค.8 มุมเริ่มต้นของลูกตุ้มอันล่างเป็น 15 องศา และลูกตุ้มอันบนเป็น 10 องศา

รูปที่ ค.9 เป็นผลการจำลองสถานการณ์เมื่อมุมเริ่มต้นของลูกตุ้มทั้งสองเท่ากัน คือ 15 องศา นั่นคือก้านลูกตุ้มทั้งสองอยู่ในแนวเดียวกัน ซึ่งจะสังเกตเห็นว่า ในช่วงแรกตัวรถจะเคลื่อนที่ไปทางด้านซ้ายเล็กน้อย ส่งผลให้ลูกตุ้มอันล่างหมุนตามเข็มนาฬิกา ทำให้มุมของลูกตุ้มอันล่างเพิ่มขึ้นเป็นประมาณ 17 องศา และลูกตุ้มอันบนจะหมุนทวนเข็มนาฬิกา ทำให้มุมของลูกตุ้มอันบนลดลงเหลือประมาณ 12.5 องศา จากนั้นตัวควบคุมนิวโร-ตาบิว-ฟิชซึ่งจะสร้างเสถียรภาพให้ระบบลูกตุ้ม ผกผันแบบข้อต่อเดียวในลักษณะเช่นเดียวกับสถานการณ์ในรูปที่ ค.8 ซึ่งใช้เวลาในการเข้าสู่เสถียรภาพประมาณ 6.0 วินาที



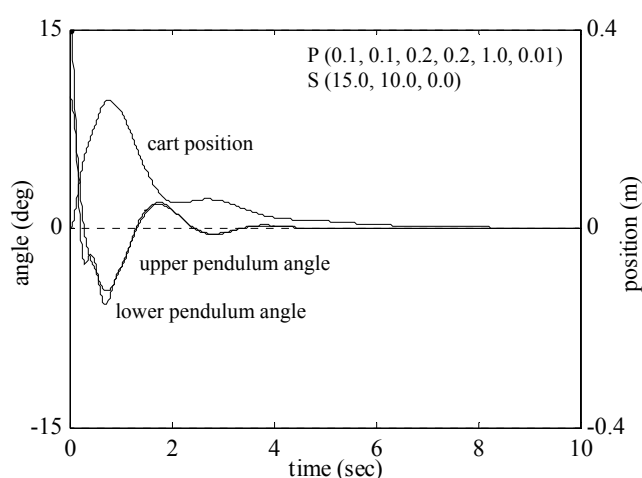
รูปที่ ค.9 มุมเริ่มต้นของลูกตุ้มทั้งสองเป็น 15 องศา



รูปที่ ค.10 มุมเริ่มต้นของลูกตุ้มอันล่างเป็น 15 องศา และลูกตุ้มอันบนเป็น 20 องศา

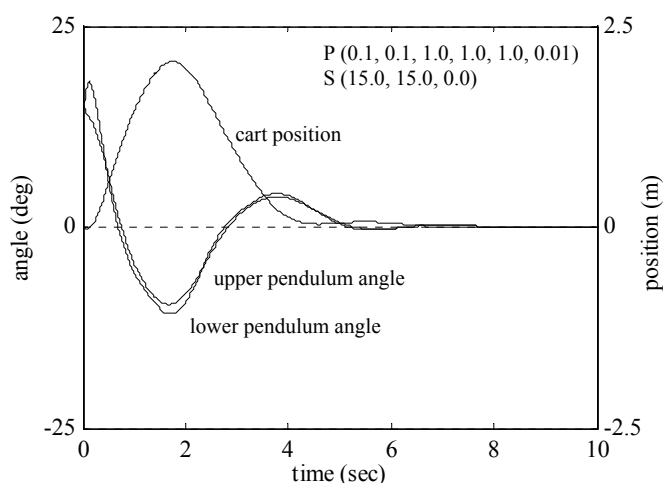
รูปที่ ค.10 เป็นการจำลองสถานการณ์เมื่อผลต่างเชิงมุมมีค่าเป็นบวก กล่าวคือ มุมของลูกตุ้มอันบนมีค่าเป็น 20 องศา ส่วนมุมของลูกตุ้มอันล่างเป็น 15 องศา จากผลการจำลองสถานการณ์พบว่า ตัวรถจะเคลื่อนที่ไปทางด้านซ้ายเล็กน้อย เพื่อให้ลูกตุ้มอันล่างหมุนตามเข็ม ส่งผลให้มุมของลูกตุ้มอันล่างเพิ่มขึ้นจนเกือบถึง 25 องศา และทำให้ลูกตุ้มอันบนหมุนในทิศทวนเข็มนาฬิกา จนมุมของลูกตุ้มอันบนลดลง ทั้งนี้จะสังเกตเห็นว่าสถานการณ์ในการควบคุมคล้ายคลึงกับสถานการณ์แรก ดังนั้นจึงสามารถสร้างเสถียรภาพให้ระบบได้ในแนวทางเดียวกัน ซึ่งใช้เวลาทั้งสิ้นประมาณ 5.36 วินาที

ต่อมาเป็นการจำลองสถานการณ์เมื่อปรับลดมวลและความยาวก้านของลูกตุ้มทั้งสองให้น้อยลง โดยที่มุมเริ่มต้นของลูกตุ้มอันล่างเป็น 15 องศา และมุมเริ่มต้นของลูกตุ้มอันบนเป็น 10 องศา ทั้งนี้มวลของตัวรถยังคงเท่าเดิม ซึ่งเมื่อพิจารณาผลการจำลองสถานการณ์ในรูปที่ ค.11 จะสังเกตเห็นว่า ในช่วงแรกมุมของลูกตุ้มอันล่างจะมีการสั่นเกิดขึ้น จากนั้นก้านลูกตุ้มทั้งสองจะวางตัวในแนวเดียวกัน จนกระทั่งตั้งตรงที่ตำแหน่งศูนย์องศา และตัวรถก็สามารถเคลื่อนที่กลับมายังกึ่งกลางของรางเส้น โดยใช้เวลาประมาณ 6.67 วินาที



รูปที่ ค.11 ปรับลดความยาวก้านลูกตุ้มทั้งสองเป็น 0.2 เมตร

ส่วนรูปที่ ค.12 เป็นผลการจำลองสถานการณ์เมื่อปรับเพิ่มความยาวของก้านลูกตุ้มทั้งสองให้ยาวขึ้นเป็น 1 เมตร โดยที่มุมเริ่มต้นของลูกตุ้มทั้งสองเท่ากัน คือ 15 องศา ทั้งนี้มวลของก้านลูกตุ้มทั้งสองกำหนดให้มีค่าเช่นเดียวกับสถานการณ์ในรูปที่ ค.11 จากผลการจำลองสถานการณ์ที่ได้ พบว่า ตัวควบคุมนิวโร-ตาบ-ฟัซซี่ สามารถสร้างเสถียรภาพให้ระบบได้ในเวลา 8.41 วินาที และลักษณะการสร้างเสถียรภาพให้ระบบยังคงเป็นไปในแนวทางเดียวกันกับสถานการณ์ในรูปที่ ค.9



รูปที่ 12 ปรับเพิ่มความยาวก้านลูกตุ้มทั้งสองเป็น 1.0 เมตร

ผลการเปรียบเทียบเวลาในการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบสองข้อต่อ โดยใช้ตัวควบคุมนิวโร-ตาบิว-ฟิชชี่ กับตัวควบคุมฟิชชี่แบบกฎอินพุตเดียว (Yi, Yubazaki, and Hirota, 2001) สามารถพิจารณาได้จากตารางที่ ค.3 ซึ่งทดสอบที่สถานการณ์ต่างๆ ตามรูปที่ ค.8, ค.9, ค.10, ค.11 และ ค.12 ตามลำดับ โดยจะสังเกตเห็นว่าตัวควบคุมนิวโร-ตาบิว-ฟิชชี่ ที่พัฒนาขึ้นสามารถสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบสองข้อต่อได้ภายในระยะเวลาที่รวดเร็วกว่าตัวควบคุมฟิชชี่แบบกฎอินพุตเดียวทุกสถานการณ์ แม้ว่าสถานการณ์ที่ควบคุมจะเปลี่ยนแปลงไป ตัวอย่างเช่น สถานการณ์แรกเป็นการสร้างเสถียรภาพให้ระบบเมื่อมุมของลูกตุ้มอันล่างเป็น 15 องศา และมุมของลูกตุ้มอันบนเป็น 10 องศา ซึ่งจะได้ว่าตัวควบคุมนิวโร-ตาบิว-ฟิชชี่ใช้เวลาในการสร้างเสถียรภาพให้ระบบเพียง 4.47 วินาที ในขณะที่ตัวควบคุมฟิชชี่แบบกฎอินพุตเดียวใช้เวลาทั้งสิ้น 7.19 วินาที

ตารางที่ ค.3 เวลาในการเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบสองข้อต่อ

สถานการณ์ที่	เวลาเข้าสู่เสถียรภาพ (วินาที)	
	ตัวควบคุมฟิชชี่แบบกฎอินพุตเดียว	ตัวควบคุมนิวโร-ตาบิว-ฟิชชี่
1	7.19	4.47
2	8.46	6.00
3	9.27	5.36
4	8.22	6.67
5	10.13	8.41

## ค.6 สรุป

จากผลการจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบสองข้อต่อ แสดงให้เห็นว่า ตัวควบคุมนิวโร-ตาบู-ฟิชชี สามารถสร้างเสถียรภาพให้ระบบได้ภายในระยะเวลาอันสั้น แม้จะปรับเปลี่ยนสถานการณ์ในการควบคุมให้แตกต่างกันไป ซึ่งระบบลูกค้ำผกผันแบบสองข้อต่อนั้นถือว่าเป็นตัวอย่างของระบบทดสอบ ที่มีความไม่เป็นเชิงเส้น ขนาดเสถียรภาพ และควบคุมได้ยาก ดังนั้นจึงอาจกล่าวได้ว่า ตัวควบคุมนิวโร-ตาบู-ฟิชชี ที่พัฒนาขึ้นมีความสามารถในการสร้างเสถียรภาพให้ระบบลูกค้ำผกผันแบบสองข้อต่อ โดยโครงสร้างของตัวควบคุมที่พัฒนาขึ้นนี้ประกอบด้วยข้อกำหนดกฎฟิชชีแบบกฎอินพุตเดียว และการเรียนรู้ของโครงข่ายประสาทเทียม เพื่อปรับค่าระดับความสำคัญพลวัตให้เหมาะสมกับสถานการณ์ที่ต้องการควบคุม ซึ่งค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมได้จากวิธีการค้นหาแบบตามูเชิงปรับตัว

ภาคผนวก ง

โครงข่ายประสาทเทียมและการเรียนรู้แบบแพร่กลับ

### ง.1 ประวัติความเป็นมาของโครงข่ายประสาทเทียม

ในปี 1943 ถือได้ว่าเป็นปีแรกแห่งการเริ่มต้นพัฒนาระบบโครงข่ายประสาทเทียมโดย วาร์เรน แม็คคัลลอค (Warren McCulloch) และ วอลเทอร์ พิตส์ (Walter Pitts) ได้เสนอแบบจำลองอย่างง่ายของเซลล์ประสาทเพื่ออธิบายคุณลักษณะของเซลล์ประสาทในรูปแบบระดับไบนารีสองสถานะ (two-state binary threshold) โดยผลคูณระหว่างค่าน้ำหนักกับสัญญาณอินพุตจะถูกนำไปเปรียบเทียบกับค่าระดับ (threshold) ซึ่งถ้ามีค่าน้อยกว่าค่าระดับ เอาต์พุตจะมีค่าเป็น -1 แต่ถ้ามีค่ามากกว่าค่าระดับจะได้เอาต์พุตเป็น +1 ทั้งนี้ค่าน้ำหนักของเซลล์ประสาทในแบบจำลองของแม็คคัลลอคและพิตส์ถูกกำหนดให้มีค่าเป็นฟังก์ชันลอจิกแบบง่าย (simple logic function) โดยค่าน้ำหนักเหล่านี้ยังไม่มีการเรียนรู้หรือปรับตัวใดๆ เกิดขึ้น ซึ่งในเวลาต่อมาแนวคิดของแม็คคัลลอคและพิตส์ได้เป็นประโยชน์อย่างมากในการเป็นพื้นฐานสำหรับพัฒนาโครงข่ายประสาทเทียม

ในปี 1949 โดแนลด์ เฮบบ์ (Donald Hebb) ได้เสนอผลงานวิจัยว่า การเรียนรู้ของสมองสามารถอธิบายได้ด้วยรูปแบบของการประกอบเซลล์ประสาทเข้าด้วยกันเป็นโครงข่าย และได้เสนอกฎการเรียนรู้แบบแรกสำหรับโครงข่ายประสาทเทียม เรียกว่า กฎการเรียนรู้ของเฮบบ์ (Hebb's rule) ที่ทำให้โครงข่ายของเซลล์ประสาทเทียมที่แม็คคัลลอคและพิตส์ได้เสนอไว้สามารถเรียนรู้ปัญหาต่างๆ ได้สำเร็จ โดยกฎการเรียนรู้ของเฮบบ์ถือเป็นการเรียนรู้แบบไม่มีผู้สอน (unsupervised learning)

ในช่วงปี 1954 คอมพิวเตอร์ที่ทำงานเลียนแบบสมองหรือ neurocomputer เครื่องแรกของโลกถูกสร้างและทดสอบโดยมินสกี (Minsky) เมื่อคอมพิวเตอร์ดังกล่าวได้รับการป้อนตัวอย่างสำหรับการเรียนรู้เข้าไป ก็จะสามารถปรับการเชื่อมโยงหรือความแข็งแรงของการเชื่อมโยงระหว่างเซลล์ประสาทเทียมได้เองโดยอัตโนมัติ

ในปี 1958 แฟรงค์ โรเซนแบลทท์ (Frank Rosenblatt) ได้พัฒนาสถาปัตยกรรมโครงข่ายประสาทเทียมขึ้น เรียกว่า เพอร์เซพตรอน (perceptron) ซึ่งมีการเรียนรู้แบบมีผู้สอน (supervised learning) รวมทั้งเสนอวิธีการเรียนรู้แบบใหม่สำหรับสถาปัตยกรรมโครงข่ายประสาทเทียมดังกล่าวด้วย โดยการปรับความแข็งแรงของการเชื่อมโยง ซึ่งจะพิจารณาได้จากการเปรียบเทียบความรู้ของโครงข่ายประสาทเทียมกับความรู้ของผู้สอน (teacher)

ในช่วงปี 1960 เบอ์นาร์ด์ วิโดรว (Bernard Widrow) และมาร์เซียน ฮอฟฟ์ (Marcian Hoff) ได้นำเสนอกฎการเรียนรู้แบบใหม่ที่มีประสิทธิภาพสูงเรียกว่า กฎการเรียนรู้ของวิโดรว-ฮอฟฟ์ (Widrow-Hoff learning rule) หรือกฎการเรียนรู้เดลต้า (delta rule) ซึ่งเป็นการเรียนรู้แบบมีผู้สอน โดยอาศัยหลักการของค่าเฉลี่ยกำลังสองน้อยที่สุด (least mean square; LMS) ซึ่งการเรียนรู้วิธีนี้จะแตกต่างจากการเรียนรู้แบบเพอร์เซพตรอน กล่าวคือการเรียนรู้แบบเพอร์เซพตรอน จะทำการปรับค่าน้ำหนักการเชื่อมโยงเมื่อผลตอบสนองไม่ถูกต้อง ในขณะที่กฎการเรียนรู้แบบ



เคลต้าจะทำการปรับค่าน้ำหนักเพื่อลดผลต่างระหว่างเอาต์พุตที่ได้จากโครงข่ายประสาทเทียมกับเอาต์พุตที่ต้องการนอกจากนี้ วิโครวและฮอฟฟีได้พัฒนาโครงข่ายประสาทแบบง่ายซึ่งคล้ายคลึงกับเพอร์เซพตรอนที่เรียกว่า อดาไลน์ (ADALINE: adaptive linear combiner) ซึ่งในเวลาต่อมาได้รับการขยายแนวคิดไปเป็นมาดาไลน์ (MADALINE: many ADALINES)

ในปี 1972 ทูโว โคโฮเนน (Teuvo Kohonen) ได้นำศึกษาเกี่ยวกับหน่วยความจำสัมพันธ์ (associative memory) ของโครงข่ายประสาทเทียม และได้พัฒนาโครงข่ายการเรียนรู้แบบไม่มีผู้สอนที่เรียกว่า แผนที่คุณลักษณะการจัดการตัวเอง (self-organizing feature maps) โดยใช้การเรียนรู้แบบแข่งขันกัน (competitive learning)

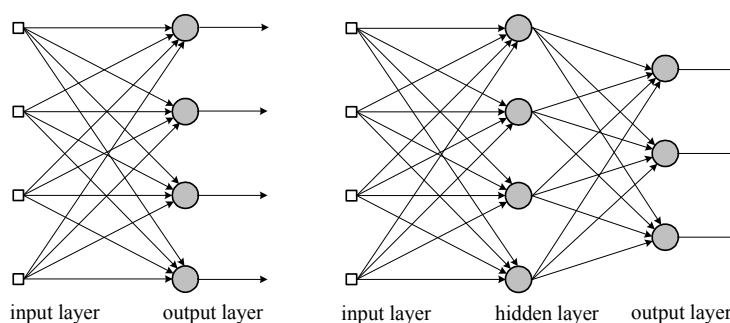
ในปี 1982 จอห์น ฮอฟฟิลด์ (John Hopfield) ได้นำเสนอโครงข่ายแบบย้อนกลับ (recurrent network) ซึ่งใช้แนวคิดของฟังก์ชันพลังงานเลียปูนอฟ (Lyapunov energy function) มาเป็นแนวทางเพื่อให้เข้าใจการคำนวณของโครงข่ายแบบย้อนกลับ ซึ่งในเวลาต่อมาเรียกโครงข่ายแบบนี้ว่าโครงข่ายฮอฟฟิลด์ (Hopfield network) ฮอฟฟิลด์ได้พัฒนาโครงข่ายประสาทเทียมซึ่งกำหนดให้ค่าน้ำหนักของการเชื่อมโยงมีค่าคงที่แต่ใช้ฟังก์ชันกระตุ้นแบบปรับตัวได้ (adaptive activation function)

ในปี 1986 เจมส์ แม็คเคลแลนด (James McClelland) และ เดวิด รุมลฮาร์ท (David Rumelhart) ได้นำเสนออัลกอริทึมการเรียนรู้แบบใหม่ซึ่งเรียกว่า การเรียนรู้แบบแพร่กลับ สำหรับปรับค่าน้ำหนักการเชื่อมโยงในโครงข่ายประสาทเทียมหลายชั้นแบบป้อนไปหน้า โดยค่าน้ำหนักเหล่านี้จะถูกปรับจากชั้นเอาต์พุต (output layer) ไปยังชั้นอินพุต (input layer) ในลักษณะชั้นต่อชั้น (layer-by-layer) เพื่อลดค่าความคลาดเคลื่อนระหว่างเอาต์พุตที่ได้จากโครงข่ายประสาทเทียมกับเอาต์พุตที่ต้องการ

## ง.2 สถาปัตยกรรมของโครงข่ายประสาทเทียม

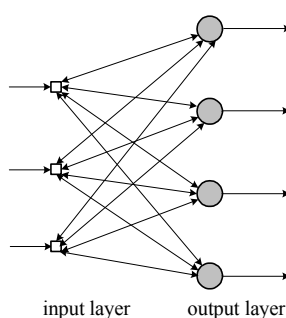
โครงข่ายประสาทเทียมประกอบไปด้วยเซลล์ประสาทเทียม (neuron) หลากๆ เซลล์เชื่อมโยงกันแบบขนานหลายๆ ชั้น ซึ่งสามารถเชื่อมโยงแบบใดก็ได้ อย่างไรก็ตามเทคนิคการเรียนรู้ของโครงข่ายประสาทเทียมมักจะถูกออกแบบมาให้ใช้งานได้กับสถาปัตยกรรมโครงข่ายประสาทเทียมที่มีลักษณะเฉพาะเท่านั้น ซึ่งโดยทั่วไปมีการจัดเซลล์ประสาทเทียมเป็นชั้นๆ ชั้นที่รับข้อมูลเข้าเรียกว่า ชั้นอินพุต ชั้นที่ผลิตผลตอบสนองของโครงข่ายเรียกว่า ชั้นเอาต์พุต ส่วนชั้นอื่นๆ ที่มีส่วนช่วยในการประมวลผลอยู่ภายในเรียกว่า ชั้นซ่อนเร้น (hidden layer) ซึ่งการกำหนดจำนวนชั้นซ่อนเร้น รวมถึงจำนวนเซลล์ประสาทเทียมในแต่ละชั้นนั้นขึ้นอยู่กับลักษณะงานแต่ละประเภท สถาปัตยกรรมพื้นฐานของโครงข่ายประสาทเทียมแสดงดังต่อไปนี้

โครงข่ายประสาทเทียมแบบป้อนไปหน้า อาจแบ่งได้เป็นสองแบบคือ มีเซลล์ประสาทชั้นเดียว (single layer) และมีเซลล์ประสาทหลายชั้น (multilayer) ซึ่งแสดงด้วยรูปที่ ง.1 โดยในแต่ละชั้นอาจประกอบด้วยเซลล์ประสาทเทียมจำนวนเท่าใดก็ได้ ปกติแล้วการเชื่อมโยงจะถูกกำหนดขึ้นระหว่างชั้นที่ติดกัน โดยจะมีการเชื่อมโยงระหว่างเซลล์ประสาทเทียมทุกตัวจากชั้นหนึ่งๆ ไปยังเซลล์ประสาทเทียมทุกตัวในชั้นต่อไป



รูปที่ ง.1 โครงข่ายประสาทเทียมแบบป้อนไปหน้า

โครงข่ายประสาทเทียมแบบมีการป้อนกลับ (recurrent or feedback network) การเชื่อมโยงที่กำหนดขึ้นระหว่างเซลล์ประสาทเทียมในชั้นหนึ่งๆ อาจย้อนกลับไปยังชั้นอื่นๆ ก่อนหน้านั้น หรือแม้แต่ภายในชั้นเดียวกันเองก็ได้ ซึ่งแสดงได้ดังรูปที่ ง.2

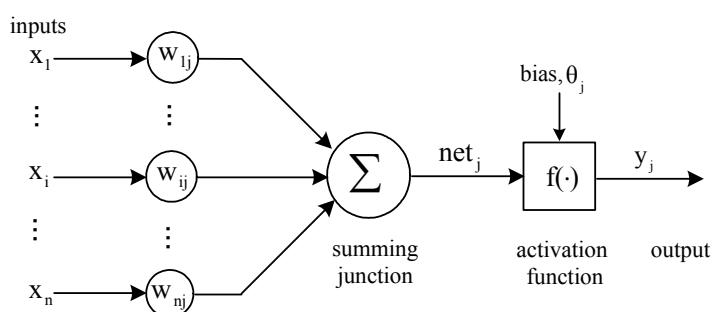


รูปที่ ง.2 โครงข่ายประสาทเทียมแบบป้อนกลับ

### ง.3 แบบจำลองของเซลล์ประสาทเทียม

เพื่อให้เกิดความเข้าใจกระบวนการเรียนรู้ของโครงข่ายประสาทเทียมจำเป็นต้องเข้าใจแบบจำลองพื้นฐานของเซลล์ประสาทเทียม ซึ่งแสดงด้วยรูปที่ ง.3 แบบจำลองดังกล่าวประกอบด้วย

องค์ประกอบที่สำคัญสามส่วน โดยส่วนแรกเป็นส่วนของไซแนปส์หรือส่วนเชื่อมโยงสัญญาณ ซึ่งแสดงคุณลักษณะของแต่ละเซลล์ประสาทเทียมด้วยค่าน้ำหนัก โดยสัญญาณอินพุต  $x_i$  ที่เชื่อมโยงกับเซลล์ประสาทเทียม  $j$  จะถูกคูณด้วยค่าน้ำหนัก  $w_{ij}$  ถ้าค่าน้ำหนักนี้มีค่าเป็นบวก หมายความว่าสัญญาณไซแนปส์เป็นสัญญาณกระตุ้น ในทางตรงกันข้ามถ้าค่าน้ำหนักดังกล่าวมีค่าเป็นลบ แสดงว่าสัญญาณไซแนปส์เป็นสัญญาณยับยั้งนั่นเอง ส่วนที่สองเป็นผลรวมของผลคูณระหว่างสัญญาณอินพุตกับค่าน้ำหนัก และส่วนสุดท้ายคือฟังก์ชันกระตุ้น (activation function) เพื่อกำหนดขนาดของสัญญาณเอาต์พุต ซึ่งขึ้นอยู่กับลักษณะของงานแต่ละประเภท ตัวอย่างเช่น ถ้าต้องการให้สัญญาณเอาต์พุตมีขอบเขตอยู่ในช่วง  $[0, 1]$  ควรเลือกใช้ฟังก์ชันกระตุ้นแบบซิกมอยด์ เป็นต้น



รูปที่ ๓.3 แบบจำลองของเซลล์ประสาทเทียม

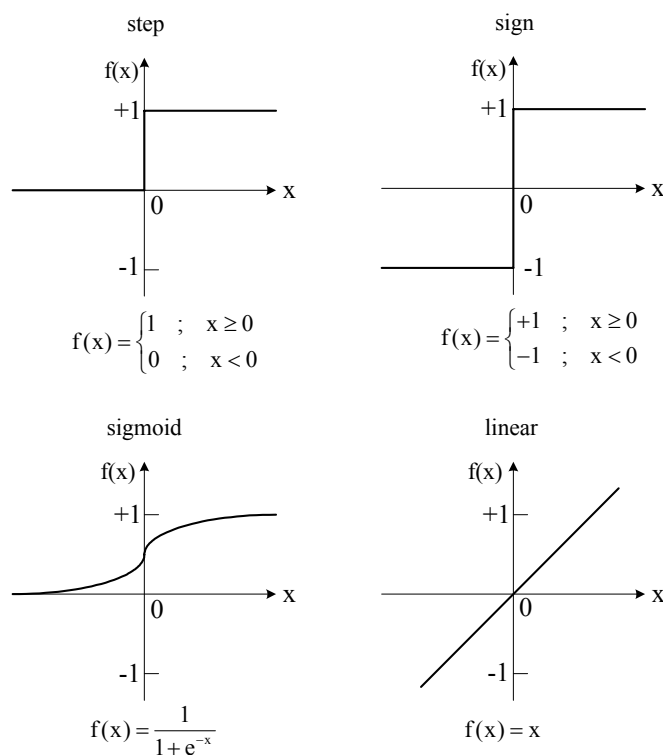
แบบจำลองทางคณิตศาสตร์ของเซลล์ประสาทเทียมดังแสดงด้วยรูปที่ ๓.3 สามารถอธิบายได้ดังสมการที่ (๓-1) และ (๓-2) ดังนี้คือ

$$\text{net}_j = \sum_{i=1}^n w_{ij} x_i \quad (3-1)$$

$$y_j = f(\text{net}_j + \theta_j) \quad (3-2)$$

- เมื่อ  $x_i$  คือ สัญญาณอินพุต  
 $y_j$  คือ สัญญาณเอาต์พุต  
 $w_{ij}$  คือ ค่าน้ำหนักจากอินพุต  $i$  ไปยังเซลล์ประสาทเทียม  $j$   
 $\theta_j$  คือ ค่าไบแอสที่เซลล์ประสาทเทียม  $j$   
 $f(\cdot)$  คือ ฟังก์ชันกระตุ้น  
 $\text{net}_j$  คือ ผลรวมของผลคูณระหว่างค่าน้ำหนักกับสัญญาณอินพุต

ตัวอย่างของฟังก์ชันกระตุ้นที่พบว่ามีมีการนำไปประยุกต์ใช้ในทางปฏิบัติ ได้แก่ ฟังก์ชันแบบขั้นบันได (step function) ฟังก์ชันเครื่องหมาย (sign function) ฟังก์ชันเชิงเส้น (linear function) และ ฟังก์ชันซิกมอยด์ (sigmoid function) ดังแสดงในรูปที่ ๓.4 โดยฟังก์ชันกระตุ้นแบบขั้นบันได และแบบเครื่องหมายนั้น เรียกว่า hard limit function ซึ่งนำไปใช้สำหรับงานแบ่งแยกประเภท (classification) และการรู้จำแบบอย่าง (pattern recognition) ในขณะที่ฟังก์ชันซิกมอยด์ใช้สำหรับโครงข่ายประสาทเทียมที่มีการเรียนรู้แบบแพร่กลับ โดยมีขอบเขตของเอาต์พุตอยู่ในช่วง  $[0, 1]$  และฟังก์ชันเชิงเส้นใช้สำหรับการประมาณค่าเชิงเส้น (linear approximation) (Negnevitsky, 2002)



รูปที่ ๓.4 รูปแบบฟังก์ชันกระตุ้น

#### ๓.4 การเรียนรู้ของโครงข่ายประสาทเทียม

คุณลักษณะที่สำคัญของโครงข่ายประสาทเทียม คือความสามารถในการเรียนรู้จากสภาวะแวดล้อม เพื่อปรับปรุงสมรรถนะต่างๆ โดยผ่านกระบวนการเรียนรู้เหล่านั้น ซึ่งโครงข่ายประสาทเทียมจะเรียนรู้โดยกระบวนการวนซ้ำ (iterative process) เพื่อทำการปรับค่าพารามิเตอร์ของโครงข่าย ได้แก่ ค่าน้ำหนัก (weight) และไบแอส (bias) จุดประสงค์ของการเรียนรู้ก็เพื่อฝึกสอน

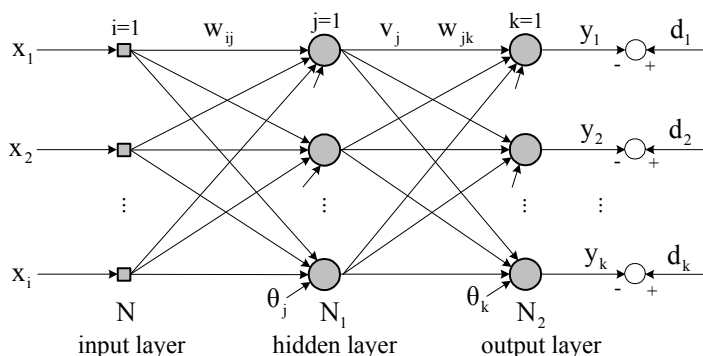
ให้โครงข่ายประสาทเทียมสามารถทำงานได้ตามที่ต้องการ โดยทั่วไปแล้วการเรียนรู้มีอยู่มากมายหลายแบบ แต่สามารถแบ่งออกได้เป็นกลุ่มใหญ่ๆ สองกลุ่ม ได้แก่

**การเรียนรู้แบบมีผู้สอน (supervised learning)** เมื่อมีสัญญาณอินพุตเข้ามาที่ชั้นอินพุตโครงข่ายประสาทเทียมจะสร้างผลตอบออกมาเป็นเอาต์พุต ผลตอบดังกล่าวจะถูกนำมาเปรียบเทียบกับเป้าหมาย (target) ซึ่งผู้สอนกำหนดขึ้น หากผลตอบทั้งสองมีความแตกต่างกัน ความคลาดเคลื่อนดังกล่าวจะถูกนำไปคำนวณเพื่อทำการปรับแต่งน้ำหนักต่างๆ ในโครงข่ายประสาทเทียมเพื่อลดความคลาดเคลื่อนให้เหลือน้อยที่สุด การปรับแต่งน้ำหนักโดยพิจารณาจากความคลาดเคลื่อนนี้จะขึ้นอยู่กับกฎการเรียนรู้ อย่างไรก็ตาม วิธีการส่วนใหญ่ของการเรียนรู้แบบมีผู้สอนจะอาศัยวิธีการทางคณิตศาสตร์ในเรื่องของเทคนิคการหาค่าเหมาะสม (optimization technique) ตัวอย่างเช่น การเรียนรู้แบบแพร่กลับ เป็นต้น

**การเรียนรู้แบบไม่มีผู้สอน (unsupervised learning)** การเรียนรู้แบบนี้ไม่จำเป็นต้องมีค่าเป้าหมาย ในระหว่างการเรียนรู้โครงข่ายประสาทเทียมจะได้รับข้อมูลในรูปแบบต่างๆ และจะทำการจัดกลุ่มรูปแบบเหล่านั้นเองตามต้องการ โดยอาศัยแนวทางในการจัดกลุ่ม เช่น รูปทรง สี เป็นต้น ดังนั้น หากไม่มีการให้แนวทางที่ชัดเจนว่าการจัดกลุ่มควรเป็นไปตามคุณลักษณะใด การจัดกลุ่มอาจไม่ประสบความสำเร็จ ในโครงข่ายประสาทเทียมบางชนิดอาจจะใช้วิธีการแข่งขันกันของกลุ่มต่างๆ เรียกว่า การเรียนรู้แบบแข่งขันกัน (competitive learning) อีกแนวทางหนึ่ง คือ การจัดกลุ่มเกิดจากการที่โครงข่ายประสาทเทียมประเมินข้อมูลต่างๆ ที่ถูกป้อนเข้าไปในระหว่างการเรียนรู้จนสร้างเป็นวิธีการจัดกลุ่มขึ้นมาได้เอง เรียกว่า การเรียนรู้แบบจัดตัวเอง (self-organizing)

### ง.5 การเรียนรู้แบบแพร่กลับ

เนื่องจากโครงข่ายประสาทเทียมแบบชั้นเดียว มีข้อจำกัดในการแก้ปัญหาที่ซับซ้อน ส่งผลให้โครงข่ายประสาทเทียมแบบหลายชั้นได้รับความนิยม และมีการนำไปใช้ในการแก้ปัญหาต่างๆ มากขึ้น วิธีการฝึกสอนโครงข่ายประสาทเทียมแบบหลายชั้นซึ่งเป็นที่รู้จักกันมาก คือ การเรียนรู้แบบแพร่กลับค่าความคลาดเคลื่อน ซึ่งประกอบด้วยสองกระบวนการสำคัญ ได้แก่ การป้อนสัญญาณอินพุตแบบป้อนไปหน้า (feedforward) ในลักษณะชั้นต่อชั้น และแพร่กลับค่าความคลาดเคลื่อนแบบย้อนกลับ (feedback) ในลักษณะชั้นต่อชั้นเช่นเดียวกัน เพื่อปรับค่าน้ำหนักและค่าไบแอสระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต รวมทั้งทำการปรับค่าน้ำหนักและค่าไบแอสระหว่างชั้นอินพุตกับชั้นซ่อนเร้นด้วย ทั้งนี้เพื่อต้องการให้ผลตอบสนองที่ได้จากเอาต์พุตของโครงข่ายประสาทเทียมใกล้เคียงกับเอาต์พุตที่ต้องการ การเรียนรู้ของโครงข่ายประสาทเทียมที่ประกอบด้วยชั้นซ่อนเร้นหนึ่งชั้นแสดงได้ด้วยรูปที่ ง.5



รูปที่ ง.5 การเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียมแบบหลายชั้น

โดยที่ค่าพารามิเตอร์ต่างๆ ที่ปรากฏในรูปที่ ง.5 มีความหมายดังนี้คือ

$x_i$  คือ สัญญาณอินพุตที่เข้าสู่ชั้นอินพุตตรงเซลล์ประสาทเทียม  $i$

$i$  คือ เซลล์ประสาทเทียมในชั้นอินพุต

$j$  คือ เซลล์ประสาทเทียมในชั้นซ่อนเร้น

$k$  คือ เซลล์ประสาทเทียมในชั้นเอาต์พุต

$N$  คือ จำนวนเซลล์ประสาทเทียมทั้งหมดในชั้นอินพุต

$N_1$  คือ จำนวนเซลล์ประสาทเทียมทั้งหมดในชั้นซ่อนเร้น

$N_2$  คือ จำนวนเซลล์ประสาทเทียมทั้งหมดในชั้นเอาต์พุต

$v_j$  คือ เอาต์พุตจากชั้นซ่อนเร้น

$y_k$  คือ เอาต์พุตจากชั้นเอาต์พุต

$d_k$  คือ เอาต์พุตที่ต้องการ

$w_{ij}$  คือ ค่าน้ำหนักที่เชื่อมโยงระหว่างชั้นอินพุตกับชั้นซ่อนเร้นจากเซลล์ประสาทเทียม

$i$  ไปยังเซลล์ประสาทเทียม  $j$

$w_{jk}$  คือ ค่าน้ำหนักที่เชื่อมโยงระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุตจากเซลล์ประสาทเทียม

$j$  ยังเซลล์ประสาทเทียม  $k$

$\theta_j$  คือ ค่าไบแอสสำหรับชั้นซ่อนเร้นที่เซลล์ประสาทเทียม  $j$

$\theta_k$  คือ ค่าไบแอสสำหรับชั้นเอาต์พุตที่เซลล์ประสาทเทียม  $k$

เมื่อพิจารณารูปที่ ง.5 จะสังเกตเห็นว่า สัญญาณอินพุต  $x_i$  ไหลจากชั้นอินพุตไปยังเซลล์ประสาทเทียมทุกตัวของชั้นซ่อนเร้น โดยมีการเชื่อมโยงระหว่างกันด้วยค่าน้ำหนัก  $w_{ij}$  จากนั้นชั้นซ่อนเร้นจะคำนวณเอาต์พุต  $v_j$  เพื่อเป็นสัญญาณอินพุตให้กับชั้นเอาต์พุตต่อไป โดยเชื่อมโยงด้วยค่าน้ำหนัก  $w_{jk}$  จนกระทั่งได้สัญญาณเอาต์พุตของโครงข่ายประสาทเทียม  $y_k$  ในระหว่างการเรียนรู้สัญญาณ

เอาต์พุตเหล่านี้จะถูกนำไปเปรียบเทียบกับสัญญาณเอาต์พุตที่ต้องการ  $d_k$  เพื่อคำนวณค่าความคลาดเคลื่อนสำหรับกระบวนการแพร่กลับ จากนั้นค่าน้ำหนักของทุกชั้นจะถูกปรับค่าให้เหมาะสมซึ่งอธิบายได้ด้วยสมการที่ (ง-3) และ (ง-4) ตามลำดับ

$$w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n) \quad (\text{ง-3})$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \quad (\text{ง-4})$$

โดยที่  $w_{jk}(n+1)$  คือ ค่าน้ำหนักจากชั้นซ่อนเร้นไปยังชั้นเอาต์พุตหลังปรับค่า  
 $w_{jk}(n)$  คือ ค่าน้ำหนักจากชั้นซ่อนเร้นไปยังชั้นเอาต์พุตก่อนปรับค่า  
 $\Delta w_{jk}(n)$  คือ ค่าน้ำหนักปรับแก้จากชั้นซ่อนเร้นไปยังชั้นเอาต์พุต  
 $w_{ij}(n+1)$  คือ ค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้นหลังปรับค่า  
 $w_{ij}(n)$  คือ ค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้นก่อนปรับค่า  
 $\Delta w_{ij}(n)$  คือ ค่าน้ำหนักปรับแก้จากชั้นอินพุตไปยังชั้นซ่อนเร้น  
 $n$  คือ จำนวนรอบ

กระบวนการเรียนรู้แบบแพร่กลับเพื่อปรับค่าน้ำหนักและค่าไบแอสของโครงข่ายประสาทเทียมจะอาศัยหลักการของเทคนิควิธีการลดระดับความชัน (gradient descent method) โดยที่  $\Delta w_{jk}(n)$  สามารถคำนวณได้จากสมการที่ (ง-5)

$$\Delta w_{jk}(n) = -\eta_1 \frac{\partial E(n)}{\partial w_{jk}(n)} \quad (\text{ง-5})$$

เมื่อ  $\eta_1$  คือ อัตราการเรียนรู้ และ  $E(n)$  คือ ค่าฟังก์ชันประเมิน (cost function) ที่เกิดจากผลรวมกำลังสองของค่าความคลาดเคลื่อน (sum square error) ซึ่งสามารถอธิบายได้ด้วยสมการที่ (ง-6) โดยตัวเลข  $\frac{1}{2}$  ที่ปรากฏในสมการกำหนดขึ้นเพื่อให้สามารถหาอนุพันธ์ได้ง่ายขึ้น (Freeman and Skapura, 1991)

$$E(n) = \frac{1}{2} \sum_{k=1}^{N_2} (e_k(n))^2 \quad (\text{ง-6})$$

โดยที่  $e_k(n) = d_k(n) - y_k(n)$  (ง-7)

จากสมการที่ (ง-5) ใช้หลักการของกฎลูกโซ่ (chain rule) จะได้สมการที่ (ง-8)

$$\frac{\partial E(n)}{\partial w_{jk}(n)} = \frac{\partial E(n)}{\partial e_k(n)} \cdot \frac{\partial e_k(n)}{\partial y_k(n)} \cdot \frac{\partial y_k(n)}{\partial \text{net}_k(n)} \cdot \frac{\partial \text{net}_k(n)}{\partial w_{jk}(n)} \quad (\text{ง-8})$$

โดยที่  $\text{net}_k(n)$  คือ ผลรวมของผลคูณระหว่างค่าน้ำหนักระหว่างชั้นซ่อนเร้นและชั้นเอาต์พุตกับสัญญาณอินพุตจากชั้นซ่อนเร้น ซึ่งอธิบายได้ด้วยสมการที่ (ง-9)

$$\text{net}_k(n) = \sum_{j=1}^{N_i} w_{jk} \cdot v_j \quad (\text{ง-9})$$

ดังนั้น  $y_k(n) = f(\text{net}_k(n))$  (ง-10)

เมื่อ  $f(\cdot)$  คือ ฟังก์ชันกระตุ้น จากสมการที่ (ง-8) ทำการคำนวณพจน์ทางด้านขวาของสมการได้ดังนี้

$$\frac{\partial E(n)}{\partial e_k(n)} = e_k(n) \quad (\text{ง-11})$$

$$\frac{\partial e_k(n)}{\partial y_k(n)} = -1 \quad (\text{ง-12})$$

$$\frac{\partial y_k(n)}{\partial \text{net}_k(n)} = f'(\text{net}_k(n)) \quad (\text{ง-13})$$

$$\frac{\partial \text{net}_k(n)}{\partial w_{jk}(n)} = v_j(n) \quad (\text{ง-14})$$

นำสมการที่ (ง-11), (ง-12), (ง-13), และ (ง-14) แทนในสมการที่ (ง-8) จะได้

$$\frac{\partial E(n)}{\partial w_{jk}(n)} = -e_k(n) \cdot f'(\text{net}_k(n)) \cdot v_j(n) \quad (\text{ง-15})$$

ดังนั้นค่าน้ำหนักปรับแก้ จากสมการที่ (ง-5) คือ



$$\Delta w_{jk}(n) = \eta_1 \cdot e_k(n) \cdot f'(\text{net}_k(n)) \cdot v_j(n) \quad (จ-16)$$

$$\Delta w_{jk}(n) = \eta_1 \cdot \delta_k(n) \cdot v_j(n) \quad (จ-17)$$

นั่นคือ  $\delta_k(n) = e_k(n) \cdot f'(\text{net}_k(n)) \quad (จ-18)$

หรือ  $\delta_k(n) = (d_k(n) - y_k(n)) \cdot f'(\text{net}_k(n)) \quad (จ-19)$

ในทำนองเดียวกันค่า  $\Delta w_{ij}(n)$  สามารถคำนวณได้ดังนี้

$$\Delta w_{ij}(n) = -\eta_2 \frac{\partial E(n)}{\partial w_{ij}(n)} \quad (จ-20)$$

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = \frac{\partial E(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial \text{net}_j(n)} \cdot \frac{\partial \text{net}_j(n)}{\partial w_{ij}(n)} \quad (จ-21)$$

โดยที่  $\text{net}_j(n)$  คือ ผลรวมของผลคูณระหว่างค่านำหนักระหว่างชั้นอินพุตและชั้นซ่อนเร้นกับสัญญาณอินพุตจากชั้นอินพุต ซึ่งอธิบายได้ด้วยสมการที่ (จ-22)

$$\text{net}_j(n) = \sum_{i=1}^N w_{ij} \cdot x_i \quad (จ-22)$$

ซึ่งจะได้  $v_j(n) = f(\text{net}_j(n)) \quad (จ-23)$

จากสมการที่ (จ-21) สามารถหาพจน์ทางด้านขวาของสมการได้ดังนี้

$$\frac{\partial E(n)}{\partial v_j(n)} = \sum_k e_k(n) \cdot \frac{\partial e_k(n)}{\partial v_j(n)} \quad (จ-24)$$

โดยที่  $\frac{\partial e_k(n)}{\partial v_j(n)} = \frac{\partial e_k(n)}{\partial \text{net}_k(n)} \cdot \frac{\partial \text{net}_k(n)}{\partial v_j(n)} \quad (จ-25)$

พิจารณาสมการที่ (ง-7) และ (ง-10) จะได้ดังสมการที่ (ง-26)

$$e_k(n) = d_k(n) - f(\text{net}_k(n)) \quad (\text{ง-26})$$

ดังนั้นจะได้ว่า

$$\frac{\partial e_k(n)}{\partial \text{net}_k(n)} = -f'(\text{net}_k(n)) \quad (\text{ง-27})$$

และจากสมการที่ (ง-9) จะได้

$$\frac{\partial \text{net}_k(n)}{\partial v_j(n)} = w_{jk}(n) \quad (\text{ง-28})$$

แทนสมการที่ (ง-27) และ (ง-28) ลงในสมการที่ (ง-25) จะได้

$$\frac{\partial e_k(n)}{\partial v_j(n)} = -f'(\text{net}_k(n)) \cdot w_{jk}(n) \quad (\text{ง-29})$$

แทนสมการที่ (ง-29) ลงในสมการที่ (ง-24) จะได้

$$\frac{\partial E(n)}{\partial v_j(n)} = -\sum_k e_k(n) \cdot f'(\text{net}_k(n)) \cdot w_{jk}(n) \quad (\text{ง-30})$$

จากสมการที่ (ง-22) และ (ง-23) จะได้

$$\frac{\partial v_j(n)}{\partial \text{net}_j(n)} = f'(\text{net}_j(n)) \quad (\text{ง-31})$$

$$\frac{\partial \text{net}_j(n)}{\partial w_{ij}(n)} = x_i(n) \quad (\text{ง-32})$$

แทนสมการที่ (ง-30), (ง-31) และ (ง-32) ลงในสมการที่ (ง-21) จะได้ว่า

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = -\sum_k e_k(n) \cdot f'(\text{net}_k(n)) \cdot w_{jk}(n) \cdot f'(\text{net}_j(n)) \cdot x_i(n) \quad (\text{ง-33})$$

ดังนั้นจากสมการที่ (ง-20) จะได้ว่า

$$\Delta w_{ij}(n) = \eta_2 \sum_k e_k(n) \cdot f'(\text{net}_k(n)) \cdot w_{jk}(n) \cdot f'(\text{net}_j(n)) \cdot x_i(n) \quad (\text{ง-34})$$

เมื่อพิจารณาสมการที่ (ง-18) จะได้ว่า

$$\Delta w_{ij}(n) = \eta_2 \sum_k \delta_k(n) \cdot w_{jk}(n) \cdot f'(\text{net}_j(n)) \cdot x_i(n) \quad (\text{ง-35})$$

$$\Delta w_{ij}(n) = \eta_2 \delta_j(n) \cdot x_i(n) \quad (\text{ง-36})$$

โดยที่ 
$$\delta_j(n) = \sum_k \delta_k(n) \cdot w_{jk}(n) \cdot f'(\text{net}_j(n)) \quad (\text{ง-37})$$

สำหรับการปรับค่าไบแอสของชั้นซ่อนเร้นและชั้นเอาต์พุต สามารถพิจารณาได้จากสมการที่ (ง-38) และ (ง-39) ตามลำดับ

$$\theta_j(n+1) = \theta_j(n) + \Delta\theta_j(n) \quad (\text{ง-38})$$

$$\theta_k(n+1) = \theta_k(n) + \Delta\theta_k(n) \quad (\text{ง-39})$$

เมื่อ 
$$\Delta\theta_j(n) = \eta_3 \delta_j(n) \quad \text{และ} \quad \Delta\theta_k(n) = \eta_4 \delta_k(n)$$

โดยที่  $\eta_1$ ,  $\eta_2$ ,  $\eta_3$  และ  $\eta_4$  คือ อัตราการเรียนรู้ เมื่อพิจารณาสมการที่ (ง-18) และ (ง-37) ซึ่งเป็นค่าพจน์ความคลาดเคลื่อน  $\delta_k$  และ  $\delta_j$  ตามลำดับ พบว่าการคำนวณค่าเหล่านี้จำเป็นต้องทราบค่าอนุพันธ์ของฟังก์ชันกระตุ้น ซึ่งฟังก์ชันกระตุ้นควรมีคุณลักษณะที่สำคัญ คือ เป็นฟังก์ชันต่อเนื่อง สามารถหาอนุพันธ์ได้ และ ไม่มีการลดระดับ (monotonically nondecreasing) นอกจากนี้เพื่อให้การคำนวณรวดเร็วยิ่งขึ้น ฟังก์ชันกระตุ้นที่เลือกใช้ควรจะง่ายต่อการคำนวณ (Fausett, 1994) ตัวอย่างของฟังก์ชันกระตุ้นที่นิยมใช้มากที่สุดสำหรับการเรียนรู้แบบแพร่กลับ คือ ฟังก์ชันแบบซิกมอยด์ ซึ่ง

เป็นฟังก์ชันต่อเนื่องที่สามารถหาอนุพันธ์ได้ และมีลักษณะไม่เป็นเชิงเส้น โดยเอาต์พุตมีขอบเขตอยู่ในช่วง  $[0, 1]$  อนุพันธ์ของฟังก์ชันกระตุ้นดังกล่าว แสดงได้ด้วยสมการที่ (ง-40)

$$f'(x) = f(x)(1 - f(x)) \quad (\text{ง-40})$$

ดังนั้นการคำนวณค่า  $\delta_k$  และ  $\delta_j$  จากสมการที่ (ง-18) และ (ง-37) ตามลำดับนั้น สามารถแสดงได้ด้วยสมการที่ (ง-41) และ (ง-42) ดังนี้คือ

$$\delta_k(n) = (d_k(n) - y_k(n)) \cdot f(\text{net}_k(n)) \cdot [1 - f(\text{net}_k(n))]$$

นั่นคือ 
$$\delta_k(n) = (d_k(n) - y_k(n)) \cdot y_k(n) \cdot (1 - y_k(n)) \quad (\text{ง-41})$$

และ 
$$\delta_j(n) = \sum_k \delta_k(n) \cdot w_{jk}(n) \cdot f(\text{net}_j(n)) \cdot [1 - f(\text{net}_j(n))]$$

$$\delta_j(n) = \sum_k \delta_k(n) \cdot w_{jk}(n) \cdot v_j(n) \cdot (1 - v_j(n)) \quad (\text{ง-42})$$

ดังนั้นกระบวนการเรียนรู้แบบแพร่กลับของโครงข่ายประสาทเทียม สามารถสรุปได้ 7 ขั้นตอน ดังนี้คือ

ขั้นตอนที่ 1 กำหนดค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม ได้แก่ ค่าน้ำหนักและไบแอส โดยการสุ่มค่า ซึ่งกำหนดให้มีค่าน้อยๆ อาจอยู่ในช่วง  $[-1, +1]$

ขั้นตอนที่ 2 คำนวณเอาต์พุตจากชั้นซ่อนเร้น

$$v_j(n) = f(\text{net}_j(n)) = \frac{1}{1 + \exp(-\text{net}_j(n))}$$

โดยที่ 
$$\text{net}_j(n) = \sum_{i=1}^N w_{ij} \cdot x_i + \theta_j \quad \text{เมื่อ } j=1, 2, 3, \dots, N_1$$

ขั้นตอนที่ 3 คำนวณค่าเอาต์พุตจากชั้นเอาต์พุต

$$y_k(n) = f(\text{net}_k(n)) = \frac{1}{1 + \exp(-\text{net}_k(n))}$$

โดยที่ 
$$\text{net}_k(n) = \sum_{j=1}^{N_1} w_{jk} \cdot v_j + \theta_k \quad \text{เมื่อ } k = 1, 2, 3, \dots, N_2$$

ขั้นตอนที่ 4 ปรับค่าน้ำหนักจากชั้นซ่อนเร้นไปยังชั้นเอาต์พุต

$$w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n)$$

โดยที่ 
$$\Delta w_{jk}(n) = \eta_1 \cdot \delta_k \cdot v_k \quad \text{และ} \quad \delta_k = (d_k(n) - y_k(n)) \cdot y_k(n) \cdot (1 - y_k(n))$$

ขั้นตอนที่ 5 ปรับค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้น

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n)$$

โดยที่ 
$$\Delta w_{ij}(n) = \eta_2 \cdot \delta_j \cdot x_i \quad \text{และ} \quad \delta_j = \sum_k \delta_k \cdot w_{jk} \cdot v_j(n) \cdot (1 - v_j(n))$$

ขั้นตอนที่ 6 ปรับค่าไบแอสที่ชั้นซ่อนเร้นและชั้นเอาต์พุต

$$\theta_j(n+1) = \theta_j(n) + \eta_3 \delta_j(n)$$

$$\theta_k(n+1) = \theta_k(n) + \eta_4 \delta_k(n)$$

ขั้นตอนที่ 7 ตรวจสอบเงื่อนไขการยุติกระบวนการเรียนรู้ ถ้ายังไม่พบให้เพิ่มจำนวนรอบ  $n = n + 1$  แล้วกลับไปขั้นตอนที่ 2

#### รายการอ้างอิง

Fausett, L. (1994). **Fundamentals of neural networks architectures, algorithms, and application.** New Jersey: Prentice Hall.

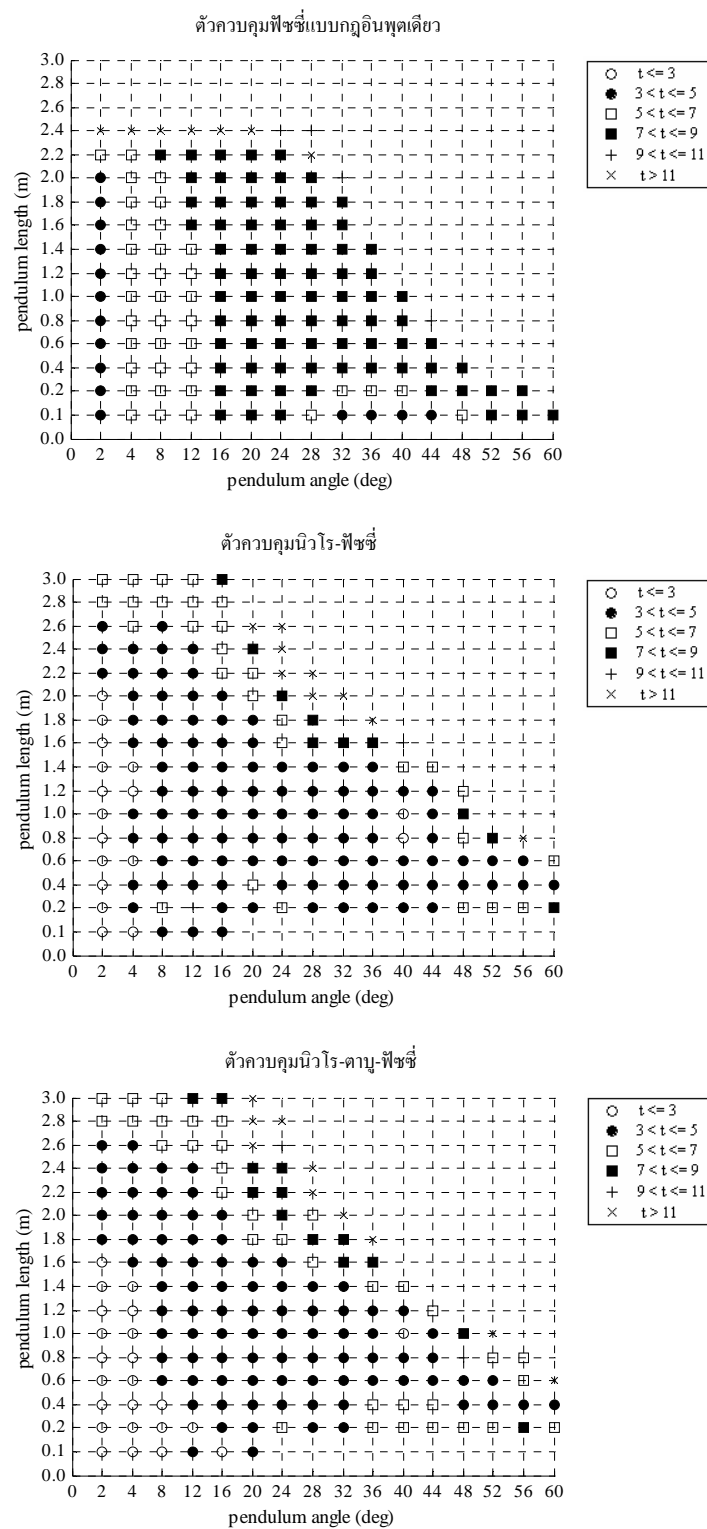
Freeman, J. A., and Skapura, D. M. (1991). **Neural networks algorithms, application, and programming techniques.** Massachusetts: Addison-Wesley.

Negnevitsky, M. (2002). **Artificial intelligence a guide to intelligent systems.** New York: Addison-Wesley.

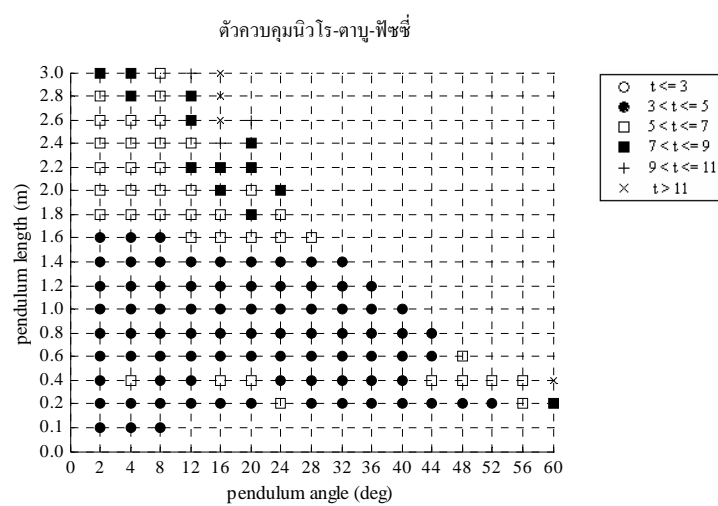
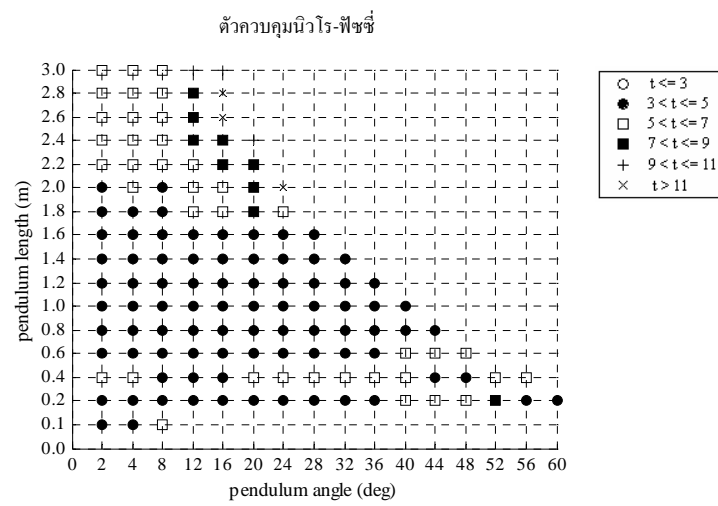
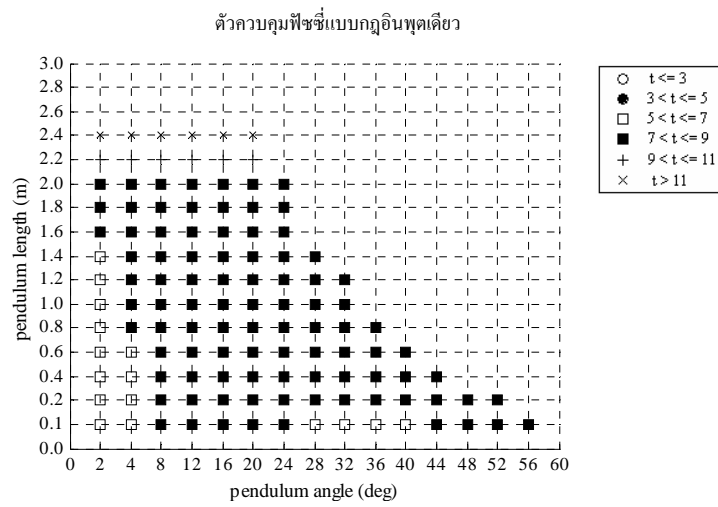
## ภาคผนวก จ

เวลาเข้าสู่เสถียรภาพของระบบถูกตุ้มผูกผันแบบข้อต่อเดียว  
เมื่อปรับเปลี่ยนค่าเริ่มต้นของระบบ โดยใช้ตัวควบคุมแบบต่างๆ

### จ.1 ผลการปรับเปลี่ยนมุมเริ่มต้นของลูกตุ้มและความยาวของก้านลูกตุ้ม



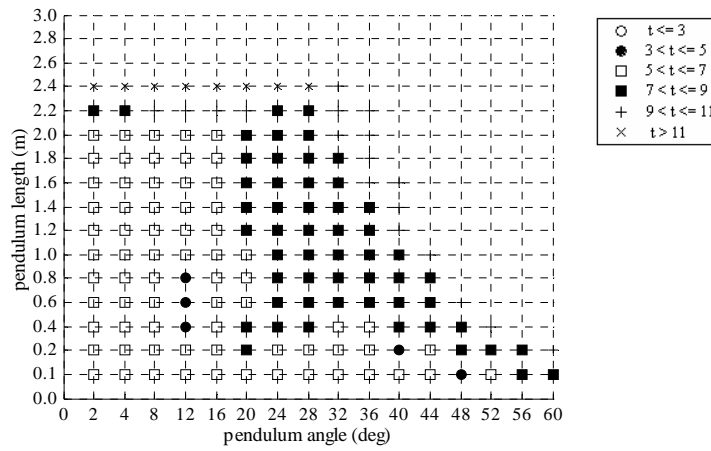
รูปที่ จ.1 เวลาเข้าสู่เสถียรภาพของระบบเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 0.0 เมตร



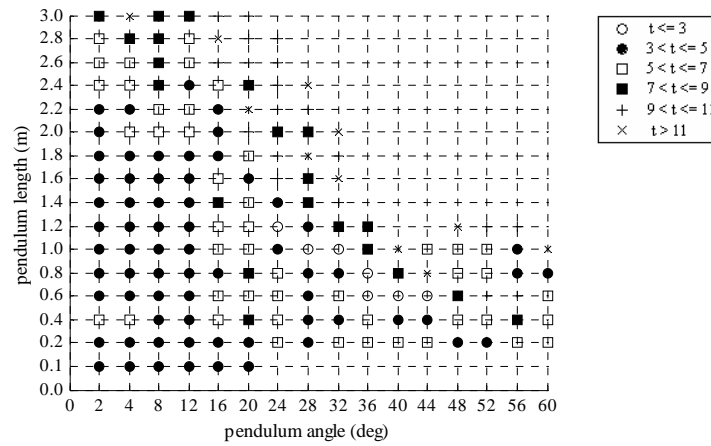
รูปที่ จ.2 เวลาเข้าสู่เสถียรภาพของระบบเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น 1.0 เมตร



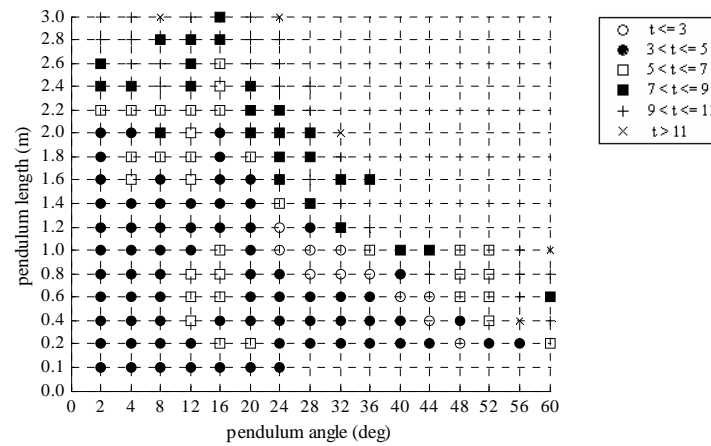
ตัวควบคุมพีชชีแบบกฎอินพุตเดียว



ตัวควบคุมนิวโร-พีชชี

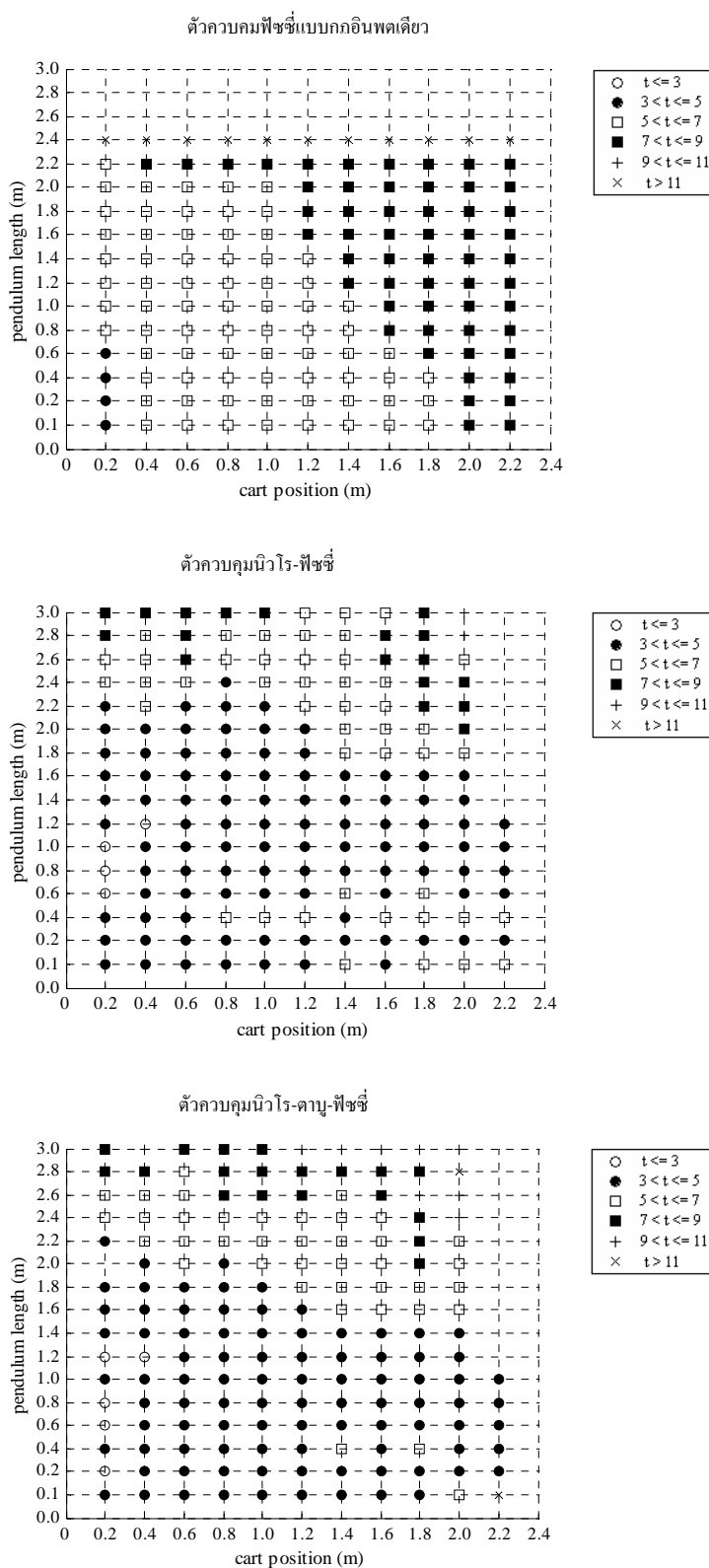


ตัวควบคุมนิวโร-คานู-พีชชี

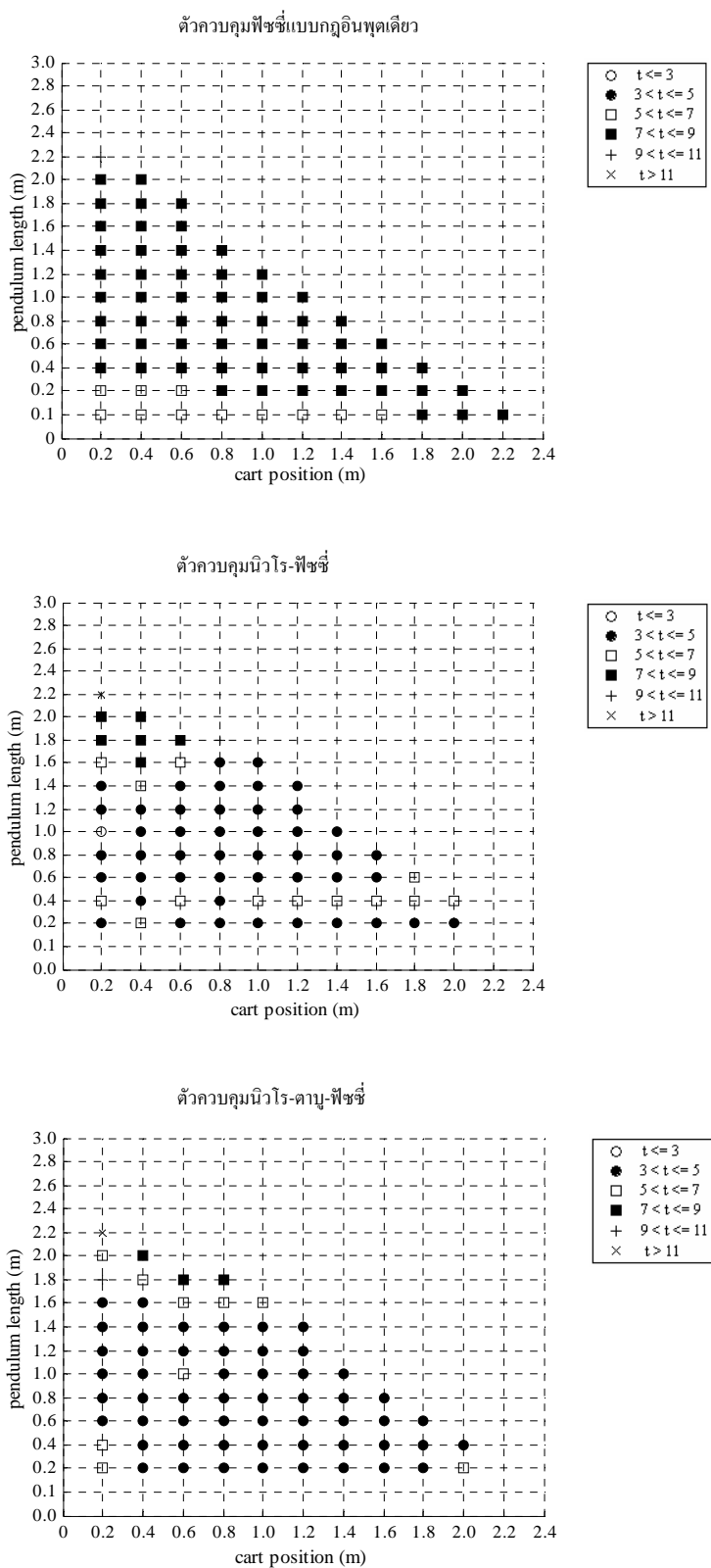


รูปที่ จ.3 เวลาเข้าสู่เสถียรภาพของระบบเมื่อตำแหน่งเริ่มต้นของตัวรถเป็น -1.0 เมตร

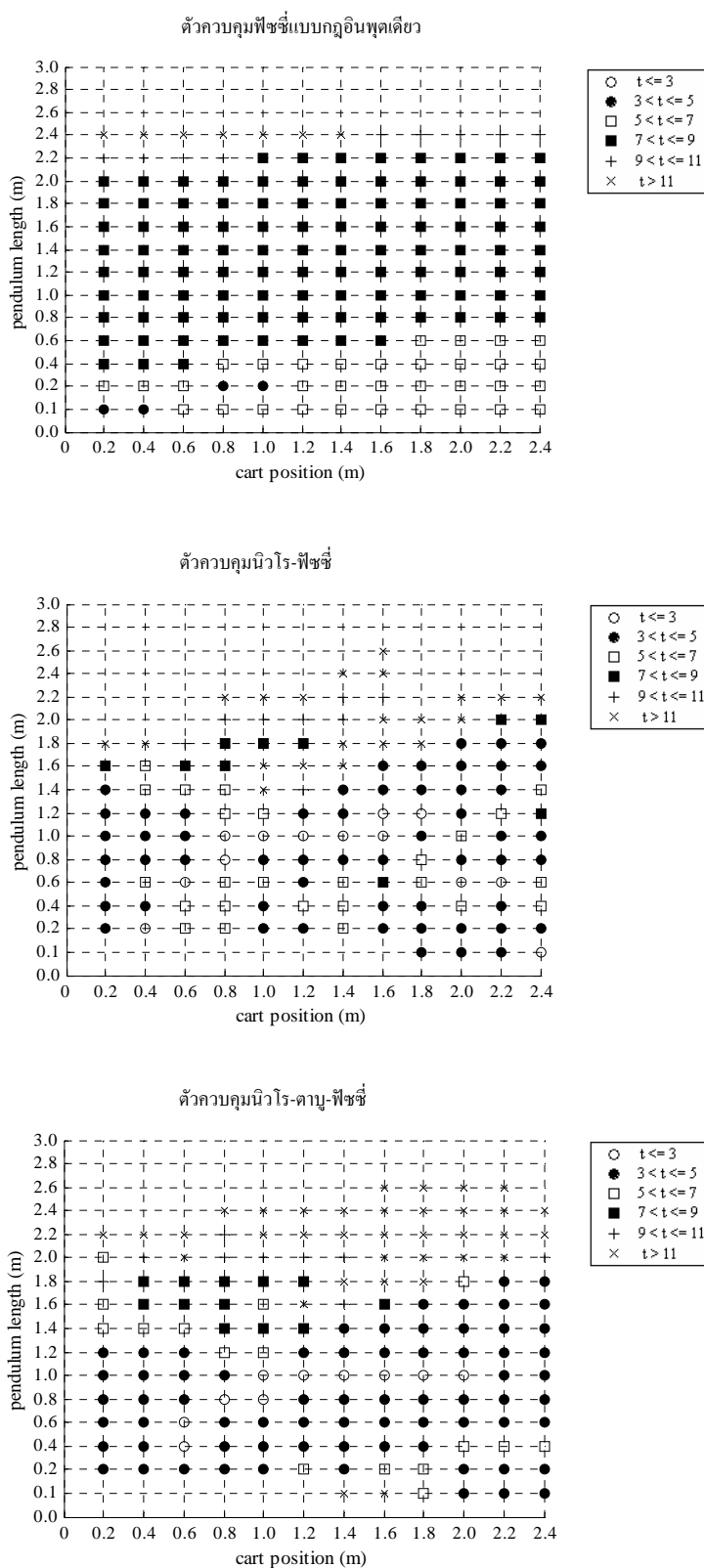
## จ.2 ผลการปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถและความยาวของก้านลูกตุ้ม



รูปที่ จ.4 เวลาเข้าสู่เสถียรภาพของระบบเมื่อมุมเริ่มต้นของลูกตุ้มเป็น 0.0 องศา

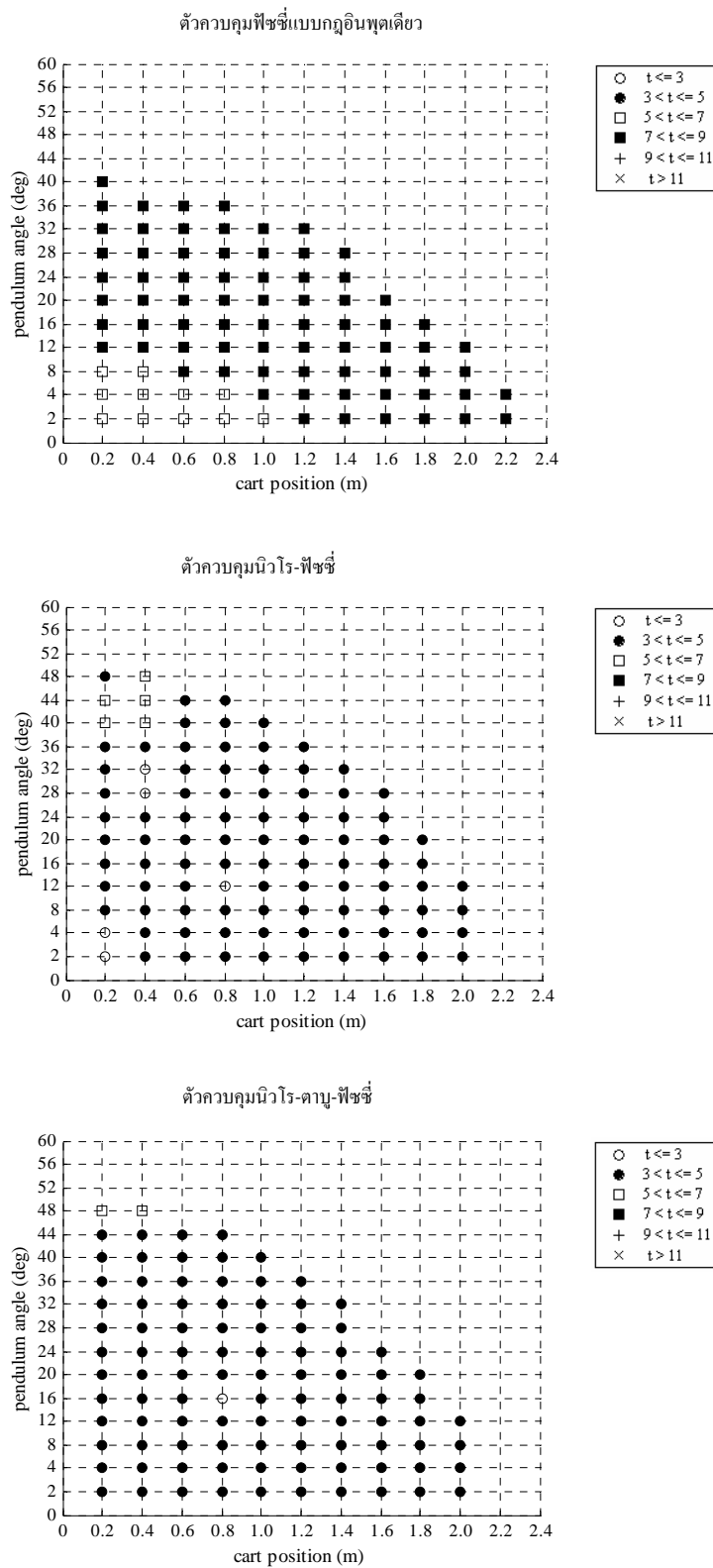


รูปที่ 5.5 เวลาเข้าสู่เสถียรภาพของระบบเมื่อมุมเริ่มต้นของลูกตุ้มเป็น 30.0 องศา

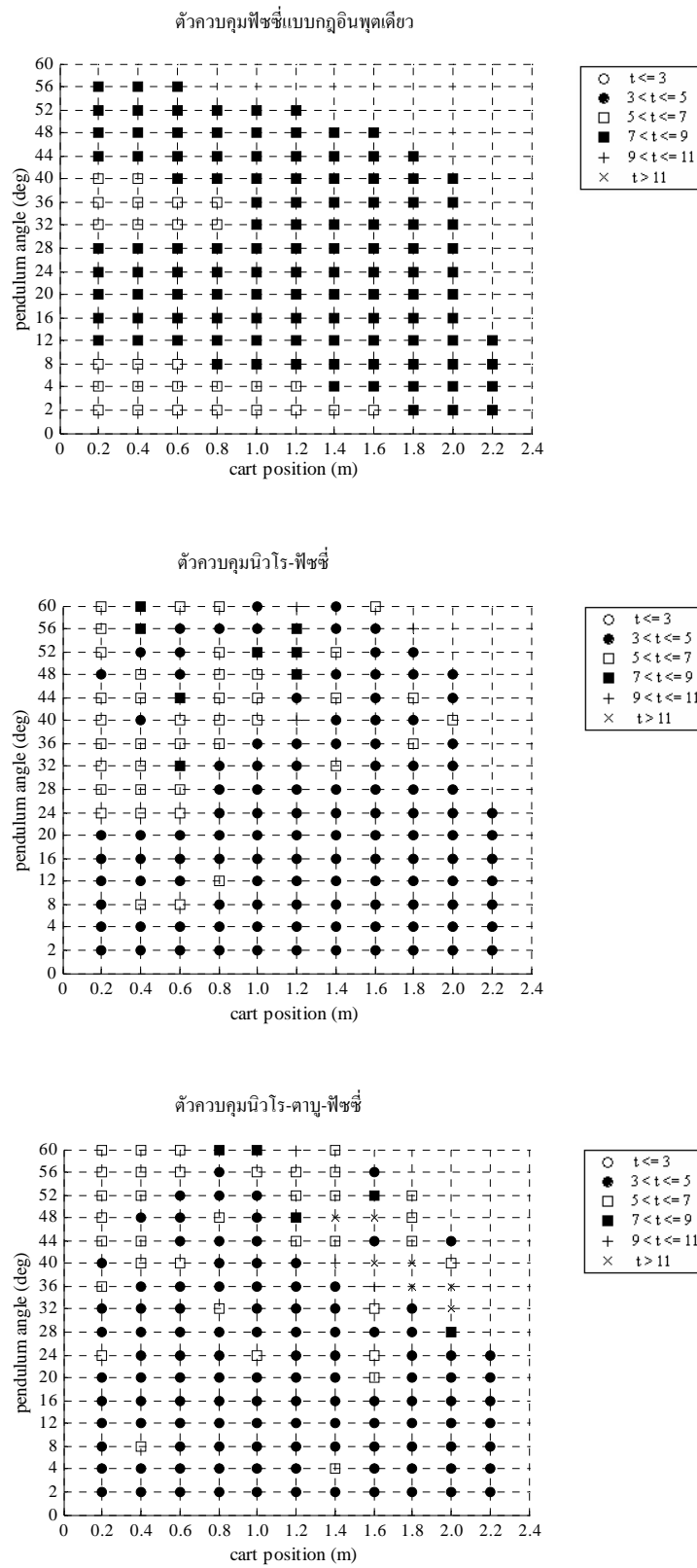


รูปที่ 6 เวลาเข้าสู่เสถียรภาพของระบบเมื่อมุมเริ่มต้นของลูกตุ้มเป็น  $-30.0$  องศา

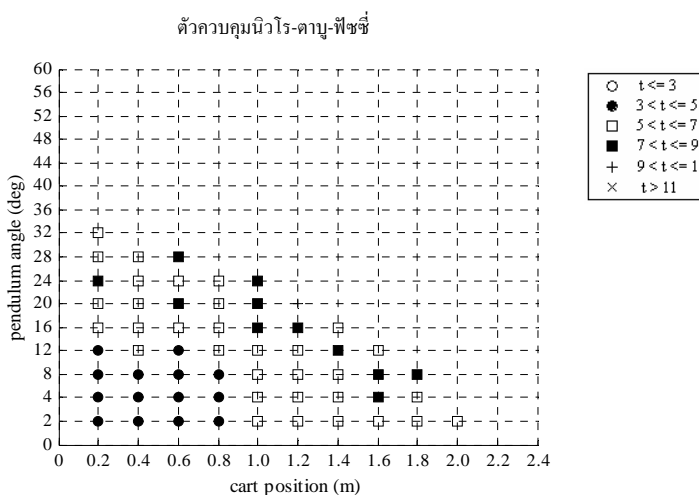
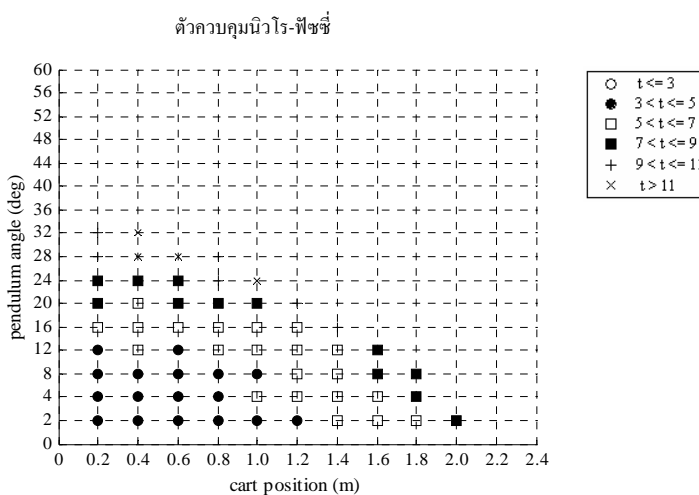
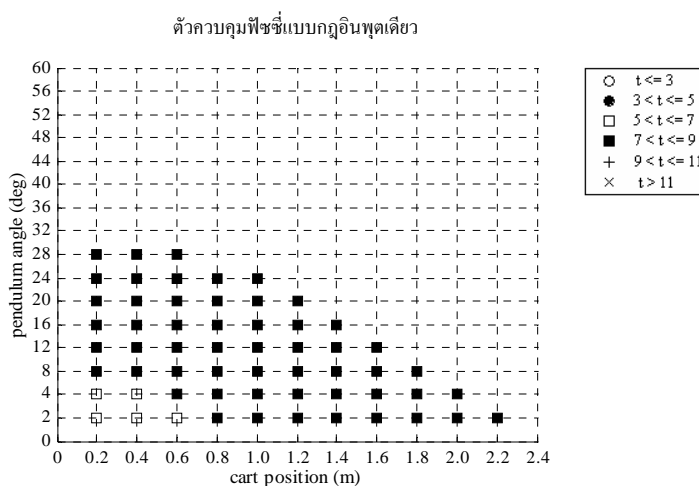
### จ.3 ผลการปรับเปลี่ยนตำแหน่งเริ่มต้นของตัวรถและมุมเริ่มต้นของลูกตุ้ม



รูปที่ จ.7 เวลาเข้าสู่เสถียรภาพของระบบเมื่อกำหนดลูกตุ้มยาว 1.0 เมตร



รูปที่ ๖.๘ เวลาเข้าสู่เสถียรภาพของระบบเมื่อกำหนดลูกตุ้มยาว 0.2 เมตร



รูปที่ ๑.๙ เวลาเข้าสู่เสถียรภาพของระบบเมื่อกำหนดลูกระยะยาว 2.0 เมตร

ภาคผนวก ฉ

โปรแกรม MATLAB สำหรับตัวควบคุมพีชชีแบบกฎอินพุตเดียว



\*\*\*\*\*

โปรแกรมสำหรับสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียวด้วยตัวควบคุมพีชชีแบบกฎอินพุตเดียว

โดย นางสาวสุภารัตน์ ขวัญอ่อน สาขาวิชาวิศวกรรมไฟฟ้า, มหาวิทยาลัยเทคโนโลยีสุรนารี, พ.ศ. 2547

\*\*\*\*\*

โปรแกรม *fuzzy\_SIRM.m*

```
function Data_system=fuzzy_SIRM(angle,position,length_p);
    % โปรแกรมหลักรับอินพุต 3 ค่า ได้แก่ มุมเริ่มต้นของลูกตุ้ม
    % ตำแหน่งเริ่มต้นของตัวรถ และความยาวก้านลูกตุ้ม
    % เอาต์พุตของโปรแกรม คือ ข้อมูลของระบบลูกตุ้มผกผันแบบข้อต่อเดียว

% กำหนดค่าพารามิเตอร์ของระบบลูกตุ้มผกผันแบบข้อต่อเดียว
Mc = 1.0; % มวลของตัวรถ (กิโลกรัม)
Mp = 0.1; % มวลของก้านลูกตุ้มผกผัน (กิโลกรัม)
g = 9.8; % ค่าแรงโน้มถ่วงของโลก (9.8 เมตร/วินาที2)
Lp = length_p/2; % ความยาวครึ่งหนึ่งของก้านลูกตุ้มผกผัน (เมตร)
N = 1000; % จำนวนจุดข้อมูลในการจำลองสถานการณ์ของระบบ
t = 10; % เวลาสิ้นสุดการจำลองสถานการณ์ของระบบ (วินาที)
dt = 0.01; % ช่วงเวลาแซมปลิง (วินาที)
s1 = 30*pi/180; % ตัวปรับค่าสำหรับมุมของลูกตุ้ม
s2 = 100*pi/180; % ตัวปรับค่าสำหรับความเร็วเชิงมุมของลูกตุ้ม
s3 = 2.4; % ตัวปรับค่าสำหรับตำแหน่งของตัวรถ
s4 = 1; % ตัวปรับค่าสำหรับความเร็วของตัวรถ
osf = 10*(Mc+Mp); % ตัวปรับค่าสำหรับแรงขับเคลื่อน
W1 = 2.00; % ค่าฐานสำหรับมุมของลูกตุ้ม
W2 = 1.50; % ค่าฐานสำหรับความเร็วเชิงมุมของลูกตุ้ม
W3 = 0.15; % ค่าฐานสำหรับตำแหน่งของตัวรถ
W4 = 0.15; % ค่าฐานสำหรับความเร็วของตัวรถ
B1 = 2.50; % ค่าขยายสำหรับมุมของลูกตุ้ม
B2 = 1.00; % ค่าขยายสำหรับความเร็วเชิงมุมของลูกตุ้ม
B3 = 0.20; % ค่าขยายสำหรับตำแหน่งของตัวรถ
B4 = 0.20; % ค่าขยายสำหรับความเร็วของตัวรถ

% กำหนดค่าเริ่มต้นในการจำลองสถานการณ์
F(1) = 0; % แรงขับเคลื่อนเริ่มต้นที่ให้กับตัวรถ
x1(1) = angle*pi/180; % มุมเริ่มต้นของลูกตุ้ม (เรเดียน)
x2(1) = 0; % ความเร็วเชิงมุมเริ่มต้นของลูกตุ้ม (เรเดียน/วินาที)
x3(1) = position % ตำแหน่งเริ่มต้นของตัวรถ (เมตร)
x4(1) = 0; % ความเร็วเริ่มต้นของตัวรถ (เมตร/วินาที)
xx2(1) = 0; % ความเร่งเชิงมุมของลูกตุ้ม (เรเดียน/วินาที2)
xx4(1) = 0; % ความเร่งของตัวรถ (เมตร/วินาที2)
for i = 2:N % เริ่มการจำลองสถานการณ์
    ti(i) = (i-1)*dt; % แปลงจุดข้อมูลเป็นเวลา
```

```

% กฎฟuzzyใน SIRM-1
R1_h = 0; R2_h = 0; R3_h = 0;
R1_c = 0; R2_c = 0; R3_c = 0;
% Rule 1
if (x1(i-1))/s1 < -1.0
    R1_h = 1.0;
    R1_c = -1.0;
elseif ((x1(i-1)/s1) >= -1.0) & ((x1(i-1)/s1) < 0)
    R1_h = abs(x1(i-1))/s1;
    R1_c = -1.0;
end;
% Rule 2
if ((x1(i-1)/s1) >= -1.0) & ((x1(i-1)/s1) < 0)
    R2_h = (x1(i-1)/s1)+1;
    R2_c = 0.0;
elseif ((x1(i-1)/s1) >= 0) & ((x1(i-1)/s1) < 1.0)
    R2_h = 1-(x1(i-1)/s1);
    R2_c = 0.0;
end;
% Rule 3
if ((x1(i-1)/s1) >= 0) & ((x1(i-1)/s1) < 1.0)
    R3_h = x1(i-1)/s1;
    R3_c = 1.0;
elseif (x1(i-1)/s1) >= 1.0
    R3_h = 1.0;
    R3_c = 1.0;
end;
f1(i) = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
% คำนวณผลการอนุมานกฎฟuzzyสำหรับ SIRM-1

% ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเงื่อนไข
% ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเอาต์พุต
% กฎที่ 1 ถ้า x1 = NB แล้ว f1 = -1.0
% กฎที่ 2 ถ้า x1 = ZO แล้ว f1 = 0.0
% กฎที่ 3 ถ้า x1 = PB แล้ว f1 = 1.0

% กฎฟuzzyใน SIRM-2
R1_h = 0; R2_h = 0; R3_h = 0;
R1_c = 0; R2_c = 0; R3_c = 0;
% Rule 4
if (x2(i-1))/s2 < -1.0
    R1_h = 1.0;
    R1_c = -1.0;
elseif ((x2(i-1)/s2) >= -1.0) & ((x2(i-1)/s2) < 0)
    R1_h = abs(x2(i-1))/s2;
    R1_c = -1.0;
end;
end;

```

```

% Rule 5                                     % กฎที่ 5 ถ้า  $x_2 = ZO$  แล้ว  $f_2 = 0.0$ 
if ((x2(i-1)/s2) >= -1.0) & ((x2(i-1)/s2) < 0 )
    R2_h = (x2(i-1)/s2)+1;
    R2_c = 0.0;
elseif (x2(i-1)/s2) >= 0 ) & ((x2(i-1)/s2) < 1.0 )
    R2_h = 1-(x2(i-1)/s2);
    R2_c = 0.0;
end;

% Rule 6                                     % กฎที่ 6 ถ้า  $x_2 = PB$  แล้ว  $f_2 = 1.0$ 
if ((x2(i-1)/s2) >= 0) & ((x2(i-1)/s2) < 1.0 )
    R3_h = x2(i-1)/s2;
    R3_c = 1.0;
elseif (x2(i-1)/s2) >= 1.0
    R3_h = 1.0;
    R3_c = 1.0;
end;

f2(i) = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
% คำนวณผลการอนุมานกฎฟuzzy สำหรับ SIRM-2

% กฎฟuzzy ใน SIRM-3
R1_h = 0; R2_h = 0; R3_h = 0;
R1_c = 0; R2_c = 0; R3_c = 0;
% Rule 7
if (x3(i-1)/s3 < -1.0
    R1_h = 1.0;
    R1_c = -1.0;
elseif ((x3(i-1)/s3) >= -1.0) & ((x3(i-1)/s3) < 0 )
    R1_h = abs(x3(i-1)/s3);
    R1_c = -1.0;
end;

% Rule 8                                     % กฎที่ 8 ถ้า  $x_3 = ZO$  แล้ว  $f_3 = 0.0$ ;
if ((x3(i-1)/s3) >= -1.0) & ((x3(i-1)/s3) < 0 )
    R2_h = (x3(i-1)/s3)+1;
    R2_c = 0.0;
elseif ((x3(i-1)/s3) >= 0 ) & ((x3(i-1)/s3) < 1.0 )
    R2_h = 1-(x3(i-1)/s3);
    R2_c = 0.0;
end;

% Rule 9                                     % กฎที่ 9 ถ้า  $x_3 = PB$  แล้ว  $f_3 = 1.0$ 
if ((x3(i-1)/s3) >= 0) & ((x3(i-1)/s3) < 1.0 )
    R3_h = x3(i-1)/s3;

```

% ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเงื่อนไข  
% ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเอาต์พุต  
% กฎที่ 7 ถ้า  $x_3 = NB$  แล้ว  $f_3 = -1.0$

```

R3_c = 1.0;
elseif (x3(i-1)/s3) >= 1.0
    R3_h = 1.0;
    R3_c = 1.0;
end;
f3(i) = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
% คำนวณผลการอนุมานกฎฟัซซี่สำหรับ SIRM-3

% กฎฟัซซี่ใน SIRM-4
R1_h = 0; R2_h = 0; R3_h = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเงื่อนไข
R1_c = 0; R2_c = 0; R3_c = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเอาต์พุต
% Rule 10 % กฎที่ 10 ถ้า x4 = NB แล้ว f4 = -1.0
if (x4(i-1))/s4 < -1.0
    R1_h = 1.0;
    R1_c = -1.0;
elseif ((x4(i-1)/s4) >= -1.0) & ((x4(i-1)/s4) < 0)
    R1_h = abs(x4(i-1))/s4;
    R1_c = -1.0;
end;
% Rule 11 % กฎที่ 11 ถ้า x4 = ZO แล้ว f4 = 0.0
if ((x4(i-1)/s4) >= -1.0) & ((x4(i-1)/s4) < 0)
    R2_h = (x4(i-1)/s4)+1;
    R2_c = 0.0;
elseif ((x4(i-1)/s4) >= 0) & ((x4(i-1)/s4) < 1.0)
    R2_h = 1-(x4(i-1)/s4);
    R2_c = 0.0;
end;
% Rule 12 % กฎที่ 12 ถ้า x4 = PB แล้ว f4 = 1.0
if ((x4(i-1)/s4) >= 0) & ((x4(i-1)/s4) < 1.0)
    R3_h = x4(i-1)/s4;
    R3_c = 1.0;
elseif (x4(i-1)/s4) >= 1.0
    R3_h = 1.0;
    R3_c = 1.0;
end;
f4(i) = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
% คำนวณผลการอนุมานกฎฟัซซี่สำหรับ SIRM-4

% กฎฟัซซี่ใน DID-1
R1_h = 0; R2_h = 0; R3_h = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเงื่อนไข
R1_c = 0; R2_c = 0; R3_c = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเอาต์พุต
% Rule 13 % กฎที่ 13 ถ้า |x1| = DS แล้ว Δw1 = 0.0
if (((abs(x1(i-1)))/s1) >= 0) & (((abs(x1(i-1)))/s1) < 0.5)

```

```

R1_h = 1-(2*(abs(x1(i-1)))/s1);
R1_c = 0.0;
end;
% Rule 14                                     % กฎที่ 14 ถ้า  $|x1| = DM$  แล้ว  $\Delta w1 = 0.5$ 
if ((abs(x1(i-1))/s1) >= 0) & ((abs(x1(i-1))/s1) < 0.5 )
    R2_h = 2*abs(x1(i-1))/s1;
    R2_c = 0.5;
elseif ((abs(x1(i-1))/s1) >= 0.5) & ((abs(x1(i-1))/s1) < 1.0 )
    R2_h = 2-(2*abs(x1(i-1))/s1);
    R2_c = 0.5;
end;
% Rule 15                                     % กฎที่ 15 ถ้า  $|x1| = DB$  แล้ว  $\Delta w1 = 1.0$ 
if ((abs(x1(i-1))/s1) >= 0.5) & ((abs(x1(i-1))/s1) < 1.0 )
    R3_h = (2*abs(x1(i-1))/s1)-1;
    R3_c = 1.0;
elseif (abs(x1(i-1))/s1) >= 1.0
    R3_h = 1.0;
    R3_c = 1.0;
end;
DW1(i) = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
                                                % คำนวณผลการอนุมานกฎฟuzzyที่ได้จากตัวแปรพลวัต  $\Delta w1$ 
DID_1(i) = W1+(B1*DW1(i));
                                                % คำนวณค่าระดับความสำคัญพลวัต DID-1

% กฎฟuzzyใน DID-2
R1_h = 0; R2_h = 0; R3_h = 0;
R1_c = 0; R2_c = 0; R3_c = 0;
% Rule 16                                     % กฎที่ 16 ถ้า  $|x1| = DS$  แล้ว  $\Delta w2 = 0.0$ 
if (((abs(x1(i-1))/s1) >= 0) & (((abs(x1(i-1))/s1) < 0.5 )
    R1_h = 1-(2*(abs(x1(i-1)))/s1);
    R1_c = 0.0;
end;
% Rule 17                                     % กฎที่ 17 ถ้า  $|x1| = DM$  แล้ว  $\Delta w2 = 0.5$ 
if ((abs(x1(i-1))/s1) >= 0) & ((abs(x1(i-1))/s1) < 0.5 )
    R2_h = 2*abs(x1(i-1))/s1;
    R2_c = 0.5;
elseif ((abs(x1(i-1))/s1) >= 0.5) & ((abs(x1(i-1))/s1) < 1.0 )
    R2_h = 2- (2*abs(x1(i-1))/s1);
    R2_c = 0.5;
end;
% Rule 18                                     % กฎที่ 18 ถ้า  $|x1| = DB$  แล้ว  $\Delta w2 = 1.0$ 
if ((abs(x1(i-1))/s1) >= 0.5) & ((abs(x1(i-1))/s1) < 1.0 )

```

```

R3_h = (2*abs(x1(i-1))/s1)-1;
R3_c = 1.0;
elseif (abs(x1(i-1))/s1) >= 1.0
    R3_h = 1.0;
    R3_c = 1.0;
end;
DW2(i) = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
% คำนวณผลการอนุมาณกฎฟuzzyที่ได้จากตัวแปรพลวัต Δw2
DID_2(i) = W2+(B2*DW2(i)); % คำนวณค่าระดับความสำคัญพลวัต DID-2

% กฎฟuzzyใน DID-3
R1_h = 0; R2_h = 0; R3_h = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเงื่อนไข
R1_c = 0; R2_c = 0; R3_c = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเอาต์พุต
% Rule 19 % กฎที่ 19 ถ้า |x1| = DS แล้ว Δw3 = 1.0
if (((abs(x1(i-1))/s1) >= 0) & ((abs(x1(i-1))/s1) < 0.5))
    R1_h = 1-(2*(abs(x1(i-1))/s1));
    R1_c = 1.0;
end;
% Rule 20 % กฎที่ 20 ถ้า |x1| = DM แล้ว Δw3 = 0.5
if ((abs(x1(i-1))/s1) >= 0) & ((abs(x1(i-1))/s1) < 0.5))
    R2_h = 2*abs(x1(i-1))/s1;
    R2_c = 0.5;
elseif ((abs(x1(i-1))/s1) >= 0.5) & ((abs(x1(i-1))/s1) < 1.0))
    R2_h = 2-(2*abs(x1(i-1))/s1);
    R2_c = 0.5;
end;
% Rule 21 % กฎที่ 21 ถ้า |x1| = DB แล้ว Δw3 = 0.0
if ((abs(x1(i-1))/s1) >= 0.5) & ((abs(x1(i-1))/s1) < 1.0))
    R3_h = (2*abs(x1(i-1))/s1)-1;
    R3_c = 0.0;
elseif (abs(x1(i-1))/s1) >= 1.0
    R3_h = 1.0;
    R3_c = 0.0;
end;
DW3(i) = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
% คำนวณผลการอนุมาณกฎฟuzzyที่ได้จากตัวแปรพลวัต Δw3
DID_3(i) = W3+ (B3*DW3(i)); % คำนวณค่าระดับความสำคัญพลวัต DID-3

% กฎฟuzzyใน DID-4
R1_h = 0; R2_h = 0; R3_h = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเงื่อนไข
R1_c = 0; R2_c = 0; R3_c = 0; % ค่าระดับความเป็นสมาชิกเริ่มต้นในส่วนตัวแปรเอาต์พุต

```



```

% ตรวจสอบข้อกำหนดในการเข้าสู่เสถียรภาพของระบบลูกคัมพคั่นแบบข้อต่อเดียว
state = [x1' x2' x3' x4']; % เก็บค่าตัวแปรสถานะของระบบลูกคัมพคั่นแบบข้อต่อเดียว
for k = 1:2 % ตรวจสอบมุมของลูกคัมและความเร็วเชิงมุมของลูกคัม
    i = N; % เริ่มพิจารณาจากจุดสุดท้ายของตัวแปรสถานะ
    y = state(:,k);
    while (abs(y(i)) <= 0.1*pi/180) & (i ~=1)
        i = i-1;
    end; % หาดำแหน่งจุดข้อมูลที่ทำให้มุมและความเร็วเชิงมุมของลูกคัมไม่เข้าสู่
    % ±0.1 องศา (องศา/วินาที)

    if i < N
        sta_time(k) = (i+1)*dt;
    else sta_time(k) = 99;
    end; % คำนวณเวลาเข้าสู่เสถียรภาพสำหรับมุมและความเร็วเชิงมุมของลูกคัม
end;

for k = 3:3 % ตรวจสอบตำแหน่งของตัวรถ
    i = N; % เริ่มพิจารณาจากจุดสุดท้ายของตำแหน่งตัวรถ
    y = state(:,k);
    while ((abs(y(i)) <= 0.01) & (max(abs(y)) <= 2.40) & (i ~= 1))
        i = i-1;
    end; % หาดำแหน่งจุดข้อมูลที่ทำให้ตำแหน่งของตัวรถไม่เข้าสู่ ±0.01 เมตร
    % และตัวรถเคลื่อนที่เกิน ±2.40 เมตร

    if i < N
        sta_time(k) = (i+1)*dt;
    else sta_time(k) = 99;
    end; % คำนวณเวลาเข้าสู่เสถียรภาพสำหรับตำแหน่งของตัวรถ
end;

for k = 4:4 % ตรวจสอบความเร็วของตัวรถ
    i = N; % เริ่มพิจารณาจากจุดสุดท้ายของความเร็วของตัวรถ
    y = state(:,k);
    while (abs(y(i)) <= 0.01) & (i = 1)
        i = i-1;
    end; % หาดำแหน่งจุดข้อมูลที่ทำให้ความเร็วของตัวรถไม่เข้าสู่
    % ±0.01 เมตร/วินาที

    if i < N
        sta_time(k) = (i+1)*dt;
    else sta_time(k) = 99;
    end; % คำนวณเวลาเข้าสู่เสถียรภาพสำหรับความเร็วของตัวรถ
end;

if max(sta_time) == 99
    time_sta = 99;
else time_sta = max(sta_time);
end; % เวลาเข้าสู่เสถียรภาพของระบบลูกคัมพคั่นแบบข้อต่อเดียว

```



```

% หาได้จากเวลาเข้าสู่เสถียรภาพสูงสุดของคัวแปรสถานะทั้งหมด
% time_sta = 99 หมายถึง ระบบไม่เข้าสู่เสถียรภาพ (unstable)

% พิจารณาค่ามากที่สุดของระบบลูกตุ้มผกผันแบบข้อต่อเดียว
os_ang = 0; % กำหนดให้มุมมากที่สุดของลูกตุ้มเป็นศูนย์องศา
ii = 1; % เริ่มพิจารณาจากจุดแรกของมุมของลูกตุ้มผกผัน
while (abs(x1(ii)) >= 5.0*pi/180)
ii = ii+1;
end; % หาดำแหน่งจุดข้อมูลที่ทำให้ขนาดมุมของลูกตุ้มนี้ยอกว่า 5 องศา
for i = ii:N
    if (abs(x1(i)) > os_ang)
        os_ang = abs(x1(i)); % หาขนาดมุมมากที่สุดของลูกตุ้มผกผัน
    end;
end;
Max_ang = os_ang*180/pi; % มุมมากที่สุดของลูกตุ้ม (องศา)
Max_angu = max(abs(x2))*180/pi; % ความเร็วเชิงมุมมากที่สุดของลูกตุ้ม (องศา/วินาที)
Max_pos = max(abs(x3)); % ตำแหน่งมากที่สุดของตัวรถ (เมตร)
Max_vel = max(abs(x4)); % ความเร็วมากที่สุดของตัวรถ (เมตร/วินาที)
Max_F = max(abs(F)); % แรงขับเคลื่อนมากที่สุดที่กระทำกับตัวรถ (นิวตัน)
Data_system = [time_sta Max_ang Max_angu Max_pos Max_vel Max_F];
% ข้อมูลของระบบที่ส่งค่ากลับเป็นเอาต์พุตให้กับฟังก์ชัน
[Ax,H1,H2] = plotyy(ti,x1*180/pi,ti,x3,'plot'); % แสดงมุมของลูกตุ้มและตำแหน่งของตัวรถในรูปแบบเดียวกัน
axes(Ax(1)); ylabel('angle (deg)'); % กำหนดแกนตั้งด้านซ้ายมือแสดงมุมของลูกตุ้ม (องศา)
axes(Ax(2)); ylabel('position (m)'); % กำหนดแกนตั้งด้านขวามือแสดงตำแหน่งของตัวรถ (เมตร)
xlabel('time (sec)'); % กำหนดแกนนอนแสดงเวลา (วินาที)

```

ภาคผนวก ข

โปรแกรมภาษาซีสำหรับตัวควบคุมนิวโร-ฟัชชี

\*\*\*\*\*

โปรแกรมสำหรับสร้างเสถียรภาพให้ระบบลูกคัมพคั่นแบบข้อต่อเดี่ยวด้วยตัวควบคุมนิวโร-ฟuzzy

โดย นางสาวสุภารัตน์ ขวัญอ่อน สาขาวิชาวิศวกรรมไฟฟ้า, มหาวิทยาลัยเทคโนโลยีสุรนารี, พ.ศ. 2547

\*\*\*\*\*

โปรแกรม *neuro\_fuzzy.cpp*

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define pi      3.14159           // ค่าคงที่  $\pi = 3.14159$ 
#define para    32                // จำนวนพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
#define I_MAX  1500              // จำนวนรอบในการเรียนรู้ของโครงข่ายประสาทเทียม
#define error_max 0.00001        // กำหนดค่าความคลาดเคลื่อนเพื่อยุติการเรียนรู้
#define num     500              // จำนวนจุดข้อมูลของระบบลูกคัมพคั่นที่ใช้ในกระบวนการเรียนรู้
#define pot     1000             // จำนวนจุดข้อมูลของระบบลูกคัมพคั่นที่ใช้ในการหาค่าต่างๆ ของระบบ
#define ang_ini 30.0             // มุมเริ่มต้นของลูกคัม (องศา)
#define pos_ini 0.0             // ตำแหน่งเริ่มต้นของลูกคัม (เมตร)
#define Lp      0.5             // ความยาวครึ่งหนึ่งของก้านลูกคัม (เมตร)

void function1(double uk1, double uk2, double uk3, double uk4, double *y_1, double *y_2, double *y_3, double *y_4);
// function1 รับค่าระดับความสำคัญพลวัตแต่ละรอบการเรียนรู้ และส่งค่า
// ความคลาดเคลื่อนมาที่โปรแกรมหลัก

void function2(double DID_1, double DID_2, double DID_3, double DID_4, double *t_sta, double *F_max, double *x1_max,
double *x2_max, double *x3_max, double *x4_max); // function 2 รับค่าระดับความสำคัญพลวัตที่ได้จากการเรียนรู้ของโครง
// ข่ายประสาทเทียม และส่งกลับค่าต่างๆ ของระบบลูกคัมพคั่น

main()
{
clrscr();

clock_t start,end; // ประกาศตัวแปรเพื่อคำนวณเวลาในการเรียนรู้
int l,m,n; // จำนวนเซลล์ประสาทเทียมในแต่ละชั้นของโครงข่าย
int i,iter // ตัวนับจำนวนรอบของการเรียนรู้
int st_ini; // ตัวแปรสำหรับรับค่าเริ่มต้นจากการสุ่มค่า
double n1,nn,nz,n1z,beta; // ค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม
double S[para]; // ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
double net_error[I_MAX]; // ค่าความคลาดเคลื่อนรวมในแต่ละรอบของการเรียนรู้
double x_l,x_h; // ขอบเขตของค่าพารามิเตอร์เริ่มต้นที่สุ่มให้กับโครงข่ายประสาทเทียม
double rd,rd_0,div; // ตัวแปรเกี่ยวกับการสุ่มค่า
double x[2]; // สัญญาณอินพุตในชั้นอินพุตของโครงข่ายประสาทเทียม
double sum,delta_u,delta_y,error_final,es[4]; // ตัวแปรในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม
double uk1,uk2,uk3,uk4; // ค่าระดับความสำคัญพลวัตในแต่ละรอบของการเรียนรู้
double y_1,y_2,y_3,y_4; // ค่าความคลาดเคลื่อนของตัวแปรสถานะแต่ละตัว
```

```

double t_sta; // เวลาในการเข้าสู่เสถียรภาพของระบบ
double F_max,x1_max,x2_max,x3_max,x4_max; // ค่ามากที่สุดของระบบลูกตุ้มผกผันแบบข้อต่อเดี่ยว
double D[4]; // ค่าระดับความสำคัญพลวัตที่ได้จากโครงข่ายประสาทเทียม
double W[2][4][I_MAX]; // ค่าน้ำหนักที่เชื่อมต่อระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
double W1[4][4][I_MAX]; // ค่าน้ำหนักที่เชื่อมต่อระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
double Z1[4][I_MAX]; // ค่าไบแอสที่ชั้นซ่อนเร้น
double Z2[4][I_MAX]; // ค่าไบแอสที่ชั้นเอาต์พุต
double delta_W2[4][4][I_MAX]; // ตัวปรับแก้ค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
double delta_W1[4][4][I_MAX]; // ตัวปรับแก้ค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
double del[4][I_MAX]; // พจน์ความคลาดเคลื่อนที่ชั้นซ่อนเร้น ( $\delta_j$ )
double del_1[4][I_MAX]; // พจน์ความคลาดเคลื่อนที่ชั้นเอาต์พุต ( $\delta_k$ )
double X1[4][I_MAX]; // เอาต์พุตของชั้นซ่อนเร้น
double X2[4][I_MAX]; // เอาต์พุตของชั้นเอาต์พุต
double yd[4][I_MAX]; // เอาต์พุตที่ต้องการของระบบ
double y[4][I_MAX]; // เอาต์พุตที่ได้ของระบบ
FILE *best_DID; // ตั้งชื่อเพิ่มข้อมูลเพื่อเก็บค่าระดับความสำคัญพลวัตว่า best_DID
x[0] = fabs(ang_ini)*pi/180.0; // เซลล์ประสาทเทียมแรกของชั้นอินพุตรับค่ามุมเริ่มต้นของลูกตุ้ม
x[1] = fabs(pos_ini); // เซลล์ประสาทเทียมที่สองของชั้นอินพุตรับค่าตำแหน่งเริ่มต้นของตัวรถ
x_1 = -1.0; // ขอบเขตล่างของค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
x_h = 1.0; // ขอบเขตบนของค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
nm = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
n1 = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
nz = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าไบแอสที่ชั้นซ่อนเร้น
n1z = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าไบแอสที่ชั้นเอาต์พุต
beta = 0.25; // ค่าคงที่โมเมนตัม
div = 32767.0; // ตัวหารค่าที่สุ่มได้เพื่อทำให้มีค่าสูงสุดเท่ากับ 1
printf("Enter seed of random number = "); // พิมพ์ตัวเลขการสุ่มเป็นจำนวนเต็ม
scanf("%d",&st_ini); // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด
srand(st_ini); // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
for(i=0;i<para;i++)
{
    rd = rand(); // ค่า rd จากการสุ่มอยู่ในช่วง 0 ถึง 32767
    rd_0 = rd/div; // ค่า rd_0 จากการสุ่มอยู่ในช่วง 0 ถึง 1
    S[i] = ((x_h-x_1)*rd_0)+x_1; // ค่าพารามิเตอร์เริ่มต้นทั้ง 32 ค่า มีขอบเขตอยู่ในช่วง -1 ถึง 1
}
i = 0; // ตัวนับลำดับของค่าพารามิเตอร์
for (l=0;l<2;l++)
{
    for(m=0;m<4;m++)
    {
        W[l][m][1] = S[i]; // กำหนดค่าเริ่มต้นให้กับค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
    }
}

```

```

        i++;
    }
}
for (m=0;m<4;m++)
{
    Z1[m][1] = S[i]; // กำหนดค่าเริ่มต้นให้กับค่าไบแอสที่ชั้นซ่อนเร้น
    i++;
}
for (m=0;m<4;m++)
{
    for (n=0;n<4;n++)
    {
        W1[m][n][1] = S[i]; // กำหนดค่าเริ่มต้นให้กับค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
        i++;
    }
}
for (n=0;n<4;n++)
{
    Z2[n][1] = S[i]; // กำหนดค่าเริ่มต้นให้กับค่าไบแอสที่ชั้นเอาต์พุต
    i++;
}
for(m=0;m<4;m++)
{
    for(n=0;n<4;n++)
    { delta_W1[m][n][0] = 0.0; } // กำหนดค่าเริ่มต้นให้กับ delta_W1
}
for(l=0;l<2;l++)
{
    for(m=0;m<4;m++)
    { delta_W[l][m][0] = 0.0; } // กำหนดค่าเริ่มต้นให้กับ delta_W
}
for(n=0;n<4;n++)
{
    yd[n][1] = 0.0; // กำหนดค่าเริ่มต้นให้กับเอาต์พุตที่ต้องการของระบบ
    X2[n][0] = 0.0; // กำหนดค่าเริ่มต้นให้กับเอาต์พุตของโครงข่ายประสาทเทียม
}
i = 2;
net_error[1] = 0.05; // กำหนดค่าเริ่มต้นให้กับค่าความคลาดเคลื่อนรวม
error_final = 500; // กำหนดค่าเริ่มต้นของความคลาดเคลื่อนสุดท้ายให้มีค่า 500 (ตัวอย่าง)
iter = 0; // กำหนดจำนวนรอบการเรียนรู้เริ่มต้นที่ให้ค่าความคลาดเคลื่อนน้อยสุด
start = clock(); // เริ่มต้นคำนวณเวลาที่ใช้ในการเรียนรู้

```

```

while((net_error[i-1]>error_max)&&(i<=I_MAX)) // ตรวจสอบเงื่อนไขการยุติกระบวนการเรียนรู้
{
for(m=0;m<4;m++) // จำนวนเอาต์พุตของชั้นซ่อนเร้น (X1j)
{
sum = 0.0;
for(l=0;l<2;l++)
{ sum = sum+W[l][m][i-1]*x[l]; }
sum = sum+Z1[m][i-1];
X1[m][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตของชั้นซ่อนเร้นเมื่อฟังก์ชันกระตุ้นเป็นแบบซิกมอยด์
}
for(n=0;n<4;n++) // จำนวนเอาต์พุตของชั้นเอาต์พุต (X2k)
{
sum = 0.0;
for(m=0;m<4;m++)
{ sum = sum+W1[m][n][i-1]*X1[m][i-1]; }
sum = sum+Z2[n][i-1];
X2[n][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตของชั้นเอาต์พุตเมื่อฟังก์ชันกระตุ้นเป็นแบบซิกมอยด์
}
uk1 = X2[0][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับมุมของลูกตุ้ม (DID-1)
uk2 = X2[1][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับความเร็วเชิงมุมของลูกตุ้ม (DID-2)
uk3 = X2[2][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับตำแหน่งของตัวรถ (DID-3)
uk4 = X2[3][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับความเร็วของตัวรถ (DID-4)
function1(uk1,uk2,uk3,uk4,&y_1,&y_2,&y_3,&y_4); // เรียกฟังก์ชันเพื่อจำลองระบบและส่งกลับค่าความคลาดเคลื่อนของระบบ
y[0][i-1] = y_1; // รับค่าความคลาดเคลื่อนมุมของลูกตุ้ม
y[1][i-1] = y_2; // รับค่าความคลาดเคลื่อนความเร็วเชิงมุมของลูกตุ้ม
y[2][i-1] = y_3; // รับค่าความคลาดเคลื่อนตำแหน่งของตัวรถ
y[3][i-1] = y_4; // รับค่าความคลาดเคลื่อนความเร็วของตัวรถ
for (n=0;n<4;n++) // ปรับค่าน้ำหนักจากชั้นซ่อนเร้นไปยังชั้นเอาต์พุต (W1jk)
{
if((X2[n][i-1]-X2[n][i-2])==0.0) // ถ้าผลต่างเอาต์พุตของโครงข่ายประสาทเทียมเป็น 0
{ delta_u = 0.000000001; } // กำหนด delta_u ≠ 0 เพื่อหลีกเลี่ยงปัญหาตัวหารเป็นศูนย์
else
{ delta_u = (X2[n][i-1]-X2[n][i-2]); } // ผลต่างเอาต์พุตของโครงข่ายประสาทเทียม
if(((y[n][i-1]-y[n][i-2])/delta_u)>=32767.0) // กำหนดขอบเขตของค่า delta_y ในกรณีค่าบวก
{ delta_y = 32767.0; }
else if(((y[n][i-1]-y[n][i-2])/delta_u)<=-32767.0)
{ delta_y = -32767.0; } // กำหนดขอบเขตของค่า delta_y ในกรณีค่าลบ
else
{ delta_y = ((y[n][i-1]-y[n][i-2])/delta_u); } // ผลต่างเอาต์พุตของระบบหารด้วยผลต่างเอาต์พุตของโครงข่าย
if((delta_y)>0.0) // ตรวจสอบเครื่องหมายของ delta_y
{ es[n] = (yd[n][i-1]-y[n][i-1]); } // ผลต่างเอาต์พุตที่ต้องการกับเอาต์พุตจริงของระบบกรณีเครื่องหมายบวก
}
}

```

```

else if((delta_y)<0.0)
{ es[n] = -1*(yd[n][i-1]-y[n][i-1]); } // ผลต่างเอาต์พุตที่ต้องการกับเอาต์พุตจริงของระบบกรณีเครื่องหมายลบ
  else if((delta_y)==0.0)
{ es[n] = 0.0; } // ผลต่างเอาต์พุตที่ต้องการกับเอาต์พุตจริงของระบบกรณี delta_y เป็นศูนย์
del_1[n][i-1] = es[n]*X2[n][i-1]*(1.0-X2[n][i-1]); // พจน์ความคลาดเคลื่อนที่ชั้นเอาต์พุต ( $\delta 1k$ )
for(m=0;m<4;m++)
{
  delta_W1[m][n][i-1] = n1*del_1[n][i-1]*X1[m][i-1]+(beta*delta_W1[m][n][i-2]);
  // ตัวปรับแก้ค่าน้ำหนักจากชั้นซ่อนเร้นไปยังชั้นเอาต์พุต
  W1[m][n][i] = W1[m][n][i-1]+delta_W1[m][n][i-1];
  // ค่าน้ำหนักจากชั้นซ่อนเร้น ไปยังชั้นเอาต์พุตที่ปรับค่าแล้ว
}
}
for (m=0;m<4;m++) // ปรับค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้น (Wij)
{
  sum = 0.0;
  for (n=0;n<4;n++)
  { sum = sum+(del_1[n][i-1]*W1[m][n][i]); }
  del[m][i-1] = sum*X1[m][i-1]*(1-X1[m][i-1]); // พจน์ความคลาดเคลื่อนที่ชั้นซ่อนเร้น ( $\delta j$ )
  for (l=0;l<2;l++)
  {
    delta_W[l][m][i-1] = nm*del[m][i-1]*x[l]+(beta*delta_W[l][m][i-2]);
    // ตัวปรับแก้ค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้น
    W[l][m][i] = W[l][m][i-1]+delta_W[l][m][i-1]; // ค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้นที่ปรับค่าแล้ว
  }
}
for (n=0;n<4;n++)
{ Z2[n][i] = Z2[n][i-1]+(n1z*del_1[n][i-1]); } // ปรับค่าไบแอสที่ชั้นเอาต์พุต (Z2k)
for (m=0;m<4;m++)
{ Z1[m][i] = Z1[m][i-1]+(nz*del[m][i-1]); } // ปรับค่าไบแอสที่ชั้นซ่อนเร้น (Z1j)
sum = 0.0;
for (n=0;n<4;n++)
{
  sum = sum+((yd[n][i-1]-y[n][i-1])*(yd[n][i-1]-y[n][i-1]));
  // ผลรวมของค่าความคลาดเคลื่อนกำลังสอง
  yd[n][i] = yd[n][i-1]; // กำหนดให้เอาต์พุตของระบบที่ต้องการมีค่าเท่าเดิม
}
net_error[i] = sum/2.0; // ความคลาดเคลื่อนรวมที่ได้จากโครงข่ายประสาทเทียม
if(net_error[i]<error_final)
{
  error_final = net_error[i]; // กรณี net_error[i] น้อยกว่า error_final ที่กำหนดไว้
}

```

```

    iter = i; // จำนวนรอบการเรียนรู้ที่ให้ค่าความคลาดเคลื่อนต่ำสุด
}
i++; // ปรับเพิ่มจำนวนรอบในการเรียนรู้
} // สิ้นสุดการวนรอบคำสั่ง while

if(iter>=2) // กรณีค่าความคลาดเคลื่อนของโครงข่ายประสาทเทียมลดลง
{
    function2(X2[0][iter-1],X2[1][iter-1],X2[2][iter-1],X2[3][iter-1],
        &t_sta,&F_max,&x1_max,&x2_max,&x3_max,&x4_max);
        // function2 รับค่าระดับความสำคัญพลวัตและส่งกลับค่าต่างๆ ของระบบ

    D[0]=X2[0][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-1)
    D[1]=X2[1][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-2)
    D[2]=X2[2][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-3)
    D[3]=X2[3][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-4)
    best_DID = fopen("G:best_rand_1.dat","wb"); // เปิดเพิ่มข้อมูล
    fwrite(&D,sizeof(double),4,best_DID); // เขียนค่าระดับความสำคัญพลวัตทั้ง 4 ค่า ลงในเพิ่มข้อมูล
    fclose(best_DID); // ปิดเพิ่มข้อมูล
    printf("%d time_stable = %f\n",st_ini,t_sta); // แสดงตัวเลขการสุ่มและเวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผัน
    printf("\nF_max=%8.4f x1_max=%8.4f x2_max=%8.4f x3_max=%8.4f x4_max=%8.4f\n",F_max,x1_max*180/pi,
        x2_max*180/pi,x3_max,x4_max); // แสดงค่ามากที่สุดของระบบ
    printf("\nD1=%9.5f ; D2=%9.5f ; D3=%9.5f ; D4=%9.5f\n",(X2[0][iter-1])*10.0, (X2[1][iter-1])*10.0,
        (X2[2][iter-1])*10.0, (X2[3][iter-1])*10.0); // แสดงค่าระดับความสำคัญพลวัต
    printf("\n error at %d=%7.6f",iter,error_final); // แสดงจำนวนรอบและค่าความคลาดเคลื่อนน้อยสุด
}
else if(iter==0) // กรณีค่าความคลาดเคลื่อนของโครงข่ายประสาทเทียมไม่ลดลง
{ printf("\n %d Not found best value #\n",st_ini); } // แสดงข้อความว่าตัวเลขสุ่มที่ใช้นั้นไม่พบคำตอบ
end = clock(); // สิ้นสุดการคำนวณเวลา
printf("Execution time=%f sec", (end-start)/CLK_TCK); // แสดงเวลาที่ใช้ในการเรียนรู้
getch();
return 0;
} // สิ้นสุดการทำงานของโปรแกรมหลัก (main)

/*-----*/

void function1(double uk1,double uk2,double uk3,double uk4,double *y_1,double *y_2,double *y_3,double *y_4)
// function1 รับค่าระดับความสำคัญพลวัตและส่งกลับค่าความ
// คลาดเคลื่อนสำหรับการเรียนรู้ของโครงข่ายประสาทเทียมในแต่ละรอบ
{ // เริ่มต้น function1
int j; // ตัวนับการวนรอบในการจำลองระบบ
double Mc, Mp,g,dt,S1,S2,S3,S4,osf; // ตัวแปรของระบบลูกตุ้มผกผันแบบข้อต่อเดียวที่ใช้ในการจำลองระบบ
double F[num]; // แรงขับเคลื่อนที่กระทำต่อตัวรถในแนวระดับ
double x1[num]; // มุมของลูกตุ้ม (เรเดียน)
double x2[num]; // ความเร็วเชิงมุมของลูกตุ้ม (เรเดียน/วินาที)

```



```

double x3[num]; // ตำแหน่งของตัวรถ (เมตร)
double x4[num]; // ความเร็วของตัวรถ (เมตร/วินาที)
double xx2[num]; // ความเร่งเชิงมุมของลูกตุ้ม (เรเดียน/วินาที2)
double xx4[num]; // ความเร่งของตัวรถ (เมตร/วินาที2)
double f1,f2,f3,f4; // ผลการอนุมานฟัซซี่ด้วยวิธี simplified fuzzy reasoning method
double R1_h,R2_h,R3_h,R1_c,R2_c,R3_c; // ค่าระดับความเป็นสมาชิกในกฎฟัซซี่
Mc = 1.0; // มวลของตัวรถ (กิโลกรัม)
Mp = 0.1; // มวลของก้านลูกตุ้ม (กิโลกรัม)
g = 9.8; // ค่าแรงโน้มถ่วงของโลก
dt = 0.01; // ช่วงเวลาแซมปลิง
S1 = 30.0*pi/180.0; // ตัวปรับค่าสำหรับมุมของลูกตุ้ม
S2 = 100.0*pi/180.0; // ตัวปรับค่าสำหรับความเร็วเชิงมุมของลูกตุ้ม
S3 = 2.4; // ตัวปรับค่าสำหรับตำแหน่งของตัวรถ
S4 = 1.0; // ตัวปรับค่าสำหรับความเร็วของตัวรถ
osf = 10.0*(Mc+Mp); // ตัวปรับค่าสำหรับแรงขับเคลื่อน
F[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับแรงขับเคลื่อน
x1[0] = ang_ini*pi/180.0; // กำหนดค่ามุมเริ่มต้นของลูกตุ้ม
x2[0] = 0.0*pi/180.0; // กำหนดค่าความเร็วเชิงมุมเริ่มต้นของลูกตุ้ม
x3[0] = pos_ini; // กำหนดค่าตำแหน่งเริ่มต้นของตัวรถ
x4[0] = 0.0; // กำหนดค่าความเร็วเริ่มต้นของตัวรถ
xx2[0] = 0.0; // กำหนดค่าความเร่งเชิงมุมเริ่มต้นของลูกตุ้ม
xx4[0] = 0.0; // กำหนดค่าความเร่งเริ่มต้นของตัวรถ
*y_1 = 0.0; // ค่าความคลาดเคลื่อนสำหรับมุมของลูกตุ้ม
*y_2 = 0.0; // ค่าความคลาดเคลื่อนสำหรับความเร็วเชิงมุมของลูกตุ้ม
*y_3 = 0.0; // ค่าความคลาดเคลื่อนสำหรับตำแหน่งของตัวรถ
*y_4 = 0.0; // ค่าความคลาดเคลื่อนสำหรับความเร็วของตัวรถ
for(j=1;j<num;j++) // เริ่มต้นการจำลองระบบ
{
/*-----SIRM-1-----*/ // กำหนดกฎฟัซซี่แบบกฎอินพุตเดียวสำหรับมุมของลูกตุ้ม
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรอินพุต
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรเอาต์พุต
// Rule 1
// กฎที่ 1 ถ้า x1 = NB แล้ว f1 = -1.0
if((x1[j-1]/S1)<-1.0)
{
R1_h = 1.0;
R1_c = -1.0;
}
else if(((x1[j-1]/S1)>=-1.0)&&(x1[j-1]/S1)<0.0) )
{
R1_h = fabs(x1[j-1]/S1);
R1_c = -1.0;
}
}

```

```

}
// Rule 2 // กฎที่ 2 ถ้า x1 = ZO แล้ว f1 = 0.0
if(((x1[j-1]/S1)>=-1.0)&&((x1[j-1]/S1)<0.0) )
{
    R2_h = (x1[j-1]/S1)+1.0;
    R2_c = 0.0;
}
else if(((x1[j-1]/S1)>=0.0)&&((x1[j-1]/S1)<1.0) )
{
    R2_h = 1.0-(x1[j-1]/S1);
    R2_c = 0.0;
}
// Rule 3 // กฎที่ 3 ถ้า x1 = PB แล้ว f1 = 1.0
if(((x1[j-1]/S1)>=0.0)&&((x1[j-1]/S1)<1.0) )
{
    R3_h = (x1[j-1]/S1);
    R3_c = 1.0;
}
else if((x1[j-1]/S1)>=1.0)
{
    R3_h = 1.0;
    R3_c = 1.0;
}
f1 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-1

/*-----SIRM-2-----*/ // กำหนดกฎฟuzzyแบบกฎอินพุตเดียวสำหรับความเร็วเชิงมุมของลูกตุ้ม
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรอินพุต
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรเอาต์พุต
// Rule 4 // กฎที่ 4 ถ้า x2 = NB แล้ว f2 = -1.0
if((x2[j-1]/S2)<-1.0)
{
    R1_h = 1.0;
    R1_c = -1.0;
}
else if(((x2[j-1]/S2)>=-1.0)&&((x2[j-1]/S2)<0.0) )
{
    R1_h = fabs(x2[j-1]/S2);
    R1_c = -1.0;
}

```

```

// Rule 5                                     // กฎที่ 5 ถ้า  $x_2 = ZO$  แล้ว  $f_2 = 0.0$ 
if(((x2[j-1]/S2)>=-1.0)&&((x2[j-1]/S2)<0.0) )
{
    R2_h = (x2[j-1]/S2)+1.0;
    R2_c = 0.0;
}
else if(((x2[j-1]/S2)>=0.0)&&((x2[j-1]/S2)<1.0) )
{
    R2_h = 1.0-(x2[j-1]/S2);
    R2_c = 0.0;
}

// Rule 6                                     // กฎที่ 6 ถ้า  $x_2 = PB$  แล้ว  $f_2 = 1.0$ 
if(((x2[j-1]/S2)>=0.0)&&((x2[j-1]/S2)<1.0) )
{
    R3_h = (x2[j-1]/S2);
    R3_c = 1.0;
}
else if((x2[j-1]/S2)>=1.0)
{
    R3_h = 1.0;
    R3_c = 1.0;
}

f2 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-2

/*-----SIRM-3-----*/
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดกฎฟuzzyแบบกฎอินพุตเดียวสำหรับตำแหน่งของตัวรถ
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรอินพุต
// Rule 7                                     // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรเอาต์พุต
if((x3[j-1]/S3)<-1.0)
{
    R1_h = 1.0;
    R1_c = -1.0;
}
else if(((x3[j-1]/S3)>=-1.0)&&((x3[j-1]/S3)<0.0) )
{
    R1_h = fabs(x3[j-1]/S3);
    R1_c = -1.0;
}

// Rule 8                                     // กฎที่ 8 ถ้า  $x_3 = ZO$  แล้ว  $f_3 = 1.0$ 
if(((x3[j-1]/S3)>=-1.0)&&((x3[j-1]/S3)<0.0) )

```

```

{
    R2_h = (x3[j-1]/S3)+1.0;
    R2_c = 0.0;
}
else if(((x3[j-1]/S3)>=0.0)&&((x3[j-1]/S3)<1.0) )
{
    R2_h = 1.0-(x3[j-1]/S3);
    R2_c = 0.0;
}
// Rule 9 // กฎที่ 9 ถ้า x3 = PB แล้ว f3 = 1.0
if(((x3[j-1]/S3)>=0.0)&&((x3[j-1]/S3)<1.0) )
{
    R3_h = (x3[j-1]/S3);
    R3_c = 1.0;
}
else if((x3[j-1]/S3)>=1.0)
{
    R3_h = 1.0;
    R3_c = 1.0;
}
f3 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-3

/*-----SIRM-4-----*/ // กำหนดกฎฟuzzyแบบกฎอินพุตเดียวสำหรับความเร็วของตัวรถ
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรอินพุต
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าระดับความเป็นสมาชิกเริ่มต้นของตัวแปรเอาต์พุต
// Rule 10 // กฎที่ 10 ถ้า x4 = NB แล้ว f4 = -1.0
if((x4[j-1]/S4)<-1.0)
{
    R1_h = 1.0;
    R1_c = -1.0;
}
else if(((x4[j-1]/S4)>=-1.0)&&((x4[j-1]/S4)<0.0) )
{
    R1_h = fabs(x4[j-1]/S4);
    R1_c = -1.0;
}
// Rule 11 // กฎที่ 11 ถ้า x4 = ZO แล้ว f4 = 1.0
if(((x4[j-1]/S4)>=-1.0)&&((x4[j-1]/S4)<0.0) )
{
    R2_h = (x4[j-1]/S4)+1.0;

```

```

R2_c = 0.0;
}
else if(((x4[j-1]/S4)>=0.0)&&((x4[j-1]/S4)<1.0) )
{
R2_h = 1.0-(x4[j-1]/S4);
R2_c = 0.0;
}
// Rule 12 // กฎที่ 12 ถ้า x4 = PB แล้ว f4 = 1.0
if(((x4[j-1]/S4)>=0.0)&&((x4[j-1]/S4)<1.0) )
{
R3_h = (x4[j-1]/S4);
R3_c = 1.0;
}
else if((x4[j-1]/S4)>=1.0)
{
R3_h = 1.0;
R3_c = 1.0;
}
f4 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมาณพีชชีสำหรับ SIRM-4
F[j] = ((f1*uk1*10.0)+(f2*uk2*10.0)+(f3*uk3*10.0)+(f4*uk4*10.0))*osf;
// แรงขับเคลื่อนที่กระทำกับตัวรถ
xx2[j] = (((Mc+Mp)*g*sin(x1[j-1]))-(F[j]+(Mp*Lp*x2[j-1]*x2[j-1]*sin(x1[j-1])))*cos(x1[j-1]))/
(((4.0/3.0)*(Mc+Mp)- (Mp*cos(x1[j-1])*cos(x1[j-1])))*Lp);
// ความเร่งเชิงมุมของลูกตุ้ม
xx4[j] = ((4.0/3.0)*(F[j]+Mp*Lp*x2[j-1]*x2[j-1]*sin(x1[j-1]))-(Mp*g*sin(x1[j-1])*cos(x1[j-1]))/
((4.0/3.0)*(Mc+Mp)-(Mp*cos(x1[j-1])*cos(x1[j-1]))));
// ความเร่งของตัวรถ
// การอินทิเกรตอาศัยการประมาณค่าด้วยสี่เหลี่ยมคางหมู (trapezoidal integration)
x2[j] = x2[j-1]+(xx2[j-1]+xx2[j])*dt/2; // ความเร็วเชิงมุมของลูกตุ้ม
x1[j] = x1[j-1]+(x2[j-1]+x2[j])*dt/2; // มุมของลูกตุ้ม
x4[j] = x4[j-1]+(xx4[j-1]+xx4[j])*dt/2; // ความเร็วของตัวรถ
x3[j] = x3[j-1]+(x4[j-1]+x4[j])*dt/2; // ตำแหน่งของตัวรถ
*y_1 = *y_1+(fabs(x1[j])); // ผลรวมค่าสัมบูรณ์มุมของลูกตุ้ม
*y_2 = *y_2+(fabs(x2[j])); // ผลรวมค่าสัมบูรณ์ความเร็วเชิงมุมของลูกตุ้ม
*y_3 = *y_3+(fabs(x3[j])); // ผลรวมค่าสัมบูรณ์ตำแหน่งของตัวรถ
*y_4 = *y_4+(fabs(x4[j])); // ผลรวมค่าสัมบูรณ์ความเร็วของตัวรถ
}
//สิ้นสุดการจำลองระบบ
*y_1 = *y_1/1000.0; // ลดระดับค่าความคลาดเคลื่อนสำหรับมุมของลูกตุ้ม
*y_2 = *y_2/1000.0; // ลดระดับค่าความคลาดเคลื่อนสำหรับความเร็วเชิงมุมของลูกตุ้ม
*y_3 = *y_3/1000.0; // ลดระดับค่าความคลาดเคลื่อนสำหรับตำแหน่งของตัวรถ
*y_4 = *y_4/1000.0; // ลดระดับค่าความคลาดเคลื่อนสำหรับความเร็วของตัวรถ

```

```

} // สิ้นสุดการทำงานของ function1

/*-----*/

void function2(double D1,double D2,double D3,double D4,double *sta_t,double *max_F,double *max_x1,double *max_x2,
              double *max_x3,double *max_x4) // รับค่าระดับความสำคัญพลวัตที่ได้จากโครงข่ายประสาทเทียมเพื่อจำลอง
// สถานการณ์การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันแบบข้อต่อเดียว
{ // เริ่มต้น function2
int j; // ตัวนับการวนรอบในการจำลองระบบ
int ii; // ตัวนับสำหรับการคำนวณเวลาเข้าสู่เสถียรภาพของระบบ
double time_sta[4]; // เวลาเข้าสู่เสถียรภาพสำหรับมุมของลูกตุ้ม ความเร็วเชิงมุมของลูกตุ้ม
// ตำแหน่งของตัวรถ และความเร็วยของตัวรถ
double Mc,Mp,g,dt,S1,S2,S3,S4,osf; // ตัวแปรของระบบลูกตุ้มผกผันแบบข้อต่อเดียวที่ใช้ในการจำลองระบบ
double F[pot]; // แรงขับเคลื่อนที่กระทำกับตัวรถในแนวระดับ
double x1[pot],x2[pot],xx2[pot]; // มุม ความเร็วเชิงมุม และความเร่งเชิงมุมของลูกตุ้ม (ตามลำดับ)
double x3[pot],x4[pot],xx4[pot]; // ตำแหน่ง ความเร็ว และความเร่งของตัวรถ (ตามลำดับ)
double f1,f2,f3,f4; // ผลการอนุมานฟัซซี่
double R1_h,R2_h,R3_h,R1_c,R2_c,R3_c; // ค่าระดับความเป็นสมาชิกของในกฎฟัซซี่
Mc = 1.0; // มวลของตัวรถ (กิโลกรัม)
Mp = 0.1; // มวลของก้านลูกตุ้ม (กิโลกรัม)
g = 9.8; // ค่าแรงโน้มถ่วงของโลก
dt = 0.01; // ช่วงเวลาแซมปลิง (วินาที)
S1 = 30.0*pi/180.0; // ตัวปรับค่าสำหรับมุมของลูกตุ้ม
S2 = 100.0*pi/180.0; // ตัวปรับค่าสำหรับความเร็วเชิงมุมของลูกตุ้ม
S3 = 2.4; // ตัวปรับค่าสำหรับตำแหน่งของตัวรถ
S4 = 1.0; // ตัวปรับค่าสำหรับความเร็วของตัวรถ
osf = 10.0*(Mc+Mp); // ตัวปรับค่าสำหรับแรงขับเคลื่อน
F[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับแรงขับเคลื่อน
x1[0] = ang_ini*pi/180.0; // กำหนดค่ามุมเริ่มต้นของลูกตุ้ม
x2[0] = 0.0*pi/180.0; // กำหนดค่าความเร็วเชิงมุมเริ่มต้นของลูกตุ้ม
x3[0] = pos_ini; // กำหนดค่าตำแหน่งเริ่มต้นของตัวรถ
x4[0] = 0.0; // กำหนดค่าความเร็วเริ่มต้นของตัวรถ
xx2[0] = 0.0; // กำหนดค่าความเร่งเชิงมุมเริ่มต้นของลูกตุ้ม
xx4[0] = 0.0; // กำหนดค่าความเร่งเริ่มต้นของตัวรถ
*max_F = fabs(F[0]); // กำหนดแรงขับเคลื่อนสูงสุดเริ่มต้นที่กระทำกับตัวรถ
*max_x1 = 0.0; // กำหนดค่าเริ่มต้นสำหรับมุมมากที่สุดของลูกตุ้ม
*max_x2 = fabs(x2[0]); // กำหนดค่าเริ่มต้นสำหรับความเร็วเชิงมุมมากที่สุดของลูกตุ้ม
*max_x3 = fabs(x3[0]); // กำหนดค่าเริ่มต้นสำหรับตำแหน่งมากที่สุดของตัวรถ
*max_x4 = fabs(x4[0]); // กำหนดค่าเริ่มต้นสำหรับความเร็วมากที่สุดของตัวรถ
for(j=1;j<pot;j++) // เริ่มต้นการจำลองระบบ
{
...

```

```

// การกำหนดกฎพีชคณิตตั้งแต่กฎที่ 1 ถึงกฎที่ 12 การคำนวณแรงขับเคลื่อน
// การได้มาของความเร่งเชิงมุมของลูกตุ้มและความเร่งของตัวรถมีรูปแบบ
// การจำลองระบบเช่นเดียวกับ function1

...

// การอินทิเกรตอาศัยการประมาณค่าด้วยสี่เหลี่ยมคางหมู (trapezoidal integration)
x2[j] = x2[j-1]+(xx2[j-1]+xx2[j])*dt/2; // ความเร็วเชิงมุมของลูกตุ้ม
x1[j] = x1[j-1]+(x2[j-1]+x2[j])*dt/2; // มุมของลูกตุ้ม
x4[j] = x4[j-1]+(xx4[j-1]+xx4[j])*dt/2; // ความเร็วของตัวรถ
x3[j] = x3[j-1]+(x4[j-1]+x4[j])*dt/2; // ตำแหน่งของตัวรถ
if(fabs(F[j])>(*max_F))
{ *max_F = fabs(F[j]); } // ค่ามากที่สุดสำหรับแรงขับเคลื่อนที่กระทำต่อตัวรถในแนวระดับ
if(fabs(x2[j])>(*max_x2))
{ *max_x2 = fabs(x2[j]); } // ค่ามากที่สุดสำหรับความเร็วเชิงมุมของลูกตุ้ม
if(fabs(x3[j])>(*max_x3))
{ *max_x3 = fabs(x3[j]); } // ค่ามากที่สุดสำหรับตำแหน่งของตัวรถ
if(fabs(x4[j])>(*max_x4))
{ *max_x4 = fabs(x4[j]); } // ค่ามากที่สุดสำหรับความเร็วของตัวรถ
} // สิ้นสุดการจำลองระบบ
// ตรวจสอบเวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว

ii = pot-1;
while( (fabs(x1[ii])<=0.1*pi/180.0)&&(ii!=0)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับมุมของลูกตุ้ม
{ ii = ii-1; }
if(ii<pot-1)
{ time_sta[0] = (ii+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับมุมของลูกตุ้ม
else
{ time_sta[0] = 99.0; } // กรณีมุมของลูกตุ้มไม่เข้าสู่เสถียรภาพตามเงื่อนไขที่กำหนด
ii = pot-1;
while((fabs(x2[ii])<=0.1*pi/180.0)&&(ii!=0)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับความเร็วเชิงมุมของลูกตุ้ม
{ ii = ii-1; }
if(ii<pot-1)
{ time_sta[1] = (ii+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับความเร็วเชิงมุมของลูกตุ้ม
else
{ time_sta[1] = 99.0; } // กรณีความเร็วเชิงมุมของลูกตุ้มไม่เข้าสู่เสถียรภาพตามเงื่อนไขที่กำหนด
ii = pot-1;
while((fabs(x3[ii])<=0.01)&&(ii!=0)&&(fabs(*max_x3)<=2.4)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับตำแหน่งของตัวรถ
{ ii = ii-1; }
if(ii<pot-1)
{ time_sta[2] = (ii+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับตำแหน่งของตัวรถ
else
{ time_sta[2] = 99.0; } // กรณีตำแหน่งของตัวรถไม่เข้าสู่เสถียรภาพตามเงื่อนไขที่กำหนด
ii = pot-1;

```

```

while((fabs(x4[ii])<=0.01)&&(ii!=0)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับความเร็วของตัวรถ
{ ii = ii-1; }
if(ii<pot-1)
{ time_sta[3] = (ii+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับความเร็วของตัวรถ
else
{ time_sta[3] = 99.0; } // กรณีความเร็วของตัวรถไม่เข้าสู่เสถียรภาพตามเงื่อนไขที่กำหนด
*sta_t = 0.0;
for(ii=0;ii<4;ii++)
{
if(time_sta[ii]>(*sta_t))
{ *sta_t = time_sta[ii]; } // เวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว
}
ii = 0;
while (fabs(x1[ii])>=5.0*pi/180.0)
{ ii = ii+1; }
for (j=ii;j<pi;j++)
{
if(fabs(x1[j])> *max_x1)
{ *max_x1 = fabs(x1[j]); } // ค่ามากที่สุดสำหรับมุมของลูกตุ้ม
}
} // สิ้นสุดการทำงานของ function2
/*-----*/

```



ภาคผนวก ซ

โปรแกรมภาษาซีสำหรับตัวควบคุมนิวโร-ตาบู-พีซซี

```

*****
โปรแกรมสำหรับสร้างเสถียรภาพให้ระบบลูกค้อนหมุนแบบข้อต่อเดียวด้วยตัวควบคุมนิวโร-ดาบู-ฟิชซี่
โดย นางสาวสุภารัตน์ ขวัญอ่อน สาขาวิชาวิศวกรรมไฟฟ้า, มหาวิทยาลัยเทคโนโลยีสุรนารี, พ.ศ. 2547
*****

โปรแกรม single_atc.cpp (ค้นหาพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม)

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define pi      3.14159           // ค่าคงที่  $\pi = 3.14159$ 
#define para    32               // กำหนดจำนวนพารามิเตอร์ที่ต้องการค้นหา
#define row     5                // กำหนดจำนวนการสุ่มคำตอบเริ่มต้นเท่ากับ 5 ชุด
#define n_ne    40               // กำหนดจำนวนคำตอบรอบข้างการค้นหาเพื่อการสุ่มเท่ากับ 40
#define error_stop 0.32         // กำหนดค่าฟังก์ชันวัตถุประสงค์เพื่อยุติการค้นหาคำตอบ
#define num     500              // กำหนดจำนวนจุดข้อมูลในการจำลองระบบ
#define ang_1   30.0             // กำหนดมุมเริ่มต้นของลูกค้อน (องศา)
#define pos_1   0.0              // กำหนดตำแหน่งเริ่มต้นของตัวรถ (เมตร)
#define Lp      0.50             // กำหนดความยาวครึ่งหนึ่งของก้านลูกค้อน (เมตร)

double function1(double s0_ini_0,double s0_ini_1,double s0_ini_2,double s0_ini_3, double s0_ini_4,double s0_ini_5,
double s0_ini_6,double s0_ini_7,double s0_ini_8,double s0_ini_9,double s0_ini_10,double s0_ini_11,
double s0_ini_12,double s0_ini_13,double s0_ini_14,double s0_ini_15,double s0_ini_16,double s0_ini_17,
double s0_ini_18,double s0_ini_19,double s0_ini_20,double s0_ini_21,double s0_ini_22,double s0_ini_23,
double s0_ini_24,double s0_ini_25,double s0_ini_26,double s0_ini_27,double s0_ini_28,double s0_ini_29,
double s0_ini_30,double s0_ini_31); // function1 รับค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมทั้ง 32 ค่า

double function2(double DID1,double DID2,double DID3,double DID4);
// function2 รับค่าระดับความสำคัญพลวัตจากโครงข่ายประสาทเทียม

main()
// เริ่มต้นโปรแกรมหลัก
{
clrscr();

clock_t start,end; // ประกาศตัวแปรเพื่อคำนวณเวลาการค้นหาคำตอบ
int st_ini, st; // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int i,j,ct,ct_0,o,i_ini,j_ini,ct_ini,ct_0_ini,count; // ตัวแปรสำหรับการนับค่าและการวนรอบ
int n; // ตัวนับจำนวนการสุ่มคำตอบ
int n_back; // ตัวนับจำนวนการใช้กลไกย้อนรอยการค้นหา
int ct_max; // จำนวนรอบการค้นหาสูงสุด
double s0_ini[row][para]; // ค่าพารามิเตอร์เริ่มต้นจำนวน 5 ชุดๆ ละ 32 ค่า
double rd,rd_0,div; // ตัวแปรเกี่ยวกับการสุ่ม
double x_l,x_h; // ขอบเขตล่างและขอบเขตบนในการค้นหาคำตอบ
double f_min_ini,f_min_best_ini[1]; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
double f_ini; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นในแต่ละรอบการค้นหา

```

```

double s0_best_ini[1][para]; // ค่าพารามิเตอร์ของ f_min_best_ini
double s; // ตัวแปรชั่วคราวทำหน้าที่รับส่งค่าพารามิเตอร์
double a,b; // ขอบของค่าสุ่ม โดยที่ a = -1.0 (ขอบล่าง) และ b = 1.0 (ขอบบน)
double c; // ค่าที่สุ่มได้จากคำสั่ง rand มีค่าเป็นบวกสูงสุดเท่ากับ 32767
double r0; // ค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1 ซึ่ง r0 = rd/div
double r; // ค่าสุ่ม r = a+[r0*(b-a)]
double radius; // รัศมีการค้นหาค่าตอบ
double f_min0[1]; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดมีค่าเท่ากับ f_min_best_ini
double s0[1][para]; // ค่าพารามิเตอร์ของ f_min0 มีค่าเท่ากับ s0_best_ini
double s1[n_ne][para]; // ค่าพารามิเตอร์ของคำตอบรอบข้างในแต่ละรอบการค้นหา
double del[1][para]; // ช่วงขอบเขตล่างถึงขอบเขตบนของค่าพารามิเตอร์แต่ละตัว
double f_min1_old; // ค่าฟังก์ชันวัตถุประสงค์เพื่อเปรียบเทียบค่าที่ดีกว่า
double f_min1[n_ne]; // ค่าตอบทั้งหมดในแต่ละรอบการค้นหา
double f_min1_best[1]; // ค่าตอบที่ดีที่สุดในแต่ละรอบการค้นหา
double s1_best[1][para]; // ค่าพารามิเตอร์ของคำตอบที่ดีที่สุดในแต่ละรอบการค้นหา
double f_min0_old; // ค่าตอบที่ดีที่สุดในอดีต
double f_min0_new; // ค่าตอบที่ดีที่สุดในปัจจุบัน
double local_f[1]; // ค่าตอบวงแคบเฉพาะถิ่น
double local_ne[1][para]; // ค่าพารามิเตอร์ของคำตอบวงแคบเฉพาะถิ่น
double global_f[1]; // ค่าตอบโกลด์วงกว้าง
double global_ne[1][para]; // ค่าพารามิเตอร์ของคำตอบโกลด์วงกว้าง
double best_error; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด
double best_neighbor[para]; // ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการค้นหา
FILE *best_s; // ตั้งชื่อเพิ่มข้อมูลเพื่อเก็บค่าพารามิเตอร์เริ่มต้นจากการค้นหาว่า best_s

/*-----สุ่มค่าคำตอบเริ่มต้น-----*/
ct = 0;
count = 0;
ct_0_ini = 0;
div = 32767.0; // ตัวหารค่าที่สุ่มได้เพื่อทำให้มีค่าสูงสุดเท่ากับ 1
x_l = -1.0; // ขอบเขตล่างของค่าพารามิเตอร์ที่ทำการค้นหา
x_h = 1.0; // ขอบเขตบนของค่าพารามิเตอร์ที่ทำการค้นหา
f_min_ini = pow(10.0,10.0); // กำหนดค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุดให้มีค่ามากๆ
printf("Enter seed of random number = ");
scanf("%d",&st_ini); // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด
srand(st_ini); // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
for(i_ini=0;i_ini<row;i_ini++)
{
    for(j_ini=0;j_ini<para;j_ini++)
    {
        rd = rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767

```

```

rd_0 = rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s = ((x_h-x_l)*rd_0)+x_l; // ค่า s มีขอบเขตจาก -1 ถึง +1
s0_ini[i_ini][j_ini] = s; // ค่าพารามิเตอร์ที่สุ่มได้ทั้ง 5 ชุดๆ ละ 32 คำตอบ
}
}
for(i_ini=0;i_ini<row;i_ini++)
{
f_ini = function1(s0_ini[i_ini][0],s0_ini[i_ini][1],s0_ini[i_ini][2],s0_ini[i_ini][3],s0_ini[i_ini][4],s0_ini[i_ini][5],
s0_ini[i_ini][6],s0_ini[i_ini][7],s0_ini[i_ini][8],s0_ini[i_ini][9],s0_ini[i_ini][10],s0_ini[i_ini][11],
s0_ini[i_ini][12],s0_ini[i_ini][13],s0_ini[i_ini][14],s0_ini[i_ini][15],s0_ini[i_ini][16],s0_ini[i_ini][17],
s0_ini[i_ini][18],s0_ini[i_ini][19],s0_ini[i_ini][20],s0_ini[i_ini][21],s0_ini[i_ini][22],s0_ini[i_ini][23],
s0_ini[i_ini][24],s0_ini[i_ini][25],s0_ini[i_ini][26],s0_ini[i_ini][27],s0_ini[i_ini][28],s0_ini[i_ini][29],
s0_ini[i_ini][30],s0_ini[i_ini][31]); // function1 รับค่าพารามิเตอร์ที่สุ่มได้และส่งกลับค่าฟังก์ชันวัตถุประสงค์

if(f_ini<=f_min_ini) // ประเมินค่าฟังก์ชันวัตถุประสงค์สำหรับค่าพารามิเตอร์แต่ละชุด
{
f_min_ini = f_ini; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด
ct_ini = ct_0_ini; // ตัวนับชุดของพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์ที่ดีขึ้น
}
ct_0_ini = ct_0_ini+1;
}
f_min_best_ini[0] = f_min_ini; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด
for(j=0;j<para;j++)
{
s0_best_ini[0][j] = s0_ini[ct_ini][j]; // ค่าเริ่มต้นของคำตอบการค้นหาที่ดีที่สุด
del[0][j] = x_h-x_l; // ช่วงของขอบเขตการค้นหาคำตอบ
s0[0][j] = s0_best_ini[0][j]; // เก็บค่าเริ่มต้นของคำตอบการค้นหาที่ดีที่สุดไว้ใน s0
local_ne[0][j] = s0[0][j]; // เก็บค่าเริ่มต้นที่เป็นคำตอบวงแคบเฉพาะถิ่นไว้ใน local_ne
global_ne[0][j] = local_ne[0][j]; // เก็บค่าเริ่มต้นที่เป็นคำตอบโกล่วงกว้างไว้ใน global_ne
best_neighbor[j] = s0[0][j]; // เก็บคำตอบเริ่มต้นในการค้นหาที่ดีที่สุดไว้ใน best_neighbor
}
f_min0[0] = f_min_best_ini[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดไว้ใน f_min0
f_min0_old = f_min0[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดเพื่อเป็นอดีตไว้ใน f_min0_old
local_f[0] = f_min0[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่เป็นคำตอบวงแคบเฉพาะถิ่น
global_f[0] = local_f[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่เป็นคำตอบโกล่วงกว้าง
best_error = f_min0[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดไว้ใน best_error
n_back = 0; // กำหนดตัวนับจำนวนการใช้กลไกการย้อนรอยเป็น 0
radius = 0.2; // กำหนดรัศมีการค้นหาเริ่มต้นเป็น 0.2
ct_0 = 0; // กำหนดตัวนับชุดของพารามิเตอร์ที่ค่าฟังก์ชันวัตถุประสงค์ลดลง
a = -1; // ขอบเขตล่างของค่าสุ่ม
b = 1; // ขอบเขตบนของค่าสุ่ม

```

```

n      = 0; // กำหนดตัวนับจำนวนการซ้ำค่าคำตอบเริ่มต้นเท่ากับ 0
printf("Enter seed of random number = ");
scanf("%d",&st); // รับค่าตัวเลขการสุ่มเริ่มต้นจากคีย์บอร์ด
srand(st); // เก็บค่าตัวเลขการสุ่มเริ่มต้นไว้ในค่าตั้ง srand
printf("Enter number of ct_max = ");
scanf("%d",&ct_max); // รับค่าจำนวนรอบสูงสุดในการค้นหาคำตอบจากคีย์บอร์ด
start = clock(); // เริ่มต้นคำนวณเวลาการค้นหาคำตอบ
loop2: for(o=0;o<=ct_max;o++) // เริ่มต้นการค้นหาคำตอบ
{
/*-----ปรับรัศมีการค้นหา-----*/
if(f_min0[0]>0.40)
{ radius = 0.2; }
else if((f_min0[0]>0.38)&&(f_min0[0]<=0.40))
{ radius = 0.15; }
else if((f_min0[0]>0.34)&&(f_min0[0]<=0.38))
{ radius = 0.1; }
else if(f_min0[0]<=0.34)
{ radius = 0.05; }
for(i=0;i<n_ne;i++) // สุ่มคำตอบรอบข้างจำนวน n_ne ชุด
{
for(j=0;j<para;j++) // แต่ละชุดของคำตอบรอบข้างประกอบด้วยคำตอบจำนวน 32 ค่า
{
c = rand(); // เก็บค่าที่สุ่มได้ซึ่งอยู่ในช่วง 0 ถึง 32767 ไว้ใน c
r0 = c/div; // ปรับค่าที่สุ่มได้ให้อยู่ในช่วง 0 ถึง 1
r = a+r0*(b-a); // ปรับค่าที่สุ่มได้ให้อยู่ในช่วง -1 ถึง 1
s1[i][j] = s0[0][j]+(r*radius*del[0][j]); // ค่าพารามิเตอร์ที่ได้จากการสุ่มค่า
if((s1[i][j]<x_l)||(s1[i][j]>x_h)) // กรณีค่าพารามิเตอร์จากการสุ่มเกินขอบเขตที่กำหนดทำการสุ่มค่าใหม่
{
while((s1[i][j]<x_l)||(s1[i][j]>x_h))
{
c = rand();
r0 = c/div;
r = a+r0*(b-a);
s1[i][j] = s0[0][j]+(r*radius*del[0][j]); // ค่าพารามิเตอร์ที่ได้จากการสุ่มซึ่งอยู่ในขอบเขตที่กำหนด
}
}
}
}
for(i=0;i<n_ne;i++)
{

```

```

f_min1[i] = function1(s1[i][0],s1[i][1],s1[i][2],s1[i][3],s1[i][4],s1[i][5],s1[i][6],s1[i][7],s1[i][8],s1[i][9],s1[i][10],s1[i][11],
                    s1[i][12],s1[i][13],s1[i][14],s1[i][15],s1[i][16],s1[i][17],s1[i][18],s1[i][19],s1[i][20],s1[i][21],s1[i][22],
                    s1[i][23],s1[i][24],s1[i][25],s1[i][26],s1[i][27],s1[i][28],s1[i][29],s1[i][30],s1[i][31]);
} // function1 ส่งกลับค่าฟังก์ชันวัตถุประสงค์ของชุดค่าตอบรอบข้างที่สุ่มได้
f_min1_old = f_min1[0]; // เก็บค่าเพื่อการเปรียบเทียบ
for(i=0;i<n_ne;i++)
{
    if(f_min1[i]<f_min1_old)
    {
        f_min1_old = f_min1[i]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดใน f_min1_old
        ct = ct_0; // นับจำนวนของค่าตอบรอบข้างที่ทำให้ฟังก์ชันวัตถุประสงค์ลดลงจากเดิม
    }
    ct_0 = ct_0+1;
}
f_min1_best[0] = f_min1_old; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดใน f_min1_best
for(j=0;j<para;j++)
{ s1_best[0][j] = s1[ct][j]; } // ค่าตอบการค้นหาที่ทำให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด
ct_0 = 0;
ct = 0;
f_min0_new = f_min1_best[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดไว้เป็นค่าปัจจุบัน
if(f_min0_new>=f_min0_old) // เปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในปัจจุบันและในอดีต
{ n = n+1; } // เพิ่มจำนวนการซ้ำค่าตอบเดิมในกรณีที่ค่าปัจจุบัน ไม่น้อยกว่าค่าในอดีต
else
{
    n = 0; // n = 0 ในกรณีที่ค่าฟังก์ชันวัตถุประสงค์ในปัจจุบันน้อยกว่าค่าในอดีต
    f_min0_old = f_min0_new; // และให้ค่าฟังก์ชันวัตถุประสงค์ในอดีตเท่ากับค่าวัตถุประสงค์ในปัจจุบัน
}
/*-----กลไกการย้อนรอย-----*/
if(n>=15) // ถ้าจำนวนการซ้ำค่าตอบเป็น 15 เรียกใช้กลไกการย้อนรอย
{
    local_f[0] = f_min0[0]; // เก็บค่า f_min0 ไว้ใน local_f
    for(j=0;j<para;j++)
    { local_ne[0][j] = s0[0][j]; } // เก็บค่า s0 ไว้ใน local_ne
    n_back = n_back+1; // เพิ่มจำนวนการใช้กลไกการย้อนรอย
    goto loop1; // กระโดดไปทำงานที่ลาเบล loop1
}
// ปรับค่าคำตอบและพารามิเตอร์
if(f_min1_best[0]<f_min0[0]) // เปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในแต่ละรอบการค้นหา
{
    f_min0[0] = f_min1_best[0]; // เก็บค่า f_min1_best ไว้ใน f_min0
}

```

```

    for(j=0;j<para;j++)
        { s0[0][j] = s1_best[0][j]; } // เก็บค่า s1_best ไว้ใน s0
    }
if(f_min0[0]<best_error) // เปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดเพื่อเก็บลงในเพิ่มข้อมูล
    {
        best_error = f_min0[0]; // เก็บค่า f_min0 ไว้ใน best_error
        for(j=0;j<para;j++)
            { best_neighbor[j] = s0[0][j]; } // เก็บค่า s0 ไว้ใน best_neighbor
    }

/*-----ตรวจสอบเงื่อนไขเพื่อยุติการค้นหา-----*/
if((count>=ct_max)||(f_min0[0]<error_stop))
    { goto loop3; } // กระโดดไปทำงานที่ลabeled loop3
printf("%d\t%.3f\t%.6f\n",count,radius,best_error); // แสดงค่า count, radius, และ best_error
count = count+1; // ปรับเพิ่มจำนวนรอบการค้นหา
} // สิ้นสุดการทำงานของลabeled loop2
loop1: if(local_f[0]<global_f[0]) // ถ้าค่าคอบวงแคบเฉพาะถิ่นน้อยกว่าค่าคอบกัลวงกว้าง
    {
        global_f[0] = local_f[0]; // เก็บค่า local_f ไว้ใน global_f
        for(j=0;j<para;j++)
            { global_ne[0][j] = local_ne[0][j]; } // เก็บค่า local_ne ไว้ใน global_ne
    }
if(global_f[0]<error_stop) // ตรวจสอบเงื่อนไขเพื่อยุติการค้นหาค่าคอบ
    { goto loop3; } // กระโดดไปทำงานที่ลabeled loop3 กรณีตรงตามเงื่อนไข
f_min0[0] = f_min1[1]; // เก็บค่าค่าคอบชุดที่ 2 ของ f_min1 ไว้ใน f_min0
for(j=0;j<para;j++)
    { s0[0][j] = s1[1][j]; } // เก็บค่า s1 ไว้ใน s0
f_min0_old = f_min0[0]; // เก็บค่า f_min0 ไว้ใน f_min0_old
n = 0; // ปรับค่าจำนวนการซ้ำค่าคอบเป็น 0
printf("%d\t%.3f\t%.6f ****back tracking****\n",count,radius,global_f[0]);
// แสดงข้อความ back tracking และ count, radius,global_f

if((count>=ct_max)||(f_min0[0]<error_stop)) // ตรวจสอบเงื่อนไขเพื่อยุติการค้นหา
    { goto loop3; } // กระโดดไปทำงานที่ลabeled loop3
count = count+1; // ปรับเพิ่มจำนวนรอบการค้นหา
goto loop2; // กระโดดไปทำงานที่ลabeled loop2
loop3: end = clock(); // สิ้นสุดการคำนวณเวลาในการค้นหา
best_s = fopen("G:/Data_ATS/n_para.dat","wb"); // เปิดเพิ่มข้อมูลเพื่อเก็บพารามิเตอร์ที่ค้นหาได้
fwrite(&best_neighbor,sizeof(double),para,best_s); // เขียนพารามิเตอร์ที่ค้นหาได้ลงในเพิ่มข้อมูล
fclose(best_s); // ปิดเพิ่มข้อมูล
printf("stoped count = %d\n",count); // แสดงจำนวนรอบที่พบค่าคอบ
printf("best_error =%.6f\n",best_error); // แสดงค่าวัตถุประสงค์ที่ดีที่สุด

```

```

printf("execution time=%f sec", (end-start)/CLK_TCK); // แสดงเวลาที่ใช้ในการค้นหา
getch();
} // สิ้นสุดการทำงานของโปรแกรมหลัก

/*-----*/

double function1(double s0_ini_0, double s0_ini_1, double s0_ini_2, double s0_ini_3, double s0_ini_4, double s0_ini_5,
                double s0_ini_6, double s0_ini_7, double s0_ini_8, double s0_ini_9, double s0_ini_10, double s0_ini_11,
                double s0_ini_12, double s0_ini_13, double s0_ini_14, double s0_ini_15, double s0_ini_16, double s0_ini_17,
                double s0_ini_18, double s0_ini_19, double s0_ini_20, double s0_ini_21, double s0_ini_22, double s0_ini_23,
                double s0_ini_24, double s0_ini_25, double s0_ini_26, double s0_ini_27, double s0_ini_28, double s0_ini_29,
                double s0_ini_30, double s0_ini_31) // รับค่าพารามิเตอร์จากการค้นหาและส่งกลับค่าฟังก์ชันวัตถุประสงค์

{
int    i; // ตัวแปรเพื่อการวนรอบ
int    l,m,n; // จำนวนเซลล์ประสาทเทียมในแต่ละชั้นของโครงข่ายประสาทเทียม
double S[para]; // พารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
double x[2]; // สัญญาณอินพุตที่ให้กับชั้นอินพุตของโครงข่ายประสาทเทียม
double sum; // ตัวแปรเพื่อหาผลรวมในกระบวนการเรียนรู้
double W[2][4][2]; // ค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
double W1[4][4][2]; // ค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
double Z1[4][2]; // ค่าไบแอสที่ชั้นซ่อนเร้น
double Z2[4][2]; // ค่าไบแอสที่ชั้นเอาต์พุต
double X1[4][2]; // เอาต์พุตของชั้นซ่อนเร้น
double X2[4][2]; // เอาต์พุตของชั้นเอาต์พุต
double uk1, uk2, uk3, uk4; // ค่าระดับความสำคัญพลวัต
double error_plant; // ค่าฟังก์ชันวัตถุประสงค์ที่ส่งกลับให้กับโปรแกรมหลัก
x[0] = fabs(ang_1)*pi/180.0; // มุมเริ่มต้นของลูกตุ้ม (เรเดียน)
x[1] = fabs(pos_1); // ตำแหน่งเริ่มต้นของตัวรอด (เมตร)
S[26] = s0_ini_26; S[27] = s0_ini_27; S[28] = s0_ini_28; S[29] = s0_ini_29; S[30] = s0_ini_30;
S[0] = s0_ini_0; S[1] = s0_ini_1; S[2] = s0_ini_2; S[3] = s0_ini_3; S[4] = s0_ini_4;
S[5] = s0_ini_5; S[6] = s0_ini_6; S[7] = s0_ini_7; S[8] = s0_ini_8; S[9] = s0_ini_9;
S[10] = s0_ini_10; S[11] = s0_ini_11; S[12] = s0_ini_12; S[13] = s0_ini_13; S[14] = s0_ini_14;
S[15] = s0_ini_15; S[16] = s0_ini_16; S[17] = s0_ini_17; S[18] = s0_ini_18; S[19] = s0_ini_19;
S[20] = s0_ini_20; S[21] = s0_ini_21; S[22] = s0_ini_22; S[23] = s0_ini_23; S[24] = s0_ini_24;
S[25] = s0_ini_25; S[26] = s0_ini_26; S[27] = s0_ini_27; S[28] = s0_ini_28; S[29] = s0_ini_29;
S[30] = s0_ini_30; S[31] = s0_ini_31; // เก็บค่า s0_ini ลงใน S

i = 0;
for (l=0;l<2;l++)
{
    for(m=0;m<4;m++)
    {
        W[l][m][1] = S[i]; // ค่าน้ำหนักระหว่างชั้นอินพุตไปยังชั้นซ่อนเร้น
    }
}

```



```

        i++;
    }
}
for (m=0;m<4;m++)
{
    Z1[m][1] = S[i];           // ค่าไบแอสที่ชั้นซ่อนเร้น
    i++;
}
for (m=0;m<4;m++)
{
    for (n=0;n<4;n++)
    {
        W1[m][n][1] = S[i];   // ค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
        i++;
    }
}
for (n=0;n<4;n++)
{
    Z2[n][1] = S[i];          // ค่าไบแอสที่ชั้นเอาต์พุต
    i++;
}
i = 2;                        // จำนวนเอาต์พุตของโครงข่ายประสาทเทียม
for(m=0;m<4;m++)
{
    sum = 0.0;
    for(l=0;l<2;l++)
    { sum = sum+W[l][m][i-1]*x[l]; }
    sum = sum+Z1[m][i-1];
    X1[m][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตจากชั้นซ่อนเร้น (X1j)
}
for(n=0;n<4;n++)
{
    sum = 0.0;
    for(m=0;m<4;m++)
    { sum = sum+W1[m][n][i-1]*X1[m][i-1]; }
    sum = sum+Z2[n][i-1];
    X2[n][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตจากชั้นเอาต์พุต (X2k)
}
uk1 = X2[0][i-1];           // ค่าระดับความสำคัญพลวัตสำหรับมุมของลูกตุ้ม
uk2 = X2[1][i-1];           // ค่าระดับความสำคัญพลวัตสำหรับความเร็วเชิงมุมของลูกตุ้ม
uk3 = X2[2][i-1];           // ค่าระดับความสำคัญพลวัตสำหรับตำแหน่งของคัตวรถ

```

```

uk4 = X2[3][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับความเร็วของตัวรถ
error_plant = function2(uk1,uk2,uk3,uk4); // ส่งค่าระดับความสำคัญพลวัตให้กับ function2
return(error_plant); // ส่งกลับค่าฟังก์ชันวัตถุประสงค์ให้โปรแกรมหลัก
} // สิ้นสุดการทำงานของ function1

/*-----*/

double function2(double DID1,double DID2,double DID3,double DID4)
// function2 รับค่าระดับความสำคัญพลวัต

{
int j,ii; // ตัวแปรเพื่อการนับค่าและการวนรอบ
double Mc,Mp,g,dt,S1,S2,S3,S4,osf; // ตัวแปรของระบบลูกคัมผกผันแบบข้อต่อเดียวที่ใช้ในการจำลองระบบ
double F[num]; // แรงขับเคลื่อนที่กระทำต่อตัวรถในแนวระดับ
double x1[num]; // มุมของลูกคัม (เรเดียน)
double x2[num]; // ความเร็วเชิงมุมของลูกคัม (เรเดียน/วินาที)
double x3[num]; // ตำแหน่งของตัวรถ (เมตร)
double x4[num]; // ความเร็วของตัวรถ (เมตร/วินาที)
double xx2[num]; // ความเร่งเชิงมุมของลูกคัม (เรเดียน/วินาที2)
double xx4[num]; // ความเร่งของตัวรถ (เมตร/วินาที2)
double f1,f2,f3,f4; // ผลการอนุมานฟัซซี่ด้วยวิธี simplified fuzzy reasoning method
double R1_h,R2_h,R3_h,R1_c,R2_c,R3_c; // ค่าระดับความเป็นสมาชิกในกฎฟัซซี่
double error_cost; // ค่าฟังก์ชันวัตถุประสงค์ที่ส่งกลับให้ function1
double sum_x1; // ผลรวมมุมของลูกคัมทุกจุดข้อมูล
double sum_x2; // ผลรวมความเร็วเชิงมุมของลูกคัมทุกจุดข้อมูล
double sum_x3; // ผลรวมตำแหน่งของตัวรถทุกจุดข้อมูล
double sum_x4; // ผลรวมตำแหน่งของตัวรถทุกจุดข้อมูล
double sta_t; // เวลาเข้าสู่สเถียรภาพของระบบลูกคัมผกผันแบบข้อต่อเดียว
double max_x3; // ตำแหน่งของตัวรถที่มากที่สุด
double max_F; // แรงขับเคลื่อนที่มากที่สุด
double os_ang; // มุมของลูกคัมที่มากที่สุด
double time_sta[4]; // เวลาเข้าสู่สเถียรภาพสำหรับมุมของลูกคัม ความเร็วเชิงมุมของลูกคัม
// ตำแหน่งของตัวรถ และความเร็วของตัวรถ

Mc = 1.0; // มวลของตัวรถ (กิโลกรัม)
Mp = 0.1; // มวลของก้านลูกคัม (กิโลกรัม)
g = 9.8; // ค่าแรงโน้มถ่วงของโลก
dt = 0.01; // ช่วงเวลาแซมปลิง
S1 = 30.0*pi/180.0; // ตัวปรับค่าสำหรับมุมของลูกคัม
S2 = 100.0*pi/180.0; // ตัวปรับค่าสำหรับความเร็วเชิงมุมของลูกคัม
S3 = 2.4; // ตัวปรับค่าสำหรับตำแหน่งของตัวรถ
S4 = 1.0; // ตัวปรับค่าสำหรับความเร็วของตัวรถ
osf = 10.0*(Mc+Mp); // ตัวปรับค่าสำหรับแรงขับเคลื่อน
F[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับแรงขับเคลื่อน

```

```

x1[0] = ang_ini*pi/180.0; // กำหนดค่ามุมเริ่มต้นของลูกตุ้ม
x2[0] = 0.0*pi/180.0; // กำหนดค่าความเร็วเชิงมุมเริ่มต้นของลูกตุ้ม
x3[0] = pos_ini; // กำหนดค่าตำแหน่งเริ่มต้นของตัวรถ
x4[0] = 0.0; // กำหนดค่าความเร็วเริ่มต้นของตัวรถ
xx2[0] = 0.0; // กำหนดค่าความเร่งเชิงมุมเริ่มต้นของลูกตุ้ม
xx4[0] = 0.0; // กำหนดค่าความเร่งเริ่มต้นของตัวรถ
sum_x1 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมมุมของลูกตุ้ม
sum_x2 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมความเร็วเชิงมุมของลูกตุ้ม
sum_x3 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมตำแหน่งของตัวรถ
sum_x4 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมความเร็วของตัวรถ
max_x3 = 0.0; // กำหนดค่าเริ่มต้นให้กับตำแหน่งของตัวรถที่มากที่สุด
max_F = 0.0; // กำหนดค่าเริ่มต้นให้กับแรงขับเคลื่อนที่มากที่สุด
os_ang = 0.0; // กำหนดค่าเริ่มต้นให้กับมุมของลูกตุ้มที่มากที่สุด
for(j=1;j<num;j++) // เริ่มต้นการจำลองระบบ
{
/*-----SIRM-1-----*/ // กำหนดกฎฟuzzyแบบกฎอินพุตเดียวสำหรับมุมของลูกตุ้ม
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต
// Rule 1 // กฎที่ 1 ถ้า x1 = NB แล้ว f1 = -1.0
if((x1[j-1]/S1)<-1.0)
{
R1_h = 1.0;
R1_c = -1.0;
}
else if(((x1[j-1]/S1)>=-1.0)&&((x1[j-1]/S1)<0.0) )
{
R1_h = fabs(x1[j-1]/S1);
R1_c = -1.0;
}
// Rule 2 // กฎที่ 2 ถ้า x1 = ZO แล้ว f1 = 0.0
if(((x1[j-1]/S1)>=-1.0)&&((x1[j-1]/S1)<0.0) )
{
R2_h = (x1[j-1]/S1)+1.0;
R2_c = 0.0;
}
else if(((x1[j-1]/S1)>=0.0)&&((x1[j-1]/S1)<1.0) )
{
R2_h = 1.0-(x1[j-1]/S1);
R2_c = 0.0;
}
}

```

```

// Rule 3 // กฎที่ 3 ถ้า x1 = PB แล้ว f1 = 1.0
if(((x1[j-1]/S1)>=0.0)&&((x1[j-1]/S1)<1.0) )
{
    R3_h = (x1[j-1]/S1);
    R3_c = 1.0;
}
else if((x1[j-1]/S1)>=1.0)
{
    R3_h = 1.0;
    R3_c = 1.0;
}
f1 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-1

/*-----SIRM-2-----*/ // กำหนดกฎฟuzzyแบบกฎอินพุตเดียวสำหรับความเร็วเชิงมุมของลูกตุ้ม
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต
// Rule 4 // กฎที่ 4 ถ้า x2 = NB แล้ว f2 = -1.0
if((x2[j-1]/S2)<-1.0)
{
    R1_h = 1.0;
    R1_c = -1.0;
}
else if(((x2[j-1]/S2)>=-1.0)&&((x2[j-1]/S2)<0.0) )
{
    R1_h = fabs(x2[j-1]/S2);
    R1_c = -1.0;
}
// Rule 5 // กฎที่ 5 ถ้า x2 = ZO แล้ว f2 = 0.0
if(((x2[j-1]/S2)>=-1.0)&&((x2[j-1]/S2)<0.0) )
{
    R2_h = (x2[j-1]/S2)+1.0;
    R2_c = 0.0;
}
else if(((x2[j-1]/S2)>=0.0)&&((x2[j-1]/S2)<1.0) )
{
    R2_h = 1.0-(x2[j-1]/S2);
    R2_c = 0.0;
}
// Rule 6 // กฎที่ 6 ถ้า x2 = PB แล้ว f2 = 1.0
if(((x2[j-1]/S2)>=0.0)&&((x2[j-1]/S2)<1.0) )

```

```

{
    R3_h = (x2[j-1]/S2);
    R3_c = 1.0;
}
else if((x2[j-1]/S2)>=1.0)
{
    R3_h = 1.0;
    R3_c = 1.0;
}
f2 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-2

/*-----SIRM-3-----*/
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดกฎฟuzzyแบบกฎอินพุตเดียวสำหรับตำแหน่งของตัวรถ
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต
// Rule 7 // กฎที่ 7 ถ้า x3 = NB แล้ว f3 = -1.0
if((x3[j-1]/S3)<-1.0)
{
    R1_h = 1.0;
    R1_c = -1.0;
}
else if(((x3[j-1]/S3)>=-1.0)&&((x3[j-1]/S3)<0.0) )
{
    R1_h = fabs(x3[j-1]/S3);
    R1_c = -1.0;
}
// Rule 8 // กฎที่ 8 ถ้า x3 = ZO แล้ว f3 = 1.0
if(((x3[j-1]/S3)>=-1.0)&&((x3[j-1]/S3)<0.0) )
{
    R2_h = (x3[j-1]/S3)+1.0;
    R2_c = 0.0;
}
else if(((x3[j-1]/S3)>=0.0)&&((x3[j-1]/S3)<1.0) )
{
    R2_h = 1.0-(x3[j-1]/S3);
    R2_c = 0.0;
}
// Rule 9 // กฎที่ 9 ถ้า x3 = PB แล้ว f3 = 1.0
if(((x3[j-1]/S3)>=0.0)&&((x3[j-1]/S3)<1.0) )
{
    R3_h = (x3[j-1]/S3);

```

```

    R3_c = 1.0;
}
else if((x3[j-1]/S3)>=1.0)
{
    R3_h = 1.0;
    R3_c = 1.0;
}
f3 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-3

/*-----SIRM-4-----*/
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดกฎฟuzzyแบบกฎอินพุตเดียวสำหรับความเร็วของตัวรถ
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต
// Rule 10 // กำหนดค่าเริ่มต้นให้กับระดับความเป็นสมาชิกของตัวแปรเอาต์พุต
if((x4[j-1]/S4)<-1.0) // กฎที่ 10 ถ้า x4 = NB แล้ว f4 = -1.0
{
    R1_h = 1.0;
    R1_c = -1.0;
}
else if(((x4[j-1]/S4)>=-1.0)&&((x4[j-1]/S4)<0.0) )
{
    R1_h = fabs(x4[j-1]/S4);
    R1_c = -1.0;
}
// Rule 11 // กฎที่ 11 ถ้า x4 = ZO แล้ว f4 = 1.0
if(((x4[j-1]/S4)>=-1.0)&&((x4[j-1]/S4)<0.0) )
{
    R2_h = (x4[j-1]/S4)+1.0;
    R2_c = 0.0;
}
else if(((x4[j-1]/S4)>=0.0)&&((x4[j-1]/S4)<1.0) )
{
    R2_h = 1.0-(x4[j-1]/S4);
    R2_c = 0.0;
}
// Rule 12 // กฎที่ 12 ถ้า x4 = PB แล้ว f4 = 1.0
if(((x4[j-1]/S4)>=0.0)&&((x4[j-1]/S4)<1.0) )
{
    R3_h = (x4[j-1]/S4);
    R3_c = 1.0;
}

```

```

else if((x4[j-1]/S4)>=1.0)
{
    R3_h = 1.0;
    R3_c = 1.0;
}
f4 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมาณพีชชีสำหรับ SIRM-4
F[i]=((f1*DID1*10.0)+(f2*DID2*10.0)+(f3*DID3*10.0)+(f4*DID4*10.0))*osf;
// แรงขับเคลื่อนที่กระทำกับตัวรถ
xx2[j] = (((Mc+Mp)*g*sin(x1[j-1]))-(F[j]+(Mp*Lp*x2[j-1]*x2[j-1]*sin(x1[j-1])))*cos(x1[j-1]))/
(((4.0/3.0)*(Mc+Mp)-(Mp*cos(x1[j-1])*cos(x1[j-1])))*Lp);
// ความเร่งเชิงมุมของลูกตุ้ม
xx4[j] = ((4.0/3.0)*(F[j]+Mp*Lp*x2[j-1]*x2[j-1]*sin(x1[j-1]))-(Mp*g*sin(x1[j-1])*cos(x1[j-1]))/
((4.0/3.0)*(Mc+Mp)-(Mp*cos(x1[j-1])*cos(x1[j-1]))));
// ความเร่งของตัวรถ

/*-----การอินทิเกรตอาศัยการประมาณค่าด้วยสี่เหลี่ยมคางหมู (trapezoidal integration)-----*/
x2[j] = x2[j-1]+(xx2[j-1]+xx2[j])*dt/2; // ความเร็วเชิงมุมของลูกตุ้ม
x1[j] = x1[j-1]+(x2[j-1]+x2[j])*dt/2; // มุมของลูกตุ้ม
x4[j] = x4[j-1]+(xx4[j-1]+xx4[j])*dt/2; // ความเร็วของตัวรถ
x3[j] = x3[j-1]+(x4[j-1]+x4[j])*dt/2; // ตำแหน่งของตัวรถ
if(fabs(x3[j])>(max_x3))
{ max_x3 = fabs(x3[j]); } // ตำแหน่งของตัวรถที่มากที่สุด
if(fabs(F[j])>(max_F))
{ max_F = fabs(F[j]); } // แรงขับเคลื่อนที่มากที่สุด
sum_x1 = sum_x1+((fabs(x1[j])/(30.0*pi/180.0))*(fabs(x1[j])/(30.0*pi/180.0)));
// ผลรวมกำลังสองสำหรับมุมของลูกตุ้ม (คิดเป็นต่อหน่วย)
sum_x2 = sum_x2+((fabs(x2[j])/(250.0*pi/180.0))*(fabs(x2[j])/(250.0*pi/180.0)));
// ผลรวมกำลังสองสำหรับความเร็วเชิงมุมของลูกตุ้ม (คิดเป็นต่อหน่วย)
sum_x3 = sum_x3+((fabs(x3[j])/1.0)*(fabs(x3[j])/1.0));
// ผลรวมกำลังสองสำหรับตำแหน่งของตัวรถ (คิดเป็นต่อหน่วย)
sum_x4 = sum_x4+((fabs(x4[j])/3.5)*(fabs(x4[j])/3.5));
// ผลรวมกำลังสองสำหรับความเร็วของตัวรถ (คิดเป็นต่อหน่วย)
}
// สิ้นสุดการจำลองระบบ

/*-----ตรวจสอบเวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผันแบบข้อต่อเดียว-----*/
j = num-1; // พิจารณาจากตำแหน่งสุดท้ายของข้อมูล
while( (fabs(x1[j])<=0.1*pi/180.0)&&(j!=0)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับมุมของลูกตุ้ม
{ j = j-1; }
if(j<num-1)
{ time_sta[0] = (j+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับมุมของลูกตุ้ม
else

```

```

{ time_sta[0] = 99.0; } // กรณีมุมของลูกตุ้มไม่เข้าสู่เสถียรภาพ
j = num-1;
while((fabs(x2[j])<=0.1*pi/180.0)&&(j!=0)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับความเร็วเชิงมุมของลูกตุ้ม
{ j = j-1; }
if(j<num-1)
{ time_sta[1] = (j+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับความเร็วเชิงมุมของลูกตุ้ม
else
{ time_sta[1] = 99.0; } // กรณีความเร็วเชิงมุมของลูกตุ้มไม่เข้าสู่เสถียรภาพ
j = num-1;
while((fabs(x3[j])<=0.01)&&(j!=0)&&(fabs(max_x3)<=2.4)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับตำแหน่งของตัวรถ
{ j = j-1; }
if(j<num-1)
{ time_sta[2] = (j+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับตำแหน่งของตัวรถ
else
{ time_sta[2] = 99.0; } // กรณีตำแหน่งของตัวรถไม่เข้าสู่เสถียรภาพ
j = num-1;
while((fabs(x4[j])<=0.01)&&(j!=0)) // ตรวจสอบเงื่อนไขการเข้าสู่เสถียรภาพสำหรับความเร็วของตัวรถ
{ j = j-1; }
if(j<num-1)
{ time_sta[3] = (j+2)*dt; } // เวลาเข้าสู่เสถียรภาพสำหรับความเร็วของตัวรถ
else
{ time_sta[3] = 99.0; } // กรณีความเร็วของตัวรถไม่เข้าสู่เสถียรภาพ
sta_t = 0.0; // กำหนดค่าเริ่มต้นให้กับเวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผัน
for(j=0;j<4;j++)
{
if(time_sta[j]>(sta_t))
{ sta_t = time_sta[j]; } // เวลาเข้าสู่เสถียรภาพของระบบลูกตุ้มผกผัน
}
ii = 0;
while (fabs(x1[ii])>=5.0*pi/180.0)
{ ii = ii+1; } // จุดข้อมูลเริ่มต้นเพื่อใช้หาค่ามุมของลูกตุ้มที่มากที่สุด
for (j=ii;j<num;j++)
{
if(fabs(x1[j])>os_ang)
{ os_ang = fabs(x1[j]); } // มุมของลูกตุ้มที่มากที่สุด
}
error_cost = ((0.25*sum_x1)+(0.25*sum_x2)+(0.25*sum_x3)+(0.25*sum_x4))/100.0;
// ค่าฟังก์ชันวัตถุประสงค์

if(sta_t>4.0)
{ error_cost = error_cost+0.2; } // ถ้าตรงตามเงื่อนไขให้ปรับเพิ่มค่าฟังก์ชันวัตถุประสงค์

```



```

if(os_ang>15.0*pi/180.0)
{   error_cost = error_cost+0.2; } // ถ้าตรงตามเงื่อนไขให้ปรับเพิ่มค่าฟังก์ชันวัตถุประสงค์
if(max_F>60.0)
{   error_cost = error_cost+0.2; } // ถ้าตรงตามเงื่อนไขให้ปรับเพิ่มค่าฟังก์ชันวัตถุประสงค์
error_cost = error_cost;
return(error_cost); // ส่งกลับค่าฟังก์ชันวัตถุประสงค์
} // สิ้นสุดการทำงานของ function2

/*-----*/

```

โปรแกรม *nn\_ats\_fuzzy.cpp* (สร้างเสถียรภาพให้ระบบลูกคัมผผันแบบข้อต่อเดียวด้วยตัวควบคุมนิวโร-ตาบู-ฟuzzy)

หลังจากที่โปรแกรม *single\_ats.cpp* ทำการค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียมได้แล้ว กระบวนการต่อมาคือ การจำลองสถานการณ์การสร้างเสถียรภาพให้ระบบลูกคัมผผันแบบข้อต่อเดียว ด้วยตัวควบคุมนิวโร-ตาบู-ฟuzzy โดยใช้โปรแกรม *nn\_ats\_fuzzy.cpp* แต่เนื่องจากรายละเอียดของโปรแกรมดังกล่าวคล้ายคลึงกับ โปรแกรม *nn\_fuzzy.cpp* ซึ่งได้อธิบายไว้ในภาคผนวก ข. ดังนั้นจึงขอละเว้นการแสดงรายละเอียดของโปรแกรม *nn\_ats\_fuzzy.cpp* โดยส่วนที่แตกต่างกันระหว่างสองโปรแกรมนี้คือ โปรแกรม *nn\_fuzzy.cpp* จะกำหนดค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมด้วยการสุ่มค่า ในขณะที่โปรแกรม *nn\_ats\_fuzzy.cpp* รับค่าพารามิเตอร์เริ่มต้นที่เหมาะสมของโครงข่ายประสาทเทียมมาจากวิธีการค้นหาแบบตามเชิงปรับตัว ดังได้กล่าวถึงรายละเอียดไว้ใน โปรแกรม *single\_ats.cpp* คำสั่งที่กำหนดค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียมของทั้งสองโปรแกรมปรากฏในตารางที่ ข.1 ซึ่งค่าพารามิเตอร์ดังกล่าวถูกเก็บไว้ในตัวแปร S

ตารางที่ ข.1 คำสั่งกำหนดค่าพารามิเตอร์เริ่มต้นให้กับโครงข่ายประสาทเทียม

<i>nn_fuzzy.cpp</i>	<i>nn_ats_fuzzy.cpp</i>
<pre> for(i=0;i&lt;para;i++) {     rd = rand();     rd_0 = rd/div;     S[i] = ((x_h-x_l)*rd_0)+x_l; } </pre>	<pre> tabu_w = fopen("G:/Data_ATS/n_para.dat","rb"); fread(&amp;S,sizeof(double),para,tabu_w); fclose(tabu_w); </pre>

ภาคผนวก ฅ

โปรแกรมภาษาซีสำหรับสร้างเสถียรภาพให้ระบบ  
ถูกคุ้มครองแบบสองข้อต่อด้วยตัวควบคุมนิวโร-ตาบู-พีซี

```

*****
โปรแกรมภาษาซีที่ทำหน้าที่สร้างเสถียรภาพให้ระบบลูกคัมผผันแบบสองข้อต่อด้วยตัวควบคุมนิวโร-คาบู-พีซซี่
โดย นางสาวสุครัตน์ ขวัญอ่อน สาขาวิชาวิศวกรรมไฟฟ้า, มหาวิทยาลัยเทคโนโลยีสุรนารี, พ.ศ. 2547
*****

โปรแกรม double_ats.cpp (ค้นหาค่าพารามิเตอร์เริ่มต้นที่เหมาะสมให้กับโครงข่ายประสาทเทียม)

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define pi      3.14159           // ค่าคงที่  $\pi = 3.14159$ 
#define para    66               // กำหนดจำนวนพารามิเตอร์ที่ต้องการค้นหา
#define row     5               // กำหนดจำนวนการสุ่มค่าตอบเริ่มต้นเท่ากับ 5 ชุด
#define n_ne    40              // กำหนดจำนวนคำตอบรอบข้างการค้นหาเพื่อการสุ่มเท่ากับ 40
#define radius  0.2             // กำหนดรัศมีการค้นหา
#define error_stop 0.035        // กำหนดค่าฟังก์ชันวัตถุประสงค์เพื่อยุติการค้นหาคำตอบ
#define num     500             // กำหนดจำนวนจุดข้อมูลในการจำลองระบบ
#define ang_up  20.0            // กำหนดมุมเริ่มต้นของลูกคัมผผันบน (องศา)
#define ang_low 15.0           // กำหนดมุมเริ่มต้นของลูกคัมผผันล่าง (องศา)
#define pos_1   0.0            // กำหนดความยาวครึ่งหนึ่งของก้านลูกคัมผผัน (เมตร)

double function1(double s0_ini_0,double s0_ini_1,double s0_ini_2,double s0_ini_3,double s0_ini_4,double s0_ini_5,
                double s0_ini_6,double s0_ini_7,double s0_ini_8,double s0_ini_9,double s0_ini_10,double s0_ini_11,
                double s0_ini_12,double s0_ini_13,double s0_ini_14,double s0_ini_15,double s0_ini_16,double s0_ini_17,
                double s0_ini_18,double s0_ini_19,double s0_ini_20,double s0_ini_21,double s0_ini_22,double s0_ini_23,
                double s0_ini_24,double s0_ini_25,double s0_ini_26,double s0_ini_27,double s0_ini_28,double s0_ini_29,
                double s0_ini_30,double s0_ini_31,double s0_ini_32,double s0_ini_33,double s0_ini_34,double s0_ini_35,
                double s0_ini_36,double s0_ini_37,double s0_ini_38,double s0_ini_39,double s0_ini_40,double s0_ini_41,
                double s0_ini_42,double s0_ini_43,double s0_ini_44,double s0_ini_45,double s0_ini_46,double s0_ini_47,
                double s0_ini_48,double s0_ini_49,double s0_ini_50,double s0_ini_51,double s0_ini_52,double s0_ini_53,
                double s0_ini_54,double s0_ini_55,double s0_ini_56,double s0_ini_57,double s0_ini_58,double s0_ini_59,
                double s0_ini_60,double s0_ini_61,double s0_ini_62,double s0_ini_63,double s0_ini_64,double s0_ini_65);
                // function1 รับค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมทั้ง 66 ค่า

double function2(double DID1,double DID2,double DID3,double DID4,double DID5,double DID6);
                // function2 รับค่าระดับความสำคัญพลวัตจากโครงข่ายประสาทเทียม

main()
{
clrscr();

clock_t start,end;           // ประกาศตัวแปรเพื่อคำนวณเวลาการค้นหาคำตอบ
int    st_ini,st;           // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int    i,j,ct,ct_0,o,i_ini,j_ini,ct_ini,ct_0_ini,count; // ตัวแปรสำหรับการนับค่าและการวนรอบ
int    ct_max;              // จำนวนรอบการค้นหาสูงสุด

```

```

int    n; // ตัวนับจำนวนการซ้ำค่าคำตอบ
int    n_back; // ตัวนับจำนวนการใช้กลไกย้อนรอยการค้นหา
double s0_ini[row][para]; // ค่าพารามิเตอร์เริ่มต้นจำนวน 5 ชุดๆ ละ 66 ค่า
double rd,rd_0,div; // ตัวแปรเกี่ยวกับการสุ่ม
double x_l,x_h; // ขอบเขตล่างและขอบเขตบนในการค้นหาคำตอบ
double f_min_ini,f_min_best_ini[1]; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
double f_ini; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นในแต่ละรอบการค้นหา
double s0_best_ini[1][para]; // ค่าพารามิเตอร์ของ f_min_best_ini
double s; // ตัวแปรชั่วคราวทำหน้าที่รับส่งค่าพารามิเตอร์
double a,b; // ขอบของค่าสุ่ม โดยที่ a = -1.0 (ขอบล่าง) และ b = 1.0 (ขอบบน)
double c; // ค่าที่สุ่มได้จากคำสั่ง rand มีค่าเป็นบวกสูงสุดเท่ากับ 32767
double r0; // ค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1 ซึ่ง r0 = rd/div
double r; // ค่าสุ่ม  $r = a + r0 * (b - a)$ 
double f_min0[1]; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดมีค่าเท่ากับ f_min_best_ini
double s0[1][para]; // ค่าพารามิเตอร์ของ f_min0 มีค่าเท่ากับ s0_best_ini
double s1[n_ne][para]; // ค่าพารามิเตอร์ของคำตอบรอบข้างในการค้นหาแต่ละรอบ
double del[1][para]; // ช่วงขอบเขตล่างถึงขอบเขตบนของค่าพารามิเตอร์แต่ละตัว
double f_min1_old; // ค่าฟังก์ชันวัตถุประสงค์เพื่อเปรียบเทียบหาค่าที่ดีกว่า
double f_min1[n_ne]; // ค่าตอบทั้งหมดในแต่ละรอบการค้นหา
double f_min1_best[1]; // ค่าตอบที่ดีที่สุดในแต่ละรอบการค้นหา
double s1_best[1][para]; // ค่าพารามิเตอร์ของคำตอบที่ดีที่สุดในแต่ละรอบการค้นหา
double f_min0_old; // ค่าตอบที่ดีที่สุดในอดีต
double f_min0_new; // ค่าตอบที่ดีที่สุดในปัจจุบัน
double local_f[1]; // ค่าตอบวงแคบเฉพาะถิ่น
double local_ne[1][para]; // ค่าพารามิเตอร์ของคำตอบวงแคบเฉพาะถิ่น
double global_f[1]; // ค่าตอบโกล่วงกว้าง
double global_ne[1][para]; // ค่าพารามิเตอร์ของคำตอบโกล่วงกว้าง
double best_error; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด
double best_neighbor[para]; // ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมที่ได้จากการค้นหา
FILE *best_ini; // ตั้งชื่อเพิ่มข้อมูลเพื่อเก็บค่าพารามิเตอร์เริ่มต้นจากการค้นหาว่า best_ini
ct      = 0;
count   = 0;
ct_0_ini = 0;
div     = 32767.0; // ตัวหารค่าที่สุ่มได้เพื่อทำให้มีค่าสูงสุดเท่ากับ 1
x_l     = -1.0; // ขอบเขตล่างของค่าพารามิเตอร์ที่ทำการค้นหา
x_h     = 1.0; // ขอบเขตบนของค่าพารามิเตอร์ที่ทำการค้นหา
f_min_ini = pow(10.0,10.0); // กำหนดค่าเริ่มต้นให้กับฟังก์ชันวัตถุประสงค์ที่ดีที่สุดให้มีค่ามากๆ
printf("Enter seed of random number = ");
scanf("%d",&st_ini); // รับค่าตัวเลขการสุ่มเริ่มต้นผ่านคีย์บอร์ด
srand(st_ini); // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
for(i_ini=0;i_ini<row;i_ini++)

```

```

{
  for(j_ini=0;j_ini<para;j_ini++)
  {
    rd = rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
    rd_0 = rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
    s = ((x_h-x_l)*rd_0)+x_l; // ค่า s มีขอบเขตจาก -1 ถึง +1
    s0_ini[i_ini][j_ini] = s; // ค่าพารามิเตอร์ที่สุ่มได้ทั้ง 5 ชุดๆ ละ 66 ค่า
  }
}
for(i_ini=0;i_ini<row;i_ini++)
{
  f_ini = function1(s0_ini[i_ini][0],s0_ini[i_ini][1],s0_ini[i_ini][2],s0_ini[i_ini][3],s0_ini[i_ini][4],s0_ini[i_ini][5],
    s0_ini[i_ini][6],s0_ini[i_ini][7],s0_ini[i_ini][8],s0_ini[i_ini][9],s0_ini[i_ini][10],s0_ini[i_ini][11],
    s0_ini[i_ini][12],s0_ini[i_ini][13],s0_ini[i_ini][14],s0_ini[i_ini][15],s0_ini[i_ini][16],s0_ini[i_ini][17],
    s0_ini[i_ini][18],s0_ini[i_ini][19],s0_ini[i_ini][20],s0_ini[i_ini][21],s0_ini[i_ini][22],s0_ini[i_ini][23],
    s0_ini[i_ini][24],s0_ini[i_ini][25],s0_ini[i_ini][26],s0_ini[i_ini][27],s0_ini[i_ini][28],s0_ini[i_ini][29],
    s0_ini[i_ini][30],s0_ini[i_ini][31],s0_ini[i_ini][32],s0_ini[i_ini][33],s0_ini[i_ini][34],s0_ini[i_ini][35],
    s0_ini[i_ini][36],s0_ini[i_ini][37],s0_ini[i_ini][38],s0_ini[i_ini][39],s0_ini[i_ini][40],s0_ini[i_ini][41],
    s0_ini[i_ini][42],s0_ini[i_ini][43],s0_ini[i_ini][44],s0_ini[i_ini][45],s0_ini[i_ini][46],s0_ini[i_ini][47],
    s0_ini[i_ini][48],s0_ini[i_ini][49],s0_ini[i_ini][50],s0_ini[i_ini][51],s0_ini[i_ini][52],s0_ini[i_ini][53],
    s0_ini[i_ini][54],s0_ini[i_ini][55],s0_ini[i_ini][56],s0_ini[i_ini][57],s0_ini[i_ini][58],s0_ini[i_ini][59],
    s0_ini[i_ini][60],s0_ini[i_ini][61],s0_ini[i_ini][62],s0_ini[i_ini][63],s0_ini[i_ini][64],s0_ini[i_ini][65]);
    // function1 รับค่าพารามิเตอร์ที่สุ่มได้และส่งกลับค่าฟังก์ชันวัตถุประสงค์
  if(f_ini<=f_min_ini) // ประเมินค่าฟังก์ชันวัตถุประสงค์สำหรับค่าพารามิเตอร์แต่ละชุด
  {
    f_min_ini = f_ini; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด
    ct_ini = ct_0_ini; // ตัวนับชุดของพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์ลดลง
  }
  ct_0_ini = ct_0_ini+1;
}
f_min_best_ini[0] = f_min_ini; // ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด
for(j=0;j<para;j++)
{
  s0_best_ini[0][j] = s0_ini[ct_ini][j]; // ค่าเริ่มต้นของพารามิเตอร์จากการค้นหาที่ดีที่สุด
  del[0][j] = x_h-x_l; // ช่วงของขอบเขตการค้นหาคำตอบ
  s0[0][j] = s0_best_ini[0][j]; // เก็บค่าเริ่มต้นของพารามิเตอร์การค้นหาที่ดีที่สุดไว้ใน s0
  local_ne[0][j] = s0[0][j]; // เก็บค่าเริ่มต้นที่เป็นคำตอบวงแคบเฉพาะถิ่นไว้ใน local_ne
  global_ne[0][j] = local_ne[0][j]; // เก็บค่าเริ่มต้นที่เป็นคำตอบใกล้เคียงกว้างไว้ใน global_ne
  best_neighbor[j] = s0[0][j]; // เก็บพารามิเตอร์เริ่มต้นในการค้นหาที่ดีที่สุดไว้ใน best_neighbor
}
f_min0[0] = f_min_best_ini[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดไว้ใน f_min0

```

```

f_min0_old = f_min0[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดเพื่อเป็นอดีตไว้ใน f_min0_old
local_f[0] = f_min0[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่เป็นค่าตอบวงแคบเฉพาะถิ่น
global_f[0] = local_f[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่เป็นค่าตอบโกล้างกว้าง
best_error = f_min0[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดไว้ใน best_error
n_back = 0; // กำหนดตัวนับจำนวนการใช้กลไกการย้อนรอยเป็น 0
ct_0 = 0; // กำหนดตัวนับชุดของพารามิเตอร์ที่ค่าฟังก์ชันวัตถุประสงค์ลดลง
a = -1; // ขอบเขตล่างของค่าสุ่ม
b = 1; // ขอบเขตบนของค่าสุ่ม
n = 0; // กำหนดตัวนับจำนวนการซ้ำค่าคำตอบเริ่มต้นเท่ากับ 0

printf("Enter seed of random number = ");
scanf("%d",&st); // รับค่าตัวเลขการสุ่มเริ่มต้นจากคีย์บอร์ด
srand(st); // เก็บค่าตัวเลขการสุ่มเริ่มต้นไว้ในคำสั่ง srand
printf("Enter number of ct_max = ");
scanf("%d",&ct_max); // รับค่าจำนวนรอบสูงสุดในการค้นหาคำตอบจากคีย์บอร์ด
start = clock(); // เริ่มต้นคำนวณเวลาการค้นหาคำตอบ
loop2: for(o=0;o<=ct_max;o++) // เริ่มต้นการค้นหาคำตอบ
{
for(i=0;i<n_ne;i++) // สุ่มคำตอบรอบข้างจำนวน n_ne ชุด
{
for(j=0;j<para;j++) // แต่ละชุดของคำตอบรอบข้างประกอบด้วยคำตอบจำนวน 66 ค่า
{
c = rand(); // เก็บค่าที่สุ่มได้ซึ่งอยู่ในช่วง 0 ถึง 32767 ไว้ใน c
r0 = c/div; // ปรับค่าที่สุ่มได้ให้อยู่ในช่วง 0 ถึง 1
r = a+r0*(b-a); // ปรับค่าที่สุ่มได้ให้อยู่ในช่วง -1 ถึง 1
s1[i][j] = s0[0][j]+(r*radius*del[0][j]); // ค่าพารามิเตอร์ที่ได้จากการสุ่ม
if((s1[i][j]<x_l)||(s1[i][j]>x_h)) // กรณีค่าพารามิเตอร์จากการสุ่มเกินขอบเขตที่กำหนดทำการสุ่มค่าใหม่
{
while((s1[i][j]<x_l)||(s1[i][j]>x_h))
{
c = rand();
r0 = c/div;
r = a+r0*(b-a);
s1[i][j] = s0[0][j]+(r*radius*del[0][j]); // ค่าพารามิเตอร์ที่ได้จากการสุ่มซึ่งอยู่ในขอบเขตที่กำหนด
}
}
}
}
}
for(i=0;i<n_ne;i++)
{
f_min1[i] = function1(s1[i][0],s1[i][1],s1[i][2],s1[i][3],s1[i][4],s1[i][5],s1[i][6],s1[i][7],s1[i][8],s1[i][9],s1[i][10],s1[i][11],
s1[i][12],s1[i][13],s1[i][14],s1[i][15],s1[i][16],s1[i][17],s1[i][18],s1[i][19],s1[i][20],s1[i][21],s1[i][22],

```

```

s1[i][23],s1[i][24],s1[i][25],s1[i][26],s1[i][27],s1[i][28],s1[i][29],s1[i][30],s1[i][31],s1[i][32],s1[i][33],
s1[i][34],s1[i][35],s1[i][36],s1[i][37],s1[i][38],s1[i][39],s1[i][40],s1[i][41],s1[i][42],s1[i][43],s1[i][44],
s1[i][45],s1[i][46],s1[i][47],s1[i][48],s1[i][49],s1[i][50],s1[i][51],s1[i][52],s1[i][53],s1[i][54],s1[i][55],
s1[i][56],s1[i][57],s1[i][58],s1[i][59],s1[i][60],s1[i][61],s1[i][62],s1[i][63],s1[i][64],s1[i][65]);
} // function1 ส่งกลับค่าฟังก์ชันวัตถุประสงค์ของชุดคำตอบรอบข้างที่สุ่มได้
f_min1_old = f_min1[0]; // เก็บค่าเพื่อการเปรียบเทียบ
for(i=0;i<n_ne;i++)
{
    if(f_min1[i]<f_min1_old)
    {
        f_min1_old = f_min1[i]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดใน f_min1_old
        ct = ct_0; // นับค่าชุดของคำตอบรอบข้างที่ทำให้ฟังก์ชันวัตถุประสงค์ลดลงจากเดิม
    }
    ct_0 = ct_0+1;
}
f_min1_best[0] = f_min1_old; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดใน f_min1_best
for(j=0;j<para;j++)
{ s1_best[0][j] = s1[ct][j]; } // คำตอบการค้นหาที่ทำให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด
ct_0 = 0;
ct = 0;
f_min0_new = f_min1_best[0]; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดไว้เป็นค่าปัจจุบัน
if(f_min0_new>=f_min0_old) // เปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในปัจจุบันและในอดีต
{ n = n+1; } // เพิ่มจำนวนการซ้ำคำตอบเดิมในกรณีที่ค่าปัจจุบัน ไม่น้อยกว่าค่าในอดีต
else
{
    n = 0; // n = 0 ในกรณีที่ค่าฟังก์ชันวัตถุประสงค์ในปัจจุบันน้อยกว่าค่าในอดีต
    f_min0_old = f_min0_new; // และให้ค่าฟังก์ชันวัตถุประสงค์ในอดีตเท่ากับค่าวัตถุประสงค์ในปัจจุบัน
}
/*-----กลไกการย้อนรอย-----*/
if(n>=15) // ถ้าจำนวนการซ้ำคำตอบเป็น 15 เรียกใช้กลไกการย้อนรอย
{
    local_f[0] = f_min0[0]; // เก็บค่า f_min0 ไว้ใน local_f
    for(j=0;j<para;j++)
    { local_ne[0][j] = s0[0][j]; } // เก็บค่า s0 ไว้ใน local_ne
    n_back = n_back+1; // เพิ่มจำนวนการใช้กลไกการย้อนรอย
    goto loop1; // กระโดดไปทำงานที่ลาเบล loop1
}
// ปรับคำตอบและพารามิเตอร์
if(f_min1_best[0]<f_min0[0]) // เปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในแต่ละรอบการค้นหา
{
    f_min0[0] = f_min1_best[0]; // เก็บค่า f_min1_best ไว้ใน f_min0
}

```

```

    for(j=0;j<para;j++)
        { s0[0][j] = s1_best[0][j]; } // เก็บค่า s1_best ไว้ใน s0
    }
    if(f_min0[0]<best_error) // เปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดเพื่อเก็บลงในแฟ้มข้อมูล
    {
        best_error = f_min0[0]; // เก็บค่า f_min0 ไว้ใน best_error
        for(j=0;j<para;j++)
            { best_neighbor[j] = s0[0][j]; } // เก็บค่า s0 ไว้ใน best_neighbor
        }
    if((count>=ct_max)||(f_min0[0]<error_stop) // ตรวจสอบเงื่อนไขเพื่อยุติการค้นหา
        { goto loop3; } // กระโดดไปทำงานที่ลาเบล loop3
    count = count+1;
    } // สิ้นสุดการทำงานของลาเบล loop2
// เปรียบเทียบค่าตอบวงแคบเฉพาะถิ่นและค่าตอบใกล้เคียงกว้าง
loop1: if(local_f[0]<global_f[0]) // ถ้าค่าตอบวงแคบเฉพาะถิ่นน้อยกว่าค่าตอบใกล้เคียงกว้าง
    {
        global_f[0] = local_f[0]; // เก็บค่า local_f ไว้ใน global_f
        for(j=0;j<para;j++)
            { global_ne[0][j] = local_ne[0][j]; } // เก็บค่า local_ne ไว้ใน global_ne
        }
    if(global_f[0]<error_stop) // ตรวจสอบเงื่อนไขเพื่อยุติการค้นหาค่าตอบ
        { goto loop3; } // กระโดดไปทำงานที่ลาเบล loop3 กรณีตรงตามเงื่อนไข
    f_min0[0] = f_min1[1]; // เก็บค่าคำตอบชุดที่ 2 ของ f_min1 ไว้ใน f_min0
    for(j=0;j<para;j++)
        { s0[0][j] = s1[1][j]; } // เก็บค่า s1 ไว้ใน s0
    f_min0_old = f_min0[0]; // เก็บค่า f_min0 ไว้ใน f_min0_old
    n = 0; // ปรับค่าจำนวนการซ้ำคำตอบเป็น 0
    if((count>=ct_max)||(f_min0[0]<error_stop) // ตรวจสอบเงื่อนไขเพื่อยุติการค้นหา
        { goto loop3; } // กระโดดไปทำงานที่ลาเบล loop3
    count = count+1; // ปรับเพิ่มจำนวนรอบการค้นหา
    goto loop2; // กระโดดไปทำงานที่ลาเบล loop2
loop3: end = clock(); // สิ้นสุดการคำนวณเวลาในการค้นหา
    best_ini = fopen("G:best_ne1.dat","wb"); // เปิดแฟ้มข้อมูลเพื่อเก็บพารามิเตอร์ที่ค้นหาได้
    fwrite(&best_neighbor,sizeof(double),para,best_ini); // เขียนพารามิเตอร์ที่ค้นหาได้ลงในแฟ้มข้อมูล
    fclose(best_ini); // ปิดแฟ้มข้อมูล
    printf("stoped count = %d\n",count); // แสดงจำนวนรอบที่พบคำตอบ
    printf("best_error =%.6f\n",best_error); // แสดงค่าวัตถุประสงค์ที่ดีที่สุด
    printf("execution time =%f sec", (end-start)/CLK_TCK); // แสดงเวลาที่ใช้ในการค้นหา
    printf("\n"); // แสดงสัญญาณเสียงเพื่อสิ้นสุดโปรแกรม
    getch();
    } // สิ้นสุดโปรแกรมหลัก

```



```

/*-----*/

double function1(double s0_ini_0,double s0_ini_1,double s0_ini_2,double s0_ini_3, double s0_ini_4,double s0_ini_5,
                double s0_ini_6,double s0_ini_7,double s0_ini_8,double s0_ini_9,double s0_ini_10,double s0_ini_11,
                double s0_ini_12,double s0_ini_13,double s0_ini_14,double s0_ini_15,double s0_ini_16,double s0_ini_17,
                double s0_ini_18,double s0_ini_19,double s0_ini_20,double s0_ini_21,double s0_ini_22,double s0_ini_23,
                double s0_ini_24,double s0_ini_25,double s0_ini_26,double s0_ini_27,double s0_ini_28,double s0_ini_29,
                double s0_ini_30,double s0_ini_31,double s0_ini_32,double s0_ini_33,double s0_ini_34,double s0_ini_35,
                double s0_ini_36,double s0_ini_37,double s0_ini_38,double s0_ini_39,double s0_ini_40,double s0_ini_41,
                double s0_ini_42,double s0_ini_43,double s0_ini_44,double s0_ini_45,double s0_ini_46,double s0_ini_47,
                double s0_ini_48,double s0_ini_49,double s0_ini_50,double s0_ini_51,double s0_ini_52,double s0_ini_53,
                double s0_ini_54,double s0_ini_55,double s0_ini_56,double s0_ini_57,double s0_ini_58,double s0_ini_59,
                double s0_ini_60,double s0_ini_61,double s0_ini_62,double s0_ini_63,double s0_ini_64,double s0_ini_65)
    // รับค่าพารามิเตอร์จากการค้นหาและส่งกลับค่าฟังก์ชันวัตถุประสงค์

{
int    i; // ตัวแปรเพื่อการวนรอบ
int    l,m,n; // จำนวนเซลล์ประสาทเทียมในแต่ละชั้นของโครงข่ายประสาทเทียม
double S[para]; // พารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
double x[3]; // สัญญาณอินพุตที่ให้กับชั้นอินพุตของโครงข่ายประสาทเทียม
double sum; // ตัวแปรเพื่อหาผลรวมในกระบวนการเรียนรู้
double W[3][6][2]; // ค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
double W1[6][6][2]; // ค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
double Z1[6][2]; // ค่าไบแอสที่ชั้นซ่อนเร้น
double Z2[6][2]; // ค่าไบแอสที่ชั้นเอาต์พุต
double X1[6][2]; // เอาต์พุตของชั้นซ่อนเร้น
double X2[6][2]; // เอาต์พุตของชั้นเอาต์พุต
double uk1,uk2,uk3,uk4,uk5,uk6; // ค่าระดับความสำคัญพลวัต
double error_plant; // ค่าฟังก์ชันวัตถุประสงค์ที่ส่งกลับให้กับโปรแกรมหลัก
x[0] = fabs(ang_up)*pi/180.0; // มุมเริ่มต้นของลูกตุ้มอันบน (เรเดียน)
x[1] = fabs(ang_low)*pi/180.0; // มุมเริ่มต้นของลูกตุ้มอันล่าง (เรเดียน)
x[2] = fabs(pos_1); // ตำแหน่งเริ่มต้นของตัวรถ (เมตร)
S[0] = s0_ini_0; S[1] = s0_ini_1; S[2] = s0_ini_2; S[3] = s0_ini_3; S[4] = s0_ini_4; S[5] = s0_ini_5;
S[6] = s0_ini_6; S[7] = s0_ini_7; S[8] = s0_ini_8; S[9] = s0_ini_9; S[10] = s0_ini_10; S[11] = s0_ini_11;
S[12] = s0_ini_12; S[13] = s0_ini_13; S[14] = s0_ini_14; S[15] = s0_ini_15; S[16] = s0_ini_16; S[17] = s0_ini_17;
S[18] = s0_ini_18; S[19] = s0_ini_19; S[20] = s0_ini_20; S[21] = s0_ini_21; S[22] = s0_ini_22; S[23] = s0_ini_23;
S[24] = s0_ini_24; S[25] = s0_ini_25; S[26] = s0_ini_26; S[27] = s0_ini_27; S[28] = s0_ini_28; S[29] = s0_ini_29;
S[30] = s0_ini_30; S[31] = s0_ini_31; S[32] = s0_ini_32; S[33] = s0_ini_33; S[34] = s0_ini_34; S[35] = s0_ini_35;
S[36] = s0_ini_36; S[37] = s0_ini_37; S[38] = s0_ini_38; S[39] = s0_ini_39; S[40] = s0_ini_40; S[41] = s0_ini_42;
S[43] = s0_ini_43; S[44] = s0_ini_44; S[45] = s0_ini_45; S[46] = s0_ini_46; S[47] = s0_ini_47; S[48] = s0_ini_48;
S[49] = s0_ini_49; S[50] = s0_ini_50; S[51] = s0_ini_51; S[52] = s0_ini_52; S[53] = s0_ini_53; S[54] = s0_ini_54;
S[55] = s0_ini_55; S[56] = s0_ini_56; S[57] = s0_ini_57; S[58] = s0_ini_58; S[59] = s0_ini_59; S[60] = s0_ini_60;
S[61] = s0_ini_61; S[62] = s0_ini_62; S[63] = s0_ini_63; S[64] = s0_ini_64; S[65] = s0_ini_65;

```

```

// เก็บค่า s0_ini ลงใน S
i = 0;
for (l=0;l<3;l++)
{
    for(m=0;m<6;m++)
    {
        W[l][m][1] = S[i]; // คำนวณน้ำหนักระหว่างชั้นอินพุตไปยังชั้นซ่อนเร้น
        i++;
    }
}
for (m=0;m<6;m++)
{
    Z1[m][1] = S[i]; // ค่าไบแอสที่ชั้นซ่อนเร้น
    i++;
}
for (m=0;m<6;m++)
{
    for (n=0;n<6;n++)
    {
        W1[m][n][1] = S[i]; // คำนวณน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
        i++;
    }
}
for (n=0;n<6;n++)
{
    Z2[n][1] = S[i]; // ค่าไบแอสที่ชั้นเอาต์พุต
    i++;
}
i = 2; // จำนวนเอาต์พุตของโครงข่ายประสาทเทียม
for(m=0;m<6;m++)
{
    sum = 0.0;
    for(l=0;l<3;l++)
    {
        sum = sum+W[l][m][i-1]*x[l];
    }
    sum = sum+Z1[m][i-1];
    X1[m][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตจากชั้นซ่อนเร้น (X1j)
}
for(n=0;n<6;n++)
{

```

```

sum = 0.0;
for(m=0;m<6;m++)
{
    sum = sum+W1[m][n][i-1]*X1[m][i-1];
}
sum = sum+Z2[n][i-1];
X2[n][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตจากชั้นเอาต์พุต (X2k)
}
uk1 = X2[0][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับผลต่างเชิงมุมของลูกคู้ม
uk2 = X2[1][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับผลต่างความเร็วเชิงมุมของลูกคู้ม
uk3 = X2[2][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับมุมของลูกคู้มอันล่าง
uk4 = X2[3][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับความเร็วเชิงมุมของลูกคู้มอันล่าง
uk5 = X2[4][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับตำแหน่งของตัวรถ
uk6 = X2[5][i-1]; // ค่าระดับความสำคัญพลวัตสำหรับความเร็วของตัวรถ
error_plant = function2(uk1,uk2,uk3,uk4,uk5,uk6); // ส่งค่าระดับความสำคัญพลวัตให้กับ function2
return(error_plant); // ส่งกลับค่าฟังก์ชันวัตถุประสงค์ให้โปรแกรมหลัก
} // สิ้นสุดการทำงานของ function1
/*-----*/

double function2(double DID_1,double DID_2,double DID_3,double DID_4,double DID_5,double DID_6)
// function2 รับค่าระดับความสำคัญพลวัตจากโครงข่ายประสาทเทียม
{
int i; // ตัวแปรเพื่อการวนรอบ
double M,m1,m2,l1,l2,L1,g,dt; // พารามิเตอร์ของระบบลูกคู้มผกผันแบบสองข้อต่อ
double S1,S2,S3,S4,S5,S6; // ตัวปรับค่าทางด้านอินพุตของกฎฟัซซี่
double osf; // ตัวปรับค่าทางด้านเอาต์พุตของกฎฟัซซี่
double R1_h,R2_h,R3_h,R1_c,R2_c,R3_c; // ค่าระดับความเป็นสมาชิกของกฎฟัซซี่
double a11,a22,a33,a12,a21,a13,a31,a23,a32,b1,b2,b3; // ค่าสัมประสิทธิ์ของสมการแบบจำลองทางคณิตศาสตร์ของระบบ
double c1,c2,d1,d2,d3,d4,e1,e2; // ตัวแปรเพื่อการหาค่าตอบของสมการ
double error_cost; // ค่าฟังก์ชันวัตถุประสงค์ที่ส่งกลับให้ function1
double F[num]; // แรงขับเคลื่อนที่กระทำกับตัวรถในแนวระดับ
double x1[num]; // ผลต่างเชิงมุมของลูกคู้ม
double x2[num]; // ผลต่างความเร็วเชิงมุมของลูกคู้ม
double x3[num]; // มุมของลูกคู้มอันล่าง
double x4[num]; // ความเร็วเชิงมุมของลูกคู้มอันล่าง
double x5[num]; // ตำแหน่งของตัวรถ
double x6[num]; // ความเร็วของตัวรถ
double xx1[num]; // ความเร่งเชิงมุมของลูกคู้มอันบน
double xx2[num]; // ความเร่งเชิงมุมของลูกคู้มอันล่าง
double x_1[num]; // ความเร็วเชิงมุมของลูกคู้มอันบน
double x_2[num]; // มุมของลูกคู้มอันบน

```

```

double xx3[num]; // ความเร่งของตัวรถ
double f1,f2,f3,f4,f5,f6; // ผลการอนุมานฟัซซี่ด้วยวิธี simplified fuzzy reasoning method
double y_1; // ผลรวมมุมของลูกตุ้มอันบน
double y_2; // ผลรวมมุมของลูกตุ้มอันล่าง
double y_3; // ผลรวมความเร็วเชิงมุมของลูกตุ้มอันบน
double y_4; // ผลรวมความเร็วเชิงมุมของลูกตุ้มอันล่าง
double y_5; // ผลรวมตำแหน่งของตัวรถ
double y_6; // ผลรวมความเร็วของตัวรถ
M = 1.0; // มวลของตัวรถ (กิโลกรัม)
m1 = 0.2; // มวลของก้านลูกตุ้มอันล่าง (กิโลกรัม)
m2 = 0.1; // มวลของก้านลูกตุ้มอันบน (กิโลกรัม)
l1 = 0.4; // ความยาวครึ่งหนึ่งของก้านลูกตุ้มอันล่าง (เมตร)
l2 = 0.2; // ความยาวครึ่งหนึ่งของก้านลูกตุ้มอันบน (เมตร)
L1 = 2.0*l1; // ความยาวของก้านลูกตุ้มอันล่าง (เมตร)
g = 9.8; // ค่าแรงโน้มถ่วงของโลก
dt = 0.01; // ช่วงเวลาแซมปลิง (วินาที)
S1 = 15.0*pi/180.0; // ตัวปรับค่าสำหรับผลต่างเชิงมุมของลูกตุ้ม
S2 = 100.0*pi/180.0; // ตัวปรับค่าสำหรับผลต่างความเร็วเชิงมุมของลูกตุ้ม
S3 = 15.0*pi/180.0; // ตัวปรับค่าสำหรับมุมของลูกตุ้มอันล่าง
S4 = 100.0*pi/180.0; // ตัวปรับค่าสำหรับความเร็วเชิงมุมของลูกตุ้มอันล่าง
S5 = 2.4; // ตัวปรับค่าสำหรับตำแหน่งของตัวรถ
S6 = 1.0; // ตัวปรับค่าสำหรับความเร็วของตัวรถ
osf = 10.0*(M+m1+m2); // ตัวปรับค่าสำหรับแรงขับเคลื่อนที่กระทำกับตัวรถ
F[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับแรงขับเคลื่อน
xx1[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับความเร็วเชิงมุมของลูกตุ้มอันบน
xx2[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับความเร็วเชิงมุมของลูกตุ้มอันล่าง
xx3[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับความเร็วของตัวรถ
x_1[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับความเร็วเชิงมุมของลูกตุ้มอันบน
x_2[0] = fabs(ang_up)*pi/180.0; // กำหนดค่าเริ่มต้นให้กับมุมของลูกตุ้มอันบน
x3[0] = fabs(ang_low)*pi/180.0; // กำหนดค่าเริ่มต้นให้กับมุมของลูกตุ้มอันล่าง
x4[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับความเร็วเชิงมุมของลูกตุ้มอันล่าง
x5[0] = fabs(pos_1); // กำหนดค่าเริ่มต้นให้กับตำแหน่งของตัวรถ
x6[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับความเร็วของตัวรถ
x1[0] = x_2[0]-x3[0]; // กำหนดค่าเริ่มต้นให้กับผลต่างเชิงมุมของลูกตุ้ม
x2[0] = 0.0; // กำหนดค่าเริ่มต้นให้กับผลต่างความเร็วเชิงมุมของลูกตุ้ม
y_1 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมมุมของลูกตุ้มอันบน
y_2 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมมุมของลูกตุ้มอันล่าง
y_3 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมความเร็วเชิงมุมของลูกตุ้มอันบน
y_4 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมความเร็วเชิงมุมของลูกตุ้มอันล่าง
y_5 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมตำแหน่งของตัวรถ
y_6 = 0.0; // กำหนดค่าเริ่มต้นให้กับผลรวมความเร็วของตัวรถ

```

```

for(i=1;i<num;i++)
{
/*-----SIRM-1-----*/
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0;
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0;
if(fabs((x1[i-1]/S1))>=32767.0)
{ goto loop4; }
// Rule 1
if((x1[i-1]/S1)<-1.0)
{
R1_h = 1.0;
R1_c = 1.0;
}
else if(((x1[i-1]/S1)>=-1.0)&&((x1[i-1]/S1)<0.0) )
{
R1_h = fabs(x1[i-1]/S1);
R1_c = 1.0;
}
// Rule 2
if(((x1[i-1]/S1)>=-1.0)&&((x1[i-1]/S1)<0.0) )
{
R2_h = (x1[i-1]/S1)+1.0;
R2_c = 0.0;
}
else if(((x1[i-1]/S1)>=0.0)&&((x1[i-1]/S1)<1.0) )
{
R2_h = 1.0-(x1[i-1]/S1);
R2_c = 0.0;
}
// Rule 3
if(((x1[i-1]/S1)>=0.0)&&((x1[i-1]/S1)<1.0) )
{
R3_h = (x1[i-1]/S1);
R3_c = -1.0;
}
else if((x1[i-1]/S1)>=1.0)
{
R3_h = 1.0;
R3_c = -1.0;
}
}
}

```

// เริ่มต้นการจำลองระบบ

// กฎพีชคณิตแบบกฎอินพุตเดียวสำหรับผลต่างเชิงมุมของลูกตุ้ม

// กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต

// กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต

// ถ้าผลต่างเชิงมุมเกินขอบเขตที่กำหนดไว้

// กระโดดไปทำงานที่ลาเบล loop4

// กฎที่ 1 ถ้า  $x1 = NB$  แล้ว  $f1 = 1.0$

// กฎที่ 2 ถ้า  $x1 = ZO$  แล้ว  $f1 = 0.0$

// กฎที่ 3 ถ้า  $x1 = PB$  แล้ว  $f1 = -1.0$

```

f1 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-1

/*-----SIRM-2-----*/
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0;
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0;
//Rule 4
if((x2[i-1]/S2)<-1.0)
{
    R1_h = 1.0;
    R1_c = 1.0;
}
else if(((x2[i-1]/S2)>=-1.0)&&((x2[i-1]/S2)<0.0) )
{
    R1_h = fabs(x2[i-1]/S2);
    R1_c = 1.0;
}
// Rule 5
if(((x2[i-1]/S2)>=-1.0)&&((x2[i-1]/S2)<0.0) )
{
    R2_h = (x2[i-1]/S2)+1.0;
    R2_c = 0.0;
}
else if(((x2[i-1]/S2)>=0.0)&&((x2[i-1]/S2)<1.0) )
{
    R2_h = 1.0-(x2[i-1]/S2);
    R2_c = 0.0;
}
// Rule 6
if(((x2[i-1]/S2)>=0.0)&&((x2[i-1]/S2)<1.0) )
{
    R3_h = (x2[i-1]/S2);
    R3_c = -1.0;
}
else if((x2[i-1]/S2)>=1.0)
{
    R3_h = 1.0;
    R3_c = -1.0;
}
f2 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-2

```

// ผลการอนุมานฟuzzyสำหรับ SIRM-1  
 // กฎฟuzzyแบบกฎอินพุตเดียวสำหรับผลต่างความเร็วเชิงมุมของลูกตุ้ม  
 // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต  
 // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต  
 // กฎที่ 4 ถ้า  $x_2 = NB$  แล้ว  $f_2 = 1.0$   
 // กฎที่ 5 ถ้า  $x_2 = ZO$  แล้ว  $f_2 = 0.0$ ;  
 // กฎที่ 6 ถ้า  $x_2 = PB$  แล้ว  $f_2 = -1.0$

```

/*-----SIRM-3-----*/
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0;
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0;
// Rule 7
if((x3[i-1]/S3)<-1.0)
{
    R1_h = 1.0;
    R1_c = 1.0;
}
else if(((x3[i-1]/S3)>=-1.0)&&((x3[i-1]/S3)<0.0) )
{
    R1_h = fabs(x3[i-1]/S3);
    R1_c = 1.0;
}
// Rule 8
if(((x3[i-1]/S3)>=-1.0)&&((x3[i-1]/S3)<0.0) )
{
    R2_h = (x3[i-1]/S3)+1.0;
    R2_c = 0.0;
}
else if(((x3[i-1]/S3)>=0.0)&&((x3[i-1]/S3)<1.0) )
{
    R2_h = 1.0-(x3[i-1]/S3);
    R2_c = 0.0;
}
// Rule 9
if(((x3[i-1]/S3)>=0.0)&&((x3[i-1]/S3)<1.0) )
{
    R3_h = (x3[i-1]/S3);
    R3_c = -1.0;
}
else if((x3[i-1]/S3)>=1.0)
{
    R3_h = 1.0;
    R3_c = -1.0;
}
f3 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);

/*-----SIRM-4-----*/
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0;

```

// กฎฟuzzyแบบกฎอินพุตเดียวสำหรับมุมของลูกตุ้มอันล่าง  
// กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต  
// กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต  
// กฎที่ 7 ถ้า  $x_3 = NB$  แล้ว  $f_3 = 1.0$

// กฎที่ 8 ถ้า  $x_3 = ZO$  แล้ว  $f_3 = 0.0$

// กฎที่ 9 ถ้า  $x_3 = PB$  แล้ว  $f_3 = -1.0$

// ผลการอนุมานฟuzzyสำหรับ SIRM-3

// กฎฟuzzyแบบกฎอินพุตเดียวสำหรับความเร็วเชิงมุมของลูกตุ้มอันล่าง  
// กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต

```

R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต
// Rule 10 // กฎที่ 10 ถ้า x4 = NB แล้ว f4 = 1.0
if((x4[i-1]/S4)<1.0)
{
    R1_h = 1.0;
    R1_c = 1.0;
}
else if(((x4[i-1]/S4)>=1.0)&&((x4[i-1]/S4)<0.0) )
{
    R1_h = fabs(x4[i-1]/S4);
    R1_c = 1.0;
}
// Rule 11 // กฎที่ 11 ถ้า x4 = ZO แล้ว f4 = 0.0
if(((x4[i-1]/S4)>=1.0)&&((x4[i-1]/S4)<0.0) )
{
    R2_h = (x4[i-1]/S4)+1.0;
    R2_c = 0.0;
}
else if(((x4[i-1]/S4)>=0.0)&&((x4[i-1]/S4)<1.0) )
{
    R2_h = 1.0-(x4[i-1]/S4);
    R2_c = 0.0;
}
// Rule 12 // กฎที่ 12 ถ้า x4 = PB แล้ว f4 = -1.0
if(((x4[i-1]/S4)>=0.0)&&((x4[i-1]/S4)<1.0) )
{
    R3_h = (x4[i-1]/S4);
    R3_c = -1.0;
}
else if((x4[i-1]/S4)>=1.0)
{
    R3_h = 1.0;
    R3_c = -1.0;
}
f4 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h); // ผลการอนุมานฟuzzyสำหรับ SIRM-4

/*-----SIRM-5-----*/ // กฎฟuzzyแบบกฎอินพุตเดียวสำหรับตำแหน่งของตัวรถ
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต
// Rule 13 // กฎที่ 13 ถ้า x5 = NB แล้ว f5 = 1.0

```



```

if((x5[i-1]/S5)<-1.0)
{
    R1_h = 1.0;
    R1_c = 1.0;
}
else if(((x5[i-1]/S5)>=-1.0)&&((x5[i-1]/S5)<0.0) )
{
    R1_h = fabs(x5[i-1]/S5);
    R1_c = 1.0;
}
// Rule 14 // กฎที่ 14 ถ้า x5 = ZO แล้ว f5 = 0.0
if(((x5[i-1]/S5)>=-1.0)&&((x5[i-1]/S5)<0.0) )
{
    R2_h = (x5[i-1]/S5)+1.0;
    R2_c = 0.0;
}
else if(((x5[i-1]/S5)>=0.0)&&((x5[i-1]/S5)<1.0) )
{
    R2_h = 1.0-(x5[i-1]/S5);
    R2_c = 0.0;
}
// Rule 15 // กฎที่ 15 ถ้า x5 = PB แล้ว f5 = -1.0
if(((x5[i-1]/S5)>=0.0)&&((x5[i-1]/S5)<1.0) )
{
    R3_h = (x5[i-1]/S5);
    R3_c = -1.0;
}
else if((x5[i-1]/S5)>=1.0)
{
    R3_h = 1.0;
    R3_c = -1.0;
}
f5 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-5

/*-----SIRM-6-----*/ // กฎฟuzzyแบบกฎอินพุตเดียวสำหรับความเร็วของตัวรถ
R1_h = 0.0; R2_h = 0.0; R3_h = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรอินพุต
R1_c = 0.0; R2_c = 0.0; R3_c = 0.0; // กำหนดค่าเริ่มต้นให้กับค่าระดับความเป็นสมาชิกของตัวแปรเอาต์พุต
// Rule 16 // กฎที่ 16 ถ้า x6 = NB แล้ว f6 = 1.0
if((x6[i-1]/S6)<-1.0)
{

```

```

R1_h = 1.0;
R1_c = 1.0;
}
else if(((x6[i-1]/S6)>=-1.0)&&((x6[i-1]/S6)<0.0) )
{
R1_h = fabs(x6[i-1]/S6);
R1_c = 1.0;
}
// Rule 17 // กฎที่ 17 ถ้า x6 = ZO แล้ว f6 = 0.0
if(((x6[i-1]/S6)>=-1.0)&&((x6[i-1]/S6)<0.0) )
{
R2_h = (x6[i-1]/S6)+1.0;
R2_c = 0.0;
}
else if(((x6[i-1]/S6)>=0.0)&&((x6[i-1]/S6)<1.0) )
{
R2_h = 1.0-(x6[i-1]/S6);
R2_c = 0.0;
}
// Rule 18 // กฎที่ 18 ถ้า x6 = PB แล้ว f6 = -1.0
if(((x6[i-1]/S6)>=0.0)&&((x6[i-1]/S6)<1.0) )
{
R3_h = (x6[i-1]/S6);
R3_c = -1.0;
}
else if((x6[i-1]/S6)>=1.0)
{
R3_h = 1.0;
R3_c = -1.0;
}
f6 = ((R1_h*R1_c)+(R2_h*R2_c)+(R3_h*R3_c))/(R1_h+R2_h+R3_h);
// ผลการอนุมานฟuzzyสำหรับ SIRM-6

F[i] = ((f1*DID_1*10.0)+(f2*DID_2*10.0)+(f3*DID_3*10.0)+(f4*DID_4*10.0)+ (f5*DID_5*10.0)+(f6*DID_6*10.0))*osf;
// แรงขับเคลื่อนที่กระทำกับตัวรถ

// คำสัมประสิทธิ์ของสมการแบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มผกผันแบบสองข้อต่อ
// (พิจารณาได้จากสมการที่ (ก-1) ในภาคผนวก ก.)
a11 = M+m1+m2;
a22 = (4.0*m1*I1/3.0)+(m2*L1*L1);
a33 = (4.0*m2*I2/3.0);
a12 = (m1*I1+m2*L1)*cos(x3[i-1]);

```

```

a21 = a12;
a13 = m2*L2*cos(x_2[i-1]);
a31 = a13;
a23 = m2*L1*L2*cos((x3[i-1])-(x_2[i-1]));
a32 = a23;
b1 = F[i]+(((m1*L1)+(m2*L1))*x4[i-1]*x4[i-1]*sin(x3[i-1]))+(m2*L2*x_1[i-1]*x_1[i-1]*sin(x_2[i-1]));
b2 = (((m1*L1)+(m2*L1))*g*sin(x3[i-1]))-(m2*L1*L2*x_1[i-1]*x_1[i-1]*sin((x3[i-1])-(x_2[i-1])));
b3 = (m2*L2*g*sin(x_2[i-1]))+(m2*L1*L2*x4[i-1]*x4[i-1]*sin((x3[i-1])-(x_2[i-1])));
// อักษรระเบียบวิธีเชิงตัวเลขแก้คำตอบของสมการ

c1 = b1-(a11*b2/a21);
c2 = b1-(a11*b3/a31);
d1 = a12-(a11*a22/a21);
d2 = a13-(a11*a23/a21);
d3 = a12-(a11*a32/a31);
d4 = a13-(a11*a33/a31);
e1 = d2-(d4*d1/d3);
e2 = c1-(c2*d1/d3);
xx1[i] = e2/e1; // ความเร่งเชิงมุมของลูกตุ้มอันบน
xx2[i] = (c1-(d2*xx1[i]))/d1; // ความเร่งเชิงมุมของลูกตุ้มอันล่าง
xx3[i] = (b1-(a12*xx2[i])-(a13*xx1[i]))/a11; // ความเร่งของตัวรถ

// การอินทิเกรตอาศัยการประมาณค่าด้วยวิธีสี่เหลี่ยมคางหมู (trapezoidal integration)
x_1[i] = x_1[i-1]+(xx1[i]+xx1[i-1])*dt/2.0; // ความเร็วเชิงมุมของลูกตุ้มอันบน
x_2[i] = x_2[i-1]+(xx2[i]+xx2[i-1])*dt/2.0; // มุมของลูกตุ้มอันบน
x4[i] = x4[i-1]+(xx3[i]+xx3[i-1])*dt/2.0; // ความเร็วเชิงมุมของลูกตุ้มอันล่าง
x3[i] = x3[i-1]+(x4[i]+x4[i-1])*dt/2.0; // มุมของลูกตุ้มอันล่าง
x6[i] = x6[i-1]+(xx3[i]+xx3[i-1])*dt/2.0; // ความเร็วของตัวรถ
x5[i] = x5[i-1]+(x6[i]+x6[i-1])*dt/2.0; // ตำแหน่งของตัวรถ
x1[i] = x_2[i]-x3[i]; // ผลต่างเชิงมุมของลูกตุ้ม
x2[i] = x_1[i]-x4[i]; // ผลต่างความเร็วเชิงมุมของลูกตุ้ม
y_1 = y_1+((fabs(x_2[i])/(15.0*pi/180.0))*(fabs(x_2[i])/(15.0*pi/180.0)));
// ผลรวมมุมของลูกตุ้มอันบน
y_2 = y_2+((fabs(x3[i])/(15.0*pi/180.0))*(fabs(x3[i])/(15.0*pi/180.0)));
// ผลรวมมุมของลูกตุ้มอันล่าง
y_3 = y_3+((fabs(x_1[i])/(250.0*pi/180.0))*(fabs(x_1[i])/(250.0*pi/180.0)));
// ผลรวมความเร็วเชิงมุมของลูกตุ้มอันบน
y_4 = y_4+((fabs(x4[i])/(250.0*pi/180.0))*(fabs(x4[i])/(250.0*pi/180.0)));
// ผลรวมความเร็วเชิงมุมของลูกตุ้มอันล่าง
y_5 = y_5+((fabs(x5[i])/1.0)*(fabs(x5[i])/1.0)); // ผลรวมตำแหน่งของตัวรถ
y_6 = y_6+((fabs(x6[i])/3.5)*(fabs(x6[i])/3.5)); // ผลรวมความเร็วของตัวรถ
} // สิ้นสุดการจำลองระบบ

```

```

if(i==num) // กรณีผลต่างเชิงมุมของลูกตุ้มมีขนาดอยู่ในขอบเขตที่กำหนดไว้
{
    error_cost = (y_1+y_2+y_3+y_4+y_5)/1000.0; // ค่าฟังก์ชันวัตถุประสงค์
    return(error_cost); // ส่งกลับค่าฟังก์ชันวัตถุประสงค์ให้กับ function1
}
else // กรณีผลต่างเชิงมุมของลูกตุ้มมีขนาดเกินขอบเขตที่กำหนดไว้
{
    loop4: error_cost = 9999.0;
    return(error_cost); // ส่งกลับค่าฟังก์ชันวัตถุประสงค์เท่ากับ 9999.0 (กำหนดให้มีค่ามาก)
}
} // สิ้นสุดการทำงานของ function2

/*-----*/

โปรแกรม d_nn_ats_fuzzy.cpp (โครงข่ายประสาทเทียมรับค่าพารามิเตอร์เริ่มต้นที่เหมาะสมเป็นค่าเริ่มต้นในกระบวนการเรียนรู้)
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define pi 3.14159 // ค่าคงที่  $\pi = 3.14159$ 
#define para 66 // จำนวนพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
#define I_MAX 300 // จำนวนรอบในการเรียนรู้ของโครงข่ายประสาทเทียม
#define num 500 // จำนวนจุดข้อมูลของระบบลูกตุ้มผกผันที่ใช้ในกระบวนการเรียนรู้
#define ang_up 20.0 // มุมเริ่มต้นของลูกตุ้มอันบน (องศา)
#define ang_low 15.0 // มุมเริ่มต้นของลูกตุ้มอันล่าง (องศา)
#define pos_1 0.0 // ตำแหน่งเริ่มต้นของตัวรถ (เมตร)
void function1(double uk1,double uk2,double uk3,double uk4,double uk5,double uk6,double *y_1,double *y_2,
double *y_3,double *y_4,double *y_5,double *y_6);
// function1 รับค่าระดับความสำคัญพลวัตแต่ละรอบการเรียนรู้ และส่งค่า
// ความคลาดเคลื่อนมาที่โปรแกรมหลัก
main() // เริ่มต้นโปรแกรมหลัก
{
    clrscr();
    clock_t start,end; // ประกาศตัวแปรเพื่อคำนวณเวลาในการเรียนรู้
    int i,iter; // ตัวนับจำนวนรอบของการเรียนรู้
    int l,m,n; // จำนวนเซลล์ประสาทเทียมในแต่ละชั้นของโครงข่าย
    double S[para]; // ค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียม
    double x[3]; // สัญญาณอินพุตในชั้นอินพุตของโครงข่ายประสาทเทียม
    double error_max; // ค่าความคลาดเคลื่อนเพื่อยุติการเรียนรู้
    double net_error[I_MAX]; // ค่าความคลาดเคลื่อนรวมในแต่ละรอบของการเรียนรู้
    double sum,delta_u,delta_y,error_final,es[6]; // ตัวแปรในกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม
    double n1,nn,nz,n1z,beta; // ค่าอัตราการเรียนรู้และค่าคงที่โมเมนตัม

```

```

double uk1,uk2,uk3,uk4,uk5,uk6; // ค่าระดับความสำคัญพลวัตในแต่ละรอบของการเรียนรู้
double y_1,y_2,y_3,y_4,y_5,y_6; // ค่าความคลาดเคลื่อนของตัวแปรสถานะแต่ละตัว
double W[3][6][I_MAX]; // ค่าน้ำหนักที่เชื่อมต่อระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
double W1[6][6][I_MAX]; // ค่าน้ำหนักที่เชื่อมต่อระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
double Z1[6][I_MAX]; // ค่าไบแอสที่ชั้นซ่อนเร้น
double Z2[6][I_MAX]; // ค่าไบแอสที่ชั้นเอาต์พุต
double delta_W[2][6][I_MAX]; // ตัวปรับแก้ค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
double delta_W1[6][6][I_MAX]; // ตัวปรับแก้ค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
double del[6][I_MAX]; // พจน์ความคลาดเคลื่อนที่ชั้นซ่อนเร้น ( $\delta_j$ )
double del_1[6][I_MAX]; // พจน์ความคลาดเคลื่อนที่ชั้นเอาต์พุต ( $\delta_k$ )
double X1[6][I_MAX]; // เอาต์พุตที่ออกจากชั้นซ่อนเร้นของโครงข่ายประสาทเทียม
double X2[6][I_MAX]; // เอาต์พุตที่ออกจากชั้นเอาต์พุตของโครงข่ายประสาทเทียม
double yd[6][I_MAX]; // เอาต์พุตที่ต้องการของระบบ
double y[6][I_MAX]; // เอาต์พุตที่ได้ของระบบ
double D2M[6]; // ค่าระดับความสำคัญพลวัตที่ได้จากโครงข่ายประสาทเทียม
FILE *best_w,*bestDID; // กำหนดชื่อแฟ้มข้อมูล
x[0] = fabs(ang_up)*pi/180.0; // เซลล์ประสาทเทียมแรกของชั้นอินพุตรับค่ามุมเริ่มต้นของลูกตุ้มอันบน
x[1] = fabs(ang_low)*pi/180.0; // เซลล์ประสาทเทียมที่สองของชั้นอินพุตรับมุมเริ่มต้นของลูกตุ้มอันล่าง
x[2] = pos_1; // เซลล์ประสาทเทียมที่สามของชั้นอินพุตรับค่าตำแหน่งเริ่มต้นของตัวรถ
nn = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าน้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
n1 = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าน้ำหนักระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
nz = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าไบแอสที่ชั้นซ่อนเร้น
n1z = 0.05; // ค่าอัตราการเรียนรู้เพื่อปรับค่าไบแอสที่ชั้นเอาต์พุต
beta = 0.25; // ค่าคงที่โมเมนตัม
best_w = fopen("G:best_ne1.dat","rb"); // เปิดแฟ้มข้อมูลเพื่ออ่านค่าพารามิเตอร์เริ่มต้นที่ได้จากการค้นหา
fread(&S,sizeof(double),para,best_w); // อ่านค่าพารามิเตอร์เริ่มต้นของโครงข่ายประสาทเทียมและเก็บไว้ใน S
fclose(best_w); // ปิดแฟ้มข้อมูล best_w
i = 0; // ตัวนับลำดับของค่าพารามิเตอร์
for (l=0;l<3;l++)
{
    for(m=0;m<6;m++)
    {
        W[l][m][1] = S[i]; // กำหนดค่าน้ำหนักเริ่มต้นที่เชื่อมต่อระหว่างชั้นอินพุตกับชั้นซ่อนเร้น
        i++;
    }
}
for (m=0;m<6;m++)
{
    Z1[m][1] = S[i]; // กำหนดค่าไบแอสเริ่มต้นให้กับชั้นซ่อนเร้น
    i++;
}

```

```

for (m=0;m<6;m++)
{
    for (n=0;n<6;n++)
    {
        W1[m][n][1] = S[i]; // กำหนดค่าน้ำหนักเริ่มต้นที่เชื่อมต่อระหว่างชั้นซ่อนเร้นกับชั้นเอาต์พุต
        i++;
    }
}
for (n=0;n<6;n++)
{
    Z2[n][1] = S[i]; // กำหนดค่าไบแอสเริ่มต้นให้กับชั้นเอาต์พุต
    i++;
}
for(m=0;m<6;m++)
{
    for(n=0;n<6;n++)
    { delta_W1[m][n][0] = 0.0; } // กำหนดค่าเริ่มต้นให้กับ delta_W1
}
for(l=0;l<3;l++)
{
    for(m=0;m<6;m++)
    { delta_W[l][m][0] = 0.0; } // กำหนดค่าเริ่มต้นให้กับ delta_W
}
for(n=0;n<6;n++)
{
    yd[n][1] = 0.0; // กำหนดค่าเริ่มต้นให้กับเอาต์พุตที่ต้องการของระบบ
    X2[n][0] = 0.0; // กำหนดค่าเริ่มต้นให้กับเอาต์พุตของโครงข่ายประสาทเทียม
}
i = 2; // จำนวนรอบเริ่มต้นในกระบวนการเรียนรู้
iter = 0; // กำหนดจำนวนรอบการเรียนรู้เริ่มต้นที่ให้ค่าความคลาดเคลื่อนน้อยสุด
net_error[1] = 0.1; // กำหนดค่าเริ่มต้นให้กับค่าความคลาดเคลื่อนรวม
error_max = 0.00001; // กำหนดค่าความคลาดเคลื่อนเพื่อยุติการเรียนรู้
error_final = 500; // กำหนดค่าเริ่มต้นของความคลาดเคลื่อนสุดท้ายให้มีค่า 500 (ตัวอย่าง)
start = clock(); // เริ่มต้นคำนวณเวลาที่ใช้ในการเรียนรู้
while((net_error[i-1]>error_max)&&(i<=L_MAX)) // ตรวจสอบเงื่อนไขการยุติกระบวนการเรียนรู้
{
    for(m=0;m<6;m++) // คำนวณเอาต์พุตของชั้นซ่อนเร้น (X1j)
    {
        sum = 0.0;
        for(l=0;l<3;l++)
        { sum = sum+W[l][m][i-1]*x[l]; }
    }
}

```

```

sum = sum+Z1[m][i-1];
X1[m][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตของชั้นซ่อนเร้นเมื่อฟังก์ชันกระตุ้นเป็นแบบซิกมอยด์
}
for(n=0;n<6;n++) // จำนวนเอาต์พุตของชั้นเอาต์พุต (X2k)
{
sum = 0.0;
for(m=0;m<6;m++)
{ sum = sum+W1[m][n][i-1]*X1[m][i-1]; }
sum = sum+Z2[n][i-1];
X2[n][i-1] = 1.0/(1.0+exp(-sum)); // เอาต์พุตของชั้นเอาต์พุตเมื่อฟังก์ชันกระตุ้นเป็นแบบซิกมอยด์
}
uk1 = X2[0][i-1]; // ค่าระดับความสำคัญพลวัตของผลต่างเชิงมุมของลูกตุ้ม (DID-1)
uk2 = X2[1][i-1]; // ค่าระดับความสำคัญพลวัตของผลต่างความเร็วเชิงมุมของลูกตุ้ม (DID-2)
uk3 = X2[2][i-1]; // ค่าระดับความสำคัญพลวัตของมุมของลูกตุ้มอันล่าง (DID-3)
uk4 = X2[3][i-1]; // ค่าระดับความสำคัญพลวัตของความเร็วเชิงมุมลูกตุ้มอันล่าง (DID-4)
uk5 = X2[4][i-1]; // ค่าระดับความสำคัญพลวัตของตำแหน่งของตัวรอด (DID-5)
uk6 = X2[5][i-1]; // ค่าระดับความสำคัญพลวัตของความเร็วของตัวรอด (DID-6)
function1(uk1,uk2,uk3,uk4,uk5,uk6,&y_1,&y_2,&y_3,&y_4,&y_5,&y_6);
// เรียกฟังก์ชันเพื่อจำลองระบบและส่งกลับค่าความคลาดเคลื่อนของระบบ
y[0][i-1] = y_1; // รับค่าความคลาดเคลื่อนผลต่างเชิงมุมของลูกตุ้ม
y[1][i-1] = y_2; // รับค่าความคลาดเคลื่อนผลต่างความเร็วเชิงมุมของลูกตุ้ม
y[2][i-1] = y_3; // รับค่าความคลาดเคลื่อนมุมของลูกตุ้มอันล่าง
y[3][i-1] = y_4; // รับค่าความคลาดเคลื่อนความเร็วเชิงมุมของลูกตุ้มอันล่าง
y[4][i-1] = y_5; // รับค่าความคลาดเคลื่อนตำแหน่งของตัวรอด
y[5][i-1] = y_6; // รับค่าความคลาดเคลื่อนความเร็วของตัวรอด
for (n=0;n<6;n++)
{
if((X2[n][i-1]-X2[n][i-2])==0) // ถ้าผลต่างเอาต์พุตของโครงข่ายประสาทเทียมเป็น 0
{ delta_u = 0.000000001; } // กำหนด delta_u ≠ 0 เพื่อหลีกเลี่ยงปัญหาตัวหารเป็นศูนย์
else
{ delta_u = (X2[n][i-1]-X2[n][i-2]); } // ผลต่างเอาต์พุตของโครงข่ายประสาทเทียม
if(((y[n][i-1]-y[n][i-2])/delta_u)>=1000.0) // กำหนดขอบเขตของค่า delta_y ในกรณีค่าบวก
{ delta_y = 1000.0; }
else if(((y[n][i-1]-y[n][i-2])/delta_u)<=-1000.0) // กำหนดขอบเขตของค่า delta_y ในกรณีค่าลบ
{ delta_y = -1000.0; }
else
{ delta_y = ((y[n][i-1]-y[n][i-2])/delta_u); } // ผลต่างเอาต์พุตของระบบหารด้วยผลต่างเอาต์พุตของโครงข่าย
if((delta_y)>0.0) // ตรวจสอบเครื่องหมายของ delta_y
{ es[n] = (yd[n][i-1]-y[n][i-1]); } // ผลต่างเอาต์พุตที่ต้องการกับเอาต์พุตจริงของระบบกรณีเครื่องหมายบวก
else if((delta_y)<0.0)
{ es[n] = -1*(yd[n][i-1]-y[n][i-1]); } // ผลต่างเอาต์พุตที่ต้องการกับเอาต์พุตจริงของระบบกรณีเครื่องหมายลบ
}

```

```

else if((delta_y)==0.0)
    { es[n] = 0.0; } // ผลต่างเอาต์พุตที่ต้องการกับเอาต์พุตจริงของระบบกรณี delta_y เป็นศูนย์
del_1[n][i-1] = es[n]*X2[n][i-1]*(1.0-X2[n][i-1]); // พจน์ความคลาดเคลื่อนที่ชั้นเอาต์พุต ( $\delta_{1k}$ )
for(m=0;m<6;m++)
    {
        delta_W1[m][n][i-1] = n1*del_1[n][i-1]*X1[m][i-1]+(beta*delta_W1[m][n][i-2]);
        // ตัวปรับแก้ค่าน้ำหนักจากชั้นซ่อนเร้น ไปยังชั้นเอาต์พุต
        W1[m][n][i] = W1[m][n][i-1]+delta_W1[m][n][i-1];
        // ค่าน้ำหนักจากชั้นซ่อนเร้น ไปยังชั้นเอาต์พุตที่ปรับค่าแล้ว
    }
}
for (m=0;m<6;m++)
    {
        sum = 0.0;
        for (n=0;n<6;n++)
            { sum = sum+(del_1[n][i-1]*W1[m][n][i]); }
        del[m][i-1] = sum*X1[m][i-1]*(1-X1[m][i-1]); // พจน์ความคลาดเคลื่อนที่ชั้นซ่อนเร้น ( $\delta_j$ )
        for (l=0;l<3;l++)
            {
                delta_W[l][m][i-1] = nn*del[m][i-1]*x[l]+(beta*delta_W[l][m][i-2]);
                // ตัวปรับแก้ค่าน้ำหนักจากชั้นอินพุตไปยังชั้นซ่อนเร้น
                W[l][m][i] = W[l][m][i-1]+delta_W[l][m][i-1]; // ค่าน้ำหนักจากชั้นอินพุต ไปยังชั้นซ่อนเร้นที่ปรับค่าแล้ว
            }
    }
for (n=0;n<6;n++)
    { Z2[n][i] = Z2[n][i-1]+(n1z*del_1[n][i-1]); } // ปรับค่าไบแอสที่ชั้นเอาต์พุต (Z2k)
for (m=0;m<6;m++)
    { Z1[m][i] = Z1[m][i-1]+(nz*del[m][i-1]); } // ปรับค่าไบแอสที่ชั้นซ่อนเร้น (Z1j)
sum = 0.0;
for (n=0;n<6;n++)
    {
        sum = sum+((yd[n][i-1]-y[n][i-1])*(yd[n][i-1]-y[n][i-1]));
        // ผลรวมของค่าความคลาดเคลื่อนกำลังสอง
        yd[n][i] = yd[n][i-1]; // กำหนดให้เอาต์พุตของระบบที่ต้องการมีค่าเท่าเดิม
    }
net_error[i] = sum/2.0; // ความคลาดเคลื่อนรวมที่ได้จากโครงข่ายประสาทเทียม
if(net_error[i]<error_final)
    {
        error_final = net_error[i]; // กรณี net_error[i] น้อยกว่า error_final ที่กำหนดไว้
        iter = i; // จำนวนรอบการเรียนรู้ที่ค่าคลาดเคลื่อนของโครงข่ายประสาทเทียมต่ำสุด
    }
}

```



```

i++; // ปรับเพิ่มจำนวนรอบการเรียนรู้
} // สิ้นสุดกระบวนการเรียนรู้
D2M[0] = X2[0][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-1)
D2M[1] = X2[1][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-2)
D2M[2] = X2[2][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-3)
D2M[3] = X2[3][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-4)
D2M[4] = X2[4][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-5)
D2M[5] = X2[5][iter-1]; // ค่าระดับความสำคัญพลวัต (DID-6)

bestDID = fopen("G:best_DID.dat","wb"); // เปิดเพิ่มข้อมูลเพื่อเก็บค่าระดับความสำคัญพลวัต
fwrite(&D2M,sizeof(double),6,bestDID); // เขียนค่าระดับความสำคัญพลวัตลงในเพิ่มข้อมูล
fclose(bestDID); // ปิดเพิ่มข้อมูล
end = clock(); // สิ้นสุดการคำนวณเวลาในการเรียนรู้
printf("Execution time =%f sec", (end-start)/CLK_TCK); // แสดงเวลาที่ใช้ในการเรียนรู้
getch();
} // สิ้นสุดการทำงานของโปรแกรมหลัก

/*-----*/

```

เนื่องจากกระบวนการต่างๆ ของ function1 ในส่วนโปรแกรม *d\_nn\_ats\_fuzzy.cpp* นั้นคล้ายคลึงกับกระบวนการที่ปรากฏใน function2 ของโปรแกรม *double\_ats.cpp* ดังนั้นจึงจะเว้นการแสดงรายละเอียดของ function1 แต่จะแสดงส่วนของคำสั่งต่างๆ ที่แตกต่างกันของฟังก์ชันทั้งสองดังปรากฏในตารางที่ ฌ.1

ตารางที่ ฌ.1 คำสั่งต่างๆ ที่แตกต่างกันระหว่าง function2 (*double\_ats.cpp*) และ function1 (*d\_nn\_ats\_fuzzy.cpp*)

function2	function1
<pre> double function2(double DID_1,double DID_2,double DID_3, double DID_4,double DID_5,double DID_6)  y_1 = 0.0; y_2 = 0.0; y_3 = 0.0; y_4 = 0.0; y_5 = 0.0; y_6 = 0.0; y_1 = y_1+((fabs(x_2[i])/(15.0*pi/180.0))* (fabs(x_2[i])/(15.0*pi/180.0))); y_2 = y_2+((fabs(x3[i])/(15.0*pi/180.0))* (fabs(x3[i])/(15.0*pi/180.0))); y_3 = y_3+((fabs(x_1[i])/(250.0*pi/180.0))* (fabs(x_1[i])/(250.0*pi/180.0))); y_4 = y_4+((fabs(x4[i])/(250.0*pi/180.0))* (fabs(x4[i])/(250.0*pi/180.0))); y_5 = y_5+((fabs(x5[i])/1.0)*(fabs(x5[i])/1.0)); y_6 = y_6+((fabs(x6[i])/3.5)*(fabs(x6[i])/3.5)); } // สิ้นสุดการจำลองระบบ </pre>	<pre> void function1(double uk1,double uk2,double uk3,double uk4,double uk5,double uk6,double *y_1,double *y_2, double *y_3,double *y_4,double *y_5,double *y_6) *y_1 = 0.0; *y_2 = 0.0; *y_3 = 0.0; *y_4 = 0.0; *y_5 = 0.0; *y_6 = 0.0; *y_1 = *y_1+(fabs(x_2[i])) ; *y_2 = *y_2+(fabs(x3[i])); *y_3 = *y_3+(fabs(x_1[i])); *y_4 = *y_4+(fabs(x4[i])); *y_5 = *y_5+(fabs(x5[i])); *y_6 = *y_6+(fabs(x6[i])); } // สิ้นสุดการจำลองระบบ </pre>

ตารางที่ ฅ.1 (ต่อ)

function2	function1
<pre> if(i==num) { error_cost = (y_1+y_2+y_3+y_4+y_5)/1000.0; return(error_cost); } else { loop4: error_cost = 9999.0; return(error_cost); } } // สิ้นสุดการทำงานของ function2 </pre>	<pre> *y_1 = *y_1/1000.0; // ปรับลดระดับค่า *y_1 *y_2 = *y_2/1000.0; // ปรับลดระดับค่า *y_2 *y_3 = *y_3/1000.0; // ปรับลดระดับค่า *y_3 *y_4 = *y_4/1000.0; // ปรับลดระดับค่า *y_4 *y_5 = *y_5/1000.0; // ปรับลดระดับค่า *y_5 *y_6 = *y_6/1000.0; // ปรับลดระดับค่า *y_6 } // สิ้นสุดการทำงานของ function1 </pre>

ภาคผนวก ญ

บทความที่ได้รับการตีพิมพ์เผยแพร่

รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา

1. “Neuro-Tabu-Fuzzy Controller to Stabilize an Inverted Pendulum System”  
Proc. TENCON 2004 IEEE International Conference on Analog and Digital  
Techniques in Electrical Engineering. pp.562-565, 21-24 November 2004.
2. “Adaptive Tabu Search and Its Advance Application” The 8<sup>th</sup> Annual  
National Symposium on Computational Science and Engineering. pp.142-143, 21-23  
July 2004.
3. “การสร้างเสถียรภาพให้ระบบลูกตุ้มผกผันด้วยตัวควบคุมนิวโร-ฟัซซี่ชนิดกฎอินพุต  
เดียว” วารสารสงขลานครินทร์ฉบับวิทยาศาสตร์และเทคโนโลยี ปีที่ 27 ฉบับที่ 1 ม.ค.-ก.พ. 2548  
หน้า 101-121.

## ประวัติผู้เขียน

นางสาวสุดารัตน์ ขวัญอ่อน เกิดเมื่อวันที่ 2 มีนาคม พ.ศ. 2523 เกิดที่อำเภออุ้มถ้อง จังหวัดสุพรรณบุรี สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย จากโรงเรียนอุ้มถ้อง จังหวัดสุพรรณบุรี และสำเร็จการศึกษาระดับปริญญาตรี วิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) จากมหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2545 ขณะศึกษาระดับปริญญาโท เกษตรอนวิชาปฏิบัติการของสาขาวิศวกรรมไฟฟ้า สำนักวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ได้แก่ ปฏิบัติการอิเล็กทรอนิกส์วิศวกรรม ปฏิบัติการวิศวกรรมไฟฟ้า 1 ปฏิบัติการระบบไฟฟ้ากำลัง 2 ปฏิบัติการวงจรและอุปกรณ์ ปฏิบัติการวงจรอิเล็กทรอนิกส์ รวมทั้งปฏิบัติการระบบควบคุม และมีผลงานทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา 3 บทความ ดังรายชื่อที่ปรากฏในภาคผนวก ญ. มีความสนใจทางด้านการควบคุมระบบด้วยวิธีแบบชาญฉลาด และการประยุกต์ใช้ปัญญาประดิษฐ์ในงานควบคุม