

การระบุเอกลักษณ์ของระบบพลวัตด้วยการประมวลผลภาพ

นายเดชา พวงดาวเรือง

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2547

ISBN 974-533-399-9

**IDENTIFICATION OF DYNAMIC SYSTEMS
VIA IMAGE PROCESSING**

Mr. Deacha Puangdownreong

A Thesis Submitted in Fulfillment of the Requirements for the

Degree of Doctor of Philosophy in Electrical Engineering

Suranaree University of Technology

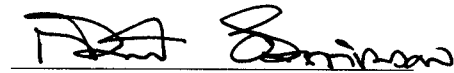
Academic Year 2004

ISBN 974-533-399-9

การระบุเอกลักษณ์ของระบบพลวัตด้วยการประมวลผลภาพ

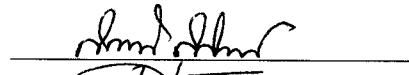
มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้บัณฑิตวิทยาลัยรับบัณฑิตศึกษาตามหลักสูตร
ปริญญาคุณวุฒิปบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์



(ผศ. ดร.กิติติ อັตถกัจมกคล)

ประธานกรรมการ



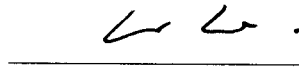
(รศ. น.ท. ดร.ศรารุฒิ สุจิตจร)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



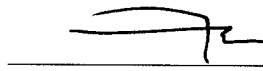
(พล.อ.ต. ดร.เพียร โตท่าโรง)

กรรมการ




(ผศ. ดร.อนันท์ อุ่นศิริไลย์)

กรรมการ



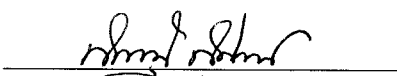
(ผศ. ดร.อาทิตย์ ศรีแก้ว)

กรรมการ



(อ. ดร.ชนัดชัย กุลวรรวานิชพงษ์)

กรรมการ



(รศ. น.ท. ดร.ศรารุฒิ สุจิตจร)

รองอธิการบดีฝ่ายวิชาการ



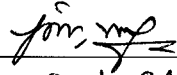
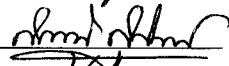
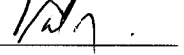
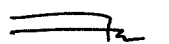
(รศ. น.อ. ดร.วรพจน์ ชำพิศ)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

เดชา พวงดาวเรือง : การระบุเอกลักษณ์ของระบบพลวัตด้วยการประมวลผลภาพ
(IDENTIFICATION OF DYNAMIC SYSTEMS VIA IMAGE
PROCESSING) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.สราวุฒิ สุจิตจร, 319 หน้า.
ISBN 974-533-399-9

งานวิจัยวิทยานิพนธ์นี้แนะนำเสนอแนวทางใหม่สำหรับการระบุเอกลักษณ์ระบบพลวัตโดยใช้เทคนิคการประมวลผลภาพ แนวทางดังกล่าวเป็นประโยชน์อย่างยิ่งในกรณีที่พลวัตของระบบไม่สามารถตรวจวัดได้โดยตัวตรวจรู้ทั่วไป โดยการใช้กล้องบันทึกภาพทำหน้าที่เป็นตัวตรวจรู้ ข้อมูลพลวัตของระบบจะได้รับการสกัดจากลำดับภาพที่บันทึกไว้ได้ วิทยานิพนธ์ได้นำเสนอรายละเอียดของการปรับเทียบกล้องบันทึกภาพ และการสกัดข้อมูลพลวัตของระบบจากลำดับภาพ การระบุเอกลักษณ์ระบบเพื่อคำนวณหาค่าพารามิเตอร์ของแบบจำลองจำเป็นต้องพึ่งพาอัลกอริทึมที่เหมาะสม สำหรับการระบุเอกลักษณ์ระบบเชิงเส้น ใช้เทคนิคการระบุเอกลักษณ์ระบบแบบดั้งเดิมซึ่งตั้งอยู่บนพื้นฐานของการวิเคราะห์แบบถดถอย การค้นหาแบบตามูเชิงปรับตัวได้รับการพัฒนาขึ้นเพื่อการระบุเอกลักษณ์ระบบเชิงเส้นและไม่เชิงเส้น การพิสูจน์คุณสมบัติการลู่เข้าหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัวได้รับการนำเสนอในวิทยานิพนธ์ แนวทางการระบุเอกลักษณ์ระบบที่นำเสนอได้ใช้ทดสอบกับระบบลูกตุ้มนาฬิกาพวงกรรณ และระบบกวดแกว่งท่อยาง ผลการระบุเอกลักษณ์ระบบมีความถูกต้องแม่นยำอย่างน่าพึงพอใจในระดับหนึ่ง

สาขาวิชา วิศวกรรมไฟฟ้า
ปีการศึกษา 2547

ลายมือชื่อนักศึกษา 
ลายมือชื่ออาจารย์ที่ปรึกษา 
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม 
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม 

DEACHA PUANGDOWNREONG : IDENTIFICATION OF DYNAMIC
SYSTEMS VIA IMAGE PROCESSING. THESIS ADVISOR : ASSOC.
PROF. SARAWUT SUJITJORN, Ph.D. 319 PP. ISBN 974-533-399-9

IDENTIFICATION/ADAPTIVE TABU SEARCH/SYSTEM MODELLING

This doctoral thesis proposes a new approach to system identification via image processing technique. The approach is particularly useful when the system of interest is unreachable by a conventional sensor. In this case, one or more cameras can be an alternative to conventional sensors. Dynamical information of the system can be extracted from the recorded images. The thesis presents the details of camera calibration, and information extraction from images. To identify model's parameters requires some appropriate algorithms. For linear models, conventional identification techniques based on regression analysis are appropriate. Adaptive Tabu Search has been developed to serve the identification of linear and nonlinear models. Its convergence proof is presented in this thesis. The proposed approaches have been tested against a cart-plus-pendulum system, and a vibrating tube, respectively. Satisfactory results have been achieved to a certain extent.

School of Electrical Engineering

Academic Year 2004

Student's Signature D. Puangdownreong

Advisor's Signature S. Sujitjorn

Co-advisor's Signature Pim Lota

Co-advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่าง ๆ ที่ได้กรุณาให้คำปรึกษา แนะนำ และช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

รองศาสตราจารย์ นาวาอากาศโท ดร.สรายุทธิ สุจิตจร ประธานกรรมการที่ปรึกษาวิทยานิพนธ์ ที่ได้กรุณามอบหมายงานวิจัยอันเป็นประโยชน์นี้ ได้อุทิศความรู้ สติปัญญา ความสามารถ ให้คำปรึกษาแนะนำ และให้กำลังใจ ด้วยความเมตตาปราณีต่อศิษย์เสมอมา

พลอากาศตรี ดร.เพียร โดท่าโรง และผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ศรีแก้ว กรรมการที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้คำปรึกษาแนะนำวิชาการทางด้านการระบุเอกลักษณ์ระบบ และการประมวลผลภาพ

ผู้ช่วยศาสตราจารย์ ดร.กิตติ อัดถกกิจมงคล รักษาการแทนหัวหน้าสาขาวิชาวิศวกรรมไฟฟ้า ที่กรุณาให้คำปรึกษาแนะนำวิชาการทางด้านการประมวลผลภาพ อาจารย์ ดร.ชนัดชัช กุลรวรานิชพงษ์ และอาจารย์กองพันธ์ อารีรักษ์ อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า ที่กรุณาให้คำแนะนำเกี่ยวกับการพิสูจน์การลู่เข้า และการโปรแกรมการค้นหาแบบตามู ตามลำดับ

อาจารย์วุฒิชัย สง่างาม อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคตะวันออกเฉียงเหนือ ที่ให้คำแนะนำเกี่ยวกับเทคนิคการประมวลผลภาพ

คุณกองพล อารีรักษ์ คุณเผด็จ เผ่าละออ คุณศิวา สอนสุจิตรา และนักศึกษาระดับบัณฑิตศึกษา สาขาวิชาวิศวกรรมไฟฟ้าทุกท่าน ที่คอยให้ความช่วยเหลือ และให้กำลังใจมาโดยตลอด

คุณประพล จาระตะคุ คุณสมิง เดิมพรมราช และเจ้าหน้าที่ศูนย์เครื่องมือวิทยาศาสตร์และเทคโนโลยีทุกท่าน ที่ให้ความช่วยเหลือเป็นอย่างดี

ท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา ที่ให้การอบรมเลี้ยงดู สนับสนุน และส่งเสริม การศึกษาอย่างดียิ่งเสมอมา ขอกราบขอบพระคุณครูบาอาจารย์ทุกท่าน ที่ให้การอบรมสั่งสอน และประสิทธิ์ประสาทวิชาความรู้ด้วยความเมตตาปราณีต่อศิษย์ และขอขอบคุณ สุกัญญา และสุคารา ภรรยาและบุตรสาว ที่คอยเป็นกำลังใจให้ตลอดมา และตลอดไป

เดชา พวงดาวเรือง

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ฉ
คำอธิบายสัญลักษณ์และคำย่อ.....	ณ
บทที่	
1 บทนำ.....	1
1.1 กล่าวนำ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.4 การจัดรูปเล่มของวิทยานิพนธ์.....	3
2 ปรัชญ่วรรณกรรมและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 กล่าวนำ.....	5
2.2 การระบุเอกลักษณ์ระบบพลวัตด้วยวิธีการดั้งเดิม.....	5
2.3 การระบุเอกลักษณ์ระบบด้วยเทคนิคทางปัญญาประดิษฐ์.....	6
2.4 การระบุเอกลักษณ์ระบบด้วยเทคนิคการประมวลผลภาพ.....	7
2.5 สรุป.....	13
3 การระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม.....	15
3.1 กล่าวนำ.....	15
3.2 แนวคิดการระบุเอกลักษณ์ระบบ.....	15
3.3 การระบุเอกลักษณ์แบบจำลองกล่องดำ.....	16
3.4 การระบุเอกลักษณ์ระบบด้วยโปรแกรมสำเร็จ.....	24
3.5 การวางแผนทดสอบระบบ.....	25

สารบัญ (ต่อ)

หน้า

3.6	สรุป	28
4	การประมวลผลภาพ.....	29
4.1	กล่าวนำ.....	29
4.2	การประมวลผลภาพ.....	29
4.2.1	ความเป็นมาของการประมวลผลภาพ.....	29
4.2.2	ภาพดิจิทัล	30
4.2.3	การแบ่งส่วนภาพ	32
4.3	การประยุกต์การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ	36
4.4	สรุป	38
5	การค้นหาแบบตามูเชิงปรับตัว.....	39
5.1	กล่าวนำ.....	39
5.2	การค้นหาแบบตามู	39
5.3	การค้นหาแบบตามูเชิงปรับตัว.....	41
5.4	การพิสูจน์การลู่เข้าหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว	47
5.5	การวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว	56
5.6	การประยุกต์การค้นหาแบบตามูเชิงปรับตัวเพื่อการระบุเอกลักษณ์ระบบ	63
5.7	สรุป	64
6	การทดสอบระบบลูกตุ้มนาฬิกาผนวกรถ	65
6.1	กล่าวนำ.....	65
6.2	การทดสอบระบบ CPP.....	65
6.3	ขั้นตอนการทดสอบระบบ CPP	68
6.4	ผลการทดสอบระบบ CPP	73
6.5	สรุป	94
7	การระบุเอกลักษณ์ระบบลูกตุ้มนาฬิกาผนวกรถ.....	95
7.1	กล่าวนำ.....	95
7.2	การระบุเอกลักษณ์ระบบ CPP เชิงเส้น	95

สารบัญ (ต่อ)

หน้า

7.2.1	การระบุเอกลักษณ์ระบบ CPP เชิงเส้นเพื่อพิจารณาอันดับที่เหมาะสม	98
7.2.2	การระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยอาศัยข้อมูลจากตัวตรวจรู้.....	99
7.2.3	การระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยอาศัยข้อมูล จากการประมวลผลภาพ.....	107
7.2.4	การระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยอาศัยข้อมูล จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน.....	108
7.2.5	การเปรียบเทียบผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้น.....	112
7.3	การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น.....	119
7.3.1	การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นเพื่อพิจารณาอันดับ ที่เหมาะสมของแรง f ที่เป็นพหุนามของ u	120
7.3.2	การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น โดยอาศัยข้อมูล จากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพ ที่ได้รับการแก้ไขความคลาดเคลื่อน	124
7.4	สรุป	125
8	การทดสอบระบบกวัดแกว่งตัวอย่าง.....	127
8.1	กล่าวนำ.....	127
8.2	การทดสอบระบบ VTS	127
8.3	ขั้นตอนการทดสอบระบบ VTS.....	130
8.4	ผลการทดสอบระบบ VTS.....	137
8.5	สรุป	140
9	การระบุเอกลักษณ์ระบบกวัดแกว่งตัวอย่าง	141
9.1	กล่าวนำ.....	141
9.2	การระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y	141
9.3	การระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x	151
9.4	สรุป	159

สารบัญ (ต่อ)

หน้า

10 บทสรุปและข้อเสนอแนะ	160
10.1 ข้อเสนอสรุป	160
10.2 ข้อเสนอแนะ	164
รายการอ้างอิง	167
ภาคผนวก	
ภาคผนวก ก. โปรแกรมการค้นหาแบบตามเชิงปรับตัว	176
ภาคผนวก ข. แบบจำลองทางคณิตศาสตร์ของระบบ CPP	193
ภาคผนวก ค. โปรแกรมประมวลผลภาพระบบ CPP	199
ภาคผนวก ง. โปรแกรมระบุเอกลักษณ์ระบบ CPP	204
ภาคผนวก จ. การควบคุมระบบ VTS	250
ภาคผนวก ฉ. การปรับเทียบกล้องวิดีโอดิจิทัล	258
ภาคผนวก ช. แบบจำลองทางคณิตศาสตร์ของระบบ VTS	272
ภาคผนวก ซ. โปรแกรมประมวลผลภาพระบบ VTS	288
ภาคผนวก ฌ. ระเบียบวิธีผลต่างสี่เหลี่ยม	294
ภาคผนวก ฉ. โปรแกรมระบุเอกลักษณ์ระบบ VTS	299
ภาคผนวก ฎ. รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์ รายการจดลิขสิทธิ์ และ CDROM ภาพวิดีโอ	316
ประวัติผู้เขียน	319

สารบัญตาราง

ตารางที่	หน้า
5.1 ผลของรัศมีการค้นหาเริ่มต้น (R)	60
5.2 ผลของจำนวนสมาชิกค่าใกล้เคียง (n)	60
5.3 ผลของจำนวนการซ้ำค่าคำตอบก่อนเรียกใช้กลไกย้อนรอยการค้นหา (n_re_Max).....	61
5.4 ผลของลำดับคำตอบที่เลือกใช้เป็นค่าเริ่มต้นในกลไกย้อนรอยการค้นหา (k^{th})	62
5.5 ผลของการปรับลดรัศมีการค้นหา ($\% reduction\ of\ R$).....	62
6.1 ค่าระดับแม่สีของกับองค์ประกอบต่าง ๆ ในระบบ CPP.....	70
6.2 ผลการแก้ไขความคลาดเคลื่อนของข้อมูลที่ได้จากการประมวลผลภาพของระบบ CPP	74
7.1 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เิงเส้นอันดับ 2 ถึง 20.....	101
7.2 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เิงเส้นโดยใช้ข้อมูลจากตัวตรวจรู้.....	102
7.3 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เิงเส้นโดยใช้ข้อมูล จากการประมวลผลภาพ	107
7.4 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เิงเส้นโดยใช้ข้อมูล จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน	112
7.5 ค่าความคลาดเคลื่อนของแบบจำลอง BJ ของระบบ CPP เิงเส้น.....	114
7.6 ผลการระบุเอกลักษณ์ค่าพารามิเตอร์ของแบบจำลองระบบ CPP ไม่เิงเส้น เมื่อแรง f เป็นพหุนามของอินพุต u อันดับ 2 ถึง 7	123
7.7 ผลการระบุเอกลักษณ์ระบบ CPP ไม่เิงเส้นโดยใช้ข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการ แก้ไขความคลาดเคลื่อน.....	126
8.1 ค่าระดับแม่สีของท่อยางในระบบ VTS	131
9.1 ผลการระบุเอกลักษณ์ค่าพารามิเตอร์ของแบบจำลองระบบ VTS ในระนาบ x, y	144
9.2 ผลการระบุเอกลักษณ์ค่าพารามิเตอร์ของแบบจำลองระบบ VTS ในระนาบ z, x	152

สารบัญรูป

รูปที่	หน้า
1.1	การระบุเอกลักษณ์ระบบ 1
1.2	การระบุเอกลักษณ์ระบบพลวัตโดยใช้การประมวลผลภาพ 2
2.1	การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ ⁽¹⁾ 9
2.2	โครงสร้างของระบบลูกค้อนมาฟิกาบนแผ่นกระดาษ ⁽¹⁾ 9
2.3	การประมวลผลภาพเพื่อระบุเอกลักษณ์ระบบแทนสั้น ⁽¹⁾ 10
2.4	การประมวลผลภาพระบบแทนสั้น ⁽¹⁾ (ก) ภาพต้นฉบับ (ข) การทำเครื่องหมายตำแหน่งที่เกิดการสั้น (ค) ผลการแบ่งส่วนภาพ (ง) ผลการสกัดตำแหน่ง 10
2.5	ผลการระบุตำแหน่งระบบแทนสั้น ⁽¹⁾ 11
2.6	โครงสร้างของระบบแผ่นยัดไม้เชิงเส้น ⁽¹⁾ 11
2.7	ผลการระบุเอกลักษณ์ระบบแผ่นยัดไม้เชิงเส้น ⁽¹⁾ 12
2.8	ระบบควบคุมครนยกโดยใช้การป้อนกลับด้วยภาพ ⁽²⁾ 13
3.1	การพัฒนาแบบจำลองทางคณิตศาสตร์ 16
3.2	โครงสร้างของแบบจำลองกล่องดำเชิงเส้น 19
3.3	การระบุเอกลักษณ์ระบบโดยใช้โปรแกรมสำเร็จรูป 24
4.1	ภาพดิจิทัลระดับเทาเทียบกับเมตริกซ์ 31
4.2	ภาพสีดิจิทัลเทียบกับเมตริกซ์ 31
4.3	การตรวจจับขอบภาพโดยใช้เกรเดียนและลาปลาเซียน 33
4.4	ฮิสโตแกรมของภาพที่มีวัตถุสีขาวอยู่บนฉากมืด 34
4.5	ฮิสโตแกรมของภาพที่มีวัตถุสีขาว 2 ชิ้นอยู่บนฉากมืด 35
4.6	ฮิสโตแกรมของวัตถุในภาพสีดิจิทัล 35
4.7	กระบวนการสกัดข้อมูลพลวัตด้วยการประมวลผลภาพแบบ off line 37
4.8	กระบวนการสกัดข้อมูลพลวัตด้วยการประมวลผลภาพแบบ on line 37
5.1	สุ่มเลือกค่าตอบเริ่มต้น, S_0 42
5.2	ผลิตค่าใกล้เคียงแบบสุ่ม 42

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.3	เปรียบเทียบค่าตอบเริ่มต้นกับค่าใกล้เคียง 42
5.4	ปรับค่าคำตอบเริ่มต้นใหม่ 42
5.5	ผลิตค่าใกล้เคียงแบบสุ่มรอบใหม่ 42
5.6	กลไกย้อนรอยการค้นหา 43
5.7	กลไกปรับรัศมีการค้นหา 44
5.8	แผนภูมิการทำงานของการค้นหาแบบตามูเชิงปรับตัว 45
5.9	แผนภูมิการทำงานของกลไกย้อนรอยการค้นหา 46
5.10	แผนภูมิการทำงานของกลไกปรับรัศมีการค้นหา 46
5.11	เส้นกราฟฟังก์ชัน TSF 58
5.12	พื้นผิวฟังก์ชัน BF 58
5.13	พื้นผิวฟังก์ชัน CF 58
5.14	การระบุเอกลักษณ์ระบบโดยการประยุกต์ใช้การค้นหาแบบตามูเชิงปรับตัว 63
6.1	ระบบ CPP 66
6.2	โครงสร้างระบบ CPP ของบริษัท Feedback 66
6.3	การทดสอบระบบ CPP เพื่อบันทึกข้อมูลโดยอาศัยตัวตรวจรู้ 67
6.4	การทดสอบระบบ CPP เพื่อบันทึกข้อมูลโดยอาศัยการประมวลผลภาพ 67
6.5	การบันทึกพลวัตของระบบ CPP 67
6.6	การจัดทำแถบสีให้กับระบบ CPP 68
6.7	ช่วงแถวที่มีการเคลื่อนที่ของรถและลูกตุ้ม 70
6.8	การกำหนดจุดต่ำสุดและจุดสูงสุดของวัตถุที่สนใจ 71
6.9	การคำนวณตำแหน่งรถและตำแหน่งเชิงมุมของลูกตุ้ม 72
6.10	ตัวอย่างการคำนวณตำแหน่งรถและตำแหน่งเชิงมุมของลูกตุ้มเมื่อมีการเคลื่อนที่ (ก) รถเคลื่อนซ้ายและลูกตุ้มเคลื่อนที่ตามเข็มนาฬิกา (ข) รถเคลื่อนซ้ายและลูกตุ้มเคลื่อนที่ทวนเข็มนาฬิกา (ค) รถเคลื่อนขวาและลูกตุ้มเคลื่อนที่ตามเข็มนาฬิกา (ง) รถเคลื่อนขวาและลูกตุ้มเคลื่อนที่ทวนเข็มนาฬิกา 75

สารบัญรูป (ต่อ)

รูปที่	หน้า
6.37 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 3 ก่อนแก้ไข	91
6.38 แบบจำลองความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 3.....	91
6.39 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 3 หลังแก้ไข.....	92
6.40 ตำแหน่งของรถจากการทดสอบตอนที่ 3 ก่อนและหลังแก้ไข.....	92
7.1 การระบุเอกลักษณ์ระบบ CPP	96
7.2 ข้อมูลอินพุตและเอาต์พุตสำหรับการระบุเอกลักษณ์ระบบ CPP เชิงเส้น (ก) ข้อมูลอินพุต (ข) ข้อมูลเอาต์พุต (ตำแหน่งเชิงมุมของลูกคีม).....	97
7.3 ผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้นอันดับ 2 ถึง 20.....	100
7.4 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้นอันดับ 2 ถึง 20	100
7.5 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เชิงเส้นอันดับ 2 ถึง 20 (ก) SAE (ข) SSE (ค) MSE (ง) RMSE	102
7.6 ผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้นโดยใช้ข้อมูลจากตัวตรวจรู้.....	104
7.7 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากตัวตรวจรู้.....	104
7.8 ผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้นโดยใช้ข้อมูลจากการประมวลผลภาพ	106
7.9 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากการประมวลผลภาพ	106
7.10 ผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้นโดยใช้ข้อมูลจากการประมวลผลภาพ ที่ได้รับการแก้ไขความคลาดเคลื่อน	110
7.11 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้นโดยใช้ข้อมูลจากการ ประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน	110
7.12 ผลการระบุเอกลักษณ์แบบจำลอง BJ ของระบบ CPP เชิงเส้น	111
7.13 ผลการตรวจสอบความถูกต้องของแบบจำลอง BJ ของระบบ CPP เชิงเส้น	111
7.14 ตำแหน่งโพลและซีโรของแบบจำลอง BJ ที่ใช้ข้อมูลจากตัวตรวจรู้.....	117
7.15 ตำแหน่งโพลและซีโรของแบบจำลอง BJ ที่ใช้ข้อมูลจากการประมวลผลภาพ	117

สารบัญรูป (ต่อ)

รูปที่	หน้า
7.16 ตำแหน่งโพลและซีโรของแบบจำลอง BJ ที่ใช้ข้อมูลจากการประมวลผลภาพ ที่ได้รับการแก้ไขความคลาดเคลื่อน	118
7.17 ผลตอบสนองในโดเมนความถี่ของแบบจำลอง BJ ของระบบ CPP เริงเส้น	118
7.18 ผลการระบุเอกลักษณ์ระบบเมื่อแรง f เป็นพหุนามของอินพุต u อันดับ 2 ถึง 7.....	121
7.19 ผลการตรวจสอบความถูกต้องของแบบจำลองเมื่อแรง f เป็นพหุนามของ อินพุต u อันดับ 2 ถึง 7.....	121
7.20 ผลการระบุเอกลักษณ์ระบบ CPP ไม่เริงเส้น.....	122
7.21 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP ไม่เริงเส้น	122
8.1 ระบบ VTS	128
8.2 ภาพถ่ายระบบ VTS	128
8.3 การติดตั้งกล้องวีดีโอดิจิทัลเพื่อบันทึกภาพการเคลื่อนที่ของระบบ VTS	129
8.4 อุปกรณ์บันทึกภาพ	129
8.5 ช่วงเวลาที่ท้อยางเคลื่อนที่เมื่อพิจารณาจากด้านข้างโดยกล้องวีดีโอดิจิทัลตัวที่ 1	132
8.6 ช่วงเวลาที่ท้อยางเคลื่อนที่เมื่อพิจารณาจากด้านบนโดยกล้องวีดีโอดิจิทัลตัวที่ 2.....	132
8.7 ตัวอย่างผลการคำนวณตำแหน่งของท้อยางจากกล้องตัวที่ 1 (ก) ภาพปรากฏความเพี้ยน (ข) ภาพภายหลังแก้ไขความเพี้ยน (ค) ผลการสกัดข้อมูล (ง) ผลการคำนวณ ตำแหน่งของท้อยาง (จ) จัดสรรข้อมูลตำแหน่งของท้อยาง	134
8.8 ตัวอย่างผลการคำนวณตำแหน่งของท้อยางจากกล้องตัวที่ 2 (ก) ภาพปรากฏความเพี้ยน (ข) ภาพภายหลังแก้ไขความเพี้ยน (ค) ผลการสกัดข้อมูล (ง) ผลการคำนวณ ตำแหน่งของท้อยาง (จ) จัดสรรข้อมูลตำแหน่งของท้อยาง	135
8.9 ตำแหน่งของท้อยางจากกล้องวีดีโอดิจิทัลตัวที่ 1 ก่อนและหลังการแก้ไขความเพี้ยน	136
8.10 ตำแหน่งของท้อยางจากกล้องวีดีโอดิจิทัลตัวที่ 2 ก่อนและหลังการแก้ไขความเพี้ยน	136
8.11 สัญญาณอินพุตของระบบ VTS	137
8.12 การเคลื่อนที่ของท้อยางในระนาบ x, y เทียบกับเวลา t	138
8.13 การเคลื่อนที่ของท้อยางในระนาบ z, x เทียบกับเวลา t	138
9.1 การระบุเอกลักษณ์ระบบ VTS	142

สารบัญรูป (ต่อ)

รูปที่	หน้า
จ.3	แผนภาพวงจรควบคุมระบบ VTS 252
จ.4	วงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับ..... 254
จ.5	วงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับจำนวน 13 ตัว..... 254
จ.6	สัญญาณทวิภาคสุ่มเทียมสำหรับควบคุมระบบ VTS..... 257
ฉ.1	ความเพี้ยนเชิงรัศมี (ก) ความเพี้ยนรูปถัง (ข) ความเพี้ยนรูปหมอนปีกเข็ม..... 260
ฉ.2	ความเพี้ยนเชิงศูนย์..... 260
ฉ.3	เรขาคณิตการฉายภาพ 261
ฉ.4	ภาพแผ่นกระดานตรวจสอบแบบเรียบที่ได้จากกล้องวิดีโอจิจิตอลทั้ง 2 ตัว (ก) ภาพจากกล้องวิดีโอจิจิตอลตัวที่ 1 (ข) ภาพจากกล้องวิดีโอจิจิตอลตัวที่ 2 267
ฉ.5	ผลการแก้ไขความเพี้ยนของภาพที่ได้จากกล้องตัวที่ 1 (ก) ภาพปรากฏความเพี้ยนรูปหมอนปีกเข็ม (ข) ภาพภายหลังการแก้ไขความเพี้ยน 271
ฉ.6	ผลการแก้ไขความเพี้ยนของภาพที่ได้จากกล้องตัวที่ 2 (ก) ภาพปรากฏความเพี้ยนรูปถัง (ข) ภาพภายหลังการแก้ไขความเพี้ยน..... 271
ช.1	ระบบ VTS แบบ 2 มิติ 274
ช.2	ระบบ VTS แบบ 2 มิติเมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง 276
ช.3	ระบบ VTS แบบ 2 มิติเมื่อกำหนดให้ $\sin(\theta) \approx \theta$ 276
ช.4	ระบบ VTS แบบ 3 มิติ 278
ช.5	ระบบ VTS แบบ 3 มิติเมื่อพิจารณาแบบ 2 มิติ (ก) พิจารณาแรงในระนาบ x, y (ข) พิจารณาแรงในระนาบ y, z 278
ช.6	ระบบ VTS แบบ 3 มิติเมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง (ก) พิจารณาแรงในระนาบ x, y (ข) พิจารณาแรงในระนาบ y, z 282
ช.7	ไดอะแกรมของแรงที่กระทำบนพื้นที่ $\Delta x \Delta y$ 283
ช.8	ภาพตัดของแรง $S \Delta y$ ในระนาบ z, x 283
ฉ.1	กริดสำหรับคำนวณผลเฉลยของสมการคลื่น..... 296
ฉ.2	การคำนวณผลเฉลย $y_{i,j+1}$ 297

คำอธิบายสัญลักษณ์และคำย่อ

a	คือ องค์ประกอบของภาพ หรือจุดภาพ, ขอบเขตความยาวของตัวอย่าง
\mathbf{a}	คือ เวกเตอร์ \mathbf{a}
α_c	คือ ค่าสัมประสิทธิ์ความเฉ
a_i	คือ ค่าสัมประสิทธิ์ของพหุนาม
$a_{i,j}$	คือ องค์ประกอบของเมตริกซ์ของภาพดิจิทัลระดับเทา
$a_{i,j,k}$	คือ องค์ประกอบของเมตริกซ์ของภาพสีดิจิทัล
a_n	คือ ค่าสัมประสิทธิ์ของพหุนามส่วนของฟังก์ชันถ่ายโอน
a_y	คือ ความเร่งในแนวแกน y
a_{11}, \dots, a_{34}	คือ ค่าพารามิเตอร์หรือองค์ประกอบของ \mathbf{A}
\mathbf{A}	คือ เมตริกซ์ของแบบจำลอง DLT
A, B	คือ ตำแหน่งปลายทั้งสองข้างของตัวอย่าง
AC	คือ เกณฑ์ปรารธนา
AI	คือ ปัญญาประดิษฐ์
AR	คือ กลไกปรับรัศมีการค้นหา
ARMAX	คือ แบบจำลอง ARMAX
ARX	คือ แบบจำลอง ARX
ATS	คือ การค้นหาแบบตามูเชิงปรับตัว
b	คือ ขอบเขตของเวลาที่พิจารณา
$best_neighbor$	คือ ค่าตอบเริ่มต้น S_0
$best_neighbor1$	คือ ค่าใกล้เคียงที่ทำให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด S_1
b_i, c_i, d_i, f_i	คือ ค่าพารามิเตอร์ของแบบจำลองกล่องดำ
b_n	คือ ค่าสัมประสิทธิ์ของพหุนามเศษของฟังก์ชันถ่ายโอน
b_1, b_2	คือ ค่าสัมประสิทธิ์ความเพี้ยนเชิงเส้นของ \mathbf{B}
B	คือ สีน้าเงิน
\mathbf{B}	คือ เมตริกซ์การชดเชยความเพี้ยนเชิงเส้นเนื่องจากความต่างของสเกล และความไม่ตั้งฉากกันระหว่างแกนภาพ
BF	คือ ฟังก์ชันโบฮาเซฟสกี

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

BJ	คือ แบบจำลอง BJ
BT	คือ กลไกย้อนรอยการค้นหา
c	คือ สัญญาณเอาต์พุต, สัญญาณเอาต์พุต (ตำแหน่งเชิงมุมของลูกตุ้ม), ค่าสัมประสิทธิ์ของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก
cc	คือ จุดमुखสำคัญ
cost	คือ ค่า cost
c_{max}	คือ ตำแหน่งคอติมันต์ที่เป็นจุดขวาสุดของวัตถุ
c_{min}	คือ ตำแหน่งคอติมันต์ที่เป็นจุดซ้ายสุดของวัตถุ
c_{ref}	คือ ตำแหน่งคอติมันต์อ้างอิง
Cart_pos	คือ ตำแหน่งรถที่เคลื่อนที่
CCD	คือ charge coupled device
CCP	คือ ระบบลูกตุ้มนาฬิกาผนวกรถ
CF	คือ ฟังก์ชันวงกลม
CG_c	คือ ตำแหน่งคอติมันต์ของจุดศูนย์กลางของวัตถุ
CG_c_cart	คือ ตำแหน่งคอติมันต์ของจุดศูนย์กลางของรถเมื่อมีการเคลื่อนที่
CG_c_pend	คือ ตำแหน่งคอติมันต์ของจุดศูนย์กลางของลูกตุ้มเมื่อมีการเคลื่อนที่
CG_r	คือ ตำแหน่งแฉกของจุดศูนย์กลางของวัตถุ
CG_r_cart	คือ ตำแหน่งแฉกของจุดศูนย์กลางของรถเมื่อมีการเคลื่อนที่
CG_r_pend	คือ ตำแหน่งแฉกของจุดศูนย์กลางของลูกตุ้มเมื่อมีการเคลื่อนที่
Count	คือ จำนวนรอบการค้นหา
ds	คือ ความยาวของท่อขางที่พิจารณาในช่วงสั้น ๆ
dx	คือ ระยะ x ที่พิจารณาในช่วงสั้น ๆ
dx	คือ ความเพี้ยนเชิงผิวสัมผัส
dy	คือ ระยะ y ที่พิจารณาในช่วงสั้น ๆ
$D(s)$	คือ พหุนามส่วนของแบบจำลอง
DF_i	คือ ตัวประกอบการลดทอนที่มีการค้นหา
DLT	คือ แบบจำลองการแปลงเชิงเส้นโดยตรง
D_u, D_v	คือ ค่าสัมประสิทธิ์การแปลงระบบพิกัดภาพ
e	คือ สัญญาณความคลาดเคลื่อน

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

$e(t)$	คือ สัญญาณรบกวน, สัญญาณรบกวนขาว
f	คือ แรง, พหุนามของอินพุต u , ความถี่สูงสุด, ความยาวโฟกัส
$f_B(x, y)$	คือ ค่าสีน้ำเงินของจุดภาพ x, y
f_c	คือ ความถี่ของการชักตัวอย่างของกล้อง
fc	คือ ความยาวโฟกัส
$f_G(x, y)$	คือ ค่าสีเขียวของจุดภาพ x, y
f_N	คือ เวกเตอร์ f_N , ความถี่ในควิสต์
$f_R(x, y)$	คือ ค่าสีแดงของจุดภาพ x, y
f_s	คือ ความถี่ของการชักตัวอย่างของตัวตรวจรู้
$f(x)$	คือ ค่า cost ของคำตอบ x , เงื่อนไขเริ่มต้นของ $y(x, t)$, เงื่อนไขเริ่มต้นของ $z(x, t)$
$f(x, y)$	คือ ฟังก์ชันของความเข้มแสงแบบ 2 มิติ, ค่าสีของจุดภาพ x, y
F	คือ แรงที่กระทำกับตัวอย่าง, ฟังก์ชันวัตถุประสงค์, แรง 2 มิติ
F	คือ เมตริกซ์ความยาวโฟกัส
FEM	คือ ระเบียบวิธีไฟไนต์เอลิเมนต์
F_x	คือ แรงในแนวแกน x
F_y	คือ แรงในแนวแกน y
g	คือ ความเร่งเนื่องจากแรงโน้มถ่วง
$g_{cart,pend}(x, y)$	คือ ค่าตรรกะที่แสดงความเป็นรถและลูกตุ้มของจุดภาพ x, y
g_i	คือ ค่าความคลาดเคลื่อนของฟังก์ชันวัตถุประสงค์
$g_{tube_cam1}(x, y)$	คือ ค่าตรรกะที่แสดงความเป็นตัวอย่างของจุดภาพ x, y ที่ได้จากกล้องตัวที่ 1
$g_{tube_cam2}(x, y)$	คือ ค่าตรรกะที่แสดงความเป็นตัวอย่างของจุดภาพ x, y ที่ได้จากกล้องตัวที่ 2
g_y	คือ ความเร่งเนื่องจากแรงโน้มถ่วงที่กระทำในแนวแกน y
$g(x)$	คือ ฟังก์ชัน, เงื่อนไขเริ่มต้นของ $y_i(x, t)$, เงื่อนไขเริ่มต้นของ $z_i(x, t)$
$g(x, y)$	คือ ค่าตรรกะของจุดภาพ x, y
G	คือ สีเขียว
$G(q, \theta)$	คือ ฟังก์ชันถ่ายโอนของแบบจำลองเอาต์พุตที่อิสระต่อสัญญาณรบกวน
$G(s)$	คือ แบบจำลองในรูปของฟังก์ชันถ่ายโอน

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

h	คือ ช่วงเวลาประวิงของรีจิสเตอร์เลื่อน, Δx
$H(q, \theta)$	คือ ฟังก์ชันถ่ายโอนของแบบจำลองการรบกวน
i	คือ จำนวนข้อมูล
i^{th}	คือ รอบการค้นหาใด ๆ
$i(x, y)$	คือ ความสว่างของแหล่งพลังงานแสง
I	คือ ข้อมูลที่ได้จากการประมวลผลภาพ
\hat{I}	คือ ข้อมูลที่ได้จากการประมวลผลภาพภายหลังการปรับแก้
<i>Iteration</i>	คือ จำนวนการสำรวจคำตอบในแต่ละ Ψ
k	คือ จำนวนคำตอบวงแคบเฉพาะถิ่นจำกัดใน Ω , จำนวนรีจิสเตอร์เลื่อนก่อนการป้อนกลับครั้งแรก, Δt
kc	คือ ค่าสัมประสิทธิ์ความเพี้ยนของภาพ
k^{th}	คือ ลำดับคำตอบที่ใช้เป็นค่าเริ่มต้นในกลไก <i>BT</i>
k_1, k_2	คือ ค่าสัมประสิทธิ์ความเพี้ยนเชิงรัศมี
KK	คือ เมตริกซ์บรรจุค่า fc , cc , และ alpha_c
K_{cart}	คือ พลังงานจลน์ของรถ
K_D	คือ อัตราขยายที่ให้การประมาณแบบจำลองของส่วนเพิ่มกำลัง
K_M	คือ อัตราขยายที่ให้การประมาณแบบจำลองของมอเตอร์ไฟฟ้ากระแสตรง
K_{pend}	คือ พลังงานจลน์ของลูกตุ้ม
$K(\cdot)$	คือ พลังงานจลน์
l	คือ ความยาวของก้านลูกตุ้ม
L	คือ เมตริกซ์ L
$L(\cdot)$	คือ ฟังก์ชันลากรางจ์
m	คือ มวลของลูกตุ้ม, จำนวนสมาชิกทั้งหมดใน Ψ , มวลของตัวอย่าง
m_{11}, \dots, m_{33}	คือ ค่าพารามิเตอร์แบบเอ็กทริเนซิกของแบบจำลองกลิ้ง
M	คือ มวลของรถ, จำนวนรอบการค้นหาที่มากเพียงพอ
M	คือ เมตริกซ์การหมุน
MATS	คือ multiple-adaptive tabu search
MSE	คือ ค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

n	คือ จำนวนสมาชิกทั้งหมดใน Ω , จำนวนสมาชิกค่าใกล้เคียง, จำนวนรีจิสเตอร์เลื่อนทั้งหมด, อันดับของฟังก์ชันถ่ายโอน
nb, nc, nd, nf	คือ อันดับของพหุนาม
nk	คือ เวลาที่จังหวะต่าง ๆ, ช่วงเวลาประวิงของการชักตัวอย่าง
n_{re}	คือ ตัวนับการซ้ำค่าคำตอบ
n_{re_Max}	คือ จำนวนการซ้ำค่าคำตอบสูงสุดที่ยอมให้เกิดขึ้น
n_re_Max	คือ จำนวนการซ้ำค่าคำตอบก่อนเรียกใช้กลไก BT
N	คือ จำนวนคำตอบที่สุ่มได้รอบ S_0 , จำนวนสมาชิกทั้งหมดของ $N_\rho(x_0)$, คาบเวลาของสัญญาณทวิภาคสุ่มเทียม, จำนวนจุดภาพ
NARMAX	คือ แบบจำลอง NARMAX
NCREE	คือ ศูนย์วิจัยวิศวกรรมแผ่นดินไหวแห่งชาติ ประเทศไต้หวัน
$N_\rho(x_0)$	คือ ค่าใกล้เคียง
$N(s)$	คือ พหุนามเศษของแบบจำลอง
ODE	คือ สมการอนุพันธ์สามัญ
OE	คือ แบบจำลอง OE
p	คือ จำนวนคำตอบสะสมใน S
p_y	คือ โมเมนต์ตัมเชิงเส้น
p_ϕ	คือ โมเมนต์ตัมเชิงมุม
p_1, p_2	คือ ค่าสัมประสิทธิ์ความเพี้ยนเชิงผิวสัมผัส
<i>pole</i>	คือ โพล
P	คือ ค่าความน่าจะเป็น
P	คือ จุดภาพ
P_i	คือ จุดวัตถุ
PC	คือ คอมพิวเตอร์ส่วนบุคคล
PDE	คือ สมการอนุพันธ์ย่อย
PID+Q	คือ ตัวควบคุม PID+Q
PRBS	คือ สัญญาณทวิภาคสุ่มเทียม
PTS	คือ การค้นหาแบบตามูเชิงน่าจะเป็น

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

PWM	คือ การมอดูเลตความกว้างพัลส์
$P(z^{-1})$	คือ พหุนาม z ของวงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับ
q	คือ ตัวปฏิบัติการเลื่อน
r	คือ รัศมีกลาง, สัญญาณอินพุตอ้างอิง, ค่าสัมประสิทธิ์ของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก
\mathbf{r}	คือ ขนาดของ \mathbf{x}, \mathbf{y}
r_i	คือ ขนาดของจุดภาพ \tilde{u}_i, \tilde{v}_i
r_{max}	คือ ตำแหน่งแถวที่เป็นจุดต่ำสุดของวัตถุ, ตำแหน่งแถวที่เป็นจุดต่ำสุดของท่อยาง
r_{min}	คือ ตำแหน่งแถวที่เป็นจุดสูงสุดของวัตถุ, ตำแหน่งแถวที่เป็นจุดสูงสุดของท่อยาง
r_{ref}	คือ ตำแหน่งแถวอ้างอิง
$r(x, y)$	คือ การสะท้อนของวัตถุรับแสง
R	คือ รัศมีการค้นหา, รัศมีการค้นหาเริ่มต้น, สีแดง, กริดคำนวณผลเฉลย
$R^{(new)}$	คือ ค่ารัศมีการค้นหาหลังการปรับลด
$R^{(old)}$	คือ ค่ารัศมีการค้นหาก่อนการปรับลด
R_N	คือ เมตริกซ์ R_N
Rc_1, Rc_2, ...	คือ เมตริกซ์การหมุน
R, G, B	คือ ค่าระดับแม่สีแดง เขียว น้ำเงิน
RMSE	คือ ค่ารากที่สองของค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง
RTS	คือ การค้นหาแบบตามูเชิงรีแอกทีฟ หรือการค้นหาแบบตามูชนิดตอบโต้ได้
s_u	คือ ตัวประกอบสเกล
S	คือ ข้อมูลที่ได้จากตัวตรวจรู้
S	คือ การสะสมของคำตอบ, แรง 3 มิติ
SAE	คือ ผลรวมของค่าความคลาดเคลื่อนสัมบูรณ์
SD	คือ ค่าเบี่ยงเบนมาตรฐาน
SSE	คือ ผลรวมค่าความคลาดเคลื่อนกำลังสอง
S_x	คือ แรงในแนวแกน x
S_y	คือ แรงในแนวแกน y

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

S_z	คือ แรงในแนวแกน z
S_0	คือ ค่าตอบเริ่มต้น
S_1	คือ ค่าใกล้เคียงที่ทำให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด
$S_1(r)$	คือ เซตที่บรรจุค่าใกล้เคียง
$S\Delta x$	คือ แรงที่กระทำในระนาบ y, z
$S\Delta y$	คือ แรงที่กระทำในระนาบ z, x
t	คือ เวลา
$theta$	คือ ตำแหน่งเชิงมุมของลูกตุ้มเมื่อมีการเคลื่อนที่
T	คือ จุดเปลี่ยนค่า, คาบเวลาของการชักตัวอย่าง
\mathbf{T}	คือ เมตริกซ์การเลื่อนขนาน
T_{B1}, T_{B2}	คือ จุดเปลี่ยนค่าที่ใช้พิจารณาค่าสีน้ำเงิน $f_B(x, y)$ ของจุดภาพ x, y
T_{G1}, T_{G2}	คือ จุดเปลี่ยนค่าที่ใช้พิจารณาค่าสีเขียว $f_G(x, y)$ ของจุดภาพ x, y
T_{R1}, T_{R2}	คือ จุดเปลี่ยนค่าที่ใช้พิจารณาค่าสีแดง $f_R(x, y)$ ของจุดภาพ x, y
$Time(x)$	คือ เวลาที่ใช้ในการสำรวจคำตอบ x
T_j	คือ ตำแหน่งของตัวอย่างเมื่อเปรียบเทียบกับตำแหน่งอ้างอิง
T'_j	คือ จุดกึ่งกลางของกลุ่มจุดภาพที่เป็นตัวอย่างในแต่ละคอสมันน์
T'_1	คือ ตำแหน่งของตัวอย่างคอสมันน์ที่ 1 (ตำแหน่งอ้างอิง)
$\mathbf{Tc}_1, \mathbf{Tc}_2, \dots$	คือ เวกเตอร์การเลื่อนขนาน
TC	คือ เกณฑ์ยุติการค้นหา
TL	คือ รายชื่อตาม
TS	คือ การค้นหาแบบตาม
TSF	คือ ฟังก์ชันผลรวมตรีโกณมิติแบบไม่สมมาตร
TV tuner adapter	คือ ตัวปรับทีวี
u	คือ สัญญาณอินพุต, สัญญาณอินพุตของมอเตอร์ไฟฟ้ากระแสตรง, จำนวนสมาชิกใน $N_p(x_t)$ ที่ทำให้ $f(x) < f(x_t)$
$u(t)$	คือ สัญญาณอินพุต
u_0, v_0	คือ จุดमुखสำคัญ, จุดศูนย์กลางภาพ
u_i, v_i	คือ จุดภาพในระบบพิกัดภาพ

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

u'_i, v'_i	คือ จุดภาพที่ได้รับการแปลงให้อยู่ในรูปทั่วไป
\tilde{u}_i, \tilde{v}_i	คือ จุดภาพที่ได้รับการฉายภาพด้วยแบบจำลองรูเข็ม
$\hat{u}_i, \hat{v}_i, r_i, a_i, G$	คือ ตัวประกอบการคำนวณการฉายภาพกลับ
U_i, V_i	คือ จุดภาพที่ได้จากการสังเกต
v_{cart}	คือ ความเร็วของรถ
v_{pend}	คือ ความเร็วของลูกตุ้ม
$v_{pend_horizontal}$	คือ ความเร็วของลูกตุ้มในแนวแกนราบ
$v_{pend_vertical}$	คือ ความเร็วของลูกตุ้มในแนวแกนตั้ง
\mathbf{V}	คือ เมตริกซ์การเลื่อนของจุดกำเนิดภาพ
$V_N(\theta)$	คือ ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย
$V'_N(\theta)$	คือ อนุพันธ์อันดับหนึ่งของ $V_N(\theta)$ เทียบกับ θ
$V''_N(\theta)$	คือ อนุพันธ์อันดับสองของ $V_N(\theta)$ เทียบกับ θ
$V(\cdot)$	คือ พลังงานศักย์
VCR	คือ เครื่องเล่นวีดีโอ
VTs	คือ ระบบกวัดแกว่งตัวอย่าง
w	คือ จำนวนสมาชิกทั้งหมดใน Λ
w_i	คือ องค์ประกอบภาพในแนวแกน z
x	คือ คำตอบ, สัญญาณ x , ระยะ x , ความยาวของตัวอย่าง
\tilde{x}	คือ คำตอบที่ผลิตขึ้นแบบสุ่ม
x^*	คือ คำตอบวงกว้าง
x'	คือ คำตอบที่ให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด
\hat{x}	คือ คำตอบวงแคบเฉพาะถิ่น
$x^{(i)}$	คือ คำตอบครั้งที่ i
$x^{(i+1)}$	คือ คำตอบครั้งที่ $i+1$
x_B	คือ จุดขอบของปริภูมิ
\mathbf{x}_d	คือ จุดภาพที่รวมความเพี้ยนของเลนส์
x_h	คือ คำตอบใน TL ที่ได้รับการเลือกจากกลไก BT
\mathbf{x}_n	คือ จุดภาพที่ได้จากการฉายภาพตามแบบจำลองรูเข็ม

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

$x_{t=0}$	คือ คำตอบเริ่มต้น
x_0	คือ คำตอบปัจจุบัน
$x_{0,Count}$	คือ คำตอบเริ่มต้น
x, y	คือ คู่ลำดับในปริภูมิของภาพ, จุดภาพ, ระนาบ x, y
x, y, z	คือ ระบบพิกัด, มิติการเคลื่อนที่ของระบบ VTS, มิติการเคลื่อนที่ของระบบพลวัต
x_i, y_i, z_i	คือ ระบบพิกัดกล้อง
x_0, y_0, z_0	คือ จุดศูนย์กลางการฉายภาพ
x_pixel	คือ จุดภาพในระบบพิกัดภาพ
XX	คือ ระบบพิกัดวัตถุ
XX_c	คือ ระบบพิกัดกล้อง
X;Y;Z	คือ ตำแหน่งของจุดภาพในระบบพิกัดวัตถุ XX
X_c;Y_c;Z_c	คือ ตำแหน่งของจุดภาพในระบบพิกัดกล้อง XX_c
X_i, Y_i, Z_i	คือ ตำแหน่งของ P_i
y	คือ ตำแหน่งของรถ, สัญญาณ y , ตำแหน่งของท่ออย่างที่พิจารณาแบบ 2 มิติ, ระยะ y , ผลเฉลยของการคำนวณที่แสดงตำแหน่งท่ออย่างในระนาบ x, y
y_i	คือ ข้อมูลเอาต์พุตในระนาบ x, y ที่ได้จากการทดสอบตำแหน่ง i
\dot{y}	คือ อนุพันธ์อันดับหนึ่งของ y เทียบกับ t
\ddot{y}	คือ อนุพันธ์อันดับสองของ y เทียบกับ t
\hat{y}_i	คือ ข้อมูลเอาต์พุตในระนาบ x, y ที่ได้จากแบบจำลองตำแหน่ง i
$y(t)$	คือ สัญญาณเอาต์พุต
$y(x, t)$	คือ ตำแหน่งของท่ออย่างที่แปรผันตามระยะ x และเวลา t , ตำแหน่งของท่ออย่างในระนาบ x, y , ข้อมูลเอาต์พุตในระนาบ x, y ที่ได้จากการทดสอบ
$\hat{y}(t \theta)$	คือ การทำนายค่าเอาต์พุต $y(t)$
$\hat{y}(x, t)$	คือ ข้อมูลเอาต์พุตในระนาบ x, y ที่ได้จากแบบจำลอง
$y_t(x, t)$	คือ อนุพันธ์ย่อยอันดับหนึ่งของ $y(x, t)$ เทียบกับ t
$y_{tt}(x, t)$	คือ อนุพันธ์ย่อยอันดับสองของ $y(x, t)$ เทียบกับ t
$y_{xx}(x, t)$	คือ อนุพันธ์ย่อยอันดับสองของ $y(x, t)$ เทียบกับ x

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

z	คือ ตำแหน่งของท่ออย่างที่พิจารณาแบบ 3 มิติ, ผลเฉลยของการคำนวณที่แสดงตำแหน่งท่ออย่างในระนาบ z, x
z_i	คือ ข้อมูลเอาต์พุตในระนาบ z, x ที่ได้จากการทดสอบตำแหน่ง i
\hat{z}_i	คือ ข้อมูลเอาต์พุตในระนาบ z, x ที่ได้จากแบบจำลองตำแหน่ง i
z^{-1}	คือ ช่วงเวลาประวิงของรีจิสเตอร์เลื่อน
$z(x, t)$	คือ ตำแหน่งของท่ออย่างในระนาบ z, x , ข้อมูลเอาต์พุตในระนาบ z, x ที่ได้จากการทดสอบ
$\hat{z}(x, t)$	คือ ข้อมูลเอาต์พุตในระนาบ z, x ที่ได้จากแบบจำลอง
$z_t(x, t)$	คือ อนุพันธ์ย่อยอันดับหนึ่งของ $z(x, t)$ เทียบกับ t
$z_{tt}(x, t)$	คือ อนุพันธ์ย่อยอันดับสองของ $z(x, t)$ เทียบกับ t
$z_{xx}(x, t)$	คือ อนุพันธ์ย่อยอันดับสองของ $z(x, t)$ เทียบกับ x
$z(x, y, t)$	คือ ตำแหน่งของท่ออย่างที่แปรผันตามระยะ x ระยะ y และเวลา t
$zero$	คือ ซีโร
z, x	คือ ระนาบ z, x
α	คือ จำนวนรอบของการค้นหาทั้งหมดที่ลดลง
α_n	คือ ค่าสัมประสิทธิ์ของแบบจำลองความคลาดเคลื่อน
α, β, γ, D	คือ ค่าสัมประสิทธิ์ของฟังก์ชันลากรานจ์
β	คือ มุมของแผ่นกระดานที่กระทำกับแกนตั้ง
δ	คือ จำนวนน้อย ๆ ของค่าความคลาดเคลื่อนมากที่สุดที่ยอมให้เกิดของ x'_i
δ_i	คือ ค่าฟังก์ชันวัตถุประสงค์กำหนด
$\delta u_i^{(r)}, \delta v_i^{(r)}$	คือ ความเพี้ยนเชิงรัศมี
$\delta u_i^{(t)}, \delta v_i^{(t)}$	คือ ความเพี้ยนเชิงผิวสัมผัส
ε	คือ ค่าฟังก์ชันวัตถุประสงค์, จำนวนน้อย ๆ ของค่าความคลาดเคลื่อนมากที่สุดที่ยอมให้เกิดของ $f(x'_i)$
$\varepsilon(t, \theta)$	คือ ค่าความคลาดเคลื่อน
ξ_i	คือ ระยะห่างระหว่าง x'_i กับ \hat{x}
γ	คือ ค่าสูงสุดที่ยอมให้เกิดขึ้นได้
λ	คือ ตัวประกอบสเกลรวม
$\hat{\lambda}_N$	คือ ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

μ	คือ ช่วงของการปรับการคำนวณแบบวนซ้ำ, ตัวประกอบการปรับรัศมี, มวลของท่ออย่างต่อเนื่องหนึ่งหน่วย
$\eta(t)$	คือ แบบจำลองเอาต์พุตที่อิสระต่อสัญญาณรบกวน
$\omega(t)$	คือ แบบจำลองการรบกวน
ω, φ, κ	คือ มุมของออยเลอร์
$\varphi(t)$	คือ เวกเตอร์ถดถอย
ρ	คือ ค่ารัศมี
ϕ	คือ มุมการแกว่งไกวของลูกตุ้ม
$\dot{\phi}$	คือ อนุพันธ์อันดับหนึ่งของ ϕ เทียบกับ t
$\ddot{\phi}$	คือ อนุพันธ์อันดับสองของ ϕ เทียบกับ t
σ	คือ มวลของท่ออย่างต่อเนื่องหนึ่งหน่วยในแนวแกน x และ y
θ	คือ เวกเตอร์ของพารามิเตอร์ในแบบจำลอง, มุมของท่อ, มุมของลูกตุ้มที่กระทำกับแกนตั้ง, การสะสมของปฏิภูมิการค้นหาย่อย
$\hat{\theta}^{(i)}$	คือ ค่าพารามิเตอร์ในแบบจำลองที่ได้จากการคำนวณครั้งที่ i
$\hat{\theta}^{(i+1)}$	คือ ค่าพารามิเตอร์ในแบบจำลองที่ได้จากการคำนวณครั้งที่ $i+1$
$\hat{\theta}_N$	คือ ค่าประมาณของ θ
θ_x	คือ มุมของท่อในแนวแกน x
θ_y	คือ มุมของท่อในแนวแกน y
\emptyset	คือ เซตว่าง, เส้นผ่าศูนย์กลาง
Δc	คือ ค่าผลต่างของตำแหน่งคอคอดค้ำของรถและลูกตุ้ม
Δr	คือ ค่าผลต่างของตำแหน่งแฉกของรถและลูกตุ้ม, ความเพี้ยนเชิงรัศมี
Δt	คือ เวลา t ที่พิจารณาในช่วงสั้น ๆ, ความเพี้ยนเชิงผิวสัมผัส
Δx	คือ ระยะ x ที่พิจารณาในช่วงสั้น ๆ
Δy	คือ ตำแหน่งของท่อที่พิจารณาในช่วงสั้น ๆ
$\Delta \theta$	คือ มุมของท่อที่พิจารณาในช่วงแคบ ๆ
$\Delta \theta_x$	คือ มุมของท่อในแนวแกน x ที่พิจารณาในช่วงแคบ ๆ
$\Delta \theta_y$	คือ มุมของท่อในแนวแกน y ที่พิจารณาในช่วงแคบ ๆ
Λ	คือ ปฏิภูมิคอนเวกซ์

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

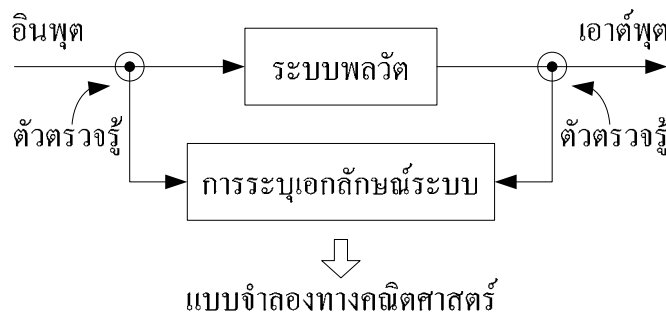
Ψ	คือ ปริภูมิการค้นหาย่อย
Ω	คือ ปริภูมิการค้นหาที่มีขอบเขตจำกัด
\mathfrak{R}^+	คือ ปริภูมิของจำนวนจริงบวก
Γ	คือ แบบจำลองความคลาดเคลื่อน
$O(k^2)$	คือ ส่วนเหลือของผลต่างแบบตรงกลางเพื่อประมาณเทอม $y_{tt}(x,t)$
$O(h^2)$	คือ ส่วนเหลือของผลต่างแบบตรงกลางเพื่อประมาณเทอม $y_{xx}(x,t)$
\oplus	คือ ปฏิบัติการของพีชคณิตมอดุโล
% reduction of R	คือ ปริมาณการลดรัศมีการค้นหาเปรียบเทียบกับรัศมีการค้นหาก่อนหน้า

บทที่ 1

บทนำ

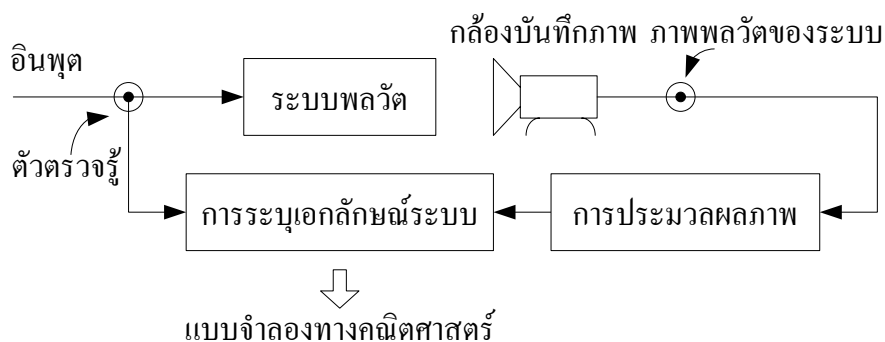
1.1 กล่าวนำ

การระบุเอกลักษณ์ระบบ (system identification) เป็นกระบวนการที่สำคัญอย่างยิ่ง กระบวนการหนึ่งในบริบทระบบควบคุม เป็นกระบวนการคำนวณหาแบบจำลองทางคณิตศาสตร์ที่สมบูรณ์ของระบบ โดยอาศัยข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบที่ได้จากการวัดโดยตรงจากตัวตรวจรู้ที่ติดตั้งอยู่ตามตำแหน่งต่าง ๆ อย่างเหมาะสม ดังอาจแสดงได้ในรูปที่ 1.1



รูปที่ 1.1 การระบุเอกลักษณ์ระบบ

ในวงรอบการควบคุมทั่วไป ข้อมูลความรู้เกี่ยวกับอินพุตของระบบมักจะสามารถตรวจวัดได้จากแหล่งจ่าย แต่ข้อมูลความรู้เกี่ยวกับเอาต์พุตของระบบอาจไม่เป็นเช่นนั้น โดยเฉพาะอย่างยิ่งในระบบที่ไม่สามารถติดตั้งตัวตรวจรู้เข้ากับระบบได้ เช่น ในระบบที่มีการเคลื่อนไหวและอ่อนไหวอยู่ตลอดเวลา หรือในระบบที่มีพื้นที่ในการติดตั้งตัวตรวจรู้จำกัดและอันตราย ในกรณีที่พฤติกรรมทางพลวัตของระบบสามารถสังเกตได้ด้วยสายตาของมนุษย์ตลอดเวลา แนวทางหนึ่งที่เป็นไปได้สำหรับการตรวจจับพฤติกรรมทางพลวัตของระบบที่มีการเคลื่อนไหวและอ่อนไหวคือการใช้กล้องบันทึกภาพทำหน้าที่เป็นตัวตรวจรู้ โดยทำการบันทึกภาพการเคลื่อนไหวของระบบในช่วงเวลาที่กำหนด ดังแสดงในรูปที่ 1.2 จากนั้นอาศัยเทคนิคการประมวลผลภาพที่เหมาะสมเพื่อสกัดข้อมูลพลวัตของระบบจากลำดับของภาพที่บันทึกได้ ข้อมูลดังกล่าวคือข้อมูลความรู้เกี่ยวกับเอาต์พุตของระบบนั่นเอง และในท้ายที่สุดการระบุเอกลักษณ์ระบบพลวัตจะสามารถดำเนินการสำเร็จลงได้โดยอาศัยข้อมูลความรู้เกี่ยวกับอินพุตจากตัวตรวจรู้ และข้อมูลความรู้เกี่ยวกับเอาต์พุตจากภาพหรือลำดับของภาพ



รูปที่ 1.2 การระบุเอกลักษณ์ระบบพลวัตโดยใช้การประมวลผลภาพ

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อระบุเอกลักษณ์ระบบพลวัตโดยใช้เทคนิคการประมวลผลภาพ
2. เพื่อพัฒนาเทคนิคการประมวลผลภาพสำหรับสกัดข้อมูลพลวัตจากลำดับของภาพเพื่อการระบุเอกลักษณ์ระบบ
3. เพื่อพัฒนาการค้นหาแบบตามูเชิงปรับตัวสำหรับค้นหาพารามิเตอร์ในแบบจำลองพิสตุงก์การลู่เข้าผลเฉลย และวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว
4. เพื่อสร้างระบบกวดแกว่งท่อย่างต้นแบบ และพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบดังกล่าว เพื่อใช้เป็นระบบทดสอบการระบุเอกลักษณ์ระบบด้วยการประมวลผลภาพ

จากวัตถุประสงค์ของการวิจัยที่กล่าวมา สามารถแบ่งกรอบงานวิจัยออกได้เป็น 4 ส่วนด้วยกันคือ การพัฒนาอัลกอริทึมของการค้นหาแบบตามูเชิงปรับตัว การพัฒนาเทคนิคการประมวลผลภาพ การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบพลวัต และการระบุเอกลักษณ์ระบบพลวัต ซึ่งในแต่ละกรอบงานวิจัยจะมีกิจกรรมต่าง ๆ ดังต่อไปนี้

กรอบงานวิจัยที่ 1 การพัฒนาอัลกอริทึมของการค้นหาแบบตามูเชิงปรับตัว

1. พัฒนาอัลกอริทึมของการค้นหาแบบตามูเชิงปรับตัว
2. พิสตุงก์การลู่เข้าผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว
3. วิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว

กรอบงานวิจัยที่ 2 การพัฒนาเทคนิคการประมวลผลภาพ

1. ศึกษาการแบ่งส่วนภาพ และกลไกการกำหนดจุดเปลี่ยนค่า
2. ศึกษา และทำการปรับเทียบกล้องบันทึกภาพ
3. ทำการสร้างลำดับของภาพจากกล้องบันทึกภาพ
4. ทำการสกัดข้อมูลพลวัตจากลำดับของภาพ

กรอบงานวิจัยที่ 3 การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบพลวัต

1. ทำการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบลูกตุ้มนาฬิกาผนวกรวด
2. ทำการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบกวัดแกว่งท่อยาง

กรอบงานวิจัยที่ 4 การระบุเอกลักษณ์ระบบพลวัต

1. ทำการระบุเอกลักษณ์ระบบลูกตุ้มนาฬิกาผนวกรวด
2. ทำการระบุเอกลักษณ์ระบบกวัดแกว่งท่อยาง

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถพัฒนาเทคนิคการประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบพลวัตได้
2. สามารถพัฒนาการค้นหาแบบตามูเชิงปรับตัวเพื่อใช้ค้นหาค่าพารามิเตอร์ของแบบจำลองทางคณิตศาสตร์ได้

1.4 การจัดรูปเล่มของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย 10 บท และ 11 ภาคผนวก นอกจากบทที่ 1 ซึ่งเป็นบทนำแล้ว ยังมีบทอื่น ๆ ที่มีเนื้อหาโดยสรุปดังต่อไปนี้

บทที่ 2 กล่าวถึงการปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง เพื่อใช้เป็นข้อมูลอ้างอิงและเป็นแนวทางสำหรับการดำเนินงานวิจัยต่อไป

บทที่ 3 กล่าวถึงทฤษฎีการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม การระบุเอกลักษณ์แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ การระบุเอกลักษณ์ระบบด้วยโปรแกรมสำเร็จ และการวางแผนทดสอบระบบ

บทที่ 4 กล่าวถึงการประมวลผลภาพ การแบ่งส่วนภาพ การระบุตำแหน่งวัตถุในภาพด้วยการกำหนดจุดเปลี่ยนค่า และการประยุกต์การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ

บทที่ 5 กล่าวถึงการค้นหาแบบตามู (TS) การค้นหาแบบตามูเชิงปรับตัว (ATS) การพิสูจน์การลู่เข้าผลเฉลย และการวิเคราะห์สมรรถนะการค้นหาผลเฉลยของ ATS รวมทั้งการประยุกต์ ATS เพื่อการระบุเอกลักษณ์ระบบ

บทที่ 6 กล่าวถึงการทดสอบระบบลูกตุ้มนาฬิกาผนวกรวด (CPP) การบันทึกผลการทดสอบผ่านตัวตรวจรู้ และกล้องบันทึกภาพ การสกัดข้อมูลพลวัตของระบบ CPP จากภาพที่บันทึกได้ การแก้ไขความคลาดเคลื่อนของข้อมูลที่ได้จากกล้องบันทึกภาพ และให้การอภิปรายผล

บทที่ 7 กล่าวถึงการระบุเอกลักษณ์ระบบ CPP เชิงเส้นและไม่เชิงเส้น การวิเคราะห์แบบจำลองที่ได้ และให้การอภิปรายผล

บทที่ 8 กล่าวถึงการทดสอบระบบกวัดแกว่งท่อยาง (VTS) การบันทึกผลการทดสอบผ่านกล้องบันทึกภาพ การสกัดข้อมูลพลวัตของระบบ VTS จากภาพที่บันทึกได้ และอภิปรายผล

บทที่ 9 กล่าวถึงการระบุเอกลักษณ์ระบบ VTS การวิเคราะห์แบบจำลองที่ได้ และให้การอภิปรายผล

บทที่ 10 เป็นบทสรุปของงานวิจัย และข้อเสนอแนะ

ภาคผนวก ก. แสดงรายละเอียดโปรแกรม ATS ที่พัฒนาขึ้นด้วย MATLAB และ C

ภาคผนวก ข. แสดงรายละเอียดการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ CPP

ภาคผนวก ค. แสดงรายละเอียดโปรแกรมการประมวลผลภาพของระบบ CPP

ภาคผนวก ง. แสดงรายละเอียดโปรแกรมการระบุเอกลักษณ์ระบบ CPP

ภาคผนวก จ. แสดงรายละเอียดการควบคุมระบบ VTS

ภาคผนวก ฉ. แสดงรายละเอียดการปรับเทียบกล้องวีดีโอดิจิทัล

ภาคผนวก ช. แสดงรายละเอียดการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS

ภาคผนวก ซ. แสดงรายละเอียดโปรแกรมการประมวลผลภาพของระบบ VTS

ภาคผนวก ฌ. แสดงรายละเอียดการคำนวณเชิงตัวเลขตามระเบียบวิธีผลต่างสืบเนื่อง

ภาคผนวก ฎ. แสดงรายละเอียดโปรแกรมการระบุเอกลักษณ์ระบบ VTS

ภาคผนวก ฏ. แสดงรายชื่อบทความวิจัยที่ได้รับการตีพิมพ์ รายการจดลิขสิทธิ์ และ CDROM ภาพวีดีโอ

บทที่ 2

ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 กล่าวนำ

ในบทที่ 2 นี้จะกล่าวถึงการปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับการระบุเอกลักษณ์ระบบพลวัตโดยใช้เทคนิคการประมวลผลภาพ โดยสามารถแบ่งออกได้เป็นสามแนวทาง คือ การระบุเอกลักษณ์ระบบพลวัตด้วยวิธีการดั้งเดิม การระบุเอกลักษณ์ระบบด้วยเทคนิคทางปัญญาประดิษฐ์ และการระบุเอกลักษณ์ระบบด้วยเทคนิคการประมวลผลภาพ โดยใช้ฐานข้อมูลที่เป็นแหล่งสะสมรายงานวิจัยและวรรณกรรมที่เกี่ยวข้องทางวิศวกรรมศาสตร์และเทคโนโลยี อาทิ เช่น ฐานข้อมูลจาก IEEE IEE (IEL online) IEICE และ ScienceDirect เป็นต้น ผลจากการปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องจะถูกใช้เป็นข้อมูลอ้างอิง และเป็นแนวทางสำหรับการดำเนินงานวิจัยต่อไป

2.2 การระบุเอกลักษณ์ระบบพลวัตด้วยวิธีการดั้งเดิม

การระบุเอกลักษณ์ระบบพลวัตได้มีการศึกษาวิจัยมาอย่างช้านาน และแตกแขนงออกไปอย่างกว้างขวาง โดยอาจเริ่มต้นศึกษาจากตำราการระบุเอกลักษณ์ระบบพื้นฐาน อาทิเช่น Ljung (1987), Ljung และ Glad (1994), Landau (1990), และ Johansson (1993) เป็นต้น ซึ่งกล่าวถึงการระบุเอกลักษณ์ระบบพลวัตด้วยวิธีการดั้งเดิม ว่าจำเป็นต้องอาศัยข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ จากการทดสอบ และการตรวจวัดอย่างเหมาะสม การคำนวณหาค่าพารามิเตอร์ของแบบจำลองทางคณิตศาสตร์ ใช้แนวทางการถดถอยเชิงเส้น (linear regression) ที่อาศัยการคำนวณเชิงตัวเลขแบบกำลังสองน้อยที่สุด (least squares) ดัชนีบ่งชี้คุณภาพของแบบจำลองทางคณิตศาสตร์ ใช้ค่าความคลาดเคลื่อนระหว่างค่าเอาต์พุตจริงที่ได้จากการทดสอบ และค่าเอาต์พุตที่ได้มาจกแบบจำลองทางคณิตศาสตร์ แนวความคิดและขั้นตอนการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม ได้รับการทบทวนไว้เป็นระยะ ๆ ดังปรากฏในวรรณกรรมวิจัยต่าง ๆ เช่น Ljung (1994) และ Ljung (2001) เป็นต้น

การระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิมในช่วงทศวรรษที่ 90 ยึดแนวทางการคำนวณในลักษณะการถดถอยเชิงเส้น ที่อาศัยการคำนวณเชิงตัวเลขแบบกำลังสองน้อยที่สุด เป็นตัวปฏิบัติการหลัก การระบุเอกลักษณ์ระบบในลักษณะวงรอบเปิด ดังปรากฏใน Van Den Hof (1991), Vries และ Van Den Hof (1995), และ Van Den Hof, Heuberger, และ Bokor (1994) และ

การระบุเอกลักษณ์ระบบในลักษณะวงรอบปิด ดังปรากฏใน Van Den Hof, Schrama, และ Bosgar (1992), Forssell และ Ljung (2000), และ Ljung และ Forssell (1999) แบบจำลองทางคณิตศาสตร์ที่ได้จากการระบุเอกลักษณ์ระบบทั้งสองแนวทางข้างต้น มีลักษณะเป็นแบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ ในรูปของพหุนามและฟังก์ชันถ่ายโอน นอกจากนี้ยังมีการระบุเอกลักษณ์ระบบที่ให้แบบจำลองทางคณิตศาสตร์ในรูปของปริภูมิสถานะเชิงเส้น ดังปรากฏใน Ljung (1991), Viberg และคณะ (1991), และ McKelvey, Akcay, และ Ljung (1996) เป็นต้น

มีคณะผู้วิจัยอีกกลุ่มหนึ่งที่ทำให้ความสนใจในเรื่องของการพัฒนาลักษณะการคำนวณและรูปแบบของแบบจำลองทางคณิตศาสตร์ เพื่อการระบุเอกลักษณ์ระบบ ดังเช่น Niu, Ljung, และ Bjorck (1996) ได้ดำเนินงานระบุเอกลักษณ์ระบบโดยใช้เทคนิคหลายแบบจำลอง (multiple models) ที่ตั้งอยู่ในรูปของเมตริกซ์ เปรียบเทียบกับการใช้วิธีกำลังสองน้อยที่สุดแบบเดิม ซึ่งพบว่าวิธีดังกล่าวสะดวกกว่า และอ่อนตัวต่อการอนุวัตมากกว่า งานวิจัยของ Wallin, Isaksson, และ Ljung (2000) ได้มีการดัดแปลงการคำนวณแบบวิธีกำลังสองน้อยที่สุด เพื่อการคำนวณค่าพารามิเตอร์ให้กับแบบจำลองชนิด ARX เมื่อข้อมูลเกิดความไม่สมบูรณ์ งานวิจัยของ Forssell และ Ljung (1998) และ Forssell และ Ljung (2000) ได้พัฒนาเทคนิคการประมาณค่าพารามิเตอร์ในแบบจำลองชนิด OE และ BJ ให้สามารถดำเนินการระบุเอกลักษณ์ระบบที่ขาดเสถียรภาพได้ หรือในงานวิจัยของ Sun, Liu, และ Sano (1997) ได้พัฒนาการระบุเอกลักษณ์ระบบโดยใช้เทคนิคการชักตัวอย่างสัญญาณเอาต์พุตด้วยความถี่ที่สูงกว่าการชักตัวอย่างสัญญาณอินพุต และดำเนินการในลักษณะวงรอบปิด นอกจากนี้ยังมีงานวิจัยของ Boaghe และคณะ (2002) ได้ทำการระบุเอกลักษณ์เครื่องยนต์กังหันแก๊สในโดเมนเวลาและโดเมนความถี่ โดยอาศัยแบบจำลองกล่องดำไม่เชิงเส้นชนิด NARMAX รูปแบบของแบบจำลอง NARMAX ที่เสนอโดยคณะผู้วิจัยดังกล่าว อยู่ในรูปของ ผลคูณระหว่างแบบจำลอง ARMAX กับฟังก์ชันไม่เชิงเส้นใด ๆ ที่สอดคล้องกับกายภาพและพลวัตของระบบ

2.3 การระบุเอกลักษณ์ระบบด้วยเทคนิคทางปัญญาประดิษฐ์

จากการปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง พบว่ามีงานวิจัยที่ประยุกต์เทคนิคทางปัญญาประดิษฐ์ (artificial intelligent, AI) เพื่อใช้ค้นหาค่าพารามิเตอร์ในแบบจำลองทางคณิตศาสตร์ ทั้งที่เป็นเชิงเส้นและไม่เชิงเส้น ดังปรากฏใน Ljung และ Sjoberg (1992), Karaboga และ Kalinli (1997) และ Lee, Jun, และ Kim (2000) ซึ่งได้ประยุกต์โครงข่ายประสาท (neural network) เพื่อการระบุเอกลักษณ์ระบบ โครงข่ายประสาทดังกล่าวได้รับการฝึกฝน โดยการค้นหาแบบตาบ (Tabu search, TS) งานวิจัยของ Lin และ Miller (2000)

ได้ประยุกต์การค้นหาแบบตามูเพื่อระบุเอกลักษณ์ระบบทางเคมี และงานวิจัยของกองพัน อารีรักษ์ (2544) ได้ใช้การค้นหาแบบตามูเพื่อระบุเอกลักษณ์ความไม่เป็นเชิงเส้นของระบบสองมวล ความเฉื่อยที่ปรากฏกำหนดองเชิงกล แบบจำลองทางคณิตศาสตร์ที่ได้จากการระบุเอกลักษณ์ระบบ อธิบายพจน์ความไม่เป็นเชิงเส้นแบบอิมพัลในวงรอบการควบคุม

2.4 การระบุเอกลักษณ์ระบบด้วยเทคนิคการประมวลผลภาพ

การวินิจฉัยการเคลื่อนที่ของวัตถุจากลำดับของภาพ มีปรากฏตั้งแต่ช่วงกลางทศวรรษที่ 70 ซึ่งมีเป้าหมายหลักคือ เพื่อระบุตำแหน่งและท่าทาง (orientation) ของวัตถุในปริภูมิ 3 มิติ งานวิจัยในช่วงทศวรรษที่ 80 ดังปรากฏใน Roach และ Aggarwal (1980) และ Weng, Huang, และ Ahuja (1987) ใช้อัลกอริทึมพื้นฐานอย่างกำลังสองน้อยที่สุด เพื่อคำนวณหาแบบจำลองที่สามารถระบุตำแหน่งและท่าทางของวัตถุประเภทแข็งเกร็ง (rigid) และได้มีการเรียบเรียงไว้อีกครั้งใน Aggarwal และ Nandhakumar (1988) งานวิจัยที่มีการประยุกต์การประมวลผลภาพหรือลำดับของภาพเพื่อระบุพลวัตของระบบ ดังปรากฏใน Yeung และ Li (1996) เป็นการระบุเอกลักษณ์ระบบไม่เชิงเส้นหลายอินพุตโดยใช้ข้อมูลภาพ แบบจำลองที่ได้อยู่ในรูปของปริภูมิสถานะ อัลกอริทึมที่ใช้คำนวณเป็นโครงข่ายเวฟเล็ก ข้อมูลที่ใส่อยู่ในรูปของอนุกรมเวลา (time series) และแบบจำลองที่ได้ถูกใช้เป็นเครื่องมือสำหรับพยากรณ์พายุฝนในช่วงเวลาสั้น ๆ งานวิจัยของ Forstor, Jezek, และ Sohn (1998) ได้ใช้ภาพถ่ายแผ่นน้ำแข็งจากเรดาร์ที่ติดอยู่กับดาวเทียม เพื่อพิจารณาพลวัตของธารน้ำแข็ง อันจะส่งผลกระทบต่อเปลี่ยนแปลงของภูมิอากาศในอนาคต ข้อมูลจากภาพที่ถูกนำไปใช้คือความเร็วของแผ่นธารน้ำแข็ง (ice velocity) และอัตราความเครียด (strain rate) ซึ่งพิจารณาจากคอนทัวร์ของภาพถ่าย นอกจากนี้ยังมีคณะวิจัยที่นำทีมโดย Ghosh ดังที่ปรากฏใน Ghosh และ Loucks (1996) และ Ghosh, Inaba, และ Takahashi (2000) ได้พัฒนาพีชคณิตที่เป็นประโยชน์ต่อการระบุตำแหน่งและท่าทางของวัตถุในปริภูมิ 3 มิติจากข้อมูลภาพ ซึ่งเป็นประโยชน์ต่อการมองเห็นของเครื่องจักรกล (machine vision) งานวิจัยที่มีการพัฒนาตัวสังเกต (observer) การเคลื่อนที่ของวัตถุ และระบุพลวัตของระบบ ดังปรากฏใน Chen, Kano, และ Fukuda (2000) มีพื้นฐานการออกแบบตัวสังเกตโดยใช้เทคนิคโหมดการเลื่อนไถล (sliding mode) งานวิจัยของ Rousseau, Desrochers และ Krouglicof (2001) ได้นำเสนอการระบุเอกลักษณ์ค่าพารามิเตอร์ของหุ่นยนต์อุตสาหกรรม โดยใช้ภาพที่ได้จากกล้องที่ติดอยู่ตรงปลายแขนหุ่นยนต์ อัลกอริทึมได้รับการพัฒนามาจากวิธีกำลังสองน้อยที่สุด ค่าพารามิเตอร์ที่ได้จากการระบุเอกลักษณ์ระบบ แสดงระยะห่างขององค์ประกอบต่าง ๆ ของหุ่นยนต์กับวัตถุ งานวิจัยที่ใช้กล้อง 2 ตัว เพื่อคอยตรวจจับวัตถุแข็งเกร็งที่เคลื่อนที่ในปริภูมิ เพื่อระบุตำแหน่งและท่าทางของวัตถุ ดังปรากฏใน Kano และ Ghosh (2000) มีความซับซ้อนของอัลกอริทึมที่ใช้อยู่มาก และ

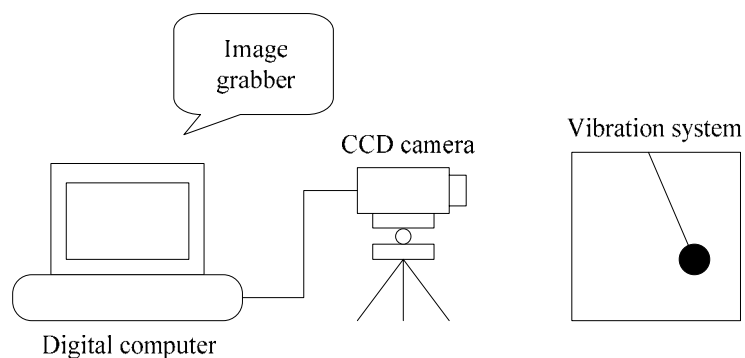
แบบจำลองที่ได้มีลักษณะตายตัว ยังมีงานวิจัยเกี่ยวกับการประมาณทิศทางการเคลื่อนที่ของวัตถุจากความสว่างของเส้นขอบภาพแบบ 2 มิติ ดังปรากฏใน Guerrero และ Sagues (2001) ซึ่งให้ผลออกมาน่าพอใจ เมื่อวัตถุเคลื่อนที่ในช่วงสั้น ๆ แต่มิได้นำเสนอการทดสอบแบบ 3 มิติ นอกจากนี้ยังมีรายงานวิจัยที่ได้เปรียบเทียบเทคนิคการประมาณการเคลื่อนที่ของวัตถุทั้งแบบ 2 มิติและ 3 มิติ จากลำดับของภาพ ดังปรากฏใน Stiller และ Konrad (1999)

ในปี ค.ศ. 2004 มีงานวิจัยของ Chung และคณะ (2004) ได้ทำการระบุเอกลักษณ์ระบบพลวัตด้วยเทคนิคการประมวลผลภาพ งานวิจัยดังกล่าวมีความน่าสนใจมาก เพราะมีความใกล้เคียงกับงานวิจัยที่จะดำเนินการ ซึ่งโครงสร้างการประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบแสดงดังรูปที่ 2.1 มีการใช้กล้องบันทึกภาพดิจิทัลทำหน้าที่เป็นตัวตรวจรู้ เพื่อทำการบันทึกพลวัตของระบบ ภาพจากกล้องบันทึกภาพได้รับการอ่านเข้าสู่คอมพิวเตอร์ผ่านอุปกรณ์รับภาพ (image grabber) และได้รับการสกัดรายละเอียดพลวัต ด้วยเทคนิคการประมวลผลภาพที่เหมาะสม เพื่อระบุเอกลักษณ์ระบบต่อไป Chung และคณะ ได้นำเสนอการระบุเอกลักษณ์ระบบด้วยเทคนิคการประมวลผลภาพไว้ 3 การทดลองด้วยกัน ดังต่อไปนี้

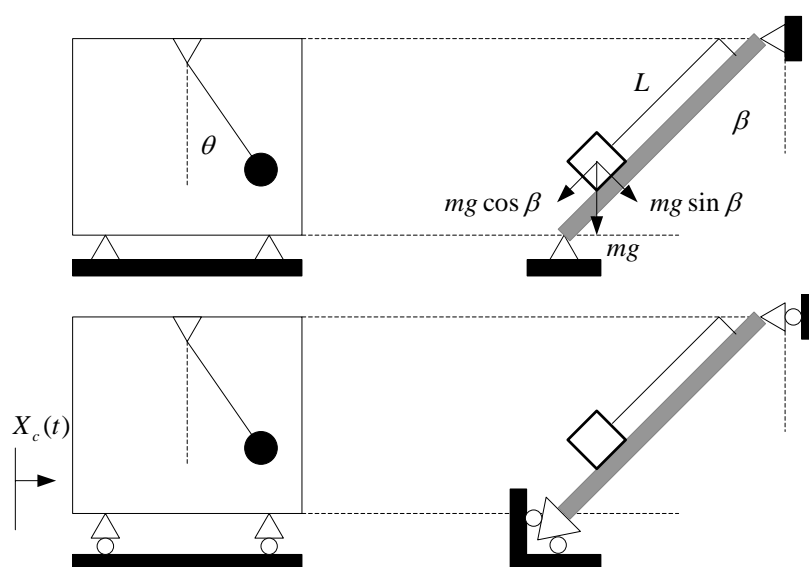
การทดลองที่ 1 เป็นการระบุเอกลักษณ์ระบบลูกตุ้มนาฬิกาบนแผ่นกระดาน เพื่อคำนวณค่าสัมประสิทธิ์ความฝืด (friction coefficient) อันเกิดจากน้ำหนักของลูกตุ้มที่กดลงบนแผ่นกระดาน โครงสร้างของระบบแสดงดังรูปที่ 2.2 ลูกตุ้มทำมุมกับแกนตั้งด้วยมุม θ และแผ่นกระดานทำมุมเอียงกับแกนตั้งด้วยมุม β การทดสอบได้แบ่งออกเป็น 2 กรณีคือ กรณีแรกกำหนดให้แผ่นกระดานอยู่กับที่ และในกรณีหลัง กำหนดให้แผ่นกระดานเคลื่อนที่ ในแต่ละกรณีได้ทำการปรับค่ามุม β ให้เปลี่ยนแปลงไป เพื่อดูผลกระทบที่เกิดขึ้นกับค่าสัมประสิทธิ์ความฝืด การทดลองได้ใช้กล้อง CCD (charge coupled device) ที่มีอัตราภาพ 30 ภาพต่อวินาที ทำหน้าที่เป็นตัวตรวจรู้ การคำนวณเพื่อระบุเอกลักษณ์ระบบอาศัยการวิเคราะห์เชิงถดถอย (regression analysis) และแบบจำลองของระบบลูกตุ้มนาฬิกา อยู่ในรูปของสมการอนุพันธ์ไม่เชิงเส้น มวลของลูกตุ้มเท่ากับ 0.13 กิโลกรัม และความยาวของลวดที่ใช้หึงลูกตุ้มเท่ากับ 0.365 เมตร ภายหลังจากการระบุเอกลักษณ์ระบบ ได้มีการตรวจสอบผลย้อนกลับโดยการคำนวณเชิงตัวเลขด้วยระเบียบวิธีรุงเง-คุดตา (Runge-Kutta method) ซึ่งพบว่าให้ผลที่น่าพึงพอใจ

การทดลองที่ 2 เป็นการระบุเอกลักษณ์ระบบแท่นสั่น (shaking table) ซึ่งเป็นระบบที่ใช้ศึกษาการเกิดแผ่นดินไหวของศูนย์วิจัยวิศวกรรมแผ่นดินไหวแห่งชาติ (national center for research on earth quake engineering, NCREE) ประเทศไต้หวัน ในการทดลองนี้ Chung และคณะ ได้เปลี่ยนมาใช้กล้องวิดีโอที่มีอัตราภาพ 15 ภาพต่อวินาที เพื่อการบันทึกภาพ จากนั้นนำวิดีโอเทปที่ได้ไปอ่านด้วยเครื่องเล่นวิดีโอ (VCR) เพื่อส่งสัญญาณภาพเข้าสู่คอมพิวเตอร์โดยผ่านตัวปรับทีวี (TV tuner adapter) ดังแสดงในรูปที่ 2.3 การทดสอบต้องการระบุระยะขจัด

(displacement) ของระบบแทนสั้น การประมวลผลภาพอาศัยการแบ่งส่วนภาพ และสกัดตำแหน่งที่เกิดการสั่น เปรียบเทียบกับตำแหน่งที่อยู่กับที่ หรือตำแหน่งอ้างอิงของระบบ ตำแหน่งทั้งสองได้รับการทำเครื่องหมายเพื่อให้สะดวกในการประมวลผล และระบุระยะขจัดที่เกิดขึ้น ดังแสดงในรูปที่ 2.4 ผลที่ได้จากการประมวลผลภาพ เปรียบเทียบกับผลที่ได้จากการวัดด้วยตัวตรวจรู้ มีความใกล้เคียงกันอย่างน่าพอใจ ดังแสดงในรูปที่ 2.5

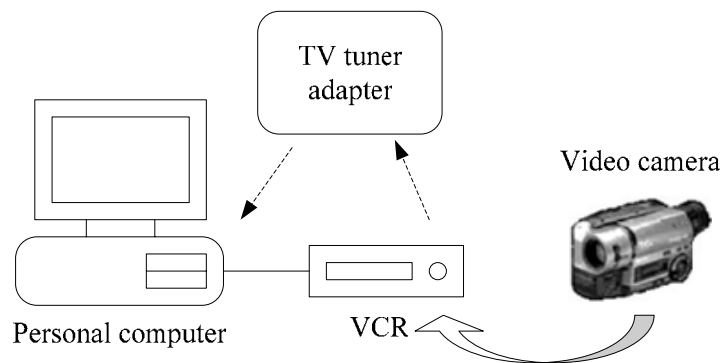


รูปที่ 2.1 การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ⁽¹⁾

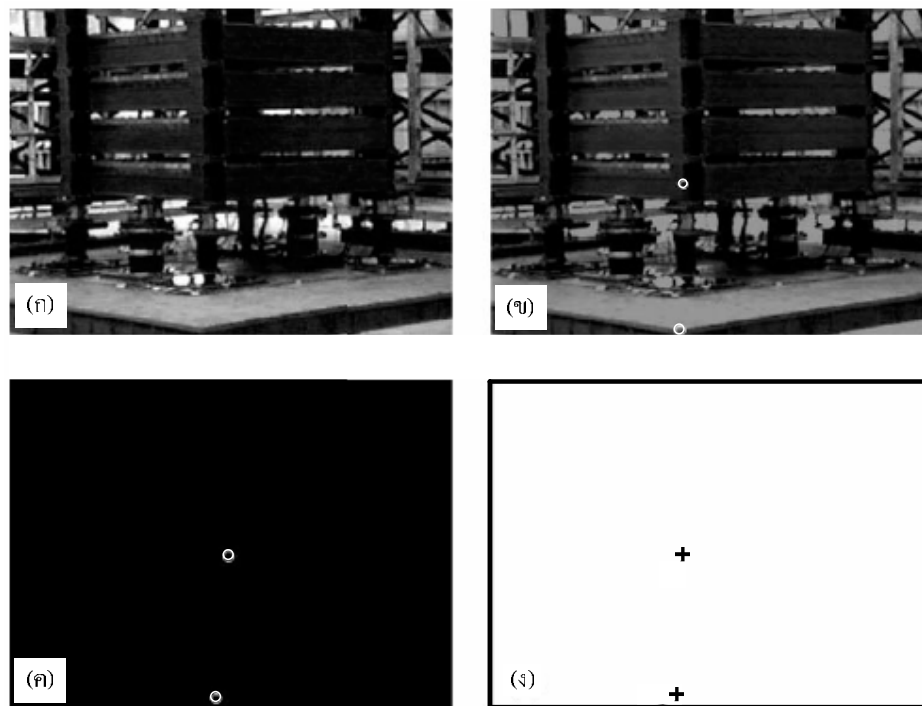


รูปที่ 2.2 โครงสร้างของระบบลูกตุ้มนาฬิกาบนแผ่นกระดาน⁽¹⁾

⁽¹⁾หมายเหตุ จาก “Digital image processing for non-linear system identification,” โดย Chung, Liang, Kushiya, and Shinozuka, 2004, *Non-Linear Mechanics*, 39, 691-707.



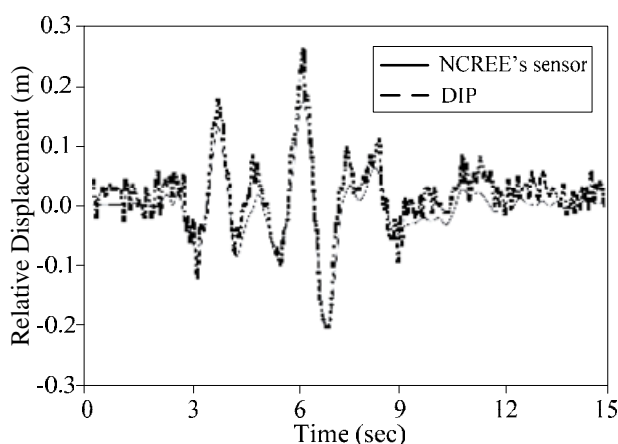
รูปที่ 2.3 การประมวลผลภาพเพื่อระบุเอกลักษณ์ระบบแทนสั้น⁽¹⁾



รูปที่ 2.4 การประมวลผลภาพระบบแทนสั้น⁽¹⁾

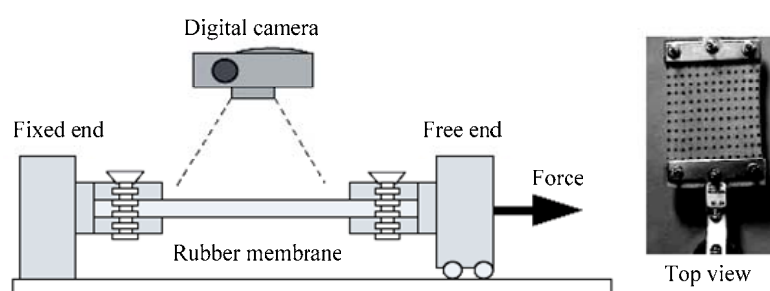
- (ก) ภาพต้นฉบับ (ข) การทำเครื่องหมายตำแหน่งที่เกิดการสั่น
(ค) ผลการแบ่งส่วนภาพ (ง) ผลการสกัดตำแหน่ง

⁽¹⁾หมายเหตุ จาก “Digital image processing for non-linear system identification,” โดย Chung, Liang, Kushiya, and Shinozuka, 2004, Non-Linear Mechanics, 39, 691-707.



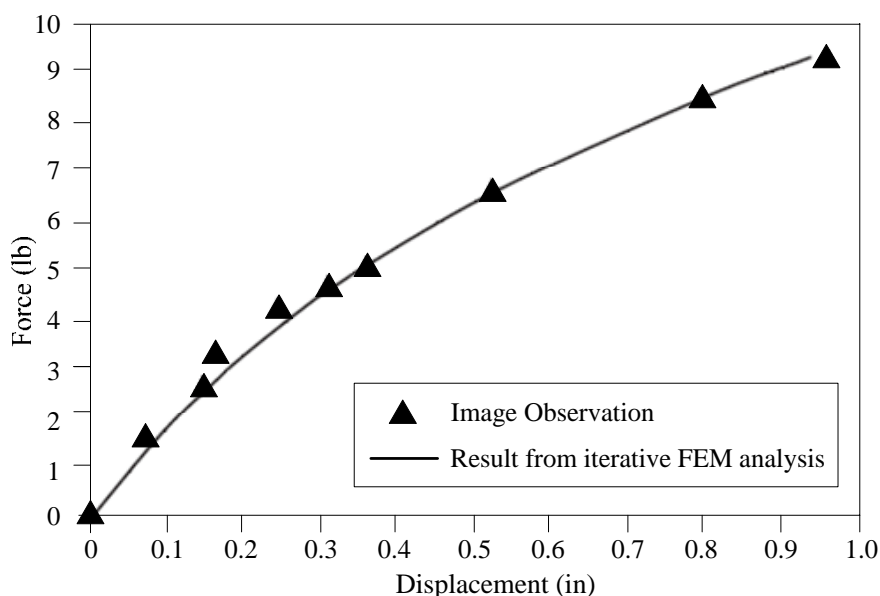
รูปที่ 2.5 ผลการระบุตำแหน่งระบบแทนสั้น⁽¹⁾

การทดลองที่ 3 เป็นการระบุเอกลักษณ์ระบบแผ่นยืดไม่เชิงเส้น (non-linear elastomeric membrane) การทดลองนี้ใช้กล้องบันทึกภาพดิจิทัล (camcorder) ของบริษัท Sony รุ่น TRV11 เพื่อบันทึกภาพการยึดตัวของแผ่นยืดภาพได้แรงดึง โครงสร้างของระบบแสดงดังรูปที่ 2.6 จะเห็นได้ว่าปลายด้านหนึ่งของแผ่นยืดถูกยึด ขณะที่ปลายอีกด้านหนึ่งถูกปล่อยอิสระเพื่อรองรับแรงดึง พื้นผิวของแผ่นยืดได้รับการทำให้เป็นจุดในลักษณะเมตริกซ์ เพื่อความสะดวกในการประมวลผล การทดสอบต้องการระบุความสัมพันธ์ระหว่างแรง กับระยะขจัดที่เกิดขึ้นบนแผ่นยืด การประมวลผลภาพอาศัยการแบ่งส่วน และสกัดตำแหน่ง ผลที่ได้จากการประมวลผลภาพได้รับการเปรียบเทียบกับผลการคำนวณเชิงตัวเลขด้วยวิธี FEM (finite-element method) ซึ่งพบว่า การประมวลผลภาพสามารถระบุเอกลักษณ์ระบบได้อย่างน่าพึงพอใจ ดังแสดงในรูปที่ 2.7



รูปที่ 2.6 โครงสร้างของระบบแผ่นยืดไม่เชิงเส้น⁽¹⁾

⁽¹⁾หมายเหตุ จาก “Digital image processing for non-linear system identification,” โดย Chung, Liang, Kushiya, and Shinozuka, 2004, *Non-Linear Mechanics*, 39, 691-707.



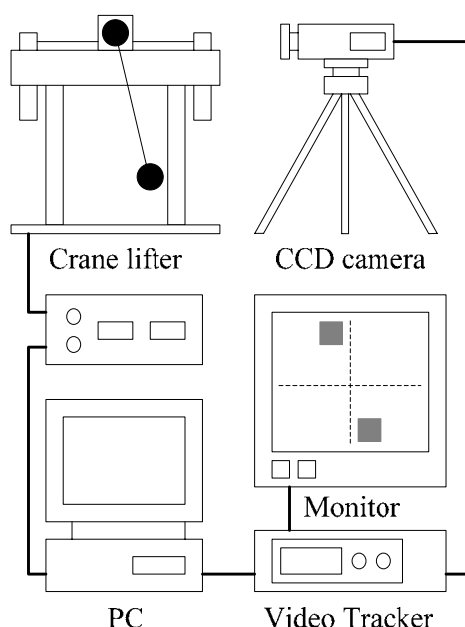
รูปที่ 2.7 ผลการระบุเอกลักษณ์ระบบแผ่นยึดไม้เชิงเส้น⁽¹⁾

⁽¹⁾หมายเหตุ จาก “Digital image processing for non-linear system identification,” โดย Chung, Liang, Kushiya, and Shinozuka, 2004, Non-Linear Mechanics, 39, 691-707.

ในปี ค.ศ. 2004 เช่นเดียวกัน มีงานวิจัยเกี่ยวกับการป้อนกลับด้วยภาพ (visual feedback) เพื่อควบคุมการทำงานของเครนยก (crane lifter) โดยใช้ตัวควบคุม PID+Q ที่ออกแบบภายใต้บริบทการควบคุมแบบคงทน (robust control) งานวิจัยดังกล่าวเป็นของ Matsuo และคณะ (2004) ซึ่งได้มีการใช้กล้อง CCD ทำงานร่วมกับ Video Tracker รุ่น 220 เพื่อทำหน้าที่เป็นตัวตรวจรู้ของระบบ กล้องและ Video Tracker ซึ่งมีความแยกชัด 1/60 วินาที จะตรวจจับภาพการเคลื่อนที่ของรถและเครน ดังรูปที่ 2.8 ทั้งรถและเครนได้รับการทำแถบสีให้มีความแตกต่างไปจากส่วนอื่น ๆ ของระบบอย่างมีนัยสำคัญ ตัวเฝ้าตรวจ (monitor) การเคลื่อนที่ของ Video Tracker แสดงการเคลื่อนที่ของรถและเครนด้วยหน้าต่างสี่เหลี่ยม Video Tracker จะทำการประมวลผลภาพ เพื่อสกัดข้อมูลตำแหน่งของรถและมุมของเครน ในฐานะข้อมูลพลวัตที่สนใจ จากนั้นข้อมูลดังกล่าวจะถูกส่งไปยังตัวควบคุม PID+Q ที่อนุวัตบนคอมพิวเตอร์ และตัวควบคุม PID+Q จะส่งสัญญาณควบคุมผ่านตัวขับเราไปยังระบบเครนยก เพื่อควบคุมให้เครนเกิดการสั่นไกวน้อยที่สุด

การใช้กล้อง CCD ที่ทำงานร่วมกับ Video Tracker ในฐานะตัวตรวจรู้ ก่อให้เกิดช่วงเวลาที่ประวิงขึ้นในระบบ ทั้งนี้เพราะต้องใช้เวลาในการประมวลผลภาพ และถ่ายโอนข้อมูล นอกจากนั้นแล้ว ข้อมูลที่ได้ อาจมีความคลาดเคลื่อนมากกว่าการใช้ตัวตรวจรู้แบบดั้งเดิม ทั้งช่วงเวลาประวิง

และความคลาดเคลื่อนที่เกิดขึ้น จะไปรวมอยู่ในแบบจำลองทางคณิตศาสตร์ของระบบ ซึ่งอาจกล่าวได้ว่า ระบบป้อนกลับด้วยภาพจะมีแบบจำลองทางคณิตศาสตร์ที่ปรากฏช่วงเวลาประวิง และความไม่แน่นอน (uncertainty) อย่างไรก็ตาม ปัญหาดังกล่าวสามารถแก้ไขได้โดยใช้ตัวควบคุมที่ออกแบบภายใต้บริบทการควบคุมแบบคงทน ดังปรากฏในงานของ Matsuo และคณะ สำหรับ Video Tracker รุ่น 220 เป็นอุปกรณ์สำเร็จรูป ที่มีกลไกการประมวลผลภาพสำหรับสกัดตำแหน่งของวัตถุที่มีการเคลื่อนที่แบบ 2 มิติ เวลาที่ใช้ทำการประมวลผลภาพ และถ่ายโอนข้อมูลเท่ากับ 1/30 วินาที และมีความคลาดเคลื่อนของตำแหน่งวัตถุที่ตรวจจับเท่ากับ 5 ถึง 10 เปอร์เซ็นต์ อย่างไรก็ตามในงานวิจัยของ Matsuo และคณะ มิได้แสดงรายละเอียดกลไกการประมวลผลภาพของ Video Tracker รุ่น 220 เอาไว้



รูปที่ 2.8 ระบบควบคุมเครนยกโดยใช้การป้อนกลับด้วยภาพ⁽²⁾

⁽²⁾หมายเหตุ จาก “Nominal performance recovery by PID+Q controller and its application to antisway control of crane lifter with visual feedback,” โดย Matsuo, Yoshino, Suemitsu, and Nakano, 2004, IEEE Control System Technology, 12(1), 156-166.

2.5 สรุป

บทที่ 2 นี้ได้รายงานผลการปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้องกับการระบุเอกลักษณ์ระบบพลวัตโดยใช้เทคนิคการประมวลผลภาพ โดยอาศัยฐานข้อมูลจาก IEEE IEE

(IEL online) IEICE และ ScienceDirect การปฏิทัศน์วรรณกรรมแบ่งออกเป็นสามแนวทางคือ การระบุเอกลักษณ์ระบบพลวัตด้วยวิธีการดั้งเดิม การระบุเอกลักษณ์ระบบด้วยเทคนิคทางปัญญาประดิษฐ์ และการระบุเอกลักษณ์ระบบด้วยเทคนิคการประมวลผลภาพ ผลการปฏิทัศน์วรรณกรรมทำให้ทราบถึงแนวทางวิจัยที่เกี่ยวข้อง ระเบียบวิธีที่ใช้ และผลการดำเนินงานจากคณະนักวิจัยต่าง ๆ ตั้งแต่อดีตเป็นต้นมา ซึ่งอาจสรุปได้ดังนี้

การระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม อาศัยข้อมูลความรู้ที่เกี่ยวข้องกับสัญญาณอินพุตและเอาต์พุตของระบบ การคำนวณใช้คณิตศาสตร์แบบการถดถอยเชิงเส้น และกำลังสองน้อยที่สุด ดัชนีสมรรถนะที่ใช้บ่งชี้คุณภาพของแบบจำลองทางคณิตศาสตร์ คือค่าความคลาดเคลื่อนระหว่างค่าเอาต์พุตจริงที่ได้จากการทดสอบ กับค่าเอาต์พุตที่ได้จากแบบจำลองทางคณิตศาสตร์

การระบุเอกลักษณ์ระบบด้วยเทคนิคทางปัญญาประดิษฐ์ มีการใช้โครงข่ายประสาทเป็นแบบจำลองของระบบพลวัต ที่ได้รับการฝึกสอนโดยการค้นหาแบบตาบอด นอกจากนี้ยังมีการใช้การค้นหาแบบตาบอดเพื่อการระบุเอกลักษณ์ระบบพลวัตโดยตรงอีกด้วย

การระบุเอกลักษณ์ระบบด้วยเทคนิคการประมวลผลภาพ มีการใช้เทคนิคการประมวลผลภาพแบบดิจิทัลเพื่อระบุเอกลักษณ์ระบบพลวัต โดยอาศัยกล้องบันทึกภาพทำหน้าที่เป็นตัวตรวจรู้ นอกจากนี้ยังมีงานวิจัยเกี่ยวกับการควบคุมป้อนกลับด้วยภาพ ที่ใช้กล้องบันทึกภาพทำงานร่วมกับ Video Tracker เพื่อทำหน้าที่เป็นตัวตรวจรู้ของระบบอีกด้วย

ผลจากการปฏิทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง จะถูกใช้เป็นข้อมูลอ้างอิง และเป็นแนวทางสำหรับการดำเนินงานวิจัยต่อไป

บทที่ 3

การระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม

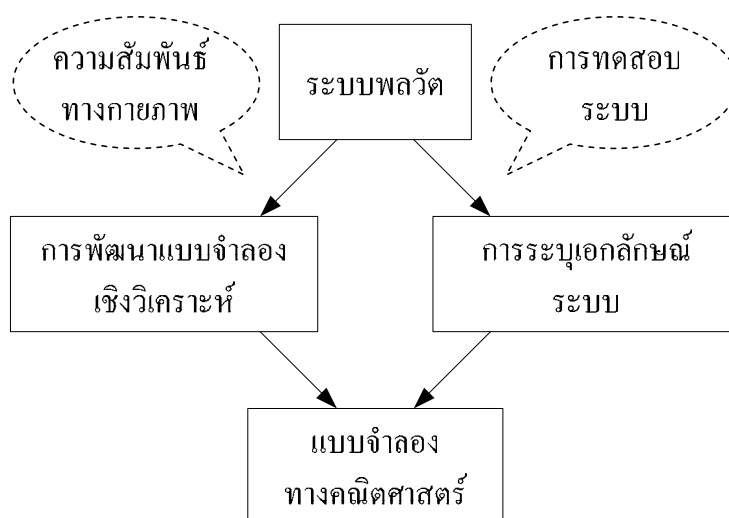
3.1 กล่าวนำ

การระบุเอกลักษณ์ระบบด้วยระเบียบวิธีการดั้งเดิม จะอาศัยข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ โดยทั่วไป แบบจำลองที่ได้จากการระบุเอกลักษณ์ระบบ ได้รับการแบ่งออกเป็นสองลักษณะ (Ljung and Glad, 1994) คือ แบบจำลองชนิดที่ปรากฏค่าพารามิเตอร์ (parametric model) และแบบจำลองชนิดที่ไม่ปรากฏค่าพารามิเตอร์ (non-parametric model) ในวิทยานิพนธ์นี้ให้ความสนใจเฉพาะในแบบจำลองชนิดที่ปรากฏค่าพารามิเตอร์เท่านั้น ซึ่งได้รับการแบ่งออกเป็นสองประเภทคือ แบบจำลองเฉพาะกรณี (tailor-made model) และแบบจำลองทั่วไป (ready-made model) แบบจำลองเฉพาะกรณีได้รับการพัฒนาขึ้นจากความสัมพันธ์ทางกายภาพพื้นฐานของระบบ ค่าพารามิเตอร์ที่ปรากฏในแบบจำลองประเภทนี้ สื่อความหมายทางกายภาพของระบบเป็นสำคัญ สำหรับแบบจำลองทั่วไป เป็นแบบจำลองที่ได้รับการประยุกต์ใช้งานอย่างกว้างขวาง ค่าพารามิเตอร์ที่ปรากฏในแบบจำลองดังกล่าว ไม่สื่อความหมายทางกายภาพของระบบ แต่ถูกใช้เพื่ออธิบายถึงความสัมพันธ์ของตัวแปรอินพุตและเอาต์พุตของระบบ แบบจำลองประเภทนี้เป็นที่รู้จักกันในนามแบบจำลองกล่องดำ (black-box model) การระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม โดยเฉพาะการระบุเอกลักษณ์แบบจำลองกล่องดำ ซึ่งเป็นรูปลักษณะที่ใช้ประโยชน์อย่างกว้างขวางในวิทยาการระบบควบคุม จะได้รับการนำเสนอในบทที่ 3 นอกจากนี้ยังได้นำเสนอแนวทางการระบุเอกลักษณ์ระบบด้วยโปรแกรมสำเร็จ และการวางแผนทดสอบระบบ เพื่อบันทึกข้อมูลอินพุตและเอาต์พุตอย่างถูกต้องและเพียงพอ สำหรับการระบุเอกลักษณ์ระบบต่อไป

3.2 แนวคิดการระบุเอกลักษณ์ระบบ

โดยทั่วไป การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบพลวัต จะอาศัยความสัมพันธ์ทางกายภาพของระบบ ซึ่งเป็นกฎทางธรรมชาติ ที่สามารถอธิบายพฤติกรรมทางพลวัตด้วยสมการคณิตศาสตร์ ดังเช่น กฎการเคลื่อนที่ของนิวตัน เป็นต้น วิธีการดังกล่าวเป็นพื้นฐานของการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบพลวัตต่าง ๆ ซึ่งอาจเรียกแบบจำลองที่ได้จากวิธีการดังกล่าวว่า แบบจำลองเชิงทฤษฎี (theoretical model) หรือแบบจำลองเชิงวิเคราะห์ (analytical model) ในระบบที่มีความซับซ้อนสูง อาจประสบปัญหาในเรื่องของการหาสมการทางคณิตศาสตร์

มาใช้อธิบายความสัมพันธ์ทางกายภาพของระบบ ส่งผลให้ไม่สามารถพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบอย่างถูกต้องได้ วิธีการหนึ่งที่ถูกนำมาใช้แก้ปัญหาดังกล่าวคือ การทดสอบระบบด้วยสัญญาณอินพุตที่เหมาะสม สัญญาณอินพุตและเอาต์พุตของระบบได้รับการบันทึก และคำนวณด้วยกระบวนการทางคณิตศาสตร์ เพื่อพัฒนาแบบจำลองที่เหมาะสมที่สุดให้กับระบบ วิธีการพัฒนาแบบจำลองทางคณิตศาสตร์ดังกล่าวเรียกว่า การระบุเอกลักษณ์ระบบ (system identification) รูปที่ 3.1 อธิบายกระบวนการพัฒนาแบบจำลองทางคณิตศาสตร์ทั้งสองแนวทางที่กล่าวมาข้างต้น



รูปที่ 3.1 การพัฒนาแบบจำลองทางคณิตศาสตร์

แบบจำลองทางคณิตศาสตร์ของระบบพลวัต อาจมีรูปแบบที่แตกต่างกันออกไป เช่น อาจอยู่ในรูปของฟังก์ชันถ่ายโอน สมการอนุพันธ์สามัญ (ODE) หรือสมการอนุพันธ์ย่อย (PDE) ทั้งนี้รูปแบบของแบบจำลองทางคณิตศาสตร์ของระบบ จะขึ้นอยู่กับธรรมชาติของระบบเป็นสำคัญ ดังที่ได้กล่าวมาข้างต้นแล้วว่า แบบจำลองทางคณิตศาสตร์ที่ให้ความสนใจในขณะนี้ เป็นแบบจำลองกล่องดำ ที่มีรูปแบบเป็นฟังก์ชันถ่ายโอน สามารถประยุกต์กับระบบพลวัตเชิงเส้นใด ๆ ก็ได้ ดังนั้นในหัวข้อต่อไป จะกล่าวถึงการระบุเอกลักษณ์แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ ซึ่งจะใช้เป็นพื้นฐานในการระบุเอกลักษณ์ระบบพลวัตในงานวิจัยนี้ต่อไป

3.3 การระบุเอกลักษณ์แบบจำลองกล่องดำ

การระบุเอกลักษณ์ระบบพลวัตอาจแสดงได้ดังรูปที่ 1.1 โดยอาศัยข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ ผ่านกระบวนการทางคณิตศาสตร์ เพื่อพัฒนาแบบจำลองที่เหมาะสม

โดยปกติแบบจำลองกล่องดำจะแสดงในรูปของเวลาเต็มหน่วย ซึ่งข้อมูลอินพุตและเอาต์พุตของระบบได้รับการบันทึกแบบซีกตัวอย่าง โดยใช้ตัวปฏิบัติการเลื่อน q (shift operator) กระทำกับสมการผลต่างเชิงเส้น (linear difference equation) โดยที่ $q^{nk} y(t) = y(t + nk)$ และ $q^{-nk} y(t) = y(t - nk)$ สัญลักษณ์ t ที่ปรากฏ มีความหมายเป็นเวลาเต็มหน่วย (discrete time) แบบจำลองกล่องดำเชิงเส้นชนิดเวลาเต็มหน่วยแสดงดังสมการที่ (3-1)

$$y(t) = \eta(t) + \omega(t) \quad (3-1)$$

เมื่อพจน์ $\eta(t)$ คือ แบบจำลองเอาต์พุตที่อิสระต่อสัญญาณรบกวน (noise-free output model) และพจน์ $\omega(t)$ คือ แบบจำลองการรบกวน (disturbance model) ซึ่งแสดงในสมการที่ (3-2) และ (3-3) ตามลำดับ

$$\eta(t) = G(q, \theta)u(t) \quad (3-2)$$

$$\omega(t) = H(q, \theta)e(t) \quad (3-3)$$

เมื่อ $u(t)$ คือ สัญญาณอินพุต $e(t)$ คือ สัญญาณรบกวน $y(t)$ คือ สัญญาณเอาต์พุต และ θ คือ เวกเตอร์ของพารามิเตอร์ในแบบจำลอง

กำหนดให้ $G(q, \theta)$ และ $H(q, \theta)$ เป็นฟังก์ชันถ่ายโอนในโดเมน q ดังสมการที่ (3-4) และ (3-5) ตามลำดับ

$$G(q, \theta) = \frac{B(q)}{F(q)} = \frac{b_1 q^{-nk} + b_2 q^{-nk-1} + \dots + b_{nb} q^{-nk-nb+1}}{1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf}} \quad (3-4)$$

$$H(q, \theta) = \frac{C(q)}{D(q)} = \frac{1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}}{1 + d_1 q^{-1} + \dots + d_{nd} q^{-nd}} \quad (3-5)$$

ดังนั้นสมการที่ (3-2) อาจเขียนในอีกรูปแบบหนึ่งได้ดังสมการที่ (3-6)

$$\eta(t) + f_1 \eta(t-1) + \dots + f_{nf} \eta(t-nf) = b_1 u(t-nk) + \dots + b_{nb} u[t - (nb + nk - 1)] \quad (3-6)$$

เมื่อ nk คือช่วงเวลาประวิง (delay time) ของการชักตัวอย่าง กำหนดให้คาบเวลาของการชักตัวอย่างเป็น T ซึ่งมีค่าคงที่ และเมื่อ $e(t)$ เป็นสัญญาณรบกวนขาว (white noise) สมการที่ (3-1) สามารถเขียนใหม่ได้ดังสมการที่ (3-7)

$$\begin{aligned} y(t) &= G(q, \theta)u(t) + H(q, \theta)e(t) \\ &= \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \end{aligned} \quad (3-7)$$

เมื่อ θ คือ เวกเตอร์ของพารามิเตอร์ b_i, c_i, d_i , และ f_i ของฟังก์ชันถ่ายโอน จะเห็นว่าแบบจำลองกล่องดำ ถูกกำกับไว้ด้วยอันดับของพหุนามจำนวน 5 ค่า คือ nb, nc, nd, nf , และ nk แบบจำลองในสมการที่ (3-7) เป็นที่รู้จักกันในนามแบบจำลอง BJ (Box-Jenkins model) โครงสร้างของแบบจำลอง BJ แสดงด้วยบล็อกไดอะแกรมในรูปที่ 3.2

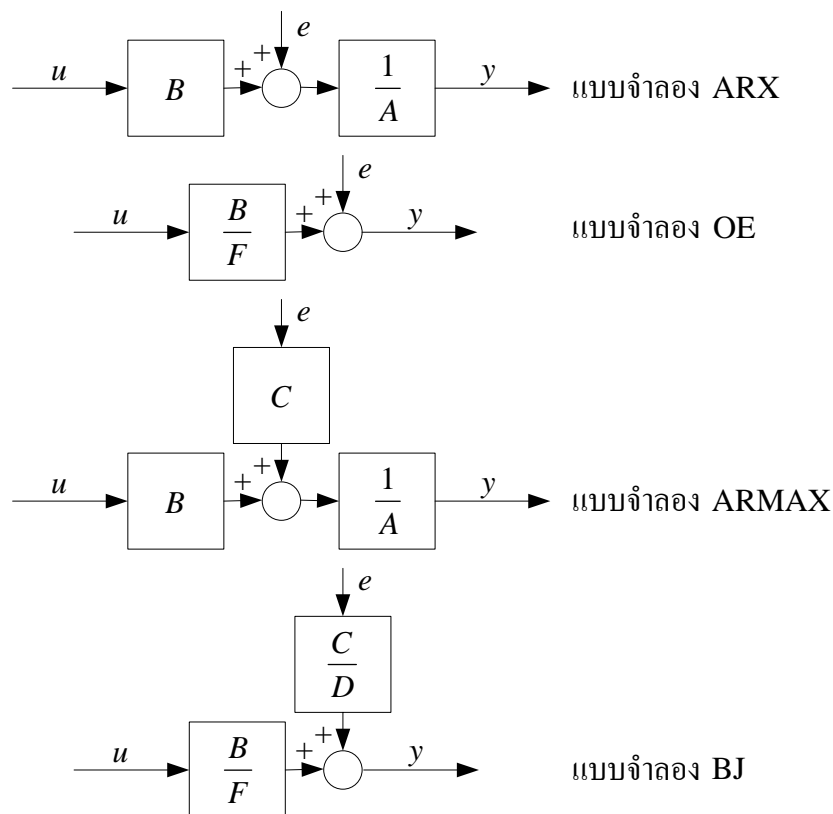
ในกรณีที่ $H(q, \theta) = 1$ นั่นคือ $nc = nd = 0$ สมการที่ (3-7) สามารถเขียนใหม่ได้ดังสมการที่ (3-8) ซึ่งเป็นที่รู้จักกันในนามแบบจำลอง OE (output error model) โครงสร้างของแบบจำลอง OE แสดงด้วยบล็อกไดอะแกรมในรูปที่ 3.2 เช่นเดียวกัน

$$\begin{aligned} y(t) &= G(q, \theta)u(t) + e(t) \\ &= \frac{B(q)}{F(q)}u(t) + e(t) \end{aligned} \quad (3-8)$$

เมื่อกำหนดให้พจน์ G และ H มีพหุนามในเทอมส่วนเป็น $A(q)$ ตามสมการที่ (3-9) และเมื่อคูณสมการที่ (3-7) ด้วย $A(q)$ จะได้สมการที่ (3-10) ซึ่งแบบจำลองในสมการที่ (3-10) รู้จักกันในนามแบบจำลอง ARMAX (autoregression, moving average of white noise, extra (exogenous) input model) โครงสร้างของแบบจำลอง ARMAX แสดงด้วยบล็อกไดอะแกรมในรูปที่ 3.2 เช่นกัน

$$A(q) = F(q) = D(q) = 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \quad (3-9)$$

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (3-10)$$



รูปที่ 3.2 โครงสร้างของแบบจำลองกล่องดำเชิงเส้น

ในทางกายภาพ แบบจำลอง ARMAX มีพลวัตของสัญญาณอินพุต และสัญญาณรบกวนที่เหมือนกัน (มีโพลเหมือนกัน) ซึ่งจะมีความสมเหตุสมผลในกรณีที่แบบจำลองดังกล่าวถูกใช้เพื่ออธิบายพลวัตของระบบที่มีสัญญาณอินพุต และสัญญาณรบกวน เข้ามากระทำพร้อมกัน ในกรณีของแบบจำลอง ARMAX ที่มีพจน์ $C(q) = 1$ นั่นคือ $nc = 0$ แบบจำลองที่ได้แสดงในสมการที่ (3-11) ซึ่งเป็นที่รู้จักกันในนามแบบจำลอง ARX (autoregression, extra input model) โครงสร้างของแบบจำลอง ARX แสดงด้วยบล็อกไดอะแกรมในรูปที่ 3.2 เช่นเดียวกัน

$$A(q)y(t) = B(q)u(t) + e(t) \quad (3-11)$$

ในการทำนายค่าเอาต์พุต $y(t)$ จะเริ่มพิจารณาจากแบบจำลอง OE ดังสมการที่ (3-8) โดยกำหนดให้การทำนายค่าเอาต์พุต $y(t)$ ของแบบจำลอง OE คือ $\hat{y}(t|\theta)$ ดังสมการที่ (3-12)

$$\hat{y}(t|\theta) = G(q, \theta)u(t) \quad (3-12)$$

จากแบบจำลอง ARX ในสมการที่ (3-11) สามารถเขียนสมการเอาต์พุต $y(t)$ ได้เป็น

$$y(t) = -a_1 y(t-1) - \dots - a_{na} y(t-na) + b_1 u(t-nk) + \dots + b_{nb} u(t-nk-nb+1) + e(t) \quad (3-13)$$

การทำนายค่าเอาต์พุต $y(t)$ ของแบบจำลอง ARX คือ $\hat{y}(t|\theta)$ ดังสมการที่ (3-14)

$$\hat{y}(t|\theta) = -a_1 y(t-1) - \dots - a_{na} y(t-na) + b_1 u(t-nk) + \dots + b_{nb} u(t-nk-nb+1) \quad (3-14)$$

ความแตกต่างระหว่างสมการที่ (3-12) กับ (3-14) คือ ในแบบจำลอง OE การทำนายค่าเอาต์พุตจะใช้ข้อมูลอินพุตแต่เพียงอย่างเดียว ขณะที่แบบจำลอง ARX จะใช้ค่าเอาต์พุตเก่าประกอบการทำนายค่าเอาต์พุตปัจจุบันด้วย

ดังนั้นสมการที่ (3-7) สามารถเขียนฟังก์ชันของค่าเอาต์พุต $y(t)$ ในรูปทั่วไปได้โดยทำการหารสมการที่ (3-7) ด้วย $H(q, \theta)$ ดังสมการที่ (3-15)

$$H^{-1}(q, \theta) y(t) = H^{-1}(q, \theta) G(q, \theta) u(t) + e(t) \\ y(t) = [1 - H^{-1}(q, \theta)] y(t) + H^{-1}(q, \theta) G(q, \theta) u(t) + e(t) \quad (3-15)$$

เมื่อ $nc \geq nd$ พจน์ $[1 - H^{-1}(q, \theta)]$ จะพิจารณาได้ดังสมการที่ (3-16)

$$1 - H^{-1}(q, \theta) = 1 - \frac{D(q)}{C(q)} = \frac{(c_1 - d_1)q^{-1} + \dots + (c_{nc} - d_{nd})q^{-nc}}{1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}} \quad (3-16)$$

จะเห็นว่าพจน์ $[1 - H^{-1}(q, \theta)] y(t)$ ในสมการที่ (3-15) ใช้ค่าเอาต์พุตเก่าในการคำนวณและการทำนายค่าเอาต์พุต $y(t)$ ดังนั้นอาจเขียน $\hat{y}(t|\theta)$ ได้ดังสมการที่ (3-17)

$$\hat{y}(t|\theta) = [1 - H^{-1}(q, \theta)] y(t) + H^{-1}(q, \theta) G(q, \theta) u(t) \quad (3-17)$$

จากสมการที่ผ่านมา ฟังก์ชันการทำนายค่าเอาต์พุตยังคงมีความซับซ้อนอยู่ ดังนั้นเพื่อให้สะดวกในการพิจารณา การทำนายค่าเอาต์พุต $\hat{y}(t|\theta)$ จะได้รับการแสดงในรูปของฟังก์ชันเชิงเส้นของค่าพารามิเตอร์ θ ดังสมการที่ (3-18)

$$\hat{y}(t|\theta) = \theta^T \varphi(t) \quad (3-18)$$

เมื่อ θ เป็น คอลัมน์เวกเตอร์ (column vector) ของพารามิเตอร์ที่ไม่ทราบค่า ในขณะที่ $\varphi(t)$ เป็น คอลัมน์เวกเตอร์ ที่สร้างมาจากค่าอินพุตและเอาต์พุตเก่า แบบจำลองนี้ได้รับการเรียกขานว่า แบบจำลองการถดถอยเชิงเส้น เรียก $\varphi(t)$ ว่าเวกเตอร์ถดถอย (regression vector) และเรียกสมาชิกใน $\varphi(t)$ ว่าตัวถดถอย (regressors)

แบบจำลอง ARX สามารถเขียนให้อยู่ในรูปของการถดถอยเชิงเส้นได้ดังสมการที่ (3-19)

$$\theta = [a_1 \ a_2 \ \dots \ a_{na} \ b_1 \ \dots \ b_{nb}]^T$$

$$\varphi(t) = [-y(t-1) \ \dots \ -y(t-na) \ u(t-nk) \ \dots \ u(t-nk-nb+1)]^T \quad (3-19)$$

การพิจารณาความถูกต้องของแบบจำลองที่ได้จากการระบุเอกลักษณ์ระบบ จะอาศัยค่าความคลาดเคลื่อนระหว่างค่าจริงกับค่าที่ทำนาย เป็นเกณฑ์ในการพิจารณา ค่าความคลาดเคลื่อน $\varepsilon(t, \theta)$ และค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (mean square error) $V_N(\theta)$ แสดงดังสมการที่ (3-20) และ (3-21) ตามลำดับ

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta) \quad (3-20)$$

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \theta) \quad (3-21)$$

การคำนวณหาค่าพารามิเตอร์ของแบบจำลองซึ่งบรรจุอยู่ใน θ เป็นการเลือกค่าต่าง ๆ ใน θ ที่ทำให้ $V_N(\theta)$ มีค่าน้อยที่สุด ซึ่งอาจเขียนแสดงได้ดังสมการที่ (3-22)

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta) \quad (3-22)$$

เมื่อ θ ถูกประมาณด้วย $\hat{\theta}_N$ ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย อาจเขียนแทนด้วย $\hat{\lambda}_N$ ดังสมการที่ (3-23)

$$\hat{\lambda}_N = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \hat{\theta}_N) \quad (3-23)$$

สำหรับแบบจำลองการถดถอยเชิงเส้น ค่าความคลาดเคลื่อน $\varepsilon(t, \theta)$ และค่าความคลาดเคลื่อนกำลังสองเฉลี่ย $V_N(\theta)$ แสดงดังสมการที่ (3-24) และ (3-25) ตามลำดับ

$$\varepsilon(t, \theta) = y(t) - \theta^T \varphi(t) \quad (3-24)$$

$$\begin{aligned} V_N(\theta) &= \frac{1}{N} \sum_{t=1}^N [y(t) - \theta^T \varphi(t)]^2 \\ &= \frac{1}{N} \sum_{t=1}^N y^2(t) - \frac{1}{N} \sum_{t=1}^N 2\theta^T \varphi(t) y(t) + \frac{1}{N} \sum_{t=1}^N \theta^T \varphi(t) \varphi^T(t) \theta \\ &= \frac{1}{N} \sum_{t=1}^N y^2(t) - 2\theta^T f_N + \theta^T R_N \theta \end{aligned} \quad (3-25)$$

เมื่อ f_N เป็น คอลัมน์เวกเตอร์ และ R_N เป็น เมทริกซ์ ดังสมการที่ (3-26)

$$f_N = \frac{1}{N} \sum_{t=1}^N \varphi(t) y(t), \quad R_N = \frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \quad (3-26)$$

ถ้า R_N เป็นเมทริกซ์ที่สามารถผกผันได้ สมการที่ (3-25) สามารถเขียนใหม่ได้ดังสมการที่ (3-27)

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N y^2(t) - f_N^T R_N^{-1} f_N + \left[(\theta - R_N^{-1} f_N)^T R_N (\theta - R_N^{-1} f_N) \right] \quad (3-27)$$

ค่า $V_N(\theta)$ ในสมการที่ (3-27) จะมีค่าน้อยที่สุดเมื่อพจน์ทางขวาสุดมีค่าเท่ากับศูนย์ นั่นคือ

$$\theta = \hat{\theta}_N = R_N^{-1} f_N \quad (3-28)$$

ค่า $\hat{\theta}_N$ จะได้รับการคำนวณจากสมการที่ (3-26) และ (3-28) ซึ่งมักใช้วิธีระบบสมการเชิงเส้น

ในกรณีแบบจำลอง ARX สมาชิกในเมตริกซ์ R_N และในเวกเตอร์ f_N สามารถเขียนอยู่ในรูปต่าง ๆ ดังสมการที่ (3-29) ซึ่ง $\hat{\theta}_N$ ได้มาจากฟังก์ชันแปรปรวนร่วม (covariance function) ระหว่างค่าอินพุตและเอาต์พุตของระบบ

$$\frac{1}{N} \sum_{t=1}^N y(t-j)u(t-k), \frac{1}{N} \sum_{t=1}^N y(t-j)y(t-k), \text{ หรือ } \frac{1}{N} \sum_{t=1}^N u(t-j)u(t-k) \quad (3-29)$$

การคำนวณ $\hat{\theta}_N$ เพื่อให้ $V_N(\theta)$ มีค่าน้อยที่สุด จะอาศัยการคำนวณเชิงตัวเลข ซึ่งโดยทั่วไปจะใช้ระเบียบวิธีของนิวตัน-ราฟสัน (Newton-Raphson method) พิจารณาสมการที่ (3-30)

$$g(x) = 0 \quad (3-30)$$

ผลเฉลยของสมการที่ (3-30) คือ x ที่ได้จากการคำนวณแบบวนซ้ำ ดังสมการที่ (3-31)

$$x^{(i+1)} = x^{(i)} - [g'(x^{(i)})]^{-1} g(x^{(i)}) \quad (3-31)$$

เมื่อ $g'(x)$ คืออนุพันธ์ของ $g(x)$ เทียบกับ x

กำหนดให้ μ คือ ช่วงของการปรับการคำนวณแบบวนซ้ำ เพื่อให้ค่า $x^{(i+1)}$ เป็นคำตอบที่ดีกว่า $x^{(i)}$ ดังนั้นสมการที่ (3-31) จึงเขียนใหม่ได้ดังสมการที่ (3-32)

$$x^{(i+1)} = x^{(i)} - \mu [g'(x^{(i)})]^{-1} g(x^{(i)}) \quad (3-32)$$

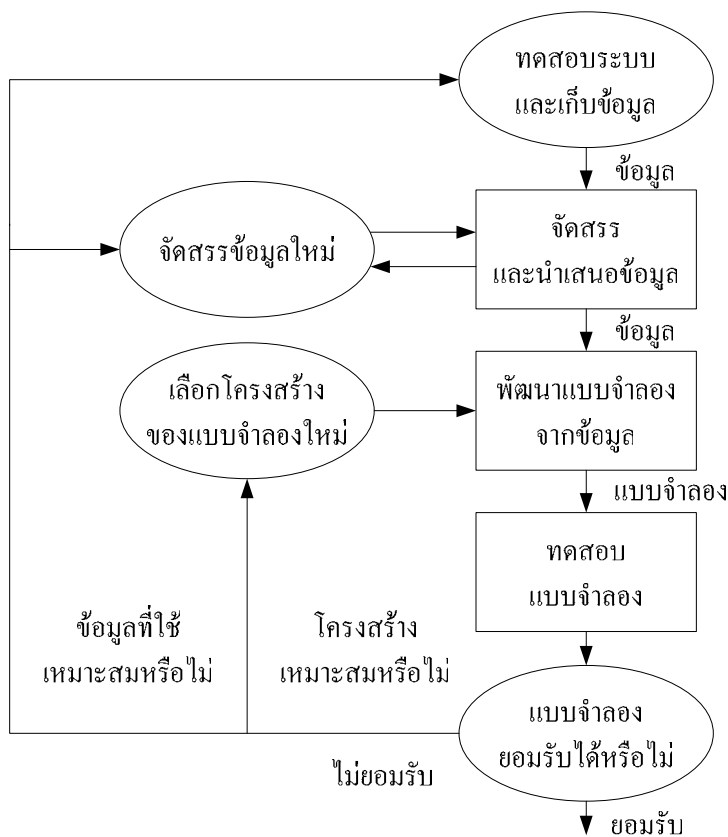
เมื่อพิจารณาระเบียบวิธีของนิวตัน-ราฟสัน การทำให้ $V_N(\theta)$ มีค่าน้อยที่สุด จะเป็นดังสมการที่ (3-33)

$$\frac{d}{d\theta} V_N(\theta) = 0 \quad (3-33)$$

โดยที่ $V_N(\theta)$ เป็นฟังก์ชันค่าจริงที่ได้รับการทำอนุพันธ์เทียบกับ θ ซึ่งมีผลเฉลยเป็นคอลัมน์เวกเตอร์ ผลเฉลย $\hat{\theta}^{(i+1)}$ แสดงดังสมการที่ (3-34)

$$\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} - \mu^{(i)} [V_N''(\hat{\theta}^{(i)})]^{-1} V_N'(\hat{\theta}^{(i)}) \tag{3-34}$$

เมื่อ $V_N'(\theta)$ คืออนุพันธ์อันดับหนึ่งของ $V_N(\theta)$ เทียบกับ θ และ $V_N''(\theta)$ คืออนุพันธ์อันดับสองของ $V_N(\theta)$ เทียบกับ θ ขณะที่ช่วงของการปรับการคำนวณแบบวนซ้ำ $\mu^{(i)}$ ได้รับการกำหนดเพื่อให้ $V_N(\hat{\theta}^{(i+1)})$ มีค่าน้อยกว่า $V_N(\hat{\theta}^{(i)})$



รูปที่ 3.3 การระบุเอกลักษณ์ระบบโดยใช้โปรแกรมสำเร็จรูป

3.4 การระบุเอกลักษณ์ระบบด้วยโปรแกรมสำเร็จ

สำหรับการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิมที่กล่าวในหัวข้อ 3.3 ได้มีการพัฒนาเป็นโปรแกรมสำเร็จรูป ในฐานะเครื่องมือสำหรับการพัฒนาแบบจำลองทางคณิตศาสตร์ โปรแกรมสำเร็จรูปที่ได้รับการใช้งานอย่างแพร่หลายคือ Math Works SYSTEM IDENTIFICATION TOOLBOX ที่ปฏิบัติงานบน MATLAB (Ljung, 1995) โปรแกรมดังกล่าวสามารถทำการระบุเอกลักษณ์แบบจำลองชนิดที่ปรากฏพารามิเตอร์ และแบบจำลองชนิดที่ไม่ปรากฏพารามิเตอร์ การระบุเอกลักษณ์ระบบโดยใช้โปรแกรมสำเร็จรูปอาจแสดงได้ดังรูปที่ 3.3 ซึ่งรูปสี่เหลี่ยมผืนผ้าเป็นกระบวนการที่กระทำโดยคอมพิวเตอร์ และรูปวงรีเป็นกระบวนการที่กระทำโดยมนุษย์

จากรูปที่ 3.3 แสดงขั้นตอนการระบุเอกลักษณ์ระบบโดยอาศัยโปรแกรมสำเร็จ เริ่มต้นจากการทดสอบระบบเพื่อเก็บข้อมูลอินพุตและเอาต์พุต ซึ่งในขั้นตอนนี้ดำเนินการโดยมนุษย์ จากนั้นทำการจัดสรรและนำเสนอข้อมูลที่มีนัยสำคัญต่อการพัฒนาแบบจำลอง ซึ่งขั้นตอนนี้ดำเนินการโดยคอมพิวเตอร์ ข้อมูลที่ได้รับการจัดสรรอย่างเหมาะสม จะถูกนำไปใช้เพื่อพัฒนาแบบจำลอง ซึ่งขั้นตอนนี้ดำเนินการโดยคอมพิวเตอร์เช่นกัน หลังจากนั้นแบบจำลองจะได้รับการตรวจสอบความถูกต้องด้วยคอมพิวเตอร์ ขั้นตอนต่อมาคือ การพิจารณาว่าแบบจำลองยอมรับได้หรือไม่โดยมนุษย์ ถ้าผลการตรวจสอบความถูกต้องของแบบจำลองเป็นที่น่าพอใจ จะถือว่าแบบจำลองนั้นได้รับการยอมรับให้ใช้งานต่อไปได้ แต่ถ้าผลการตรวจสอบความถูกต้องของแบบจำลองไม่เป็นที่น่าพอใจ จะถือว่าแบบจำลองนั้นไม่เป็นที่ยอมรับและต้องทำการปรับปรุงแก้ไข ซึ่งอาจเลือกแนวทางปฏิบัติได้สามแนวทางคือ เลือกโครงสร้างของแบบจำลองใหม่ จัดสรรข้อมูลใหม่ หรือทดสอบและเก็บข้อมูลใหม่ ซึ่งขั้นตอนการแก้ไขแบบจำลองที่กล่าวมาจะดำเนินการโดยมนุษย์

3.5 การวางแผนทดสอบระบบ

เพื่อให้ได้ข้อมูลพลวัตของระบบที่มีความครบถ้วนสำหรับการระบุเอกลักษณ์ระบบ การวางแผนทดสอบระบบจึงมีความสำคัญ และจำเป็นต้องคำนึงถึงปัจจัยต่าง ๆ ได้แก่ การเลือกสัญญาณอินพุตในการทดสอบ การเลือกช่วงเวลาชักตัวอย่างสัญญาณ การปรุงแต่งข้อมูล การเลือกโครงสร้างแบบจำลอง และการตรวจสอบความถูกต้องของแบบจำลอง ดังรายละเอียดต่อไปนี้

การเลือกสัญญาณอินพุต สัญญาณอินพุตที่เหมาะสม ควรเป็นสัญญาณที่กระตุ้นระบบอย่างสม่ำเสมอ เช่นสัญญาณรายคาบต่าง ๆ เป็นต้น สำหรับการทดสอบระบบในโดเมนความถี่ ควรเลือกสัญญาณอินพุตที่มีการเลื่อนขนาดสองระดับอย่างสุ่ม (Eykhoff, 1974) สัญญาณดังกล่าวจะครอบคลุมทุกความถี่ ในกรณีการทดสอบระบบไม่เชิงเส้น ระดับของสัญญาณควรมากกว่าสองระดับ สำหรับการทดสอบระบบในโดเมนเวลา ควรเริ่มต้นทดสอบด้วยอินพุตแบบขั้นบันได เพื่อพิจารณาค่าคงที่เวลา (time constant) ของระบบ จากนั้นอาจประยุกต์สัญญาณขบวนพัลส์ (pulse train) ที่มีช่วงความกว้างแตกต่างกัน และในขบวนพัลส์ดังกล่าวควรมีอย่างน้อยหนึ่งช่วงความกว้างที่สามารถใช้แทนสัญญาณแบบขั้นบันไดได้

ในทางปฏิบัติ สัญญาณที่มักถูกใช้เป็นอินพุตสำหรับทดสอบระบบเพื่อการระบุเอกลักษณ์ระบบคือสัญญาณทวิภาคสุ่มเทียม หรือสัญญาณ PRBS (pseudo random binary signal) ซึ่งมีลักษณะเป็นลำดับพัลส์สองระดับที่ผ่านการมอดูเลตเชิงความกว้างอย่างสุ่ม และมีความเป็นรายคาบที่ยาวนานมาก สัญญาณทวิภาคสุ่มเทียมสามารถผลิตได้จากรีจิสเตอร์เลื่อนที่มีการป้อนกลับ ควบคู่ไปกับพีชคณิตมอดูโล (Modulo algebra) (Landau, 1990; Eykhoff, 1974; Johansson, 1993; และ Davies, 1970) ดังนั้นการเลือกสัญญาณอินพุตเพื่อการทดสอบระบบอาจกล่าวได้ว่า สัญญาณ

ทวิภาค หรือสัญญาณสองระดับ เหมาะสมสำหรับการทดสอบระบบเชิงเส้น การเลือกย่านความถี่ของสัญญาณอินพุต ควรเลือกให้ครอบคลุมย่านความถี่ที่ระบบสามารถปฏิบัติงานได้ โดยอาจพิจารณาจาก แบนด์วิดท์ (bandwidth) ของผลตอบสนองทางความถี่ของระบบ สำหรับการเลือกอินพุตเป็นสัญญาณขบวนพัลส์ ต้องมีช่วงความกว้างของพัลส์ที่แตกต่างกัน และมีอย่างน้อยหนึ่งช่วงความกว้างพัลส์ที่สามารถใช้แทนสัญญาณแบบขั้นบันได เพื่อตรวจสอบค่าคงที่เวลาของระบบ

การเลือกช่วงเวลาชักตัวอย่างสัญญาณ ในทฤษฎีบทการชักตัวอย่างสัญญาณ (sampling theorem) ความถี่ในการชักตัวอย่างสัญญาณต้องมีค่ามากกว่าความถี่ในควิสต์ (Nyquist frequency) ไม่เช่นนั้นแล้ว ข้อมูลที่ได้จะปรากฏการเคลือบแฝง (alias) ซึ่งไม่สามารถใช้แทนพลวัตของระบบที่แท้จริงได้ ในทางปฏิบัติ ช่วงเวลาชักตัวอย่างสัญญาณ (sampling interval) สามารถเลือกให้มีการชักตัวอย่างสัญญาณได้อย่างน้อย 5 ถึง 8 ครั้งภายในช่วงเวลาขึ้น (rise time) ของผลตอบสนองในโดเมนเวลาของระบบ ซึ่งจะกำหนดให้มีการชักตัวอย่างสัญญาณที่ละเอียดกว่านี้ก็ได้ ความถี่ของการชักตัวอย่างสัญญาณควรมีค่าประมาณ 10 เท่าของแบนด์วิดท์ของระบบ (Ljung and Glad, 1994)

การปรุงแต่งข้อมูล ในการทดสอบระบบและเก็บข้อมูลนั้น ข้อมูลบางส่วนอาจมีความไม่เหมาะสม เช่น ระดับของข้อมูลเกิดการขยับเลื่อนไปจากกลุ่มข้อมูลหลัก หรือข้อมูลได้รับการรบกวนด้วยสัญญาณรบกวนความถี่สูง เป็นต้น ดังนั้นข้อมูลจะต้องได้รับการปรุงแต่งให้เหมาะสมก่อนที่จะนำไปทำการระบุเอกลักษณ์ระบบ การปรุงแต่งข้อมูลอาจใช้ตัวกรองแบบต่าง ๆ เช่น ใช้ตัวกรองผ่านต่ำ (low-pass filter) เพื่อกรองสัญญาณรบกวนความถี่สูงออกไป ในบางกรณีอาจต้องพึ่งพาเทคนิคเชิงตัวเลข ตลอดจนเทคนิคการทำนายข้อมูล เป็นต้น

การเลือกโครงสร้างแบบจำลอง การเลือกโครงสร้างแบบจำลองที่เหมาะสม เป็นอีกปัจจัยหนึ่งที่สำคัญอย่างยิ่งในการระบุเอกลักษณ์ระบบ ปัญหาการเลือกโครงสร้างของแบบจำลอง อาจพบได้หลายลักษณะ เช่น การเลือกโครงสร้างแบบจำลองระหว่าง แบบจำลองเฉพาะกรณี และแบบจำลองทั่วไป การเลือกแบบจำลองกล่องดำระหว่างแบบจำลอง ARX ARMAX OE หรือ BJ และการเลือกอันดับ (order) ที่เหมาะสมของแบบจำลอง

แบบจำลองเฉพาะกรณี จะเหมาะกับระบบที่สามารถวิเคราะห์แบบจำลองได้โดยตรง โดยอาศัยความสัมพันธ์ทางกายภาพ ค่าพารามิเตอร์ที่ปรากฏในแบบจำลองจะสื่อความหมายทางกายภาพของระบบโดยตรง ซึ่งอาจได้มาจากการทดสอบตามแบบแผน หรือจากข้อกำหนดที่ปรากฏในใบแสดงข้อมูล (data sheet) อย่างไรก็ตาม แบบจำลองประเภทนี้อาจประสบปัญหาในกรณีที่ระบบมีความซับซ้อน จนไม่สามารถหาความสัมพันธ์ทางกายภาพเชิงวิเคราะห์ได้ สำหรับแบบจำลองทั่วไป หรือแบบจำลองกล่องดำ เป็นแบบจำลองที่ได้รับการประยุกต์ใช้งานอย่าง

กว้างขวาง ค่าพารามิเตอร์ที่ปรากฏในแบบจำลองจะไม่สื่อความหมายทางกายภาพของระบบ แต่ถูกใช้เพื่ออธิบายความสัมพันธ์ของอินพุตและเอาต์พุตของระบบ การเลือกใช้แบบจำลอง ก่อตั้งขึ้นอยู่กับระบบ และสถานการณ์ที่เกิดขึ้นกับระบบเป็นสำคัญ ในเบื้องต้นอาจสรุปได้ ดังนี้คือ แบบจำลอง ARX เป็นแบบจำลองที่ง่ายที่สุด และเหมาะที่จะใช้เป็นแบบจำลองเริ่มต้นการ ระบุเอกลักษณ์ระบบ แบบจำลอง ARMAX เป็นแบบจำลองที่มีความอ่อนตัวกว่าแบบจำลอง ARX และสามารถจัดการปัญหาเกี่ยวกับพลวัตของการรบกวนได้ แบบจำลอง OE มีข้อดีคือ แบบจำลองดังกล่าวมีเพียงพจน์ของแบบจำลองเอาต์พุตที่อิสระต่อสัญญาณรบกวน ซึ่งจะเหมาะสม กับระบบที่มีการปฏิบัติงานภายใต้เงื่อนไขสภาวะแวดล้อมเช่นนั้น แบบจำลอง BJ เป็นแบบจำลอง ที่สมบูรณ์แบบที่สุดในบรรดาแบบจำลองก่อดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ เพราะพจน์ ของแบบจำลองเอาต์พุตที่อิสระต่อสัญญาณรบกวน และพจน์ของแบบจำลองการรบกวน สามารถ แยกพิจารณาพลวัตออกจากกันได้อย่างอิสระ จากที่กล่าวมาจะพบว่าแบบจำลอง ARX และ ARMAX มีการพิจารณาพลวัตร่วมกันระหว่างสัญญาณอินพุตกับสัญญาณรบกวน ซึ่งเหมาะกับ ระบบที่มีสัญญาณทั้งสองเข้ามากระทำกับระบบพร้อมกัน ขณะที่แบบจำลอง BJ จะเหมาะกับระบบ ที่มีสัญญาณอินพุตและสัญญาณรบกวนเข้ามากระทำกับระบบไม่พร้อมกัน สำหรับการเลือกอันดับ ของแบบจำลอง จำเป็นต้องพิจารณาแยกออกเป็นสองส่วน กล่าวคือ ส่วนที่เป็นอันดับของ แบบจำลอง และส่วนที่เป็นช่วงเวลาประวิง (delay time) ซึ่งการมีช่วงเวลาประวิงปรากฏอาจต้อง ตั้งเกิดจากผลตอบสนองต่ออินพุตแบบขึ้นบันได การเลือกอันดับของแบบจำลองควรพิจารณาให้ เหมาะสมกับระบบที่ทดสอบ โดยอาจอาศัยความรู้พื้นฐานทางกายภาพของระบบ หรือองค์ประกอบ ต่าง ๆ ของระบบ เพื่อช่วยในการคาดคะเนอันดับของแบบจำลองที่ควรจะเป็น โดยเริ่มต้นปรับ อันดับของแบบจำลองจากน้อยไปสู่มาก (จากอันดับต่ำสู่อันดับสูง) ถ้าแบบจำลองมีอันดับสูงมาก อาจจำเป็นต้องพิจารณาถึงความเป็นไปได้ที่โพลและซีโรของแบบจำลองจะสามารถหักล้างกัน เพื่อ ลดอันดับของแบบจำลอง อย่างไรก็ตาม การพิจารณาถึงความถูกต้องเหมาะสมของโครงสร้าง แบบจำลอง รวมถึงอันดับของแบบจำลอง จะต้องให้ความสนใจในเรื่องคุณภาพของแบบจำลอง เป็นสำคัญ ดังจะกล่าวต่อไป

การตรวจสอบความถูกต้องของแบบจำลอง เป็นการทดสอบความถูกต้องของแบบจำลอง เพื่อดูความสมเหตุสมผลในการใช้แบบจำลองอธิบายพลวัตของระบบ การตรวจสอบจะสอดคล้องกับ การพิจารณาคุณภาพของแบบจำลอง โดยให้การพิจารณาถึงสองประเด็นต่อไปนี้ (1) แบบจำลอง สามารถพัฒนาขึ้นจากชุดข้อมูลที่แตกต่างกันหรือคนละชุดกัน (จากระบบพลวัตระบบเดียวกัน) โดยที่แบบจำลองที่พัฒนาขึ้นจะต้องมีคุณสมบัติที่เหมือนกัน หรือคล้ายคลึงกัน กล่าวอีกนัยหนึ่ง ก็คือ ไม่ว่าโครงสร้างของแบบจำลองจะมีลักษณะแตกต่างกันอย่างไร จะต้องสามารถสื่อความ หมายถึงพลวัตของระบบได้อย่างครอบคลุม แม้ว่าแบบจำลองนั้น ๆ จะพัฒนาขึ้นมาจากข้อมูล

คนละชุดกันก็ตาม (2) แบบจำลองจะต้องแสดงถึงพฤติกรรมทางพลวัตของระบบได้อย่างถูกต้อง ซึ่งหมายความว่าเอาต์พุตของแบบจำลองจะต้องสอดคล้องกับเอาต์พุตของระบบที่ได้จากการทดสอบ แม้ว่าข้อมูลอินพุตที่ใช้ตรวจสอบความถูกต้องของแบบจำลองเพื่อดูพฤติกรรมทางพลวัตนั้น จะเป็นคนละชุดกับข้อมูลอินพุตที่ใช้ในการระบุเอกลักษณ์ระบบก็ตาม

แนวทางปฏิบัติทั่วไปนั้น ข้อมูลที่ได้จากการทดสอบจะถูกแบ่งออกเป็นสองส่วน ข้อมูลส่วนแรก ใช้เพื่อการระบุเอกลักษณ์แบบจำลองของระบบ ข้อมูลส่วนหลัง ใช้เพื่อตรวจสอบความถูกต้องของแบบจำลอง หากผลการตรวจสอบความถูกต้องของแบบจำลองด้วยข้อมูลส่วนหลังนี้ มีความสมเหตุสมผล จะถือว่าแบบจำลองดังกล่าวสามารถใช้อธิบายพฤติกรรมทางพลวัตของระบบ และสามารถใช้แทนระบบพลวัตตามวัตถุประสงค์ในบริบทระบบควบคุมต่อไปได้

3.6 สรุป

ในบทนี้ได้กล่าวถึงการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม ที่ต้องอาศัยข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ กล่าวถึงการระบุเอกลักษณ์แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ แบบจำลองดังกล่าวมีอยู่ด้วยกัน 4 รูปแบบคือ แบบจำลอง ARX ARMAX OE และ BJ โครงสร้างของแบบจำลองจะอยู่ในรูปของฟังก์ชันถ่ายโอนโดเมน q กล่าวถึงแนวทางการระบุเอกลักษณ์ระบบด้วยโปรแกรมสำเร็จ ซึ่งเป็นกระบวนการที่มีปฏิสัมพันธ์ระหว่างมนุษย์กับคอมพิวเตอร์ และกล่าวถึงการวางแผนทดสอบระบบ อันประกอบด้วย การเลือกสัญญาณอินพุต การเลือกช่วงเวลาชักตัวอย่างสัญญาณ การปรุงแต่งข้อมูลที่ได้จากการทดสอบ การเลือกโครงสร้างแบบจำลอง และการตรวจสอบความถูกต้องของแบบจำลองที่ได้จากการระบุเอกลักษณ์ระบบ

บทที่ 4

การประมวลผลภาพ

4.1 กล่าวนำ

ดังได้กล่าวมาแล้วในบทที่ 1 การระบุเอกลักษณ์ระบบที่ไม่สามารถติดตั้งตัวตรวจรู้เข้ากับระบบได้ จำเป็นต้องอาศัยกล้องบันทึกภาพทำหน้าที่เป็นตัวตรวจรู้ เพื่อทำการบันทึกภาพการเคลื่อนไหวของระบบในช่วงเวลาที่สนใจ จากนั้นประยุกต์เทคนิคการประมวลผลภาพที่เหมาะสมเพื่อสกัดข้อมูลพลวัตของระบบจากภาพหรือลำดับของภาพที่บันทึกได้ และนำข้อมูลพลวัตที่สกัดได้จากภาพหรือลำดับของภาพ มาทำการระบุเอกลักษณ์ระบบต่อไป ดังนั้นในบทที่ 4 นี้ จะกล่าวถึงเทคนิคการประมวลผลภาพที่เอื้อประโยชน์ต่อการสกัดข้อมูลพลวัต มีเนื้อหาเริ่มต้นจากความเป็นมาของการประมวลผลภาพ การแบ่งส่วนภาพ (image segmentation) การระบุตำแหน่งวัตถุในภาพโดยใช้กลไกการกำหนดจุดเปลี่ยนค่า (thresholding) และการประยุกต์การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ

4.2 การประมวลผลภาพ

ในหัวข้อนี้ กล่าวถึงความเป็นมาของการประมวลผลภาพเบื้องต้น รูปแบบของภาพดิจิทัล การแบ่งส่วนภาพ และกลไกการกำหนดจุดเปลี่ยนค่า ดังรายละเอียดต่อไปนี้

4.2.1 ความเป็นมาของการประมวลผลภาพ

การประมวลผลภาพมีวัตถุประสงค์หลักสองประการคือ เพื่อการปรับปรุงข้อมูลภาพสำหรับแปลความหมาย และเพื่อการประมวลผลข้อมูลภาพจากการมองเห็นของเครื่องจักรกล

การปรับปรุงภาพในยุคแรก ๆ จะเกี่ยวข้องกับกระบวนการรับภาพ และการกระจายระดับความสว่างภายในภาพ สำหรับกระบวนการรับภาพในยุคแรก ๆ นั้นจะใช้วิธีพื้นฐานเพื่อการสร้างภาพใหม่ จากแผ่นเทปแบบมีรูที่อยู่บนเครื่องรับสัญญาณระยะไกล ส่วนการกระจายระดับความสว่างภายในภาพในยุคแรก ๆ นั้น มีความสามารถในการให้ความสว่างของภาพได้เพียง 5 ระดับ จนถึงปี ค.ศ. 1929 จึงเพิ่มขึ้นเป็น 15 ระดับ

ในปี ค.ศ. 1964 ได้มีการพัฒนาการปรับปรุงภาพด้วยคอมพิวเตอร์เกิดขึ้น ซึ่งเริ่มต้นในห้องปฏิบัติการเครื่องบินไอพ่นแบบขับเคลื่อน (Jet-Propulsion laboratory) ที่รัฐแคลิฟอร์เนีย สหรัฐอเมริกา (Rafael and Richard, 1992) ภาพดวงจันทร์ที่ถูกส่งลงมาโดยเครื่อง Ranger 7 ได้รับการประมวลผลเพื่อปรับแก้ความเพี้ยน (distortion) ของภาพ อันเนื่องมาจากคุณสมบัติของ

กล้องบันทึกภาพ เทคนิคดังกล่าวถูกใช้เป็นพื้นฐานของการปรับปรุงภาพ และการสร้างภาพใหม่ในการสำรวจดวงจันทร์ และดาวอังคาร รวมถึงภารกิจในการเดินทางสำรวจอวกาศด้วยยานอพอลโล (Apollo) อีกด้วย

จากปี ค.ศ. 1964 ถึงปัจจุบัน เทคนิคการประมวลผลภาพแบบดิจิทัลได้รับการนำไปประยุกต์ใช้เพื่อแก้ปัญหาในงานต่าง ๆ มากมาย อาทิเช่น ด้านการแพทย์ ได้มีการนำเอาเทคนิคการประมวลผลภาพมาใช้ในการปรับปรุงความเปรียบต่าง (contrast) หรือระดับความเข้มของสี จากภาพเอกซเรย์ (X-rays) หรือภาพถ่ายทางการแพทย์อื่น ๆ เพื่อความสะดวกในการแปลความหมายและวินิจฉัยโรคของแพทย์ ด้านภูมิศาสตร์ ได้มีการนำเอาเทคนิคการประมวลผลภาพมาใช้ในการศึกษากระแสนมลภาวะ (pollution patterns) จากภาพถ่ายทางอากาศ และจากภาพถ่ายดาวเทียม ด้านโบราณคดี ได้มีการนำเอาเทคนิคการประมวลผลภาพมาใช้ในการสร้างภาพที่ไม่ชัดหรือภาพที่ชำรุดทรุดโทรมขึ้นมาใหม่ ซึ่งภาพเหล่านี้มีคุณค่าทางประวัติศาสตร์อย่างยิ่ง ด้านฟิสิกส์ ได้มีการนำเอาเทคนิคการประมวลผลภาพมาใช้ในการปรับปรุงคุณภาพของภาพจากการทดลองต่าง ๆ เช่น ภาพผลการทดสอบพลาสมาพลังงานสูง (high-energy plasmas) และภาพถ่ายอิเล็กตรอนไมโครสโคป (electron microscopy) นอกจากนี้ ยังมีการประยุกต์ใช้เทคนิคการประมวลผลภาพในงานด้านดาราศาสตร์ นิวเคลียร์ ชีววิทยา ภูมิศาสตร์ และอุตสาหกรรมอีกด้วย

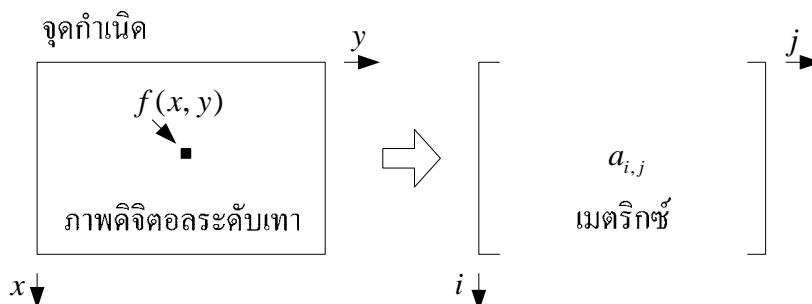
สำหรับการประมวลผลข้อมูลภาพจากการมองเห็นของเครื่องจักรกล จะให้ความสนใจต่อการสกัด หรือถอดรายละเอียดข้อมูลภาพด้วยคอมพิวเตอร์ การประยุกต์การประมวลผลข้อมูลภาพในลักษณะดังกล่าว ได้แก่ การรู้จำคุณลักษณะแบบอัตโนมัติ การมองเห็นของเครื่องจักรกลอุตสาหกรรมเพื่อการประกอบ หรือเพื่อตรวจสอบผลิตภัณฑ์ งานรู้จำภาพถ่ายทางทหาร การประมวลผลภาพถ่ายนิ้วมือมนุษย์ การกรองภาพเอกซเรย์และภาพถ่ายอย่างเล็อด และการประมวลผลภาพถ่ายทางอากาศ หรือภาพถ่ายดาวเทียม เพื่อการพยากรณ์อากาศ และเพื่อศึกษาพืชผลทางการเกษตร เป็นต้น

4.2.2 ภาพดิจิทัล

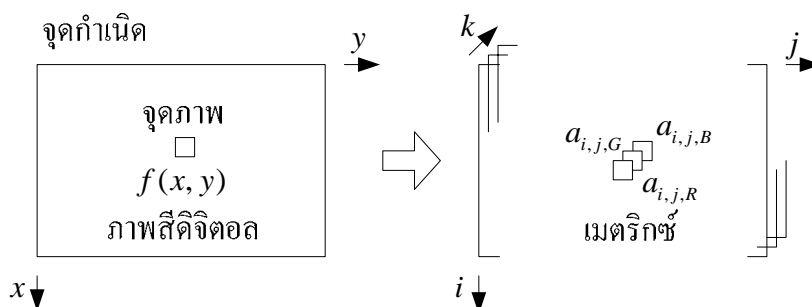
ในกรณีของภาพดิจิทัลระดับเทา (gray scale) หรือภาพโมนochrome) ฟังก์ชันของความเข้มแสงแบบ 2 มิติ กำหนดให้เป็น $f(x, y)$ เมื่อ x และ y คือคู่ลำดับในปริภูมิของภาพ ค่า f ที่จุด x, y สัมพันธ์กับความสว่างของภาพที่จุดภาพนั้น ๆ ดังนั้นภาพดิจิทัลระดับเทาสามารถพิจารณาเป็นเมตริกซ์ ซึ่งมีองค์ประกอบเป็น $a_{i,j}$ เมื่อ i, j แสดงถึงตำแหน่งของคู่ลำดับ x, y โดยที่ a คือ f ดังแสดงในรูปที่ 4.1 และอาจเรียก a ว่าเป็นองค์ประกอบของภาพหรือจุดภาพ (pixels)

ในกรณีภาพสี (color image) จะพิจารณาแม่สีตามมาตรฐานของระบบกล้องวิดีโอ โดยกำหนดให้แต่ละจุดภาพประกอบด้วยแม่สีทั้งสามคือ สีแดง (R) สีเขียว (G) และสี

น้ำเงิน (B) ภาพสีดิจิทัลสามารถพิจารณาเป็นเมตริกซ์ 3 มิติ ดังรูปที่ 4.2 โดยที่ค่า f ที่จุด x, y คือ $a_{i,j,k}$ เมื่อ i, j คือ คู่ลำดับ x, y และ k คือ ค่าสีแดง เขียว และน้ำเงิน ตามลำดับ



รูปที่ 4.1 ภาพดิจิทัลระดับเทาเทียบกับเมตริกซ์



รูปที่ 4.2 ภาพสีดิจิทัลเทียบกับเมตริกซ์

ค่า f ทั้งในกรณีของภาพดิจิทัลระดับเทา และภาพสีดิจิทัล มีปริมาณจำกัด เพราะแสงเป็นพลังงานชนิดหนึ่ง ดังนั้น $f(x, y)$ สามารถเขียนได้ดังสมการที่ (4-1)

$$0 < f(x, y) < \infty \quad (4-1)$$

โดยธรรมชาติ ค่า $f(x, y)$ จะขึ้นกับความสว่างของแหล่งพลังงานแสง และการสะท้อนของวัตถุรับแสง เมื่อกำหนดให้ $i(x, y)$ คือ ความสว่างของแหล่งพลังงานแสง และ $r(x, y)$ คือ การสะท้อนของวัตถุรับแสง ค่า $f(x, y)$ สามารถเขียนได้ดังสมการที่ (4-2)

$$f(x, y) = i(x, y)r(x, y) \quad (4-2)$$

$$0 < i(x, y) < \infty \quad (4-3)$$

$$0 < r(x, y) < 1 \quad (4-4)$$

จากสมการที่ (4-3) ความสว่างของแหล่งพลังงานแสงถูกกำหนดให้อยู่ในช่วง 0 ถึง ∞ ยกตัวอย่างเช่น ดวงอาทิตย์จะให้แสงสว่างสู่ผิวโลกมากกว่า 9,000 แสงเทียนต่อตารางฟุต (foot-candles) ในวันฟ้าโปร่ง แต่จะลดลงเหลือน้อยกว่า 1,000 แสงเทียนต่อตารางฟุต ในวันที่ท้องฟ้ามีเมฆปกคลุม ดวงจันทร์เต็มดวงจะให้ความสว่างเท่ากับ 0.01 แสงเทียนต่อตารางฟุต ขณะที่ความสว่างของห้องทำงานปกติจะมีค่าประมาณ 100 แสงเทียนต่อตารางฟุต เป็นต้น จากสมการที่ (4-4) การสะท้อนของวัตถุรับแสงถูกกำหนดให้มีค่าเท่ากับ 0 เมื่อมีการดูดซับแสงไว้ทั้งหมด นั่นคือไม่มีการสะท้อนของแสงออกมาจากวัตถุรับแสงเลย และมีค่าเท่ากับ 1 เมื่อแสงมีการสะท้อนออกมาจากวัตถุรับแสงทั้งหมด ยกตัวอย่างเช่น กำมะหยี่สีดำ มีค่า $r(x, y)$ เท่ากับ 0.01 เหล็กสแตนเลส มีค่า $r(x, y)$ เท่ากับ 0.65 ผนังเรียบสีขาว มีค่า $r(x, y)$ เท่ากับ 0.80 โลหะแผ่นสีเงิน มีค่า $r(x, y)$ เท่ากับ 0.91 และหิมะ มีค่า $r(x, y)$ เท่ากับ 0.93 เป็นต้น

ในปัจจุบัน การประมวลผลภาพดิจิทัลได้แตกแขนงออกไปมากมาย เช่น การแปลงภาพ (image transforms) การปรับปรุงภาพ (image enhancement) การสร้างภาพใหม่ (image restoration) การบีบอัดภาพ (image compression) การแบ่งส่วนภาพ (image segmentation) การรู้จำและแปลความภาพ (image recognition and interpretation) และการแทนและพรรณนาภาพ (image representation and description) เป็นต้น อย่างไรก็ตาม ในงานวิจัยนี้ให้ความสนใจการแบ่งส่วนภาพ เพราะเกี่ยวข้องและเป็นประโยชน์ต่องานวิจัยที่จะดำเนินการ ผู้อ่านสามารถศึกษาเทคนิคการประมวลผลภาพแบบต่าง ๆ ได้จากตำราการประมวลผลภาพดิจิทัลทั่วไป เช่น AI (2000) และ Rafael และ Richard (1992) เป็นต้น

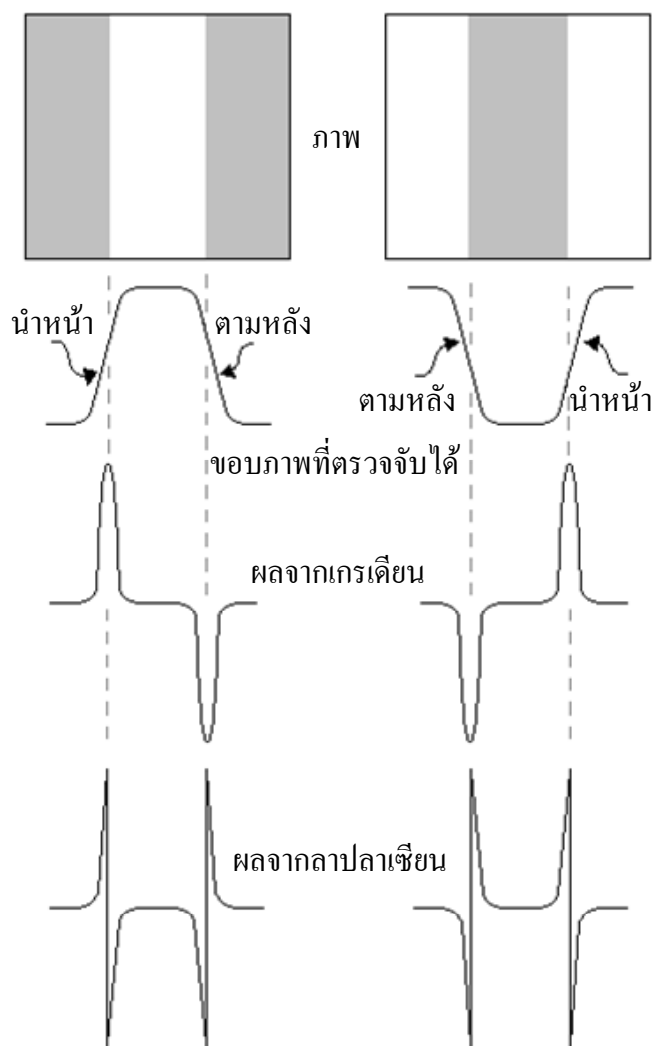
4.2.3 การแบ่งส่วนภาพ

การแบ่งส่วนภาพเป็นการวิเคราะห์ภาพเพื่อสกัดข้อมูลความรู้จากภาพ อัลกอริทึมอาศัยคุณสมบัติความไม่ต่อเนื่อง (discontinuity) และความคล้ายคลึงกัน (similarity)

คุณสมบัติความไม่ต่อเนื่อง ประกอบด้วย การตรวจจับจุดภาพ การตรวจจับเส้นภาพ และการตรวจจับขอบภาพ

- การตรวจจับจุดภาพ (point detection) ใช้หลักการพิจารณาจุดภาพที่สนใจ และจุดภาพใกล้เคียง (neighbors) ที่อยู่ติดกับจุดภาพที่พิจารณา โดยใช้ค่าสีเป็นเกณฑ์ในการพิจารณา

- การตรวจจับเส้นภาพ (line detection) ใช้หลักการเดียวกันกับการตรวจจับจุดภาพ โดยจะเพิ่มการพิจารณาความต่อเนื่อง หรือความเหมือนกันของค่าสีของจุดภาพที่อยู่ติดกัน ค่าสีของจุดภาพที่สนใจจะถูกใช้เป็นเกณฑ์ในการพิจารณาความต่อเนื่องของจุดภาพ ซึ่งสื่อความหมายของการเป็นเส้นภาพนั่นเอง



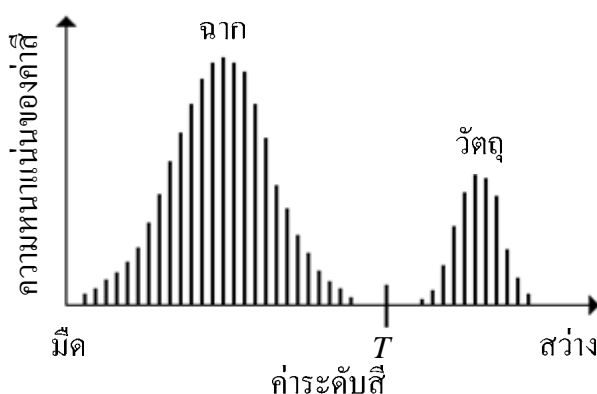
รูปที่ 4.3 การตรวจจับขอบภาพโดยใช้เกรเดียนและลาปลาเซียน

- การตรวจจับขอบภาพ (edge detection) มีหลายเทคนิควิธีด้วยกัน แต่หลักการพื้นฐานคือการทำอนุพันธ์เส้นภาพ ตัวปฏิบัติการทางคณิตศาสตร์สำหรับการทำอนุพันธ์เส้นภาพคือเกรเดียน (Gradient) ซึ่งเป็นการทำอนุพันธ์อันดับหนึ่งให้กับเส้นภาพ และลาปลาเซียน (Laplacian) ซึ่งเป็นการทำอนุพันธ์อันดับสองให้กับเส้นภาพ ดังแสดงในรูปที่ 4.3 การทำอนุพันธ์อันดับหนึ่งของค่าสีระดับเทาโดยเกรเดียน จะให้ค่าบวกเมื่อขอบภาพนำหน้า (leading edge) จะให้ค่าลบเมื่อขอบภาพตามหลัง (tailing edge) และจะให้ค่าศูนย์เมื่อค่าสีคงที่ สำหรับการทำอนุพันธ์อันดับสองของค่าสีระดับเทาโดยลาปลาเซียน จะให้ค่าบวกเมื่อมีการเปลี่ยนแปลงทางด้านมืดของขอบภาพ จะให้ค่าลบเมื่อมีการเปลี่ยนแปลงทางด้านสว่างของขอบภาพ และจะให้ค่าศูนย์

เมื่อค่าสีคงที่ ผลการทำอนุพันธ์อันดับหนึ่ง สามารถตรวจจับขอบภาพที่ปรากฏได้ และเครื่องหมาย (sign) จากผลการทำอนุพันธ์อันดับสอง สามารถวินิจฉัยขอบภาพด้านมืดและด้านสว่างได้

คุณสมบัติความคล้ายคลึงกัน มีหลายเทคนิควิธี เช่น การกำหนดจุดเปลี่ยนค่า (thresholding) เพื่อใช้ตรวจจับวัตถุในภาพ เมื่อค่าสีของกลุ่มของจุดภาพที่สนใจมีค่าต่างกัน แต่ในความเป็นจริงแล้วเป็นวัตถุชิ้นเดียวกัน หรือการแบ่งส่วนของพื้นที่วัตถุ (region-oriented segmentation) ซึ่งใช้พีชคณิตเกี่ยวกับเซต (set) ของจุดภาพเพื่อพิจารณาขอบเขตของพื้นที่ของวัตถุที่สนใจในภาพ และการใช้การเคลื่อนที่เพื่อแบ่งส่วนภาพ (the use of motion in segmentation) ซึ่งจะใช้แบ่งส่วนภาพในกรณีที่วัตถุมีการเคลื่อนที่แบบ 3 มิติ อย่างไรก็ตามงานวิจัยนี้ให้ความสนใจการแบ่งส่วนภาพโดยใช้การกำหนดจุดเปลี่ยนค่า ซึ่งมีรายละเอียดดังต่อไปนี้

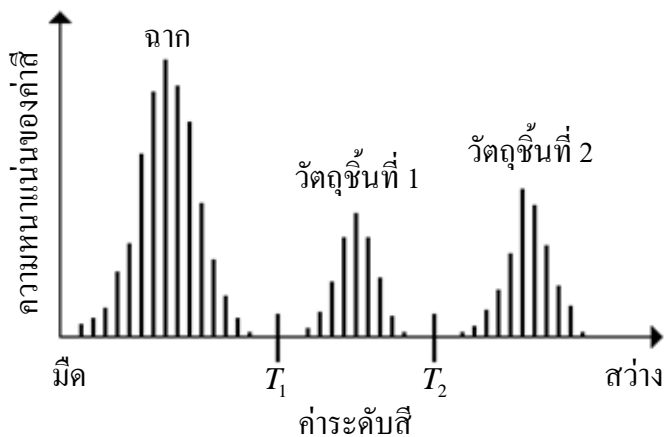
การกำหนดจุดเปลี่ยนค่า อาศัยฮิสโตแกรม (histogram) ซึ่งเป็นกราฟที่แสดงถึงความถี่ หรือความหนาแน่นของค่าสีในกลุ่มของจุดภาพที่สนใจ รูปที่ 4.4 แสดงฮิสโตแกรมของค่าสีของวัตถุที่มีความสว่างอยู่บนฉากมืด กำหนดให้วัตถุและฉากมีค่าสีระดับเทา ซึ่งสามารถแบ่งส่วนภาพออกได้ 2 ส่วนโดยอาศัยจุดเปลี่ยนค่า T กล่าวคือ ถ้าจุดภาพ x, y ใด ๆ ทำให้ $f(x, y) > T$ จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของวัตถุ และในทางตรงข้าม ถ้าจุดภาพ x, y ใด ๆ ทำให้ $f(x, y) \leq T$ จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของฉาก



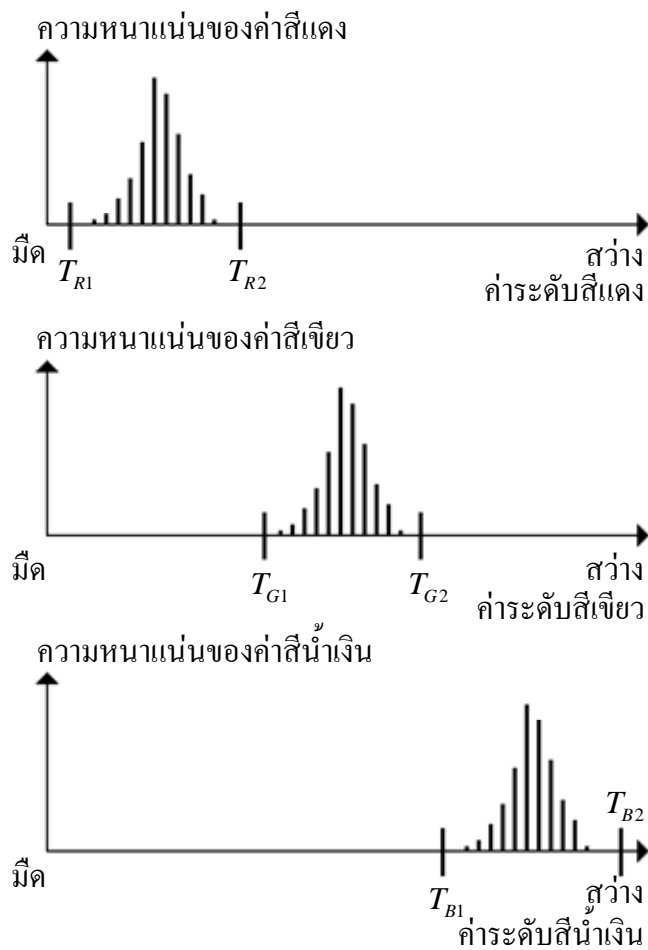
รูปที่ 4.4 ฮิสโตแกรมของภาพที่มีวัตถุสีสว่างอยู่บนฉากมืด

ในกรณีที่วัตถุ 2 ชิ้น ซึ่งค่าสีของวัตถุทั้งสองมีความสว่างต่างกันอยู่บนฉากมืด ฮิสโตแกรมแสดงดังรูปที่ 4.5 ซึ่งสามารถแบ่งส่วนภาพออกได้ 3 ส่วนโดยอาศัยจุดเปลี่ยนค่า T_1 และ T_2 กล่าวคือ ถ้าจุดภาพ x, y ใด ๆ ทำให้ $T_1 < f(x, y) \leq T_2$ จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของวัตถุ

ชั้นที่ 1 ถ้าจุดภาพ x, y ใด ๆ ทำให้ $f(x, y) > T_2$ จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของวัตถุชั้นที่ 2 และถ้าจุดภาพ x, y ใด ๆ ทำให้ $f(x, y) \leq T_1$ จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของฉาก



รูปที่ 4.5 ฮิสโตแกรมของภาพที่มีวัตถุสีสว่าง 2 ชั้นอยู่บนฉากมืด



รูปที่ 4.6 ฮิสโตแกรมของวัตถุในภาพสีดิจิทัล

ในกรณีภาพสีดิจิทัล การกำหนดจุดเปลี่ยนค่าแสดงดังรูปที่ 4.6 ซึ่งฮิสโตแกรมจะแยกพิจารณาตามแม่สีในกลุ่มของจุดภาพที่สนใจ จากรูปที่ 4.6 (T_{R1}, T_{R2}) , (T_{G1}, T_{G2}) , และ (T_{B1}, T_{B2}) คือจุดเปลี่ยนค่าที่ใช้พิจารณาค่าสีแดง $f_R(x, y)$, สีเขียว $f_G(x, y)$, และสีน้ำเงิน $f_B(x, y)$ ของจุดภาพ x, y ตามลำดับ ถ้าจุดภาพ x, y ใด ๆ ทำให้ $T_{R1} < f_R(x, y) \leq T_{R2}$ และ $T_{G1} < f_G(x, y) \leq T_{G2}$ และ $T_{B1} < f_B(x, y) \leq T_{B2}$ จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของวัตถุที่สนใจ ไม่เช่นนั้นแล้ว จะถือว่าจุดภาพ x, y ดังกล่าวอยู่ในกลุ่มของวัตถุอื่น หรืออยู่ในกลุ่มของฉากที่กำหนดดังกล่าวอาจเขียนแสดงด้วยฟังก์ชัน $g(x, y)$ ดังสมการที่ (4-5)

$$g(x, y) = \begin{cases} 1, & \begin{cases} T_{R1} < f_R(x, y) \leq T_{R2} \\ T_{G1} < f_G(x, y) \leq T_{G2} \\ T_{B1} < f_B(x, y) \leq T_{B2} \end{cases} \\ 0, & \text{otherwise} \end{cases} \quad (4-5)$$

เมื่อ $g(x, y)$ คือ ค่าตรรกะของจุดภาพ x, y ที่ทำให้ $f_j(x, y)$, $j = R, G, B$ มีค่าอยู่ในช่วงที่กำกับโดยจุดเปลี่ยนค่า กล่าวคือ ถ้า $g(x, y)$ เท่ากับ 1 จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของวัตถุที่สนใจ แต่ถ้า $g(x, y)$ เท่ากับ 0 จะถือว่าจุดภาพนั้นอยู่ในกลุ่มของวัตถุอื่น หรืออยู่ในกลุ่มของฉากนั่นเอง ในกรณีที่ภาพสีดิจิทัล มีวัตถุที่สนใจมากกว่าหนึ่งชิ้น ฟังก์ชัน $g(x, y)$ จะแสดงดังสมการที่ (4-6) เมื่อ i คือจำนวนของวัตถุในภาพ

$$g_i(x, y) = \begin{cases} 1, & \begin{cases} T_{R1,i} < f_{R,i}(x, y) \leq T_{R2,i} \\ T_{G1,i} < f_{G,i}(x, y) \leq T_{G2,i} \\ T_{B1,i} < f_{B,i}(x, y) \leq T_{B2,i} \end{cases} \\ 0, & \text{otherwise} \end{cases} \quad (4-6)$$

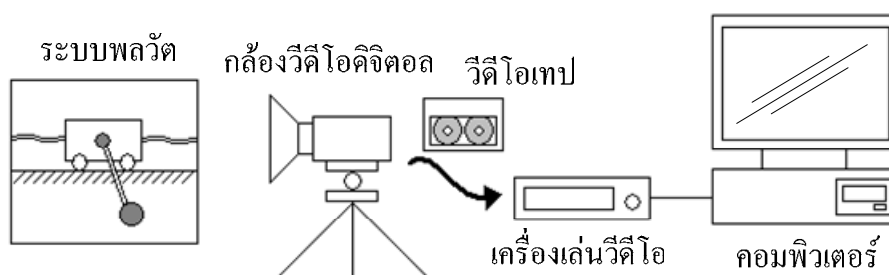
4.3 การประยุกต์การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ

การประยุกต์การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ ในงานวิจัยนี้หมายถึงการนำเอาเทคนิคการประมวลผลภาพไปทำการสกัดข้อมูลพลวัตของระบบ จากภาพหรือลำดับของภาพ เป็นกระบวนการทำงานร่วมกันระหว่างกล้องบันทึกภาพ และเทคนิคการประมวลผลภาพที่เหมาะสมแทนตัวจริงทั่วไป การได้มาซึ่งข้อมูลพลวัตของระบบจากกระบวนการดังกล่าว อาจกระทำได้สองแนวทางดังนี้

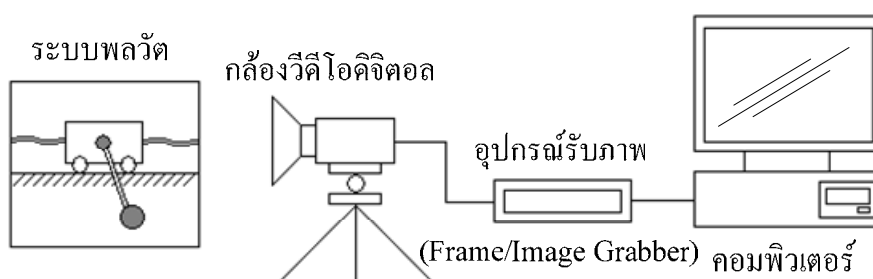
แนวทางแรก ดังแสดงในรูปที่ 4.7 อาศัยกล้องวีดีโอดิจิทัล เพื่อทำการบันทึกภาพการเคลื่อนที่ของระบบพลวัตในช่วงเวลาที่สนใจ การเคลื่อนที่ดังกล่าวได้รับการบันทึกไว้ในวีดีโอเทป

จากนั้นข้อมูลภาพในวิดีโอเทปจะได้ระบบการอ่านผ่านเครื่องเล่นวิดีโอ และส่งเข้าสู่คอมพิวเตอร์ เพื่อการประมวลผลภาพด้วยเทคนิคที่เหมาะสม และในท้ายที่สุดจะได้ข้อมูลพลวัตของระบบ กระบวนการสกัดข้อมูลพลวัตด้วยการประมวลผลภาพตามแนวทางที่กล่าวมานี้ ถือเป็นการทำงานแบบ off line

แนวทางต่อมา ดังแสดงในรูปที่ 4.8 อาศัยกล้องวิดีโอดิจิทัล เพื่อทำการบันทึกภาพการเคลื่อนที่ของระบบพลวัตในช่วงเวลาที่สนใจเช่นเดียวกับแนวทางแรก หากแต่ข้อมูลภาพที่ได้จากกล้องบันทึกภาพ มิได้รับการบันทึกลงในวิดีโอเทป แต่จะถูกส่งไปยังอุปกรณ์รับภาพ (frame/image grabber) และจากนั้นข้อมูลภาพจะได้รับการส่งต่อไปยังคอมพิวเตอร์ เพื่อการประมวลผลภาพด้วยเทคนิคที่เหมาะสม และในท้ายที่สุดจะได้ข้อมูลพลวัตของระบบ กระบวนการสกัดข้อมูลพลวัตด้วยการประมวลผลภาพตามแนวทางที่กล่าวมานี้ ถือเป็นการทำงานแบบ on line



รูปที่ 4.7 กระบวนการสกัดข้อมูลพลวัตด้วยการประมวลผลภาพแบบ off line



รูปที่ 4.8 กระบวนการสกัดข้อมูลพลวัตด้วยการประมวลผลภาพแบบ on line

การประมวลผลภาพเพื่อสกัดข้อมูลพลวัตจากภาพหรือลำดับของภาพที่ใช้ในงานวิจัยนี้ คือ การแบ่งส่วนภาพโดยใช้การกำหนดจุดเปลี่ยนค่า สิ่งที่ต้องดำเนินการในขั้นเตรียมการมีดังนี้

ประการแรก ระบบพลวัตเองจะต้องได้รับการจัดทำแถบสีที่แตกต่างกัน ในแต่ละองค์ประกอบของระบบ อันอาจประกอบไปด้วยฉากหลัง องค์ประกอบของระบบที่อยู่กับที่ และ

องค์ประกอบของระบบที่เคลื่อนที่ องค์ประกอบเหล่านี้จะปรากฏเป็นวัตถุในภาพหรือลำดับของภาพ วัตถุที่มีการเคลื่อนที่ถือเป็นตัวแปรพลวัตของระบบที่สนใจ ซึ่งจำเป็นจะต้องได้รับการจัดทำแถบสีที่แตกต่างกับวัตถุที่อยู่กับที่ และจากหลัง อย่างมีนัยสำคัญ ทั้งนี้เพื่อความสะดวกในการสกัดข้อมูลพลวัตด้วยการกำหนดจุดเปลี่ยนค่า

ประการต่อมา กล้องบันทึกภาพที่ใช้จะต้องมีความแยกชัด หรือมีอัตราภาพ (frame rate) สูงเพียงพอที่จะสามารถบันทึกข้อมูลได้อย่างครบถ้วน ภายในช่วงเวลาที่น่าสนใจ กล้องบันทึกภาพต้องมีความถูกต้องสูง ต่อคุณสมบัติข้อนี้ กล้องบันทึกภาพอาจต้องได้รับการปรับเทียบกล้อง (calibration) ด้วยเทคนิควิธีที่เหมาะสมก่อนที่จะนำไปบันทึกภาพ

ประการสุดท้าย ในกรณีที่ระบบพลวัตมีการเคลื่อนที่แบบ 2 มิติ สามารถใช้กล้องบันทึกภาพเพียงหนึ่งตัวสำหรับการบันทึกภาพ แต่ในกรณีที่ระบบพลวัตมีการเคลื่อนที่แบบ 3 มิติ จำเป็นต้องใช้กล้องบันทึกภาพอย่างน้อย 2 ตัวสำหรับการบันทึกภาพ สิ่งที่จะต้องคำนึงถึงคือ ตำแหน่งของกล้องบันทึกภาพแต่ละตัว แนวทางปฏิบัติโดยทั่วไปคือ ตำแหน่งของกล้องบันทึกภาพจะต้องตั้งฉากกับระนาบการเคลื่อนที่ของระบบ ในกรณีที่ระบบมีการเคลื่อนที่แบบ 2 มิติ เช่นเคลื่อนที่ในระนาบ x, y ตำแหน่งของกล้องบันทึกภาพควรอยู่ในแนวแกน z เพื่อให้สามารถบันทึกการเคลื่อนที่ของระบบได้อย่างครบถ้วนมากที่สุด ในกรณีที่ระบบมีการเคลื่อนที่แบบ 3 มิติ เช่นเคลื่อนที่ในระนาบ x, y, z และมีการใช้กล้องบันทึกภาพ 2 ตัว ตำแหน่งของกล้องบันทึกภาพตัวที่ 1 ควรอยู่ในแนวแกน z เพื่อให้สามารถบันทึกการเคลื่อนที่ของระบบในระนาบ x, y และตำแหน่งของกล้องบันทึกภาพตัวที่ 2 ควรอยู่ในแนวแกน y เพื่อให้สามารถบันทึกการเคลื่อนที่ของระบบในระนาบ z, x ในกรณีนี้ข้อมูลภาพที่ได้จากกล้องบันทึกภาพทั้ง 2 ตัว จะเป็นข้อมูลแบบ 2 มิติที่มีความสัมพันธ์กัน (มิติ x) จากความสัมพันธ์ดังกล่าว สามารถสร้างเป็นข้อมูลการเคลื่อนที่ของระบบแบบ 3 มิติได้ โดยอาศัยการคำนวณเชิงตัวเลขที่เหมาะสม สิ่งสำคัญอีกประการหนึ่งสำหรับการใช้กล้องบันทึกภาพมากกว่า 1 ตัวคือ คุณสมบัติของกล้องบันทึกภาพ อุปกรณ์รับภาพ และคอมพิวเตอร์ รวมทั้งรายละเอียดการปรับตั้ง (set up) กล้องบันทึกภาพแต่ละตัว จะต้องเหมือนกันทุกประการ

4.4 สรุป

ในบทที่ 4 นี้ ได้กล่าวถึงการประยุกต์เทคนิคการประมวลผลภาพเพื่อสกัดข้อมูลพลวัตของระบบ โดยเริ่มต้นจากความเป็นมาของการประมวลผลภาพ รูปแบบของภาพดิจิทัล เทคนิคการแบ่งส่วนภาพ โดยอาศัยคุณสมบัติความไม่ต่อเนื่อง และความคล้ายคลึงกันของค่าสีในจุดภาพ เทคนิคการกำหนดจุดเปลี่ยนค่า และการประยุกต์การประมวลผลภาพเพื่อการระบุเอกลักษณ์ระบบ รวมทั้งสิ่งที่ควรคำนึงถึงในการใช้กล้องบันทึกภาพการเคลื่อนที่ของระบบทั้งแบบ 2 มิติและ 3 มิติ

บทที่ 5

การค้นหาแบบตาบู่เชิงปรับตัว

5.1 กล่าวนำ

การระบุเอกลักษณ์ระบบไม่เชิงเส้นในงานวิจัยนี้ ประยุกต์ใช้การค้นหาแบบตาบู่เชิงปรับตัว เพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง การค้นหาแบบตาบู่เชิงปรับตัวเป็นเทคนิคการค้นหาทางปัญญาประดิษฐ์ที่ทรงประสิทธิภาพเทคนิคหนึ่ง ซึ่งได้รับการพัฒนามาจากการค้นหาแบบตาบู่ ดังนั้นเนื้อหาที่ปรากฏในบทที่ 5 นี้ จะกล่าวถึงการค้นหาแบบตาบู่เชิงปรับตัว โดยมีเนื้อหาเริ่มต้นจากการค้นหาแบบตาบู่ การพัฒนาการค้นหาแบบตาบู่เชิงปรับตัว อัลกอริทึม การพิสูจน์การลู่เข้าผลเฉลย และการวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตาบู่เชิงปรับตัว รวมทั้งแนวทางการประยุกต์การค้นหาแบบตาบู่เชิงปรับตัวเพื่อการระบุเอกลักษณ์ระบบ

5.2 การค้นหาแบบตาบู่

การค้นหาแบบตาบู่ (Tabu search, TS) (Glover, 1986) เป็นวิธีการค้นหาคำตอบที่เหมาะสมที่สุดด้วยคอมพิวเตอร์ในปัญหาแบบไม่ต่อเนื่องเชิงผสมผสาน การค้นหาแบบตาบู่อาศัยกระบวนการค้นหาค่าใกล้เคียง (neighborhood search) และรายชื่อตาบู่ (Tabu list, TL) ซึ่งจะทำหน้าที่เก็บค่าคำตอบในอดีต คำตอบใน TL สามารถใช้ประโยชน์ในกรณีการค้นหาเกิดการลู่เข้าโดยคำตอบวงแคบเฉพาะถิ่น (local solution) โดยอาศัยเกณฑ์ปรารถนา (aspiration criteria, AC) กระบวนการค้นหาจะเคลื่อนที่แบบดิเทอร์มินิสติก ซึ่งจะถูกกำหนดไว้อย่างมีระเบียบแบบแผน โดยทั่วไปจะถูกกำกับไว้ด้วยเงื่อนไขความคงอยู่ล่าสุด (recency) และเงื่อนไขความซ้ำซาก (frequency) (Glover, 1989; Glover, 1990; Glover and Laguna, 1997)

กล่าวสำหรับเงื่อนไขความคงอยู่ล่าสุด คือเงื่อนไขที่ใช้พิจารณาทิศทางการค้นหาคำตอบในช่วงเวลาที่ผ่านมา เมื่อกลไกการค้นหาพบคำตอบด้วยทิศทางการค้นหานั้น ๆ เงื่อนไขความคงอยู่ล่าสุด จะกำหนดให้ทิศทางดังกล่าวมีสถานะต้องห้าม (Tabu) ในช่วงระยะเวลาหนึ่ง ต่อเมื่อเวลาผ่านไปมากเพียงพอจนมั่นใจได้ว่าการค้นหาจะไม่วนกลับไปพบคำตอบเดิมอีกแล้ว เงื่อนไขความคงอยู่ล่าสุดจะยกเลิกสถานะต้องห้ามของทิศทางการค้นหาดังกล่าว ด้วยเงื่อนไขความคงอยู่ล่าสุดนี้ จะเป็นการประกันว่ากลไกการค้นหาจะไม่เข้าไปทำการค้นหาในทิศทางเดิมหรือพื้นที่เดิมที่ดำเนินการค้นหาผ่านมาแล้ว เงื่อนไขนี้จะช่วยให้การค้นหาสามารถพบคำตอบที่ดีอื่น ๆ แม้ว่าในจำนวนคำตอบเหล่านั้นอาจเป็นคำตอบที่ไม่ดีกว่าคำตอบเดิมที่เคยพบแล้วก็ตาม

สำหรับเงื่อนไขความซ้ำซาก คือเงื่อนไขที่ใช้พิจารณาจำนวนการเคลื่อนที่เข้าหาคำตอบของ กลไกการค้นหาด้วยทิศทางเดิม กล่าวคือเมื่อทิศทางเคลื่อนที่เข้าหาคำตอบใด ๆ ถูกเรียกใช้ ด้วยจำนวนครั้งที่มากเกินไปจนเกินกำหนด เงื่อนไขความซ้ำซากจะกำหนดให้ทิศทางเคลื่อนที่ดังกล่าวนั้นมีสถานะต้องห้ามในช่วงระยะเวลาหนึ่ง เพื่อเป็นการประกันว่ากลไกการค้นหาคำตอบจะไม่ดำเนินการ ค้นหาในลักษณะวนรอบอยู่กับที่ อันจะส่งผลทำให้เกิดการล๊อคของคำคำตอบ

เงื่อนไขความคงอยู่ล่าสุด และเงื่อนไขความซ้ำซาก จะถูกประยุกต์ใช้งานร่วมกันเสมอ ในลักษณะการดำเนินงานด้วยหน้าที่ตรงกันข้ามอย่างสัมพันธ์กัน กล่าวคือ เมื่อเงื่อนไขหนึ่ง ทำหน้าที่กำหนดสถานะต้องห้ามให้กับกลไกการค้นหา อีกเงื่อนไขหนึ่งจะทำหน้าที่ยกเลิกสถานะ ต้องห้ามนั้นอย่างสอดคล้องและสัมพันธ์กัน อีกทั้งยังต้องสัมพันธ์กับ TL ด้วย

การค้นหาแบบตามนัยยังมีกลไกที่คอยทำหน้าที่เพิ่มสมรรถนะการค้นหาคำตอบ อีกสองกลไก ด้วยกัน คือ กลไกความเข้มข้น (intensification) และกลไกความหลากหลาย (diversification) (Glover, 1994; Glover and Laguna, 1997) ซึ่งสรุปได้ดังนี้

กลไกความเข้มข้น จะเป็นการค้นหาคำตอบในกลุ่มคำตอบที่พบแล้วในอดีตที่ได้รับการ บันทึกไว้ใน TL ซึ่งกลุ่มคำตอบดังกล่าวได้รับการพิจารณาแล้วว่าเป็นกลุ่มคำตอบที่ดี กลไกความเข้มข้นจะกำหนดให้กลไกการค้นหาคำตอบดำเนินการค้นหาคำตอบซ้ำในพื้นที่เดิมที่เคยพบกลุ่มคำตอบที่ดี อย่างละเอียดมากยิ่งขึ้น ซึ่งอาจส่งผลทำให้การค้นหาสามารถพบคำตอบที่ดีกว่าเดิมได้

กลไกความหลากหลาย จะเป็นการค้นหาคำตอบในปริภูมิหรือพื้นที่การค้นหาที่ยังไม่เคย ได้รับการสำรวจหรือค้นหามาก่อน กลไกความหลากหลายจะกำหนดให้กลไกการค้นหาคำตอบ ค้นหาคำตอบในพื้นที่อื่นหรือทิศทางสำรวจอื่นที่แตกต่างไปจากเดิม ซึ่งอาจส่งผลทำให้การค้นหาสามารถพบคำตอบที่ดีกว่าเดิมได้เช่นเดียวกัน

ด้วยเพราะขั้นตอนที่ค่อนข้างง่าย การค้นหาแบบตามนัยจึงได้รับความนิยมอย่างแพร่หลาย และ ถูกนำไปประยุกต์ใช้แก้ปัญหาต่าง ๆ มากมาย อาทิเช่น การแก้ปัญหาเกี่ยวกับการเงิน (Glover et al, 1994) กระบวนการผลิตอาหาร (Zhang et al, 2003) การวางแผนการผลิตและจำหน่ายกำลังไฟฟ้า (Mantawy et al, 1998; Nara et al, 2001; Silva et al, 2001; Kulworawanichpong and Sujitjorn, 2002) การระบุเอกลักษณ์ระบบพลวัต (Lin and Miller, 2000; Yao et al, 2001; กองพัน อารีรักษ์, 2544) การจราจร (Cordeau and Laporte, 2003) การทำ course schedule (Hertz, 1992) การแก้ปัญหา flow shop (Nowicki and Smutnicki, 1996) และการทำ graph coloring (Hertz and Werra, 1987) เป็นต้น ยิ่งกว่านั้น การค้นหาแบบตามนัยได้รับการพิสูจน์ว่ามีคุณสมบัติการลู่เข้าหาผลเฉลยอีกด้วย (Hanafi, 2000; Glover and Hanafi, 2001) การค้นหาแบบตามนัยได้รับการพัฒนาให้มีประสิทธิภาพในการค้นหาคำตอบมากยิ่งขึ้น จากคณะวิจัยต่าง ๆ ซึ่งจะขอยกตัวอย่างพอสังเขปดังนี้

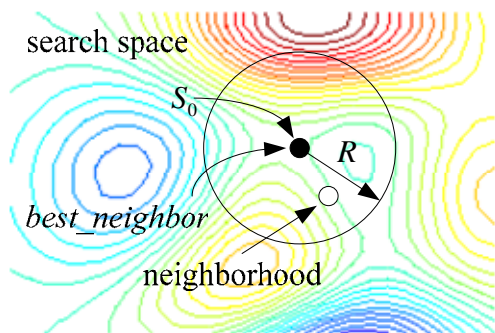
การค้นหาแบบตามูเชิงรีแอกทีฟ หรือการค้นหาแบบตามูชนิดตอบโต้ได้ (Reactive Tabu Search หรือ RTS) (Battiti and Tecchiolli, 1994; Battiti, 1996) คือการค้นหาแบบตามูที่เพิ่มกลไกป้อนกลับ (feedback mechanism) หรือกลไกรีแอกทีฟ (reactive mechanism) กลไกดังกล่าวจะถูกเรียกใช้เมื่อการค้นหาเกิดการล็อกโดยคำตอบวงแคบเฉพาะถิ่น กลไกรีแอกทีฟจะมองย้อนกลับไปในพื้นที่เดิมที่มีการค้นหาผ่านมาแล้ว ซึ่งโดยปกติจะเป็นเขตพื้นที่ต้องห้าม และจะใช้คำตอบเดิมที่ถูกบันทึกไว้ใน TL เพื่อนำมาใช้เป็นคำตอบเริ่มต้นในรอบการค้นหาถัดไป ซึ่งพบว่าการเพิ่มกลไกในลักษณะดังกล่าว เป็นการป้องกันการล็อกโดยคำตอบวงแคบเฉพาะถิ่นได้อย่างมีประสิทธิภาพ นอกจากนี้ RTS ยังมีการปรับโครงสร้างของขนาดและความยาวของ TL ให้สอดคล้องกับปัญหาและสถานการณ์ของการค้นหาที่พบ RTS ได้รับการนำไปประยุกต์ใช้ในงานระบบไฟฟ้ากำลัง (Toune et al, 1998; Fukuyama, 2000) อีกด้วย

การค้นหาแบบตามูเชิงน่าจะเป็น (Probabilistic Tabu Search หรือ PTS) (Kochetov and Goncharov, 2000) คือการค้นหาแบบตามูที่มีกระบวนการค้นหาตั้งอยู่บนกระบวนการสุ่มค่าใกล้เคียงในแต่ละรอบการค้นหาจะถูกผลิตขึ้นมาแบบสุ่มโดยมีค่าความน่าจะเป็นกำกับ ซึ่งพบว่าการค้นหาในลักษณะนี้สามารถพบคำตอบได้อย่างรวดเร็ว ในงานวิจัยของ Kochetov และคณะ ได้มีการพิสูจน์การเข้าสู่หาคำตอบวงกว้างโดยอาศัยระเบียบวิธีลูกโซ่มาร์คอฟ (Markov chains) และได้ประยุกต์ใช้ PTS เพื่อการจัดตำแหน่งสิ่งอำนวยความสะดวกอย่างเหมาะสม (facility location) อีกด้วย

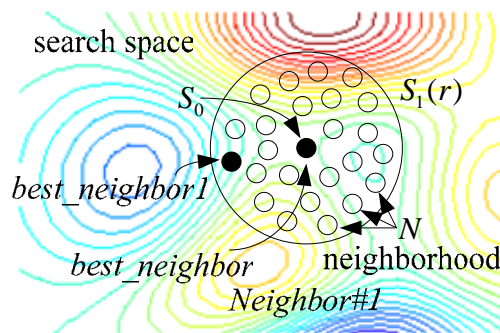
5.3 การค้นหาแบบตามูเชิงปรับตัว

ในปัญหาที่มีความซับซ้อนสูง และปรากฏคำตอบวงแคบเฉพาะถิ่นจำนวนมาก การค้นหาแบบตามูอาจประสบปัญหาเรื่องของการล็อกโดยคำตอบวงแคบเฉพาะถิ่น ทำให้ไม่สามารถเข้าสู่หาคำตอบวงกว้าง (global solution) ได้อย่างแท้จริง (กองพัน อารีรักษ์, 2544) ในงานวิจัยนี้จึงได้พัฒนาการค้นหาแบบตามูเชิงปรับตัว (adaptive Tabu search, ATS) โดยเพิ่มกลไกย้อนรอยการค้นหา (back-tracking mechanism) เพื่อแก้ปัญหาการล็อกโดยคำตอบวงแคบเฉพาะถิ่น และกลไกปรับรัศมีการค้นหา (adaptive radius mechanism) เพื่อเพิ่มความเร็วให้กับกระบวนการค้นหาคำตอบ นอกจากนั้นแล้ว การเคลื่อนที่ของกระบวนการค้นหา หรือการผลิตค่าใกล้เคียงยังได้ถูกปรับให้เป็นแบบสุ่ม (random) เพื่อช่วยเพิ่มโอกาสในการค้นพบคำตอบวงกว้างอีกด้วย อัลกอริทึมของการค้นหาแบบตามูเชิงปรับตัวมีรายละเอียดดังต่อไปนี้

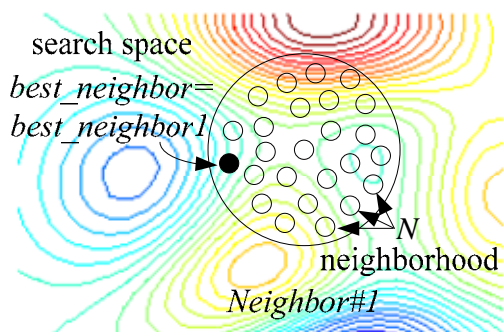
ขั้นตอนที่ 1. กำหนดขอบเขตปริภูมิการค้นหา (search space), รัศมีการค้นหา (R), ตัวนับ (counter), TL, และเกณฑ์ยุติการค้นหา (termination criteria, TC)



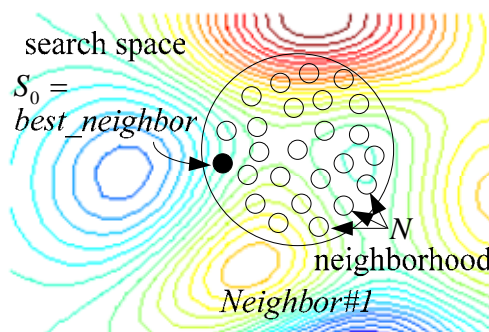
รูปที่ 5.1 สุ่มเลือกคำตอบเริ่มต้น, S_0



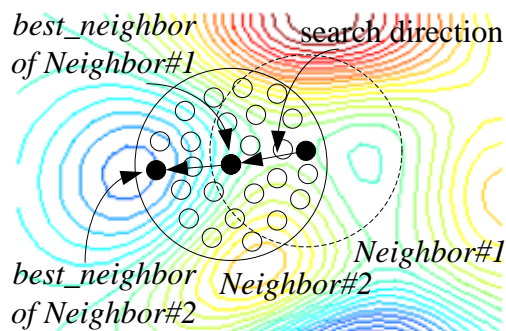
รูปที่ 5.2 ผลิตค่าใกล้เคียงแบบสุ่ม



รูปที่ 5.3 เปรียบเทียบคำตอบเริ่มต้นกับค่าใกล้เคียง



รูปที่ 5.4 ปรับค่าคำตอบเริ่มต้นใหม่



รูปที่ 5.5 ผลิตค่าใกล้เคียงแบบสุ่มรอบใหม่

ขั้นตอนที่ 2. สุ่มเลือกคำตอบเริ่มต้น, S_0 , ภายในปริภูมิการค้นหา โดยกำหนดให้ S_0 มีค่าเป็น $best_neighbor$ ปัจจุบัน ดังแสดงในรูปที่ 5.1

ขั้นตอนที่ 3. ผลิตค่าใกล้เคียง (neighborhood) แบบสุ่มรอบคำตอบเริ่มต้น, S_0 , ภายในปริภูมิการค้นหารัศมี R จำนวน N ตัว แล้วเก็บไว้ในเซต $S_1(r)$ ดังแสดงในรูปที่ 5.2

ขั้นตอนที่ 4. ประเมินค่าใกล้เคียงด้วยฟังก์ชันวัตถุประสงค์ (cost function หรือ objective function) โดยกำหนดให้ S_1 คือค่าใกล้เคียงที่ทำให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด และให้ S_1 มีค่าเป็น $best_neighbor1$ ดังแสดงในรูปที่ 5.2

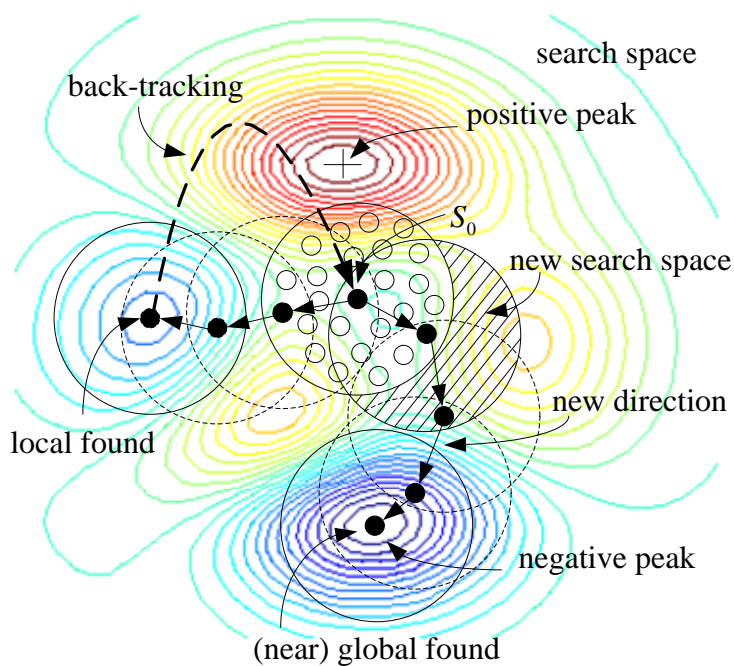
ขั้นตอนที่ 5. ถ้า $S_1 < S_0$ แล้วทำการเก็บ S_0 ไว้ใน TL จากนั้นปรับค่า $S_0 = S_1$ ไม่เช่นนั้นแล้วทำการเก็บ S_1 ไว้ใน TL แทน ดังแสดงในรูปที่ 5.3 ถึง 5.5

ขั้นตอนที่ 6. เรียกใช้กลไกย้อนรอยการค้นหา เมื่อกระบวนการค้นหาเกิดการล็อกโดยคำตอบวงแคบเฉพาะถิ่น ดังแสดงในรูปที่ 5.6

ขั้นตอนที่ 7. ตรวจสอบ TC ถ้า TC สอดคล้อง ให้ยุติการกระบวนการค้นหา ค่า S_0 คือคำตอบที่ดีที่สุด ไม่เช่นนั้นให้ทำขั้นตอนต่อไป

ขั้นตอนที่ 8. เรียกใช้กลไกปรับรัศมีการค้นหา เมื่อกระบวนการค้นหาเข้าใกล้คำตอบดังแสดงในรูปที่ 5.7

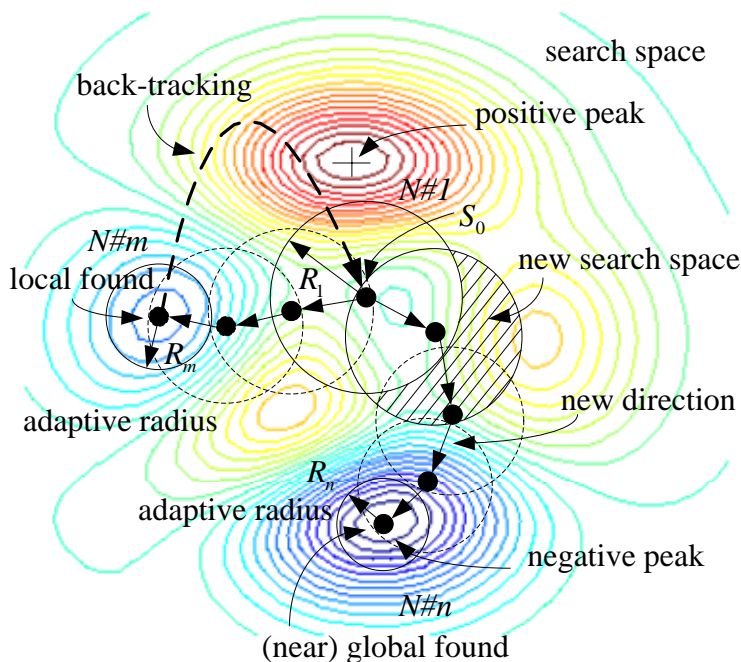
ขั้นตอนที่ 9. ปรับเพิ่มค่าตัวนับ และวนกลับไปยังขั้นตอนที่ 3 เพื่อดำเนินกระบวนการค้นหาในรอบต่อไป



รูปที่ 5.6 กลไกย้อนรอยการค้นหา

กลไกย้อนรอยการค้นหา เป็นกระบวนการเรียกใช้คำตอบเก่าที่บันทึกไว้ใน TL เมื่อพบว่าคำตอบที่ได้จากการค้นหามีค่าซ้ำกันตามจำนวนครั้งที่กำหนด ซึ่งทำให้เกิดความไม่แน่ใจว่าเป็นคำตอบวงกว้าง หรือเป็นการล็อกโดยคำตอบวงแคบเฉพาะถิ่น ดังแสดงในรูปที่ 5.6 เมื่อกระบวนการค้นหาพบคำตอบวงแคบเฉพาะถิ่น และเกิดการล็อกโดยคำตอบวงแคบเฉพาะถิ่นขึ้น กลไกการย้อนรอยค้นหาจะอนุญาตให้ใช้คำตอบที่ได้มีการค้นหาไว้แล้วซึ่งบรรจุอยู่ใน TL เพื่อใช้เป็นคำตอบเริ่มต้นสำหรับการค้นหาในรอบต่อไป ซึ่งจะทำให้เกิดปริภูมิการค้นหาใหม่ เป็นการ

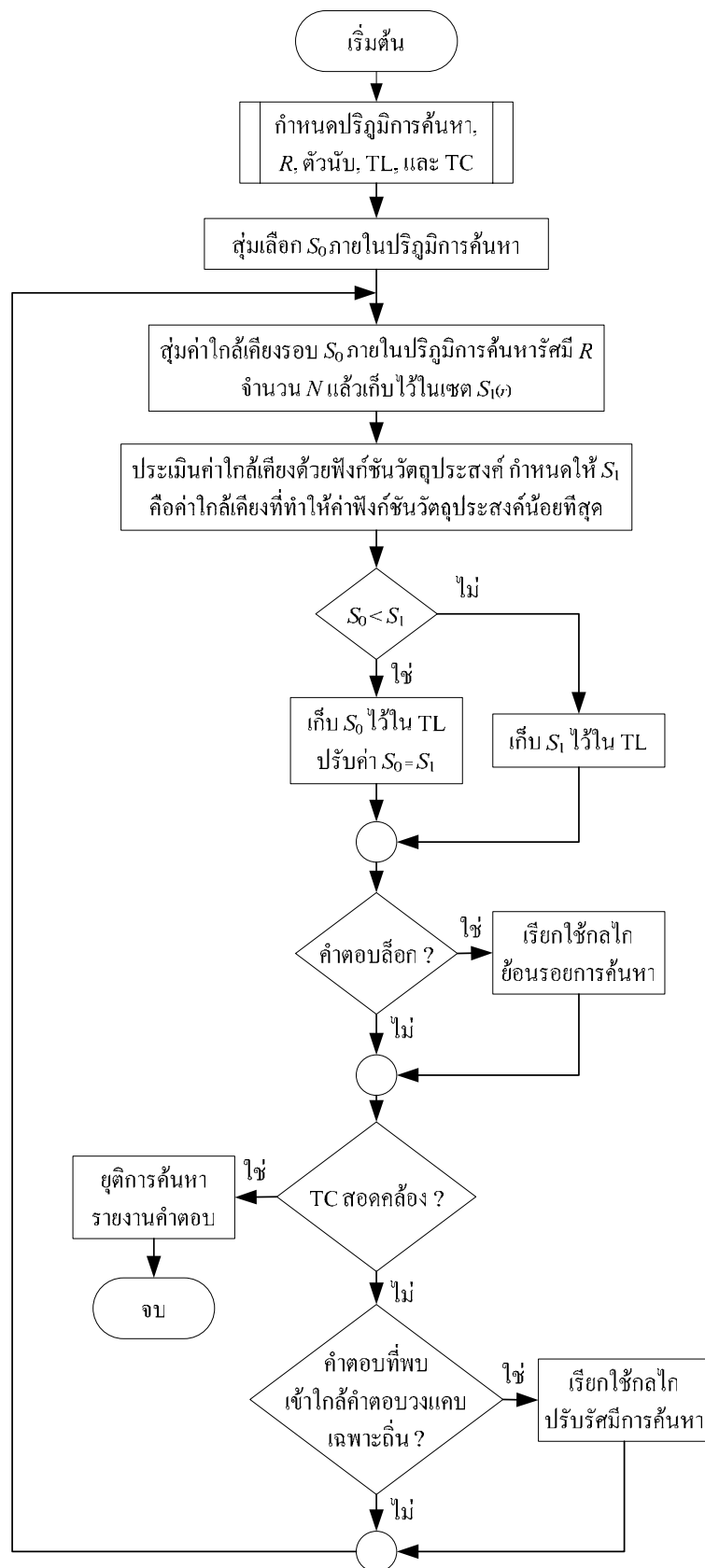
เปิดโอกาสให้กระบวนการค้นหาเปลี่ยนทิศทาง และสามารถหลุดออกจากการล็อกโดยคำตอบวงแคบเฉพาะถิ่นได้ จากนั้นกระบวนการค้นหาจะดำเนินต่อไปจนพบกับคำตอบวงแคบเฉพาะถิ่นอื่น ๆ และในที่สุดก็จะพบกับคำตอบวงกว้าง



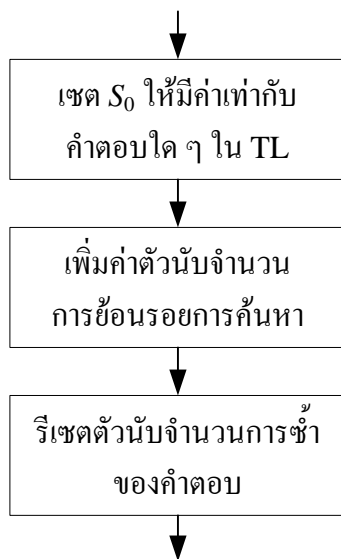
รูปที่ 5.7 กลไกปรับรัศมีการค้นหา

สำหรับกลไกปรับรัศมีการค้นหา ดังแสดงในดังรูปที่ 5.7 เป็นกระบวนการเร่งความเร็วให้กับกระบวนการค้นหา กลไกดังกล่าวจะดำเนินการปรับลดรัศมีการค้นหาลงเมื่อกระบวนการค้นหาพบคำตอบที่ใกล้คำตอบวงแคบเฉพาะถิ่นหรือคำตอบวงกว้าง เมื่อรัศมีการค้นหาที่มีค่าลดลงขณะที่จำนวนค่าใกล้เคียงเท่าเดิม จะทำให้กระบวนการค้นหาสามารถพิจารณาค่าใกล้เคียงได้อย่างละเอียดมากยิ่งขึ้น กล่าวอีกนัยหนึ่งก็คือ เมื่อกระบวนการค้นหาเข้าใกล้คำตอบ รัศมีการค้นหาที่จะยิ่งเล็กลง และค่าใกล้เคียงที่ถูกผลิตขึ้นมาแบบสุ่มก็จะยิ่งมีความละเอียดมากขึ้น ทำให้กระบวนการค้นหาพบคำตอบได้อย่างรวดเร็ว เงื่อนไขการปรับรัศมีการค้นหาอาจกำหนดโดยใช้ค่าฟังก์ชันวัตถุประสงค์ ดังแสดงในสมการที่ (5-1) เมื่อ ϵ คือ ค่าฟังก์ชันวัตถุประสงค์ที่พบ δ_i คือ ค่าฟังก์ชันวัตถุประสงค์กำหนด DF_i คือ ตัวประกอบการลดรัศมีการค้นหา (decreasing factor) และ $i = 1, \dots, n$ คือ จำนวนเงื่อนไข ค่า δ , DF , และ i ขึ้นอยู่กับปัญหาที่สนใจ ในแต่ละกรณีปัญหานั้น ไม่จำเป็นจะต้องมีค่าเท่ากัน

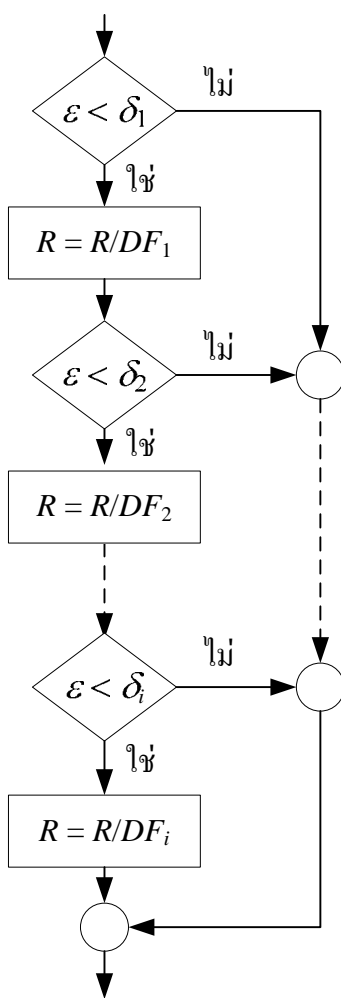
$$\text{if } [\epsilon \leq \delta_i], \text{ then } [R = \frac{R}{DF_i}] \tag{5-1}$$



รูปที่ 5.8 แผนภูมิการทำงานของการค้นหาแบบตามูเชิงปรับตัว



รูปที่ 5.9 แผนภูมิการทำงานของกลไกย้อนรอยการค้นหา



รูปที่ 5.10 แผนภูมิการทำงานของกลไกปรับรัศมีการค้นหา

การพัฒนาโปรแกรมการค้นหาแบบตามูเชิงปรับตัว จะอาศัยแผนภูมิการดำเนินงานของ อัลกอริทึม ดังแสดงในรูปที่ 5.8 โดยที่รายละเอียดของกลไกย้อนรอยการค้นหาคำตอบ แสดงด้วย แผนภูมิในรูปที่ 5.9 และรายละเอียดของกลไกปรับรัศมีการค้นหาแสดง ด้วยแผนภูมิในรูปที่ 5.10 ตามลำดับ โปรแกรมการค้นหาแบบตามูเชิงปรับตัว ได้รับการพัฒนาขึ้นด้วย MATLABTM และ C ดังรายละเอียดที่ปรากฏในภาคผนวก ก.

5.4 การพิสูจน์การลู่เข้าหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว

การพิสูจน์การลู่เข้าหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว อาศัยตรรกะทาง คณิตศาสตร์บริสุทธิ์ร่วมกับตรรกะเชิงศึกษาสำนึก (heuristic logic) เพื่อพิสูจน์ว่าการค้นหา แบบตามูเชิงปรับตัว มีคุณสมบัติการลู่เข้าหาคำตอบวงกว้าง ภายใต้ नियามและทฤษฎีบท ดังรายละเอียดต่อไปนี้

นิยาม 1 กำหนดให้ Ω เป็นปริภูมิการค้นหาที่มีขอบเขตจำกัด (finite search space) ซึ่งมี สมาชิก x ทั้งหมดจำนวน n ตัว, $x_i, i=1, \dots, n$, โดยที่ $n < \infty$

นิยาม 2 กำหนดให้ภายใน Ω มีคำตอบวงแคบเฉพาะถิ่นจำกัด จำนวน k ตัว และ Ω ถูก แบ่งออกเป็นปริภูมิคอนเวกซ์จำนวน k ปริภูมิ ซึ่งแทนด้วย $\Lambda_i, i=1, \dots, k$ ในแต่ละปริภูมิ คอนเวกซ์, Λ , จะมีสมาชิกทั้งหมดจำนวน w ตัว ($w < n$ หรือ $n = \sum_{i=1}^k w_i$) แต่จะมีสมาชิกที่เป็น คำตอบวงแคบเฉพาะถิ่นเพียงตัวเดียว และสมาชิกดังกล่าวจะต้องไม่อยู่บนขอบของปริภูมิ ทั้งนี้ เพื่อให้สอดคล้องกับฟังก์ชันคอนเวกซ์ (convex function) (Webster, 1994; Kreyszig, 1978; Chong and Zak, 2001; วัชรพงษ์ โขวิฑูรกิจ, 2546)

จากนิยาม 2 จะพบว่าแต่ละปริภูมิคอนเวกซ์จะไม่มี การเกยทับกัน และขอบของแต่ละปริภูมิ คอนเวกซ์จะแนบชิดกัน ดังนั้นจึงทำให้สามารถกำหนดคุณสมบัติของปริภูมิคอนเวกซ์ใน Ω ได้ สองประการดังนี้คือ $\bigcup_{i=1}^k \Lambda_i = \Omega$ และ $\bigcap_{i=1}^k \Lambda_i = \emptyset$ เมื่อ \emptyset คือเซตว่าง

นิยาม 3 กำหนดให้ Ψ เป็นปริภูมิการค้นหาย่อย (sub search space) ใน Ω , $\Psi \subset \Omega$, ในแต่ละ Ψ จะมีสมาชิก x จำกัดจำนวน m ตัว, $x_i, i=1, \dots, m$, โดยที่ $m < w < n$ และ m เป็น ค่าคงที่ (จำนวนเต็มบวกจำกัด)

กลไกการก่อรูป Ψ ถูกกำหนดดังนี้คือ ในกระบวนการค้นหาค่าเหมาะที่สุด คำตอบปัจจุบัน ถูกกำหนดโดย x_0 ปริภูมิการค้นหาย่อย, Ψ , จะก่อรูปในลักษณะสุ่มอย่างสม่ำเสมอ (uniform-random) รอบๆ x_0 ภายในวงกลม $\|\tilde{x} - x_0\| \leq \rho$ เมื่อ \tilde{x} คือคำตอบเชิงสุ่มที่ผลิตขึ้นภายในวงกลม รัศมี ρ

นิยาม 4 กำหนดให้ลำดับจำกัด $S = \{x_{0,i}\}, i=1, \dots, p$, เป็นการสะสมของคำตอบ (collection of solutions) x_0 ซึ่งจะมีคำตอบจำนวน p คำตอบในการเข้าสู่คำตอบวงกว้าง ($k \leq p$)

จากนิยาม 4 ที่ว่าด้วยลำดับจำกัด S , การสะสมของปริภูมิการค้นหาย่อย (collection of sub search space), $\theta = \{\Psi_i\}, i=1, \dots, p$, จะก่อรูปขึ้นตามมา ซึ่งในการค้นหาคำตอบวงกว้าง ปริภูมิการค้นหาไม่จำเป็นที่จะต้องถูกสำรวจจนครบทั้งหมด นั่นคือ $\bigcup_{i=1}^p \Psi_i \subset \Omega$ การก่อรูปของปริภูมิการค้นหาย่อยถัดไป Ψ_{i+1} ตามกลไกของการค้นหาแบบตามเชิงปรับตัว จะอาศัยค่าคำตอบที่ดีที่สุดที่ถูพบในปริภูมิการค้นหาย่อยก่อนหน้า Ψ_i ในฐานะจุดศูนย์กลาง $x_{0,i+1}$ ด้วยกระบวนการสุ่ม ทำให้ $\Psi_i \cap \Psi_{i+1} \neq \emptyset$ ทั้งนี้เพราะว่าอย่างน้อยที่สุดจุดศูนย์กลาง $x_{0,i+1}$ จะวางตัวอยู่ที่ทั้งใน Ψ_i และ Ψ_{i+1}

นิยาม 5 กำหนดให้ $Time(x)$ เป็นเวลาที่ใช้ในการสำรวจคำตอบ x ใด ๆ ในปริภูมิการค้นหา Ω ค่า $Time(x)$ ถูกกำหนดให้มีความคงและเท่ากันทุกครั้งที่ในการสำรวจแต่ละคำตอบ $x \in \Omega$ นั่นคือ $Time(x_i) = Time(x) > 0, i=1, \dots, n$

จากนิยาม 5 การสำรวจคำตอบทุกคำตอบภายในปริภูมิการค้นหาย่อย Ψ ซึ่งถูกผลิตขึ้นแบบสุ่มจำนวน m คำตอบ จะใช้เวลา $m \cdot Time(x)$ ดังนั้นเวลารวมสำหรับการสำรวจคำตอบทั้งหมดในทุก ๆ ปริภูมิการค้นหาย่อย จะมีค่าเป็น $p \cdot m \cdot Time(x)$ และจากการค้นหาที่นำเสนอ เวลารวมที่ใช้สำรวจคำตอบทั้งหมดในทุก ๆ ปริภูมิการค้นหาย่อย จะต้องมีความน้อยกว่าเวลารวมที่ใช้สำรวจคำตอบทั้งหมดในปริภูมิการค้นหา เพื่อแสดงให้เห็นถึงการลู่เข้าที่รวดเร็วและคงทน (fast and robust convergence) นั่นคือ $n \cdot Time(x) > p \cdot m \cdot Time(x) > m \cdot Time(x) > Time(x) > 0$

นิยาม 6 กำหนดให้ $Iteration$ เป็นจำนวนรอบการค้นหาหรือการสำรวจคำตอบในแต่ละปริภูมิการค้นหาย่อย เป็นค่าที่ใช้แสดงจำนวนคำตอบในปริภูมิการค้นหาย่อย Ψ ใด ๆ ที่ได้รับการสำรวจเรียบร้อยแล้ว ค่า $Iteration$ จะถูกปรับให้มีค่าเป็นศูนย์ (reset) ก่อนเริ่มดำเนินการสำรวจ เมื่อคำตอบใด ๆ ในปริภูมิการค้นหาย่อยได้รับการสำรวจ $Iteration = Iteration + 1$ และหลังจากทุกคำตอบในปริภูมิการค้นหาย่อยได้รับการสำรวจจนครบ ค่า $Iteration = m$ และค่าเวลารวมที่ใช้สำรวจคือ $m \cdot Time(x)$

นิยาม 7 กำหนดให้ $Count$ เป็นจำนวนรอบการค้นหาของการสำรวจปริภูมิการค้นหาย่อย เป็นค่าที่ใช้แสดงจำนวนปริภูมิการค้นหาย่อยใน Ω ที่ได้รับการสำรวจเรียบร้อยแล้ว ค่า $Count$ จะถูกปรับให้มีค่าเป็นศูนย์ (reset) ก่อนเริ่มดำเนินการสำรวจ เมื่อทุกคำตอบในปริภูมิการค้นหาย่อยใด ๆ ได้รับการสำรวจจนครบ $Count = Count + 1$ และหลังจากทุกปริภูมิการค้นหาย่อยใน θ ถูกสำรวจจนครบ ค่า $Count = p$ และค่าเวลารวมที่ใช้สำรวจคือ $p \cdot m \cdot Time(x)$

นิยาม 8 กำหนดให้ BT คือกลไกย้อนรอยการค้นหาที่อนุญาตให้มีการเรียกใช้คำตอบวงแคบเฉพาะถิ่นใด ๆ ที่บรรจุอยู่ใน TL เพื่อใช้เป็นคำตอบเริ่มต้นใหม่แทนคำตอบที่เพิ่งค้นพบจากกระบวนการค้นหา กลไกนี้ถูกพัฒนาขึ้นเพื่อปรับปรุงความสามารถในการหลุดออกจากการล็อกโดยคำตอบวงแคบเฉพาะถิ่น

นิยาม 9 กำหนดให้ AR คือกลไกปรับรัศมีการค้นหา เพื่อให้ปริภูมิการค้นหาหยุดถัดไปมีความละเอียดมากยิ่งขึ้น ส่งผลให้กระบวนการค้นหาเข้าถึงคำตอบวงแคบเฉพาะถิ่นได้ด้วยเวลาที่ลดลง โดยกำหนดให้ $\rho = \mu \cdot r$ คือรัศมีที่ปรับตัว ค่า r คือรัศมีกลางที่กำหนดแบบไม่เจาะจงหรือรัศมีที่ระบุแบบไม่เจาะจง (arbitrary nominal radius) ขณะที่ $0 < \mu \leq 1$ รัศมีนี้จะถูกใช้เพื่อกำหนดค่าใกล้เคียงรอบคำตอบปัจจุบัน

การพิสูจน์การเข้าสู่ผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว เป็นการพิสูจน์ที่ตั้งอยู่บนมูลฐานสามประการดังนี้

1. ให้ประกันคำตอบวงแคบเฉพาะถิ่น เป็นการพิสูจน์เพื่อแสดงให้เห็นว่าการค้นหาแบบตามูเชิงปรับตัว มีคุณสมบัติเข้าสู่คำตอบวงแคบเฉพาะถิ่นในแต่ละปริภูมิคอนเวกซ์ Λ การพิสูจน์นี้แสดงไว้ในทฤษฎีบท ก.

2. ให้ประกันความเร็วของกระบวนการค้นหา เป็นการพิสูจน์เพื่อแสดงให้เห็นว่าการค้นหาแบบตามูเชิงปรับตัวที่มีกลไก AR สามารถเข้าสู่คำตอบวงแคบเฉพาะถิ่นด้วยเวลาที่เร็วขึ้น การพิสูจน์นี้แสดงไว้ในทฤษฎีบท ข.

3. ให้ประกันคำตอบวงกว้าง เป็นการพิสูจน์เพื่อแสดงให้เห็นว่าการค้นหาแบบตามูเชิงปรับตัวที่มีกลไก BT สามารถพบคำตอบวงแคบเฉพาะถิ่นอื่น ๆ ที่เหลือในปริภูมิการค้นหา Ω และเมื่อกระบวนการค้นหายุติลง หนึ่งในคำตอบวงแคบเฉพาะถิ่นที่พบคือคำตอบ (ใกล้) วงกว้าง (near global solution) การพิสูจน์นี้แสดงไว้ในทฤษฎีบท ค.

อัลกอริทึมของการค้นหาแบบตามูเชิงปรับตัว ได้รับการทบทวนไว้เพื่อประกอบการพิจารณาความเข้าใจในการพิสูจน์การเข้าสู่ผลเฉลย ดังต่อไปนี้

ขั้นตอนที่ 1. กำหนดขอบเขตปริภูมิการค้นหา Ω , รัศมีการค้นหา r , TL เกณฑ์ยุติการค้นหา (termination criteria, TC), $Count = 0$, และ $Iteration = 0$

ขั้นตอนที่ 2. สุ่มเลือกคำตอบเริ่มต้น $x_{0,Count}$, ภายใน Ω โดยกำหนดให้มีค่าเป็นคำตอบวงกว้างเริ่มต้น x^* เวลาที่ใช้ในการสำรวจคือ $Time(x)$

ขั้นตอนที่ 3. ปรับค่า $Count = Count + 1$ แล้วก่อรูปปริภูมิการค้นหาย่อย $\Psi \subset \Omega$

ขั้นตอนที่ 4. ประเมินค่าฟังก์ชันวัตถุประสงค์ของ $\forall x \in \Psi_{Count}$ เมื่อค่า x ใด ๆ ถูกสำรวจหรือประเมิน ปรับค่า $Iteration = Iteration + 1$ และเมื่อทุกคำตอบใน Ψ_{Count} ถูกสำรวจจนครบ

$Iteration = m$ เวลารวมที่ใช้ในการสำรวจคือ $m \cdot Time(x)$ จากนั้นกำหนดให้ค่าคำตอบที่ให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุดเป็น x'

ขั้นตอนที่ 5. ถ้า $x' < x_{0,Count}$ แล้วทำการเก็บ $x_{0,Count}$ ไว้ใน TL จากนั้นปรับค่า $x_{0,Count} = x'$ ไม่เช่นนั้นแล้วทำการเก็บ x' ไว้ใน TL แทน จากนั้นปรับค่า $x^* = x_{0,Count}$ ถ้า $x_{0,Count} < x^*$

ขั้นตอนที่ 6. เรียกใช้กลไก BT เมื่อกระบวนการค้นหาเกิดการล็อกโดยคำตอบวงแคบเฉพาะถิ่น

ขั้นตอนที่ 7. ตรวจสอบ TC ถ้า TC สอดคล้อง ให้ยุติการกระบวนการค้นหา ค่า x^* คือคำตอบที่ดีที่สุด ค่าเวลารวมที่ใช้คือ $m \cdot Count \cdot Time(x)$ และค่า $Iteration = m \cdot Count$ (การค้นหาแบบตามเชิงปรับตัว จะดูเข้าหาคำตอบวงกว้างก็ต่อเมื่อ ปริภูมิการค้นหามีขอบเขตจำกัด มิฉะนั้นแล้วคำตอบที่พบจะเป็นเพียงคำตอบใกล้เคียงกว้าง) ถ้า TC ไม่สอดคล้องให้ทำขั้นตอนต่อไป

ขั้นตอนที่ 8. เรียกใช้กลไก AR เมื่อกระบวนการค้นหาเข้าใกล้คำตอบวงแคบเฉพาะถิ่น

ขั้นตอนที่ 9. ปรับค่า $Iteration = 0$ แล้ววนกลับไปยังขั้นตอนที่ 3 เพื่อดำเนินกระบวนการค้นหาต่อไป

ทฤษฎีบท ก. ถ้าสมาชิกทั้งหมด m ตัวในปริภูมิการค้นหาย่อย Ψ มีจำนวนมากพอที่จะให้ค่าใกล้เคียงที่ดี คำตอบวงแคบเฉพาะถิ่นบริเวณนี้จะถูกพบได้โดยการผลิตลำดับของปริภูมีย่อย

พิสูจน์ ปริภูมิการค้นหาย่อย, Ψ , ก่อรูปขึ้นรอบคำตอบเริ่มต้น, x_0 , ในลักษณะสุ่มอย่างสม่ำเสมอภายในรัศมีที่แน่นอน, ρ , เซตของคำตอบ x ทุกคำตอบที่อยู่ภายใน $\|x - x_0\| \leq \rho$ ถูกกำหนดโดย $N_\rho(x_0)$ ซึ่งเรียกว่าค่าใกล้เคียง (neighborhood) ของ x_0 นั่นก็คือ $\Psi \subset N_\rho(x_0)$ และ m จะต้องมีค่าน้อยกว่าจำนวนสมาชิกทั้งหมดใน $N_\rho(x_0)$

ให้ \hat{x} เป็นคำตอบวงแคบเฉพาะถิ่นภายในปริภูมิคอนเวกซ์, $\Lambda(x_0)$, ของ x_0 นั่นคือ $f(\hat{x}) < f(x)$ สำหรับ $\forall x \in \Lambda(x_0)$ และ $\forall x \in N_\rho(x_0)$ บอกเป็นนัยว่าทั้งใน $N_\rho(x_0)$ และในเซตของคำตอบในบริเวณนี้ วางตัวอยู่บนปริภูมิคอนเวกซ์, $\Lambda(x_0)$, เดียวกัน

เช่นเดียวกับแนวคิดของการค้นหาแบบวงแคบเฉพาะถิ่น (local search) หรืออัลกอริทึมแบบ hill-climbing การปรับค่าคำตอบปัจจุบันจะนำไปสู่คำตอบวงแคบเฉพาะถิ่นในบริเวณนี้ เมื่อปริภูมิการค้นหาย่อย Ψ_i ใด ๆ ก่อรูปขึ้น และคำตอบที่ดีที่สุดถูกค้นพบ นั่นคือ $f(x'_i) \leq f(x'_j)$ สำหรับ $i > j$ ที่รอบการค้นหา i^{th} ใด ๆ ระยะห่างระหว่าง x'_i กับคำตอบวงแคบเฉพาะถิ่น, \hat{x} , ถูกกำหนดด้วยค่า $\xi_i = \|x'_i - \hat{x}\|$ ซึ่งเป็นค่าบวก และค่าความคลาดเคลื่อนของฟังก์ชันวัตถุประสงค์ถูกกำหนดด้วย $g_i = \|f(x'_i) - f(\hat{x})\|$ ถ้ารอบการค้นหา i^{th} มีค่ามากเพียงพอโดยกำหนดให้เป็น M ค่าความคลาดเคลื่อน และค่าคำตอบที่พบจะถูกล้อมกรอบ นั่นคือ $g_i = \|f(x'_i) - f(\hat{x})\| < \varepsilon$

สำหรับ $\xi_i = \|x'_i - \hat{x}\| < \delta$ เมื่อ ε และ δ คือจำนวนบวกน้อย ๆ ของค่าความคลาดเคลื่อนมากที่สุดที่ยอมให้เกิดของ $f(x'_i)$ และ x'_i ตามลำดับ ดังนั้นข้อสรุปของการพิสูจน์ในทฤษฎีบทนี้คือ

- i. กำหนดคุณสมบัติการสืบทอดของอัลกอริทึมเป็น $f(x'_i) \leq f(x'_j)$ สำหรับ $i > j > 0$
- ii. นั่นคือ $0 \leq \|f(x'_{i+1}) - f(\hat{x})\| \leq \|f(x'_i) - f(\hat{x})\|$ สำหรับ $\forall i > 0$
- iii. อัลกอริทึมของการค้นหาแบบตามูเชิงปรับตัว มีคุณสมบัติการลู่เข้าหาคำตอบเมื่อเงื่อนไขต่อไปนี้อยู่ $\lim_{i \rightarrow M} \xi_i < \delta \rightarrow \lim_{i \rightarrow M} g_i < \varepsilon$ ที่ซึ่ง $1 \ll M < n < \infty$

iv. ถ้ามี M อย่างน้อยหนึ่งค่าที่ทำให้เงื่อนไข iii สอดคล้อง สามารถกล่าวได้ว่ากระบวนการค้นหาแบบตามูเชิงปรับตัว มีคุณสมบัติการลู่เข้าหาคำตอบวงแคบเฉพาะถิ่นแบบจำกัด (finite convergence) ด้วยเวลาที่ใช้ในการค้นหาเท่ากับ $M \cdot \text{Time}(x)$

ในทฤษฎีบท ก. สิ่งสำคัญที่ช่วยให้การพิสูจน์ทำได้โดยสะดวกคือ การกำหนดปริภูมิคอนเวกซ์ ซึ่งเป็นการประกันว่าปริภูมิการค้นหาที่ถูกแบ่ง จะมีคำตอบวงแคบเฉพาะถิ่นเพียงคำตอบเดียว แต่สำหรับปริภูมิไม่คอนเวกซ์ (non-convex region) ซึ่งมีคำตอบวงแคบเฉพาะถิ่นหลายคำตอบ การพิสูจน์ในทฤษฎีบทนี้จะยังคงคลุมเครือ กล่าวคือเมื่อคำตอบวงแคบเฉพาะถิ่นคำตอบแรกในปริภูมิถูกค้นพบ กระบวนการค้นหาจะออกจากปริภูมิดังกล่าว แม้ว่าจะยังมีคำตอบวงแคบเฉพาะถิ่นอื่นที่ยังไม่ถูกค้นพบเหลืออยู่ในปริภูมิก็ตาม กระบวนการค้นหาจะกระโดดไปยังปริภูมิข้างเคียงเพื่อดำเนินการค้นหาต่อไป สิ่งนี้อาจนำไปสู่การสูญเสียข้อมูลที่สำคัญในการค้นหาคำตอบวงแคบเฉพาะถิ่นของกระบวนการ กระนั้นก็ตาม การพิสูจน์ในทฤษฎีบทนี้ครอบคลุมกรณีของปริภูมิไม่คอนเวกซ์บางลักษณะ กล่าวโดยสรุปคือ การพิสูจน์จะเป็นจริงก็ต่อเมื่อปริภูมิไม่คอนเวกซ์ที่ถูกแบ่งส่วนมีคำตอบวงแคบเฉพาะถิ่นเพียงคำตอบเดียว

เพื่อให้การพิสูจน์ทฤษฎีบท ก. มีความเข้มแข็งมากยิ่งขึ้น การพิสูจน์การลู่เข้าหาคำตอบวงแคบเฉพาะถิ่นของการค้นหาแบบตามูเชิงปรับตัว จะใช้ระเบียบวิธีความน่าจะเป็น (probabilistic approach) เข้าช่วยดังนี้

กำหนดให้คำตอบเริ่มต้นเป็น $x_{t=0}$ ภายในปริภูมิการค้นหาย่อย $\Psi_t \subset \Omega$ เพื่อผลิตลำดับของคำตอบ x_{t+1} คุณสมบัติการสืบทอดจะได้รับการพิจารณาเพื่อเป็นการประกันว่ากระบวนการค้นหาจะลู่เข้าสู่คำตอบวงแคบเฉพาะถิ่น คำตอบปัจจุบันใด ๆ จะมีความเป็นไปได้สองลักษณะด้วยกันกล่าวคือ กระบวนการค้นหาสามารถพบคำตอบที่ดีกว่า นั่นคือ $f(x_{t+1}) < f(x_t)$ หรือกระบวนการค้นหาไม่สามารถพบคำตอบที่ดีกว่า นั่นคือ $f(x_{t+1}) \geq f(x_t)$ สำหรับการค้นหาแบบตามูเชิงปรับตัว ค่าใกล้เคียง, $N_\rho(x_t)$, ของคำตอบปัจจุบัน, x_t , จะก่อรูปขึ้น และมีสมาชิกรวมทั้งหมดจำนวน N ตัว ขณะเดียวกันปริภูมิการค้นหาย่อย, $\Psi_{t+1} \subset N_\rho(x_t)$, จะก่อรูปขึ้นในลักษณะสุ่ม และมีสมาชิกรวมทั้งหมด m ตัว โดยที่ $m < N$ และ m เป็นค่าคงที่ กระบวนการดังกล่าวตั้งอยู่บนพื้นฐานที่ว่า ไม่ใช่สมาชิกทุกตัวในค่าใกล้เคียงที่จะให้ค่าฟังก์ชันวัตถุประสงค์ที่

ดีกว่าค่าคำตอบ x_t แต่มีสมาชิกเพียง u ตัวใน $N_p(x_t)$ ที่ทำให้ $f(x) < f(x_t)$ สอดคล้อง โดยที่ $x \in \Psi_{t+1}$ ค่าความน่าจะเป็นที่กระบวนการค้นหาจะพบคำตอบที่ดีกว่า $P = P(f(x) < f(x_t))$ มีรายละเอียดดังต่อไปนี้

- กรณีที่ 1 : ($m > N - u$)

$P = 1$ ในกรณีนี้จะมีสมาชิกอย่างน้อยหนึ่งตัวในจำนวน m ที่ทำให้เงื่อนไขสอดคล้อง

- กรณีที่ 2 : ($m \leq N - u$)

ในกรณีนี้จะมีหนทางการจัดหมู่ที่เป็นไปได้เพื่อทำการสุ่มเลือกค่าสมาชิกจำนวน m ตัวจากทั้งหมด N ตัว เป็นจำนวน $\binom{N}{m} = \frac{N!}{(N-m)!m!}$ หนทาง และจะมีหนทางทั้งหมดที่เป็นไปได้ที่กระบวนการค้นหาจะไม่พบคำตอบที่ดีกว่าเป็น $\binom{N-u}{m} = \frac{(N-u)!}{(N-u-m)!m!}$ หนทาง ดังนั้นค่าความน่าจะเป็นที่กระบวนการค้นหาจะไม่พบคำตอบที่ดีกว่าแสดงในสมการที่ (5-2)

$$P = \frac{(N-u)!(N-m)!}{N!(N-u-m)!} \quad (5-2)$$

เมื่อ m และ N ต่างเป็นค่าคงที่ ดังนั้นสมการที่ (5.2) จะขึ้นกับค่า u เพียงค่าเดียว u จะมีค่ามากเมื่อคำตอบปัจจุบันเข้าใกล้คำตอบวงแคบเฉพาะถิ่น กระบวนการค้นหาจะปรับค่าคำตอบปัจจุบันด้วยคำตอบที่ดีที่สุดในแต่ละรอบของการค้นหา ดังนั้นคำตอบจะเคลื่อนเข้าสู่คำตอบวงแคบเฉพาะถิ่นเมื่อเวลาเพิ่มขึ้น นั่นคือการ $\lim_{t \rightarrow \infty} u(t) = 0$ จากสมการที่ (5-2) ค่าความน่าจะเป็นที่กระบวนการค้นหาจะไม่พบคำตอบที่ดีกว่าอีกต่อไป ซึ่งนั่นก็คือคำตอบวงแคบเฉพาะถิ่นถูกค้นพบ จะมีค่าเป็นดังสมการที่ (5-3)

$$\lim_{t \rightarrow \infty} P(t) = \lim_{t \rightarrow \infty} \frac{(N-u(t))!(N-m)!}{N!(N-u(t)-m)!} = 1 \quad (5-3)$$

ดังแสดงในสมการที่ (5-3) เมื่อกระบวนการค้นหาดำเนินไปด้วยระยะเวลาหนึ่งที่ยาวนานพออย่างสมเหตุผล ค่าความน่าจะเป็นที่กระบวนการค้นหาจะพบคำตอบวงแคบเฉพาะถิ่น จะมีค่าเข้าใกล้หนึ่ง

ทฤษฎีบท ข. กลไก AR สามารถช่วยให้กระบวนการค้นหามีความเร็วในการค้นหาคำตอบวงแคบเฉพาะถิ่นในปริภูมิคอนเวกซ์เพิ่มขึ้น ด้วยการปรับลดรัศมีการค้นหา ρ อย่างเหมาะสม

พิสูจน์ กลไก AR ถูกใช้ในการค้นหาแบบตาบอดเชิงปรับตัว เพื่อเร่งความเร็วของกระบวนการค้นหา กลไกดังกล่าวจะช่วยให้กระบวนการค้นหาเข้าสู่ค่าตอบวงแคบเฉพาะถิ่นได้เร็วกว่าการค้นหาแบบตาบอด

กำหนดให้ปริภูมิการค้นหาย่อย, Ψ , มีรัศมีการค้นหาเป็น $\rho = \mu \cdot r$ ขณะเริ่มต้นกระบวนการค้นหา ค่ารัศมีการค้นหาเริ่มต้นถูกกำหนดให้มีค่าเป็น $\rho = r, (\mu = 1.0)$ เมื่อ r คือรัศมีปกติ (normal radius) และเป็นค่าไม่เจาะจง จากทฤษฎีบท ก. เมื่อค่าตอบที่ได้รับการปรับค่าเคลื่อนที่เข้าใกล้ \hat{x} มาก ๆ นั่นคือ $\xi < \rho$ หรืออาจกล่าวได้ว่า $\hat{x} \in N_\rho(x'_i)$ ค่าความน่าจะเป็นที่จะพบค่าตอบวงแคบเฉพาะถิ่น, \hat{x} , มีค่าเป็น $\frac{m}{N}$ เมื่อ N คือจำนวนสมาชิกทั้งหมดของ $N_\rho(x'_i)$ นั่นคือเมื่อรัศมีการค้นหาที่ยังมีค่ามาก (N มาก) ค่าความน่าจะเป็นในการพบค่าตอบวงแคบเฉพาะถิ่นก็จะยิ่งน้อย ด้วยตรรกะนี้ รัศมีการค้นหาจะถูกปรับลด ($0 < \mu \leq 1$) เพื่อให้กระบวนการค้นหาสามารถเข้าสู่ค่าตอบวงแคบเฉพาะถิ่น, \hat{x} , ด้วยเวลาที่รวดเร็วยิ่งขึ้น และหลังจากที่กระบวนการค้นหาพบ \hat{x} แล้ว รัศมีการค้นหาจะต้องถูกปรับค่าให้เท่ากับค่ารัศมีการค้นหาเริ่มต้น และดำเนินกระบวนการค้นหาต่อไป ดังนั้นข้อสรุปของการพิสูจน์ในทฤษฎีบทนี้คือ

i. กำหนดให้ $\rho = \mu \cdot r, 0 < \mu \leq 1 \wedge r \in \mathcal{R}^+$, เมื่อ \mathcal{R}^+ คือปริภูมิของจำนวนจริงบวก

ii. เมื่อ $\xi < \rho$ นั่นคือ $\hat{x} \in N_\rho(x'_i) \wedge \hat{x} \notin \Psi(x'_i) \rightarrow P_\rho(\hat{x}) = \frac{m}{N}$

iii. กำหนดให้ $\xi < \rho' < \rho \wedge (N > N' \in \mathcal{I}^+) \rightarrow \left(P_{\rho'}(\hat{x}) = \frac{m}{N'} \right) > \left(P_\rho(\hat{x}) = \frac{m}{N} \right)$

เมื่อ ρ' คือ รัศมีการค้นหาที่ถูกปรับลดลง N คือ จำนวนสมาชิกทั้งหมดของ $N_\rho(x'_i)$ และ N' คือ จำนวนสมาชิกทั้งหมดของ $N_{\rho'}(x'_i)$

iv. จากการปรับลดรัศมีการค้นหาใน iii กระบวนการค้นหาสามารถเข้าสู่ด้วยเวลาที่รวดเร็วยิ่งขึ้น เนื่องจากค่าความน่าจะเป็นของการพบค่าตอบวงแคบเฉพาะถิ่นมีค่าเพิ่มขึ้นนั่นเอง

เพื่อให้การพิสูจน์ทฤษฎีบท ข. มีความเข้มแข็งมากยิ่งขึ้น การพิสูจน์จะใช้ระเบียบวิธีความน่าจะเป็นเข้าช่วยดังนี้

จากปริภูมิการค้นหาย่อย, Ψ , มีรัศมีการค้นหาเป็น $\rho = \mu \cdot r$ ขณะเริ่มต้นกระบวนการค้นหา ค่ารัศมีการค้นหาเริ่มต้นมีค่าเป็น $\rho = r, (\mu = 1.0)$ เมื่อค่าตอบที่ได้รับการปรับค่าเคลื่อนที่เข้าใกล้ \hat{x} มาก ๆ นั่นคือ $\xi < \rho$ หรืออาจกล่าวได้ว่า $\hat{x} \in N_\rho(x'_i)$ สมาชิกหนึ่งในจำนวน N ตัวจะเป็นค่าตอบวงแคบเฉพาะถิ่น, \hat{x} , จากทฤษฎีความน่าจะเป็น การจัดหมู่ที่เป็นไปได้ในการเลือกสุ่มสมาชิกจำนวน m จาก N สมาชิกคือ $\binom{N}{m} = \frac{N!}{(N-m)!m!}$ นั่นคือจะมีหนทางที่เป็นไปได้ที่จะ

พบ \hat{x} เป็น $\binom{m}{1} = \frac{m!}{(m-1)!}$ หนทาง ดังนั้นค่าความน่าจะเป็นที่กระบวนการค้นหาจะพบค่าตอบ

วงแคบเฉพาะถิ่น, \hat{x} , คือ $P = \frac{m \cdot m!}{N(N-1) \cdots (N-m+1)}$ เมื่อ N คือจำนวนสมาชิกทั้งหมดของ $N_\rho(x'_i)$ และจากทฤษฎีบท ก. ค่า $m < N$ นั้นหมายความว่า เมื่อรัศมีการค้นหาหามีค่ามาก (N มาก) ค่าความน่าจะเป็นในการพบคำตอบวงแคบเฉพาะถิ่นก็จะยิ่งน้อย ด้วยตรรกะนี้ รัศมีการค้นหาจะถูกปรับลดด้วย μ ($0 < \mu \leq 1$) เพื่อให้ทำให้กระบวนการค้นหาสามารถเข้าสู่คำตอบวงแคบเฉพาะถิ่น, \hat{x} , ด้วยเวลาที่รวดเร็ว ดังนั้นข้อสรุปของการพิสูจน์ในส่วนนี้

$$v. \text{ เมื่อ } \zeta < \rho \text{ นั่นคือ } \hat{x} \in N_\rho(x'_i) \rightarrow P_\rho(\hat{x}) = \frac{m \cdot m!}{N(N-1) \cdots (N-m+1)}$$

vi. กำหนดให้ $\zeta < \rho' < \rho \wedge (N > N' \in I^+)$ ดังนั้น

$$\left(P_{\rho'}(\hat{x}) = \frac{m \cdot m!}{N'(N'-1) \cdots (N'-m+1)} \right) > \left(P_\rho(\hat{x}) = \frac{m \cdot m!}{N(N-1) \cdots (N-m+1)} \right)$$

เมื่อ ρ' คือ รัศมีการค้นหาที่ถูกปรับลดลง N คือ จำนวนสมาชิกทั้งหมดของ $N_\rho(x'_i)$ และ N' คือ จำนวนสมาชิกทั้งหมดของ $N_{\rho'}(x'_i)$

vii. จากการปรับลดรัศมีการค้นหาใน vi กระบวนการค้นหาสามารถเข้าสู่ด้วยเวลาที่รวดเร็วยิ่งขึ้น เนื่องจากค่าความน่าจะเป็นของการพบคำตอบวงแคบเฉพาะถิ่นมีค่าเพิ่มขึ้นนั่นเอง ถ้าจำนวนรอบของการค้นหาทั้งหมดที่ใช้สำหรับการค้นหาคำตอบวงแคบเฉพาะถิ่นโดยไม่ใช้กลไก AR มีค่าเป็น M (M เป็นจำนวนเต็มบวก) เวลารวมที่ใช้สำหรับการค้นหาจะมีค่าเป็น $M \cdot \text{Time}(x)$ แต่เมื่อกระบวนการค้นหาใช้กลไก AR จำนวนรอบของการค้นหาทั้งหมดจะลดลงด้วย α ที่ซึ่ง $0 < \alpha < 1$ ดังนั้นเวลารวมที่ใช้จะมีค่าเป็น $\alpha \cdot M \cdot \text{Time}(x) < M \cdot \text{Time}(x)$

กลไก AR จะถูกเรียกใช้เมื่อคำตอบปัจจุบันเข้าใกล้คำตอบวงแคบเฉพาะถิ่น ด้วยลำดับของปริภูมิการค้นหาย่อย p จะมีคำตอบวงแคบเฉพาะถิ่นจำนวน k คำตอบที่ถูกพบ นั้นหมายความว่า กลไก AR จะถูกเรียกใช้จำนวน k ครั้ง ดังนั้นเวลาที่ใช้ในการค้นหารวมคือ

$$\sum_{j=1}^k \alpha_j \cdot M_j \cdot \text{Time}(x) < \sum_{j=1}^k M_j \cdot \text{Time}(x) = p \cdot m \cdot \text{Time}(x) < \sum_{i=1}^k w_i \cdot \text{Time}(x) \text{ หรือ}$$

$$\sum_{j=1}^k \alpha_j \cdot M_j \cdot \text{Time}(x) < \sum_{j=1}^k M_j \cdot \text{Time}(x) = p \cdot m \cdot \text{Time}(x) < n \cdot \text{Time}(x) \text{ นั่นเอง}$$

ทฤษฎีบท ก. กลไก BT ทำให้กระบวนการค้นหาสามารถพบคำตอบวงแคบเฉพาะถิ่นอื่น ๆ ในปริภูมิการค้นหา Ω และหนึ่งในคำตอบวงแคบเฉพาะถิ่นที่พบคือคำตอบวงกว้าง

พิสูจน์ ในระหว่างที่กระบวนการค้นหาดำเนินไปนั้น มีหลายสถานการณ์ที่คำตอบที่พบไม่สามารถให้ค่าฟังก์ชันวัตถุประสงค์ที่ดีกว่า x_0 ในสถานการณ์ดังกล่าว กระบวนการค้นหาจะใช้คำตอบ x_0 เป็นคำตอบเริ่มต้นสำหรับรอบการค้นหาถัดไป ซึ่งจะก่อให้เกิดการวนรอบการค้นหาที่หาค่าคำตอบที่ซ้ำค่าแบบไม่รู้จบ ทำให้เกิดการล็อกโดยคำตอบขึ้น อย่างไรก็ตาม บนพื้นฐานของกระบวนการสุ่ม รอบการค้นหาถัดไปอาจสร้างทิศทางการค้นหาใหม่ที่น่าไปสู่การเคลื่อนที่เข้าใกล้ขอบของปริภูมิคอนเวกซ์, $\Lambda(x_0)$, และเข้าใกล้ปริภูมิใกล้เคียง ทั้งนี้เพราะ $N_\rho(x_0)$ ที่ก่อรูปขึ้นรอบคำตอบ x_0 ด้วยรัศมีที่แน่นอน ρ ทำให้มันอาจทับเกยอยู่บนปริภูมิที่มากกว่าหนึ่งปริภูมิได้นั่นคือ $N_\rho(x_0) - \Lambda(x_0) \not\subset \Lambda(x_0)$ ด้วยคุณสมบัตินี้ จะสามารถทำให้กระบวนการค้นหาสามารถหลุดออกจากการล็อกคำตอบที่ถูกรับแล้วได้

กระนั้นก็ตาม กระบวนการสุ่มอาจล้มเหลวในการหลุดออกจากการล็อกคำตอบ ดังนั้นการใช้คำตอบบางค่าที่บรรจุอยู่ใน TL มาเป็นคำตอบเริ่มต้นสำหรับรอบการค้นหาถัดไปจะสามารถสร้างทิศทางการค้นหาใหม่ที่หลากหลายยิ่งขึ้น ซึ่งจะเป็นการเพิ่มโอกาสให้กระบวนการค้นหาสามารถหลุดออกจากการล็อกโดยคำตอบที่พบแล้วได้อย่างมีประสิทธิภาพมากขึ้น การกระทำเช่นนี้จะทำให้ลำดับของคำตอบที่พบ มีการกระโดดจากปริภูมิหนึ่งไปยังอีกปริภูมิหนึ่ง

การซ้ำคำตอบ (solution cycling) หมายถึงการที่กระบวนการค้นหาไม่สามารถหลุดออกจากกับดักของคำตอบวงแคบเฉพาะถิ่นที่เพิ่งถูกรับ นั่นคือคำตอบที่ได้ในรอบการค้นหาถัดไปจะเป็นการวนกลับมาพบคำตอบวงแคบเฉพาะถิ่นที่เพิ่งถูกรับเดิม กำหนดให้ n_{re} เป็นตัวนับการซ้ำคำตอบ ตัวนับ n_{re} จะเพิ่มขึ้นทุกครั้งเมื่อคำตอบที่พบในรอบการค้นหาใด ๆ มีค่าเท่ากับคำตอบที่ถูกรับแล้วก่อนหน้านี้ และกำหนดให้ n_{re_Max} เป็นจำนวนการซ้ำคำตอบสูงสุดที่ยอมให้เกิดขึ้น ดังนั้นกลไก BT จะถูกเรียกใช้ เมื่อเงื่อนไขต่อไปนี้สอดคล้อง กล่าวคือ ถ้า $n_{re} < n_{re_Max}$ ให้ดำเนินการค้นหาต่อไป ไม่ว่ากระบวนการค้นหาจะสามารถหลุดออกจากการซ้ำคำตอบเองได้หรือไม่ก็ตาม แต่ถ้า $n_{re} \geq n_{re_Max}$ ให้เรียกใช้กลไก BT

เมื่อ $n_{re} \geq n_{re_Max}$ คำตอบที่บรรจุอยู่ใน TL จะถูกเลือก เพื่อนำมาใช้เป็นคำตอบเริ่มต้นสำหรับการก่อรูปปริภูมิการค้นหาย่อย, Ψ , ถัดไป เงื่อนไข $n_{re} \geq n_{re_Max}$ อาจสามารถพิจารณาเป็นเกณฑ์ปรารณา (AC) ลักษณะหนึ่ง กลไก BT จะเลือกคำตอบ x_h ที่ซึ่ง $x_h \in TL$ และ $x_h = \max_{x_i \in TL} \|x_i - x_0\|$ ที่ซึ่ง x_h จะต้องทำให้ $f(x_0) < f(x_h)$ หลังจากเลือกคำตอบที่บรรจุอยู่ใน TL ตามเงื่อนไขข้างต้นแล้ว กำหนดให้ $x_0 = x_h$ มีฐานะเป็นคำตอบเริ่มต้นสำหรับรอบการค้นหาถัดไป ดังนั้นข้อสรุปของการพิสูจน์ในทฤษฎีบทนี้คือ

i. ถ้าคำตอบวงแคบเฉพาะถิ่น, \hat{x} , ถูกรับแล้ว และความยาวของ TL มีขนาดมากพอ จะมีอย่างน้อยหนึ่งคำตอบที่อยู่ใกล้ขอบของปริภูมิคอนเวกซ์, $\Lambda(x_0)$, นั่นคือ

$$\text{length}(\text{TL}) \gg 1 \rightarrow \exists x \in \text{TL} \wedge \|x - x_B\| < \gamma$$

เมื่อ x_B คือ จุดขอบของปริภูมิ และ γ คือ ค่าสูงสุดที่ยอมให้เกิดขึ้นได้

ii. ในระหว่างกระบวนการค้นหา ถ้าคำตอบ x_0 ปัจจุบันเข้าใกล้ขอบของปริภูมิคอนเวกซ์, $\Lambda(x_0)$, ดังแสดงใน i พร้อมกับค่ารัศมีที่แน่นอน ρ มีค่ามากพอที่จะทำให้กระบวนการค้นหาพบคำตอบบางคำตอบภายนอกปริภูมิคอนเวกซ์, $\Lambda(x_0)$, ดังนั้นคำตอบที่ดีที่สุดของปริภูมิการค้นหาย่อย, Ψ , ปัจจุบันสามารถที่จะเป็นคำตอบที่อยู่ภายนอกปริภูมิคอนเวกซ์, $\Lambda(x_0)$, ได้ด้วยความน่าจะเป็นสูง นั่นคือ

$$(\Psi(x_0) - \Lambda(x_0)) \subset (N_\rho(x_0) - \Lambda(x_0)) \not\subset \Lambda(x_0) \rightarrow \exists x \notin \Lambda(x_0)$$

iii. กระบวนการค้นหาใหม่ที่เริ่มต้นจากภายนอกปริภูมิคอนเวกซ์, $\Lambda(x_0)$, ตามที่ได้กล่าวใน ii จะเริ่มต้นกระบวนการค้นหาเพื่อเข้าสู่คำตอบวงแคบเฉพาะถิ่นอื่น ๆ ในปริภูมิคอนเวกซ์อื่น ๆ ที่เหลือในปริภูมิการค้นหา, Ω , กระบวนการค้นหาจะดำเนินการค้นหาซ้ำ ๆ อยู่ตลอดเวลา และเนื่องจากการที่คำตอบวงแคบเฉพาะถิ่นทั้งหมดในปริภูมิการค้นหา, Ω , มีจำนวน k คำตอบ กระบวนการค้นหาจะสามารถพบคำตอบวงแคบเฉพาะถิ่นทั้งหมดภายในเวลาจำกัด $p \cdot m \cdot \text{Time}(x) < \sum_{i=1}^k w_i \cdot \text{Time}(x)$ หรือ $p \cdot m \cdot \text{Time}(x) < n \cdot \text{Time}(x)$ เมื่อ TC สอดคล้อง และ ยุติการค้นหา หนึ่งในจำนวนคำตอบวงแคบเฉพาะถิ่นทั้งหมดที่พบก็คือคำตอบวงกว้างนั่นเอง

จากทฤษฎีบท ค. จำนวนของคำตอบวงแคบเฉพาะถิ่นจำกัดที่แม่นยำของแต่ละปัญหาไม่จำเป็นที่จะต้องทราบก่อน เพราะในความเป็นจริงแล้วจะไม่สามารถทราบได้ เมื่อกลไก BT ถูกเรียกใช้ จะเป็นการประกันว่าคำตอบวงแคบเฉพาะถิ่นอื่นที่ยังไม่ถูกพบจะถูกพบ ในระหว่างกระบวนการค้นหา คำตอบวงแคบเฉพาะถิ่นที่พบแล้วจะถูกสะสมไปเรื่อย ๆ เพราะฉะนั้นภายหลังจากที่กระบวนการค้นหายุติลงโดยสมบูรณ์ จำนวนของคำตอบวงแคบเฉพาะถิ่นจำกัดที่แม่นยำจะมีค่าเท่ากับจำนวนของคำตอบวงแคบเฉพาะถิ่นที่สะสมรวมทั้งหมด

5.5 การวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามเชิงปรับตัว

การวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามเชิงปรับตัว เป็นการทดสอบความสามารถในการค้นหาคำตอบวงกว้างของปัญหาที่มีคำตอบวงแคบเฉพาะถิ่นจำนวนมาก แต่มีคำตอบวงกว้างเพียงคำตอบเดียว การวิเคราะห์จะต้องกระทำกับปัญหาที่หลากหลาย ด้วยจำนวนการค้นหา (trial) ที่มากเพียงพอ และด้วยค่าคำตอบเริ่มต้นที่ไม่เท่ากัน

ในงานวิจัยนี้จะใช้โปรแกรมการค้นหาแบบตามูเชิงปรับตัว ที่พัฒนาขึ้นด้วย MATLAB™ ดำเนินการบนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกะเฮิร์ตซ์ ซึ่งมีหน่วยความจำ RAM ขนาด 256 เมกกะไบต์ สำหรับปัญหาที่ใช้ในการทดสอบจะเป็นการหาค่าต่ำสุด (minimization) ของฟังก์ชันคณิตศาสตร์แบบต่อเนื่อง (continuous) ซึ่งมีด้วยกันทั้งหมดสามฟังก์ชันดังนี้

1. ฟังก์ชันผลรวมตรีโกณมิติแบบไม่สมมาตร (unsymmetrical trigonometric sum function) หรือฟังก์ชัน TSF ดังแสดงในสมการที่ (5-4) เป็นฟังก์ชันที่มีตัวแปรต้นเพียงตัวแปรเดียวคือ x และมีค่าฟังก์ชัน $f(x)$ แสดงดังเส้นกราฟในรูปที่ 5.11 ซึ่งพบว่าฟังก์ชัน TSF ปรากฏค่าตอบวงแคบเฉพาะถิ่นจำนวนมาก แต่จะมีค่าตอบวงกว้าง หรือคำตอบ x ที่ทำให้ $f(x)$ มีค่าน้อยที่สุดอยู่เพียงคำตอบเดียว คือ $x = -0.26$ ซึ่งจะทำให้ $f(x) = 4.56 \times 10^{-5}$ ดังนั้นในการวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว ในการค้นหาค่าต่ำสุดของฟังก์ชัน TSF จะใช้ค่า $\varepsilon = 4.56 \times 10^{-5}$ เป็นหนึ่งใน TC ของกระบวนการค้นหา

2. ฟังก์ชันโบฮาเชฟสกี (Bohachevsky's function) (Bohachevsky et al, 1986) หรือฟังก์ชัน BF ดังแสดงในสมการที่ (5-5) เป็นฟังก์ชันที่มีตัวแปรต้นสองตัวแปรคือ x และ y ค่าฟังก์ชัน $f(x, y)$ แสดงดังพื้นผิวในรูปที่ 5.12 ฟังก์ชัน BF มีค่าตอบวงแคบเฉพาะถิ่นจำนวนมาก แต่จะมีค่าตอบวงกว้าง หรือคำตอบ x และ y ที่ทำให้ $f(x, y)$ มีค่าน้อยที่สุดมีอยู่เพียงคำตอบเดียว คือ $x = y = 0$ ซึ่งจะทำให้ $f(x, y) = 0$

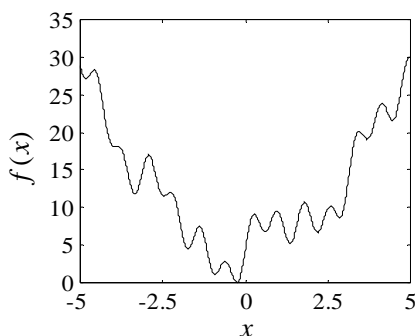
3. ฟังก์ชันวงกลม (Circle function) (Miller et al, 2002) หรือฟังก์ชัน CF ดังแสดงในสมการที่ (5-6) เป็นฟังก์ชันที่มีตัวแปรต้นสองตัวแปรเช่นเดียวกับฟังก์ชัน BF คือ x และ y มีค่าฟังก์ชัน $f(x, y)$ แสดงดังพื้นผิวในรูปที่ 5.13 ฟังก์ชัน CF มีค่าตอบวงแคบเฉพาะถิ่นจำนวนมาก แต่จะมีค่าตอบวงกว้าง หรือคำตอบ x และ y ที่ทำให้ $f(x, y)$ มีค่าน้อยที่สุดมีอยู่เพียงคำตอบเดียว คือ $x = y = 0$ ซึ่งจะทำให้ $f(x, y) = 0$

การทดสอบสมรรถนะของการค้นหาแบบตามูเชิงปรับตัวในการค้นหาค่าต่ำสุดของฟังก์ชัน BF และ CF จะใช้ค่า $\varepsilon = 1 \times 10^{-5}$ เพื่อประมาณค่าศูนย์ และใช้เป็นหนึ่งใน TC ของกระบวนการค้นหา

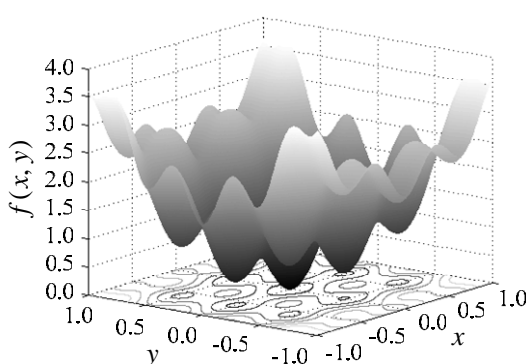
$$f(x) = (\sin(x) + 2.5 \sin(2x) + 1.5 \sin(4x) + 2 \sin(8x)) + x^2 + 4.4716 \quad (5-4)$$

$$f(x, y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7 \quad (5-5)$$

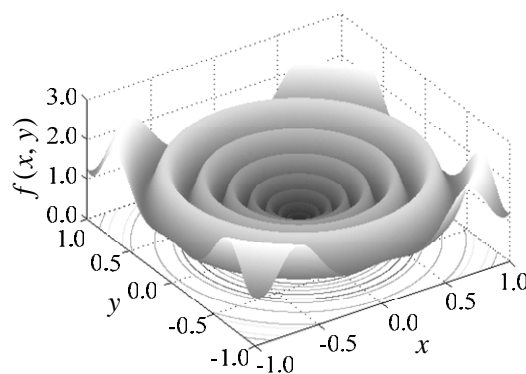
$$f(x, y) = (x^2 + y^2)^{1/4} \left(\sin^2 \left(50(x^2 + y^2)^{1/10} \right) + 0.1 \right) \quad (5-6)$$



รูปที่ 5.11 เส้นกราฟฟังก์ชัน TSF



รูปที่ 5.12 พื้นผิวฟังก์ชัน BF



รูปที่ 5.13 พื้นผิวฟังก์ชัน CF

การค้นหาแบบตาบู่เชิงปรับตัวจะมีตัวประกอบการค้นหา (search factor) หรือค่าพารามิเตอร์ของการค้นหา (search parameter) ที่ส่งผลกระทบต่อสมรรถนะของการค้นหา ดังนี้

1. R หรือรัศมีการค้นหาเริ่มต้น
2. n หรือจำนวนสมาชิกค่าใกล้เคียง
3. n_re_Max หรือจำนวนการซ้ำค่าคำตอบก่อนเรียกใช้กลไกย้อนรอยการค้นหา
4. k^{th} backward selection หรือลำดับคำตอบที่ใช้เป็นค่าเริ่มต้นในกลไกย้อนรอยการค้นหา

5. % reduction of R หรือปริมาณการลดรัศมีการค้นหาเทียบกับรัศมีการค้นหาก่อนหน้านี้ การทดสอบสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตาบู่เชิงปรับตัว จะแบ่งออกเป็นสองส่วนด้วยกัน ดังรายละเอียดต่อไปนี้

การทดสอบในส่วนแรก จะเป็นการทดสอบผลกระทบของตัวประกอบการค้นหา R , n , n_re_Max , และ k^{th} backward selection โดยที่การทดสอบในส่วนนี้จะไม่มี การปรับลดรัศมีการค้นหาแต่อย่างใด ตัวประกอบการค้นหาดังกล่าวจะถูกกำหนดให้มีค่าดังนี้

$R = 2.5, 5, 7.5, 10, 12.5, 15$, และ 20% ของปริภูมิการค้นหา

$n = 10, 20, 30, 40, 50$, และ 60 ตัว

$n_re_Max = 5, 10, 15, 20$, และ 25

$k^{th} backward selection = -1, -2, -3, -4$, และ -5

ตัวประกอบการค้นหาข้างต้นจะได้รับการปรับทีละตัว ขณะที่ตัวอื่นจะถูกคงค่าเอาไว้ การปรับค่าตัวประกอบการค้นหาแต่ละครั้ง จะทำการทดสอบสมรรถนะของการค้นหาแบบตามูเชิงปรับตัว จำนวน 1,000 ครั้ง ค่าคำตอบเริ่มต้นจะถูกกำหนดแบบสุ่มโดย MATLAB™ ทั้งนี้เพื่อเป็นการประกันว่าค่าคำตอบเริ่มต้นในแต่ละครั้งของการค้นหาจะมีค่าไม่เท่ากัน และกระบวนการค้นหาจะยุติลงเมื่อ TC สอดคล้อง ซึ่ง TC ที่ใช้มีสองเงื่อนไขดังนี้คือ

1. จำนวนรอบของการค้นหามีค่าเท่ากับจำนวนรอบของการค้นหาสูงสุด (maximum search round) ซึ่งกำหนดให้มีค่าเท่ากับ 10,000 รอบ หรือ
2. ค่าฟังก์ชัน (cost function) $\leq \varepsilon$

การทดสอบในส่วนที่สอง จะเป็นการทดสอบผลกระทบของตัวประกอบการค้นหา % reduction of R เพียงตัวเดียว สำหรับตัวประกอบการค้นหาอื่น (R, n, n_re_Max , และ $k^{th} backward selection$) จะใช้ค่าที่ทำให้ผลการทดสอบที่ดีที่สุดจากการทดสอบในส่วนแรก สำหรับเงื่อนไขการปรับปริภูมิการค้นหาที่กำหนดไว้ดังสมการที่ (5-7)

$$\begin{aligned}
 \text{(i). } & \text{if (cost function} < 10^{-1}\text{), then } \left(R^{(new)} = \frac{R^{(old)}}{DF} \right) \\
 \text{(ii). } & \text{if (cost function} < 10^{-2}\text{), then } \left(R^{(new)} = \frac{R^{(old)}}{DF} \right) \\
 \text{(iii). } & \text{if (cost function} < 10^{-3}\text{), then } \left(R^{(new)} = \frac{R^{(old)}}{DF} \right)
 \end{aligned} \tag{5-7}$$

เมื่อ $R^{(old)}$ และ $R^{(new)}$ คือ ค่าปริภูมิการค้นหา ก่อน และ หลังการปรับลด ขณะที่ค่า DF คือ ตัวประกอบปรับลด ที่กำหนดให้ค่าปริภูมิการค้นหาภายหลังจากการปรับลดแล้วมีค่าเท่ากับ 10, 15, 20, 25, และ 30% ของค่าปริภูมิการค้นหาหน้า สำหรับค่า $10^{-1}, 10^{-2}$, และ 10^{-3} ในเงื่อนไขการปรับปริภูมิการค้นหาในสมการที่ (5-7) ได้มาจากการสังเกตการลือกโดยคำตอบวงแคบเฉพาะถิ่น จากการทดสอบในส่วนแรก ผลการทดสอบสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว มีแสดงรายละเอียดดังต่อไปนี้

1. ผลของ R ที่มีต่อสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว แสดงดังตารางที่ 5.1 ซึ่งพบว่าในกรณีของฟังก์ชัน TSF ค่า R ที่ 2.5% ของปริภูมิการค้นหาจะให้

สมรรถนะในการค้นหาคำตอบวงกว้างด้วยเวลาที่เร็วที่สุด ในขณะที่ฟังก์ชัน BF และ CF ซึ่งมีความซับซ้อนกว่า ต้องการค่า R อย่างน้อย 7.5% ของปริมาณการค้นหา เพื่อประกันการลู่เข้าหาคำตอบวงกว้าง ดังนั้นโดยสรุปอาจกล่าวได้ว่า ค่า R ในช่วง 7.5 ถึง 12.5% ของปริมาณการค้นหาจะให้สมรรถนะในการค้นหาคำตอบวงกว้างที่น่าพึงพอใจ

ตารางที่ 5.1 ผลของรัศมีการค้นหาเริ่มต้น (R)

R (%)	จำนวนรอบการค้นหาเฉลี่ย			เวลาเฉลี่ย			จำนวนครั้งการค้นหา		
	ที่ใช้ในการค้นหา			ที่ใช้ในการค้นหา (วินาที)			ที่พบคำตอบวงกว้าง		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
2.5	459.45	7923.30	9341.4	1.31	48.51	52.43	1000	203	214
5.0	936.41	7156.60	6610.2	2.81	41.18	38.87	1000	296	405
7.5	897.83	3876.60	3742.3	3.28	20.95	20.63	1000	757	885
10.0	924.17	1353.10	4878.8	3.92	6.06	25.40	1000	1000	816
12.5	992.33	2263.70	5955.1	4.78	10.60	33.03	1000	988	700
15.0	933.06	3071.40	6796.2	5.02	14.91	37.60	1000	954	548
20.0	1035.5	4832.40	8038.0	6.78	24.59	47.148	1000	808	363

ตารางที่ 5.2 ผลของจำนวนสมาชิกค่าใกล้เคียง (n)

n	จำนวนรอบการค้นหาเฉลี่ย			เวลาเฉลี่ย			จำนวนครั้งการค้นหา		
	ที่ใช้ในการค้นหา			ที่ใช้ในการค้นหา (วินาที)			ที่พบคำตอบวงกว้าง		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
10	1473.5	4135.10	7134.4	2.14	10.33	21.82	998	893	526
20	707.15	2203.30	4802.4	1.44	7.72	20.23	1000	987	804
30	459.45	1353.10	3742.3	1.31	6.06	20.63	1000	1000	885
40	357.17	1089.10	3029.3	1.40	6.37	19.37	1000	1000	910
50	287.32	904.08	2928.5	1.33	6.49	25.06	1000	1000	873
60	230.81	802.53	2304.2	1.26	6.70	22.32	1000	1000	908

2. ผลของ n ที่มีต่อสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตาบู่เชิงปรับตัว แสดงดังตารางที่ 5.2 ซึ่งพบว่าจากทั้งสามฟังก์ชันที่ใช้ทดสอบ สมาชิกค่าใกล้เคียงจำนวน 30 ถึง 40 ตัว

ให้สมรรถนะในการค้นหาคำตอบวงกว้างที่น่าพึงพอใจ กล่าวคือ ในกรณีฟังก์ชัน TSF และ BF การดูเข้าหาคำตอบวงกว้างมีค่าเป็น 100% ส่วนในกรณีฟังก์ชัน CF การดูเข้าหามีค่าประมาณ 90%

3. ผลของ n_re_Max ที่มีต่อสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว แสดงดังตารางที่ 5.3 ซึ่งพบว่าจากทั้งสามฟังก์ชันที่ใช้ทดสอบ ค่า n_re_Max หรือจำนวนการซ้ำค่าคำตอบก่อนจะมีการเรียกใช้กลไกย้อนรอยการค้นหา ในระหว่าง 5 ถึง 15 ครั้ง จะให้สมรรถนะในการค้นหาคำตอบวงกว้างที่น่าพึงพอใจ ทั้งในด้านของเปอร์เซ็นต์การดูเข้าหาคำตอบวงกว้าง และเวลาที่ใช้ในการค้นหาคำตอบ

ตารางที่ 5.3 ผลของจำนวนการซ้ำค่าคำตอบก่อนเรียกใช้กลไกย้อนรอยการค้นหา (n_re_Max)

n_re_Max	จำนวนรอบการค้นหาเฉลี่ย			เวลาเฉลี่ย			จำนวนครั้งการค้นหา		
	ที่ใช้ในการค้นหา			ที่ใช้ในการค้นหา (วินาที)			ที่พบคำตอบวงกว้าง		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
5	472.52	1310.90	3029.3	1.36	5.91	19.37	1000	1000	910
10	473.65	1322.10	3279.4	1.43	5.95	22.15	1000	1000	871
15	477.45	1353.10	3438.3	1.43	6.06	23.91	1000	1000	858
20	475.71	1518.80	3466.4	1.44	6.61	24.12	1000	998	851
25	485.43	1438.80	3360.0	1.42	6.51	23.37	1000	1000	862

4. ผลของ k^{th} backward selection ที่มีต่อสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว แสดงดังตารางที่ 5.4 ซึ่งพบว่าจากทั้งสามฟังก์ชันที่ใช้ทดสอบ จะมีอัตราการดูเข้าหาคำตอบวงกว้างที่สูง อย่างไรก็ตาม ในกรณีของฟังก์ชัน BF และ CF เมื่อเลือกค่าคำตอบที่ $k^{th} = -5$ กระบวนการค้นหาจะใช้เวลาในการค้นหาคำตอบวงกว้างด้วยเวลาที่เร็วที่สุด

จากผลการทดสอบในข้อที่ 1 ถึง 4 ทำให้ทราบถึงค่าตัวประกอบการค้นหา (R , n , n_re_Max , และ k^{th} backward selection) ที่เหมาะสม อันจะส่งผลให้สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัวเป็นที่น่าพึงพอใจ ค่าตัวประกอบการค้นหาที่เหมาะสมในแต่ละฟังก์ชันทดสอบเป็นดังนี้

- i. ฟังก์ชัน TSF ใช้ $R = 2.5\%$ ของปริภูมิการค้นหา, $n = 30$ ตัว, $n_re_Max = 5$ ครั้ง, และ $k^{th} = -2$
- ii. ฟังก์ชัน BF ใช้ $R = 10.0\%$ ของปริภูมิการค้นหา, $n = 30$ ตัว, $n_re_Max = 5$ ครั้ง, และ $k^{th} = -5$

iii. ฟังก์ชัน CF ใช้ $R = 7.5\%$ ของปริภูมิการค้นหา, $n = 40$ ตัว, $n_re_Max = 5$ ครั้ง, และ $k^{th} = -5$

ค่าตัวประกอบการค้นหาที่เหมาะสมเหล่านี้จะถูกใช้เพื่อการทดสอบการปรับลดรัศมีการค้นหา (% reduction of R)

ตารางที่ 5.4 ผลของลำดับคำตอบที่เลือกใช้เป็นค่าเริ่มต้นในกลไกย้อนรอยการค้นหา (k^{th})

k^{th}	จำนวนรอบการค้นหาเฉลี่ย			เวลาเฉลี่ย			จำนวนครั้งการค้นหา		
	ที่ใช้ในการค้นหา			ที่ใช้ในการค้นหา (วินาที)			ที่พบคำตอบวงกว้าง		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
-1	461.60	1498.40	3455.1	1.49	6.33	24.09	1000	998	866
-2	493.99	1488.40	3364.4	1.47	6.68	23.65	1000	998	871
-3	477.19	1478.40	3253.7	1.57	6.90	22.21	1000	998	867
-4	464.33	1586.30	3115.4	1.52	7.31	21.45	1000	998	884
-5	470.31	1462.80	3029.3	1.53	6.21	19.37	1000	999	910

ตารางที่ 5.5 ผลของการปรับลดรัศมีการค้นหา (% reduction of R)

R ที่ถูก ปรับลด	จำนวนรอบการค้นหาเฉลี่ย			เวลาเฉลี่ย			จำนวนครั้งการค้นหา		
	ที่ใช้ในการค้นหา			ที่ใช้ในการค้นหา (วินาที)			ที่พบคำตอบวงกว้าง		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
10%	11.07	24.36	1195.4	0.03	0.09	8.70	1000	1000	892
15%	13.36	26.18	1200.9	0.04	0.10	10.52	1000	1000	887
20%	17.85	30.16	600.94	0.05	0.13	4.20	1000	1000	942
25%	22.27	38.41	601.14	0.06	0.16	4.42	1000	1000	940
30%	34.64	64.24	802.92	0.11	0.31	7.15	1000	1000	914

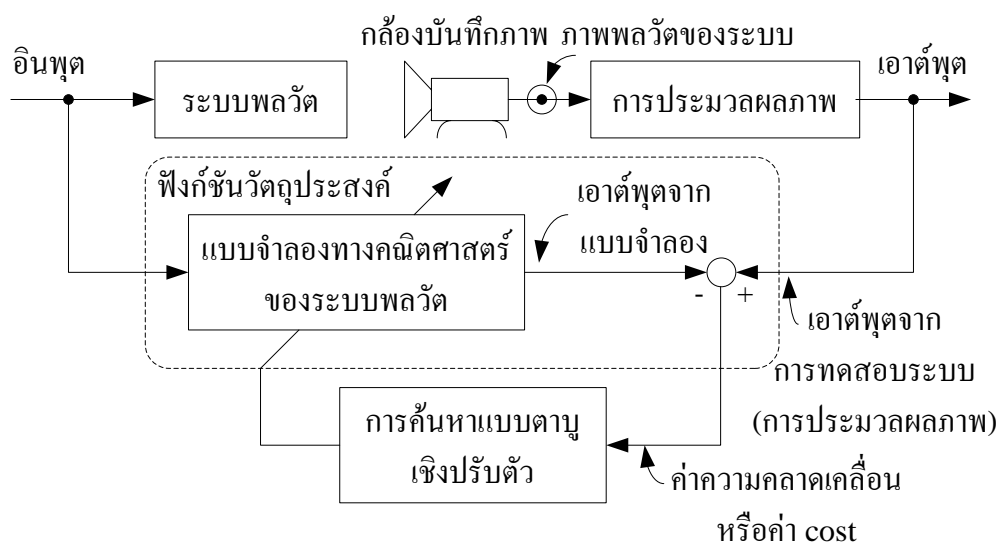
5. ผลของ % reduction of R ที่มีต่อสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามเชิงปรับตัว แสดงดังตารางที่ 5.5 จากผลการทดสอบฟังก์ชันทั้งสาม อาจกล่าวได้ว่า ค่ารัศมีการค้นหา ภายหลังจากการปรับลดแล้วมีค่าเท่ากับ 20 ถึง 25% ของค่ารัศมีการค้นหาก่อนหน้านี้ จะให้สมรรถนะในการค้นหาคำตอบวงกว้างที่น่าพึงพอใจ ทั้งในด้านของการรู้เข้าหาคำตอบวงกว้าง และเวลาที่ใช้ในการค้นหาคำตอบ

จากผลการทดสอบสมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัวที่ผ่านมา แสดงให้เห็นว่า การค้นหาแบบตามูเชิงปรับตัวที่ได้รับการพัฒนาขึ้น มีสมรรถนะการค้นหาผลเฉลยหรือคำตอบวงกว้าง ที่น่าพอใจยิ่ง และอาจสรุปแนวทางการปรับค่าตัวประกอบการค้นหา เพื่อเอื้อประโยชน์ต่อผู้ใช้ต่อไปได้ดังนี้

1. R ควรมีค่าอยู่ในช่วง 7.5 ถึง 15.0% ของปริภูมิการค้นหา
2. n ควรมีค่าอยู่ในช่วง 30 ถึง 40 ตัว
3. n_re_Max ควรมีค่าอยู่ในช่วง 5 ถึง 15 ครั้ง
4. k^{th} backward selection ควรมีค่าเท่ากับหรือประมาณค่า n_re_Max
5. % reduction of R ควรปรับให้ค่ารัศมีการค้นหาภายหลังจากการปรับลดแล้ว มีค่าเท่ากับ 20 ถึง 25% ของค่ารัศมีการค้นหาก่อนหน้า

5.6 การประยุกต์การค้นหาแบบตามูเชิงปรับตัวเพื่อการระบุเอกลักษณ์ระบบ

ในงานวิจัยนี้จะประยุกต์การค้นหาแบบตามูเชิงปรับตัวที่พัฒนาขึ้น เพื่อระบุเอกลักษณ์แบบจำลองทางคณิตศาสตร์ของระบบพลวัต กระบวนการระบุเอกลักษณ์ระบบยังคงใช้แนวคิดดั้งเดิม คืออาศัยข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ เพียงแต่กระบวนการคำนวณเพื่อการค้นหาค่าพารามิเตอร์ของแบบจำลอง จะใช้การค้นหาแบบตามูเชิงปรับตัวเป็นตัวปฏิบัติการ ซึ่งมีแนวทางปฏิบัติดังต่อไปนี้



รูปที่ 5.14 การระบุเอกลักษณ์ระบบ โดยการประยุกต์ใช้การค้นหาแบบตามูเชิงปรับตัว

1. กำหนดให้แบบจำลองทางคณิตศาสตร์ของระบบพลวัตที่สนใจ เป็นส่วนประกอบหลักของฟังก์ชันวัตถุประสงค์
2. ค่าคำตอบที่ได้จากการค้นหาแบบตามูเชิงปรับตัวคือค่าพารามิเตอร์ของแบบจำลอง
3. กลไกการระบุเอกลักษณ์ระบบยังคงใช้แนวคิดดั้งเดิม คืออาศัยข้อมูลความรู้เกี่ยวกับอินพุต (จากตัวตรวจรู้) และเอาต์พุต (จากตัวตรวจรู้หรือจากการประมวลผลภาพ) ของระบบ โดยจะประยุกต์ใช้การค้นหาแบบตามูเชิงปรับตัวเพื่อทำการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง ที่ทำให้ค่าเอาต์พุตจากการจำลองสถานการณ์มีค่าใกล้เคียงกับค่าเอาต์พุตจริงของระบบมากที่สุด โดยพิจารณาจากค่าความคลาดเคลื่อน หรือค่า cost ของกระบวนการค้นหา เมื่อการค้นหาพบค่าพารามิเตอร์ที่สามารถทำให้ค่าความคลาดเคลื่อนมีค่าน้อยที่สุดเท่าที่ยอมรับได้ นั่นคือค่าพารามิเตอร์ในแบบจำลองที่ได้จากการค้นหา รวมทั้งแบบจำลองที่ใช้ สามารถนำมาอธิบายพลวัตของระบบได้อย่างสมเหตุสมผล กระบวนการระบุเอกลักษณ์ระบบ โดยการประยุกต์ใช้การค้นหาแบบตามูเชิงปรับตัวแสดงดังรูปที่ 5.14

5.7 สรุป

ในบทที่ 5 นี้ได้กล่าวถึงการค้นหาแบบตามูเชิงปรับตัว ซึ่งได้รับการพัฒนามาจากการค้นหาแบบตามู โดยเพิ่มกลไกย้อนรอยการค้นหา เพื่อแก้ปัญหาการติดกโดยคำตอบวงแคบเฉพาะถิ่น และกลไกปรับรัศมีการค้นหา เพื่อเพิ่มความเร็วให้กับกระบวนการค้นหาคำตอบ กล่าวถึงการพิสูจน์การลู่เข้าผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว โดยใช้ตรรกะทางคณิตศาสตร์บริสุทธิ์ร่วมกับตรรกะเชิงศึกษาสำนึกและทฤษฎีความน่าจะเป็น ผลการพิสูจน์สรุปได้ว่า (1) ความน่าจะเป็นที่กระบวนการค้นหาแบบตามูเชิงปรับตัว จะพบคำตอบวงแคบเฉพาะถิ่นมีค่าเข้าใกล้หนึ่ง (2) กลไกปรับรัศมีการค้นหาทำให้การค้นหาแบบตามูเชิงปรับตัวลู่เข้าหาคำตอบด้วยเวลาที่รวดเร็วยิ่งขึ้น และ (3) กลไกย้อนรอยการค้นหาทำให้การค้นหาแบบตามูเชิงปรับตัวสามารถพบคำตอบวงแคบเฉพาะถิ่นอื่น ๆ และหนึ่งในคำตอบวงแคบเฉพาะถิ่นที่พบคือคำตอบวงกว้าง กล่าวถึงการวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว โดยการทดสอบเพื่อหาค่าต่ำสุดของฟังก์ชันผลรวมตรีโกณมิติแบบไม่สมมาตร ฟังก์ชันโบฮาเซฟสกี และฟังก์ชันวงกลม ผลการทดสอบพบว่า การค้นหาแบบตามูเชิงปรับตัวมีสมรรถนะในการค้นหาคำตอบวงกว้างอย่างน่าพึงพอใจ และยังได้กล่าวถึงการประยุกต์การค้นหาแบบตามูเชิงปรับตัว เพื่อการระบุเอกลักษณ์ระบบอีกด้วย

บทที่ 6

การทดสอบระบบลูกตุ้มนาฬิกาพวงนกรถ

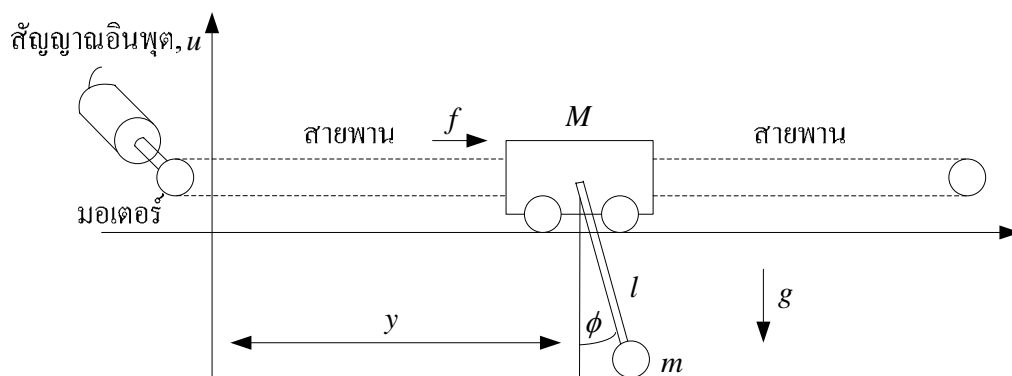
6.1 กล่าวนำ

ระบบลูกตุ้มนาฬิกาพวงนกรถ หรือระบบ CPP (cart-plus-pendulum system) เป็นระบบพลวัตระบบหนึ่งที่ถูกใช้เพื่อการระบุเอกลักษณ์ระบบในงานวิจัยนี้ ดังนั้นเนื้อหาที่ปรากฏในบทที่ 6 จะกล่าวถึงการทดสอบระบบ CPP เพื่อบันทึกข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ โดยอาศัยตัวตรวจรู้ และการประมวลผลภาพ รวมทั้งการแก้ไขความคลาดเคลื่อนของข้อมูลพลวัตที่ได้จากการประมวลผลภาพ เพื่อให้ข้อมูลอินพุตและเอาต์พุตของระบบมีความถูกต้องมากที่สุด สำหรับการระบุเอกลักษณ์ระบบต่อไป

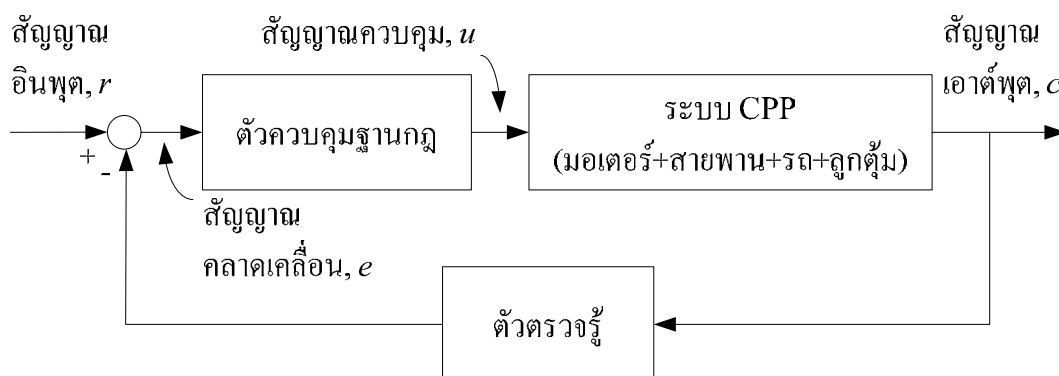
6.2 การทดสอบระบบ CPP

ระบบลูกตุ้มนาฬิกาพวงนกรถ หรือระบบ CPP แสดงในรูปที่ 6.1 เป็นระบบที่ประกอบด้วยลูกตุ้ม (pendulum) ที่ต่อเข้ากับรถ (cart) โดยที่ลูกตุ้มจะแกว่งไกวอยู่ในระนาบ x, y อย่างอิสระตามการเคลื่อนที่ของรถที่ถูกขับด้วยแรง f (นิวตัน) แรงดังกล่าวกำเนิดขึ้นจากการเคลื่อนที่ของมอเตอร์และส่งผ่านมาตามสายพาน โดยที่มุม ϕ คือมุมการแกว่งไกวของลูกตุ้ม (เรเดียน) u คือสัญญาณอินพุตของมอเตอร์ (โวลต์) y คือตำแหน่งของรถ (เมตร) M คือมวลของรถ (กิโลกรัม) m คือมวลของลูกตุ้ม (กิโลกรัม) l คือความยาวของก้านลูกตุ้ม (เมตร) และ g คือความเร่งเนื่องจากแรงโน้มถ่วง (เมตร/วินาที²)

ระบบ CPP ที่ใช้ในงานวิจัยนี้เป็นของบริษัท Feedback (Feedback Instruments Ltd.) มีโครงสร้างเป็นระบบควบคุมวงรอบปิด ดังรูปที่ 6.2 มีตัวควบคุมแบบฐานกฎ ที่แบ่งโหมดการควบคุมออกเป็นสองโหมดคือ โหมดเครน (crane mode) และ โหมดแกว่งขึ้น (swing-up mode) การควบคุมระบบจะดำเนินการผ่านโปรแกรมที่บริษัทได้พัฒนาขึ้นจาก MATLABTM และ Simulink โครงสร้างของระบบควบคุมรวมทั้งโปรแกรมการควบคุมหลักไม่ได้รับการอนุญาตให้มีการเปลี่ยนแปลง ดังนั้นแนวทางการทดสอบระบบจะใช้การทดสอบแบบวงรอบปิด จากรูปที่ 6.2 เมื่อต้องการระบุเอกลักษณ์เฉพาะระบบ CPP ข้อมูลความรู้ที่ใช้คือข้อมูลของสัญญาณควบคุม u ซึ่งถือเป็นสัญญาณอินพุตของระบบ CPP และข้อมูลของสัญญาณเอาต์พุต c (ตำแหน่งเชิงมุมของลูกตุ้ม) ซึ่งถือเป็นสัญญาณเอาต์พุตของระบบ CPP การระบุเอกลักษณ์ระบบในลักษณะดังกล่าว ถือเป็น การประมาณแบบจำลองระบบย่อย (subsystem) (Johansson, 1993)



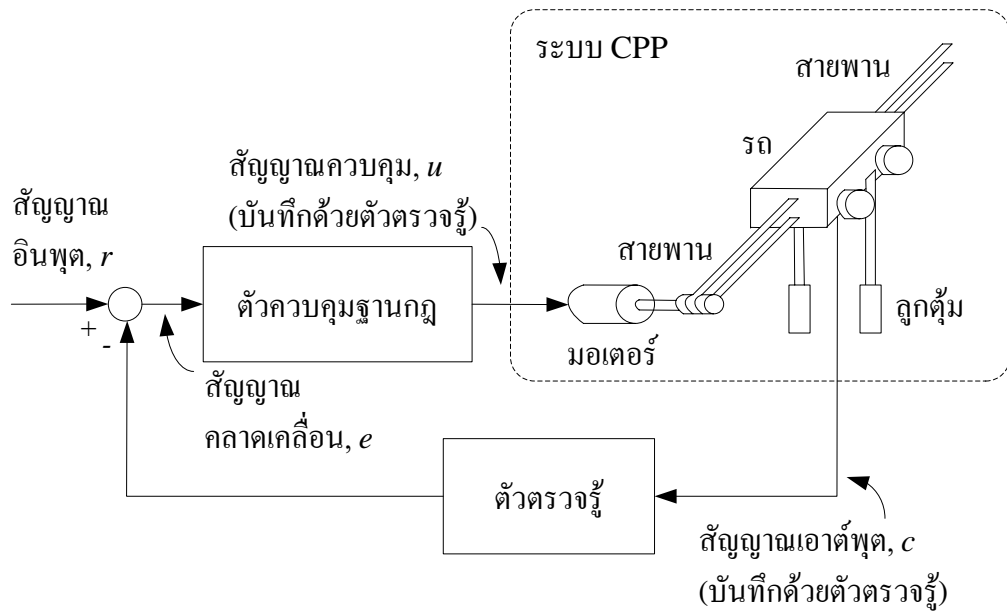
รูปที่ 6.1 ระบบ CPP



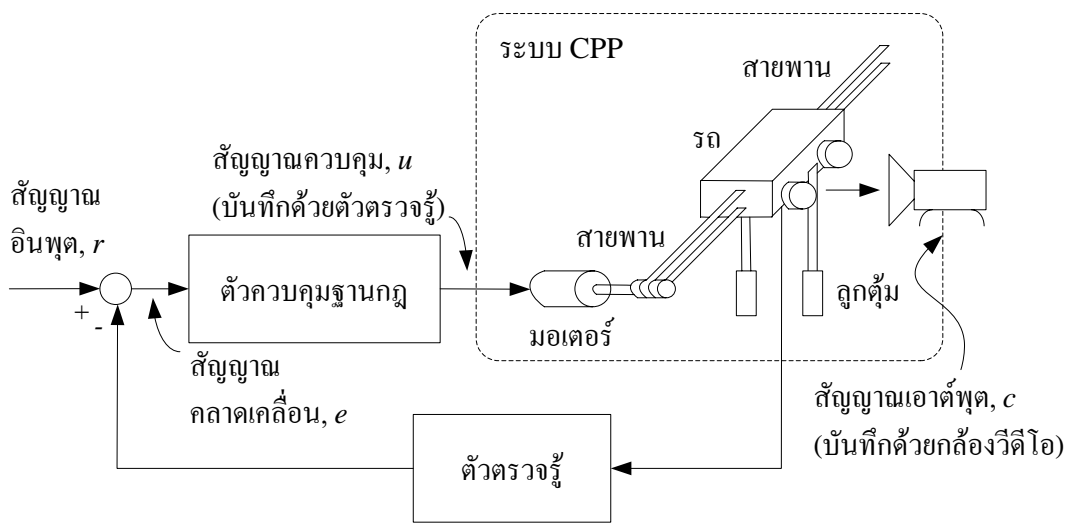
รูปที่ 6.2 โครงสร้างระบบ CPP ของบริษัท Feedback

การทดสอบระบบ CPP เพื่อบันทึกข้อมูลโดยอาศัยตัวตรวจรู้ แสดงดังรูปที่ 6.3 สัญญาณควบคุม u ซึ่งเป็นสัญญาณอินพุตของระบบ CPP และสัญญาณเอาต์พุต c ซึ่งเป็นสัญญาณเอาต์พุตของระบบ CPP (ตำแหน่งเชิงมุมของลูกตุ้ม) จะได้รับการบันทึกโดยเอ็นโคเดอแบบใช้แสง (optical encoder) ซึ่งติดตั้งอยู่ในระบบมาแต่แรก

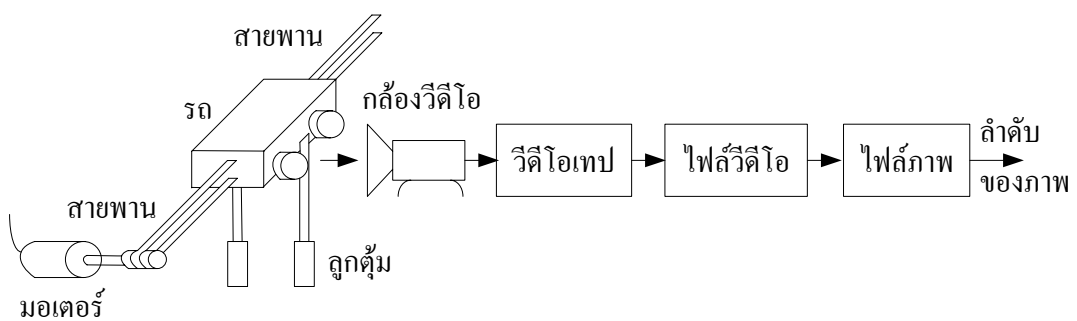
การทดสอบระบบ CPP เพื่อบันทึกข้อมูลโดยอาศัยการประมวลผลภาพ แสดงดังรูปที่ 6.4 สัญญาณควบคุม u จะถูกบันทึกโดยเอ็นโคเดอ ในขณะที่สัญญาณเอาต์พุต c จะได้รับการบันทึกด้วยกล้องวีดีโอดิจิทัล ดังรูปที่ 6.5 การเคลื่อนที่ของลูกตุ้มจะได้รับการบันทึกไว้ในวีดีโอเทป ข้อมูลภาพในวีดีโอเทปจะถูกอ่านผ่านเครื่องเล่นวีดีโอ และส่งเข้าสู่คอมพิวเตอร์เพื่อทำการแปลงเป็นไฟล์วีดีโอ (.avi) โดยอาศัยโปรแกรม Adobe Premiere[®] 6.0 จากนั้นนำไฟล์วีดีโอที่ได้ไปทำการแยกแยะลำดับภาพ (.bmp) โดยใช้ชุดคำสั่งที่พัฒนาขึ้นด้วยโปรแกรม MATLAB[™] ภาพแต่ละภาพจะได้รับการคำนวณเพื่อสกัดข้อมูลพลวัต (ตำแหน่งเชิงมุมของลูกตุ้ม) โดยใช้เทคนิคการกำหนดจุดเปลี่ยนค่า



รูปที่ 6.3 การทดสอบระบบ CPP เพื่อบันทึกข้อมูลโดยอาศัยตัวตรวจรู้

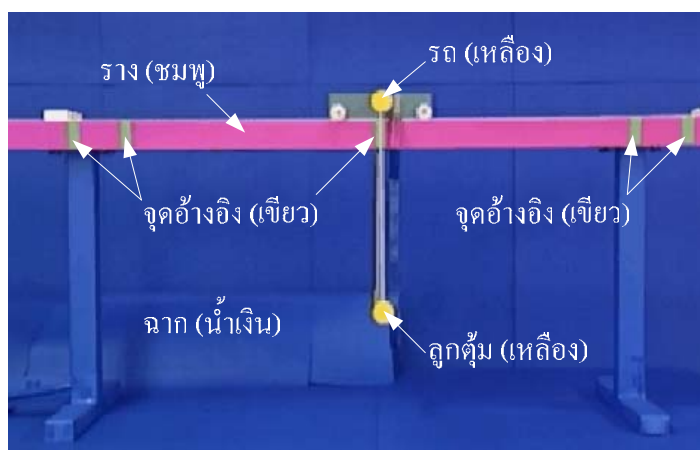


รูปที่ 6.4 การทดสอบระบบ CPP เพื่อบันทึกข้อมูลโดยอาศัยการประมวลผลภาพ



รูปที่ 6.5 การบันทึกผลวัดของระบบ CPP

จากรูปที่ 6.5 ระบบ CPP จะต้องได้รับการจัดทำแถบสีให้กับวัตถุหรือองค์ประกอบต่าง ๆ ของระบบ แถบสีของวัตถุหรือองค์ประกอบแต่ละส่วนจะต้องแตกต่างกันอย่างมีนัยสำคัญ ทั้งนี้เพื่อความสะดวกในการสกัดข้อมูลพลวัตด้วยเทคนิคการกำหนดจุดเปลี่ยนค่า การจัดทำแถบสีดังกล่าวแสดงในรูปที่ 6.6 ตัวรถและลูกตุ้มซึ่งถือเป็นตัวแปรพลวัตของระบบที่สนใจจะเป็นสีเหลือง รางจะเป็นสีชมพู เป็นสีชมพู จุดอ้างอิงจะเป็นสีเขียว และฉากจะเป็นสีน้ำเงิน



รูปที่ 6.6 การจัดทำแถบสีให้กับระบบ CPP

6.3 ขั้นตอนการทดสอบระบบ CPP

ขั้นตอนการทดสอบระบบ CPP เพื่อบันทึกข้อมูลโดยอาศัยตัวตรวจรู้ และการประมวลผลภาพ มีรายละเอียดดังต่อไปนี้

1. กำหนดให้สัญญาณอินพุตอ้างอิง (r) เป็นสัญญาณสุ่ม (random signal) ที่มีคุณสมบัติการกระจายเชิงสถิติแบบสม่ำเสมอ (uniformly distribution) มีค่าเฉลี่ยเท่ากับศูนย์ ขนาดของสัญญาณอินพุตอยู่ในช่วง ± 0.1 , ± 0.2 และ ± 0.3 โวลต์ ทั้งนี้เพื่อให้ลูกตุ้มแกว่งไกวด้วยมุม ± 0.2 , ± 0.4 , และ ± 0.6 เรเดียน (11.4286, 22.8572, และ 34.2858 องศา) ตามลำดับ

2. กำหนดเวลาที่ใช้ในการทดสอบมีค่าเท่ากับ 150 วินาที

3. สำหรับการบันทึกข้อมูลโดยอาศัยตัวตรวจรู้

3.1 กำหนดช่วงเวลาในการชักตัวอย่างข้อมูลเท่ากับ 0.15 วินาที

3.2 บันทึกสัญญาณอินพุตอ้างอิง (r) สัญญาณควบคุม (u) สัญญาณเอาต์พุต (c)

ทั้งหมด ซึ่งประกอบไปด้วยตำแหน่งของรถ ความเร็วของรถ ตำแหน่งเชิงมุมของลูกตุ้ม และความเร็วเชิงมุมของลูกตุ้ม

4. สำหรับการบันทึกข้อมูลโดยอาศัยการประมวลผลภาพ

4.1 อาศัยกล้องวิดีโอดิจิทัลยี่ห้อพานาโซนิค รุ่น DVC-PRO25 (AJ-D400) ซึ่งมีอัตราภาพ (frame rate) เท่ากับ 25 ภาพต่อวินาที ประกอบกับเลนส์ขนาด 6.5 ถึง 123 มิลลิเมตร เพื่อทำการบันทึกภาพการเคลื่อนที่ของระบบ CPP

4.2 ข้อมูลภาพในวิดีโอเทปจะถูกอ่านผ่านเครื่องเล่นวิดีโอ และถูกส่งเข้าสู่คอมพิวเตอร์เพื่อทำการแปลงเป็นไฟล์วิดีโอ (.avi) โดยอาศัยโปรแกรม Adobe Premiere[®] 6.0

4.3 แยกแยะลำดับภาพ (.bmp) จากไฟล์วิดีโอที่ได้โดยใช้ชุดคำสั่งที่พัฒนาขึ้นด้วยโปรแกรม MATLAB[™] (ดังแสดงรายละเอียดไว้ในภาคผนวก ค.)

4.4 สกัดข้อมูลพลวัตของระบบจากภาพโดยใช้เทคนิคการกำหนดจุดเปลี่ยนค่า การสกัดข้อมูลพลวัตของระบบ CPP จากลำดับภาพที่ได้โดยใช้เทคนิคการกำหนดจุดเปลี่ยนค่า มีรายละเอียดดังนี้

1. พิจารณาย่านของค่าระดับแม่สี เพื่อกำหนดจุดเปลี่ยนค่าให้กับกลไกการถอดรายละเอียดข้อมูลภาพ โดยใช้คำสั่ง `pixval` ของ MATLAB[™] เพื่อดูค่าแม่สี (R, G, B) ของจุดภาพต่าง ๆ ซึ่งผลการพิจารณาย่านของค่าระดับแม่สี แสดงในตารางที่ 6.1

2. กำหนดจุดเปลี่ยนค่าให้กับรถและลูกตุ้ม ซึ่งเป็นวัตถุที่สนใจในแต่ละภาพ โดยประยุกต์สมการที่ (4-5) ถึง (4-6) และค่าระดับแม่สีจากตารางที่ 6.1 ดังสมการที่ (6-1)

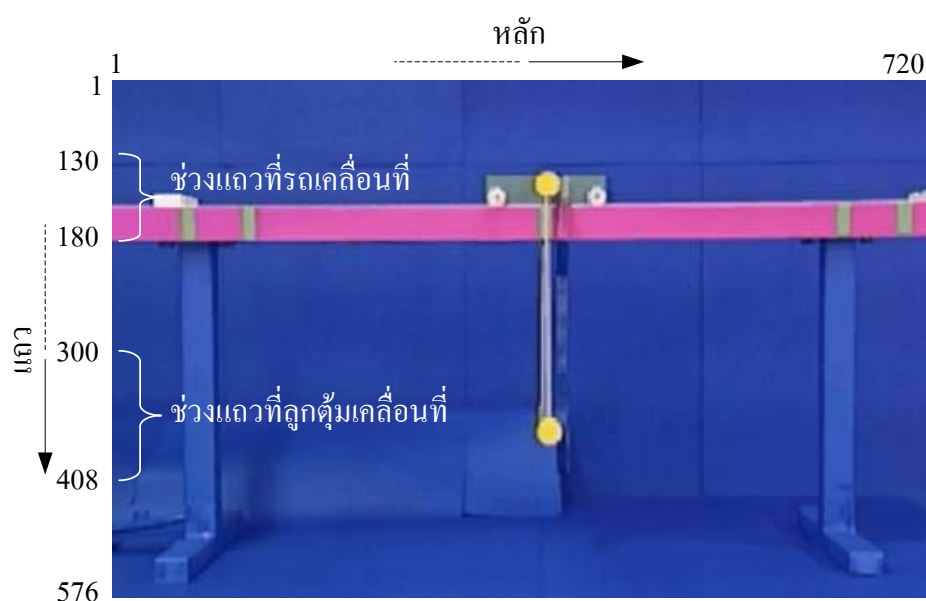
$$g_{cart, pend}(x, y) = \begin{cases} 1, & \begin{cases} 215 < f_R(x, y) \leq 248 \\ 193 < f_G(x, y) \leq 220 \\ 68 < f_B(x, y) \leq 125 \end{cases} \\ 0, & otherwise \end{cases} \quad (6-1)$$

3. ขนาดของภาพสีดิจิทัลที่ใช้มีค่าเท่ากับ 576×720 จุดภาพ เมื่อพิจารณาจากลำดับภาพทั้งหมด พบว่ารถจะมีการเคลื่อนที่อยู่ในแนวระดับ (เคลื่อนไปบนราง) ซึ่งการถอดรายละเอียดข้อมูลตำแหน่งของรถ จะพิจารณาแต่ละภาพในช่วงแถวที่ 130 ถึง 180 ดังแสดงในรูปที่ 6.7 และในส่วนของการถอดรายละเอียดตำแหน่งเชิงมุมของลูกตุ้ม เมื่อพิจารณาจากลำดับภาพทั้งหมดพบว่าลูกตุ้มจะมีการเคลื่อนที่ในช่วงแถวที่ 300 ถึง 408 ของแต่ละภาพ ดังแสดงในรูปที่ 6.7 เช่นกัน ดังนั้น จึงกำหนดให้การถอดรายละเอียดข้อมูลของรถและลูกตุ้ม พิจารณาเฉพาะช่วงแถวที่ระบุข้างต้น ทั้งนี้เพื่อให้ประมวลผลภาพในส่วนของรถและลูกตุ้มเป็นอิสระต่อกัน และยังทำให้การประมวลผลภาพสามารถดำเนินการได้รวดเร็วยิ่งขึ้น

4. กำหนดตำแหน่งอ้างอิงของรถอยู่ตำแหน่งตรงกลางของราง และตำแหน่งเชิงมุมอ้างอิงของลูกตุ้มมีค่าเท่ากับ 0 เรเดียน

ตารางที่ 6.1 ค่าระดับแม่สีของกับองค์ประกอบต่าง ๆ ในระบบ CPP

องค์ประกอบ ของภาพ	แถบสี	ค่าระดับแม่สี		
		R	G	B
ฉาก	น้ำเงิน	45-60	65-85	140-165
จุดอ้างอิง	เขียว	150-170	170-180	138-162
ราง	ชมพู	210-230	85-105	175-205
รถ และ ลูกตุ้ม	เหลือง	215-248	193-220	68-125



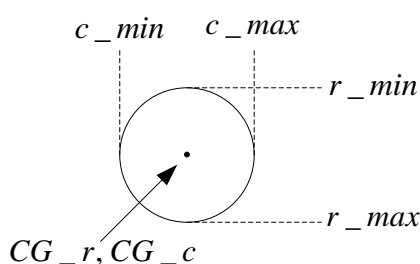
รูปที่ 6.7 ช่วงแถวที่มีการเคลื่อนที่ของรถและลูกตุ้ม

5. การคำนวณตำแหน่งของรถและตำแหน่งเชิงมุมของลูกตุ้ม มีรายละเอียดดังนี้คือ จากขั้นตอนที่ 2 และ 3 เมื่อใช้เทคนิคการกำหนดจุดเปลี่ยนค่าเพื่อหากลุ่มของจุดภาพที่แสดงถึงวัตถุที่สนใจได้แล้ว จะทำการหาจุดต่ำสุด จุดสูงสุด จุดซ้ายสุด และจุดขวาสุด ของกลุ่มจุดภาพ และเนื่องจากได้จัดทำแถบสีให้กับรถและลูกตุ้มเป็นวงกลม ดังนั้นเมื่อได้จุดต่ำสุด และจุดสูงสุดของกลุ่มจุดภาพที่สนใจแล้ว จะดำเนินการหาจุดศูนย์กลางของวัตถุโดยอาศัยสมการที่ (6-2) และ (6-3) (พิจารณารูปที่ 6.8 ประกอบ)

$$CG_r = \frac{(r_{max} - r_{min})}{2} + r_{min} \quad (6-2)$$

$$CG_c = \frac{(c_{max} - c_{min})}{2} + c_{min} \quad (6-3)$$

เมื่อ r_{min} คือ ตำแหน่งแฉกที่เป็นจุดสูงสุดของวัตถุ r_{max} คือ ตำแหน่งแฉกที่เป็นจุดต่ำสุดของวัตถุ c_{min} คือ ตำแหน่งคอคล์มันน์เป็นจุดซ้ายสุดของวัตถุ c_{max} คือ ตำแหน่งคอคล์มันน์ที่เป็นจุดขวาสุดของวัตถุ CG_r คือ ตำแหน่งแฉกของจุดศูนย์กลางของวัตถุ และ CG_c คือ ตำแหน่งคอคล์มันน์ของจุดศูนย์กลางของวัตถุ



รูปที่ 6.8 การกำหนดจุดต่ำสุดและจุดสูงสุดของวัตถุที่สนใจ

6. จากการคำนวณด้วยสมการที่ (6-2) และ (6-3) เมื่อรถอยู่บนตำแหน่งตรงกลางของราง จะได้ $CG_r = 158$ และ $CG_c = 366$ กำหนดให้ตำแหน่งดังกล่าวเป็นตำแหน่งรถอ้างอิง สำหรับการคำนวณตำแหน่งรถและตำแหน่งเชิงมุมของลูกตุ้มเมื่อมีการเคลื่อนที่ ดังสมการที่ (6-4) เมื่อ r_{ref} และ c_{ref} คือ ตำแหน่งแฉกและคอคล์มันน์อ้างอิง ตามลำดับ

$$r_{ref} = 158, c_{ref} = 366 \quad (6-4)$$

7. การคำนวณตำแหน่งรถ และตำแหน่งเชิงมุมของลูกตุ้มเมื่อมีการเคลื่อนที่ มีรายละเอียด ดังนี้ คือเมื่อทำการพิจารณาภาพการเคลื่อนที่ของระบบ CPP ในจังหวะใด ๆ อาจแสดงได้ด้วยรูปที่ 6.9 การคำนวณตำแหน่งของรถเมื่อมีการเคลื่อนที่ เทียบกับตำแหน่งรถอ้างอิง เป็นดังสมการที่ (6-5) เมื่อ $Cart_pos$ คือ ตำแหน่งรถที่เคลื่อนที่ (มิลลิเมตร) CG_c_cart คือ ตำแหน่งคอคล์มันน์ของจุดศูนย์กลางของรถเมื่อเคลื่อนที่ ค่า 2.1285 คือ ค่าคงที่ที่ได้มาจากการพิจารณาขอบเขตที่รถสามารถเคลื่อนที่ได้บนราง ซึ่งมีระยะทางเท่ากับ 977 มิลลิเมตร พิจารณาเป็นจำนวนจุดภาพได้เท่ากับ 459 จุดภาพ ดังนั้น 1 จุดภาพจึงมีค่าเท่ากับระยะทาง 2.1285 มิลลิเมตร

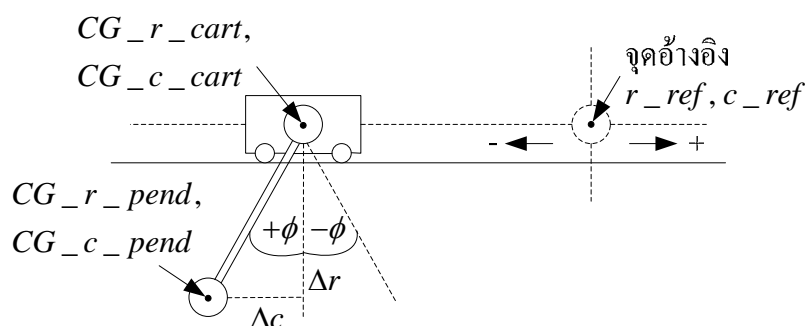
$$Cart_pos = 2.1285(CG_c_cart - c_{ref}) \quad (6-5)$$

การคำนวณตำแหน่งเชิงมุมของลูกตุ้มเมื่อมีการเคลื่อนที่ พิจารณารูปที่ 6.9 จะสามารถคำนวณได้จากสมการที่ (6-6)

$$\theta = \tan^{-1} \frac{\Delta c}{\Delta r} \quad (6-6)$$

เมื่อ θ คือ ตำแหน่งเชิงมุมของลูกตุ้มเมื่อมีการเคลื่อนที่ CG_r_cart คือ ตำแหน่งแนวของจุดศูนย์กลางของรถเมื่อมีการเคลื่อนที่ CG_c_cart คือ ตำแหน่งคอคลัมน์ของจุดศูนย์กลางของรถเมื่อมีการเคลื่อนที่ CG_r_pend คือ ตำแหน่งแนวของจุดศูนย์กลางของลูกตุ้มเมื่อมีการเคลื่อนที่ CG_c_pend คือ ตำแหน่งคอคลัมน์ของจุดศูนย์กลางของลูกตุ้มเมื่อมีการเคลื่อนที่ $\Delta r = (CG_r_pend) - (CG_r_cart)$ คือ ค่าผลต่างของตำแหน่งแนวของรถและลูกตุ้ม และ $\Delta c = (CG_c_cart) - (CG_c_pend)$ คือ ค่าผลต่างของตำแหน่งคอคลัมน์ของรถและลูกตุ้ม

โปรแกรม MATLAB™ ที่ได้รับการพัฒนาขึ้นเพื่อการถอดรายละเอียดข้อมูลพลวัตของระบบ CPP โดยอาศัยขั้นตอนดังกล่าวข้างต้น แสดงในภาคผนวก ก.



รูปที่ 6.9 การคำนวณตำแหน่งรถและตำแหน่งเชิงมุมของลูกตุ้ม

รูปที่ 6.10 แสดงตัวอย่างผลการคำนวณตำแหน่งรถและตำแหน่งเชิงมุมของลูกตุ้มเมื่อระบบมีการเคลื่อนที่ โดยอาศัยโปรแกรมที่พัฒนาขึ้น ซึ่งสามารถให้การอธิบายได้ดังนี้ จากรูปที่ 6.10 (ก) เมื่อรถเคลื่อนที่ไปทางซ้ายของจุดอ้างอิงและลูกตุ้มเคลื่อนที่ตามเข็มนาฬิกา ผลการคำนวณให้ค่าตำแหน่งรถเท่ากับ -85.14 มิลลิเมตร และตำแหน่งเชิงมุมของลูกตุ้มเท่ากับ 0.183 เรเดียน รูปที่ 6.10 (ข) เมื่อรถเคลื่อนที่ไปทางซ้ายของจุดอ้างอิงและลูกตุ้มเคลื่อนที่ทวนเข็มนาฬิกา ผลการคำนวณให้ค่าตำแหน่งรถเท่ากับ -83.0115 มิลลิเมตร และตำแหน่งเชิงมุมของลูกตุ้มเท่ากับ -0.1211 เรเดียน รูปที่ 6.10 (ค) เมื่อรถเคลื่อนที่ไปทางขวาของจุดอ้างอิงและลูกตุ้มเคลื่อนที่ตามเข็มนาฬิกา ผลการคำนวณให้ค่าตำแหน่งรถเท่ากับ 125.5815 มิลลิเมตร และตำแหน่งเชิงมุมของลูกตุ้มเท่ากับ 0.2873

เรเดียน และรูปที่ 6.10 (ง) เมื่อรถเคลื่อนที่ไปทางขวาของจุดอ้างอิงและลูกตุ้มเคลื่อนที่ทวนเข็มนาฬิกา ผลการคำนวณให้ค่าตำแหน่งรถเท่ากับ 83.0115 มิลลิเมตร และตำแหน่งเชิงมุมของลูกตุ้มเท่ากับ -0.2801 เรเดียน และเพื่อให้สามารถสังเกตผลการแบ่งส่วนภาพได้ง่ายยิ่งขึ้น รูปที่ 6.10 (ก) ถึง 6.10 (ง) จึงได้รับการแสดงด้วยภาพระดับเทา โดยกำหนดให้ภายหลังการคำนวณเพื่อแบ่งส่วนภาพด้วยเทคนิคการกำหนดจุดเปลี่ยนค่า จุดภาพใด ๆ ที่พบว่าเป็นส่วนหนึ่งของวัตถุที่สนใจ (รถหรือลูกตุ้ม) จะกำหนดให้มีค่าสีเท่ากับ 255 หรือเป็นจุดภาพที่มีสีขาว ขณะที่จุดภาพใด ๆ ที่พบว่าเป็นส่วนหนึ่งของวัตถุที่สนใจ จะกำหนดให้มีค่าสีเท่ากับ 0 หรือเป็นจุดภาพที่มีสีดำ ซึ่งผลการคำนวณตำแหน่งรถและตำแหน่งเชิงมุมของลูกตุ้มโดยโปรแกรมที่พัฒนาขึ้น ให้ผลเป็นที่น่าพอใจยิ่งทั้งในกรณีที่รถเคลื่อนที่ไปทางซ้ายและทางขวาของจุดอ้างอิง และทั้งในกรณีที่ลูกตุ้มเคลื่อนที่ในทิศทางตามเข็มนาฬิกาและทวนเข็มนาฬิกา

6.4 ผลการทดสอบระบบ CPP

การแสดงผลการทดสอบระบบ CPP จะแสดงเฉพาะข้อมูลของสัญญาณที่สนใจ ได้แก่ ตำแหน่งของรถ และตำแหน่งเชิงมุมของลูกตุ้ม ที่ได้จากการทดสอบด้วยสัญญาณอินพุตขนาด ± 0.1 , ± 0.2 และ ± 0.3 โวลต์ (ลูกตุ้มแกว่งไกวด้วยมุม ± 0.2 , ± 0.4 และ ± 0.6 เรเดียน) โดยจะเรียกการทดสอบในแต่ละครั้งว่าตอน (take) นั้นหมายความว่า การทดสอบระบบ CPP มีจำนวน 3 ตอนด้วยกัน การทดสอบตอนที่ 1 จะใช้สัญญาณอินพุตขนาด ± 0.1 โวลต์ ทั้งนี้เพื่อให้ลูกตุ้มแกว่งไกวด้วยมุมประมาณ ± 0.2 เรเดียน การทดสอบตอนที่ 2 จะใช้สัญญาณอินพุตขนาด ± 0.2 โวลต์ เพื่อให้ลูกตุ้มแกว่งไกวด้วยมุมประมาณ ± 0.4 เรเดียน และการทดสอบตอนที่ 3 จะใช้สัญญาณอินพุตขนาด ± 0.3 โวลต์ เพื่อให้ลูกตุ้มแกว่งไกวด้วยมุมประมาณ ± 0.6 เรเดียน

จำนวนข้อมูลจากการทดสอบในแต่ละตอนที่ได้จากตัวตรวจรู้ และการประมวลผลภาพ จะมีจำนวนไม่เท่ากัน ทั้งนี้เนื่องจากตัวตรวจรู้มีช่วงเวลาในการชักตัวอย่างข้อมูลเท่ากับ 0.15 วินาที นั่นคือจะได้จำนวนข้อมูลประมาณ 6.67 ข้อมูลต่อวินาที ในขณะที่กล้องวิดีโอดิจิทัลที่ใช้มีอัตราภาพเท่ากับ 25 ภาพต่อวินาที นั่นคือจะได้จำนวนข้อมูลประมาณ 25 ข้อมูลต่อวินาที เมื่อต้องการเปรียบเทียบผลการทดสอบ จำเป็นจะต้องจัดสรรข้อมูลให้มีจำนวนเท่ากัน โดยใช้จำนวนข้อมูลจากตัวตรวจรู้เป็นข้อมูลอ้างอิง ข้อมูลจากการประมวลผลภาพจะได้รับการประมาณค่าในช่วง (interpolation) เพื่อจัดสรรให้มีจำนวนข้อมูลเท่ากับจำนวนข้อมูลจากตัวตรวจรู้ ผลการทดสอบระบบ CPP เพื่อบันทึกข้อมูล โดยอาศัยตัวตรวจรู้ และการประมวลผลภาพแสดงดังรูปที่ 6.11 ถึง 6.16 ตามลำดับ

จากผลการทดสอบระบบ CPP พบว่าข้อมูลพลวัตที่ได้จากการประมวลผลภาพมีความใกล้เคียงกับข้อมูลที่ได้จากตัวตรวจรู้มาก อย่างไรก็ตาม ความคลาดเคลื่อนที่เกิดขึ้นอาจมีสาเหตุ

มาจาก การที่กล้องวิดีโอดิจิทัลที่ใช้ทำการบันทึกภาพมิได้รับการปรับเทียบ (calibration) มาก่อน การแก้ไขความคลาดเคลื่อนของข้อมูลที่ได้จากการประมวลผลภาพ จะอาศัยระเบียบวิธีกำลังสองน้อยที่สุด (least square method) เพื่อหาแบบจำลองความคลาดเคลื่อน (error model) ของข้อมูลแต่ละชุด ดังรายละเอียดต่อไปนี้

1. นำข้อมูลพลวัตที่ได้จากตัวตรวจรู้ และการประมวลผลภาพแต่ละชุดมาคำนวณหาความคลาดเคลื่อนตามระเบียบวิธีกำลังสองน้อยที่สุด เพื่อหาแบบจำลองความคลาดเคลื่อน กำหนดให้แบบจำลองความคลาดเคลื่อนมีอันดับ 1 ถึง 5 ดังสมการที่ (6-7)

$$\Gamma = \alpha_n S^n + \dots + \alpha_1 S + \alpha_0, \quad n = 1, \dots, 5 \quad (6-7)$$

เมื่อ Γ คือ แบบจำลองความคลาดเคลื่อน S คือ ข้อมูลที่ได้จากตัวตรวจรู้ (ข้อมูลอ้างอิง) α_n คือ ค่าสัมประสิทธิ์ของแบบจำลองความคลาดเคลื่อน และ i คือ จำนวนข้อมูล

2. แก้ไขความคลาดเคลื่อนของข้อมูลที่ได้จากการประมวลผลภาพตามสมการที่ (6-8)

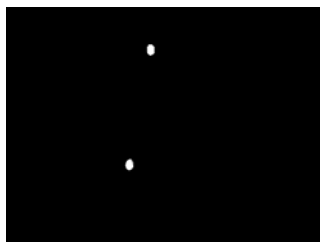
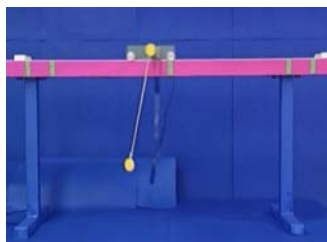
$$\hat{I}_i = I_i + \Gamma_i, \quad i = 1, \dots, n \quad (6-8)$$

เมื่อ \hat{I} คือ ข้อมูลจากการประมวลผลภาพภายหลังการปรับแก้ I คือ ข้อมูลจากการประมวลผลภาพ

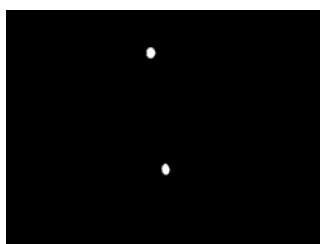
3. นำข้อมูลพลวัตที่ได้จากตัวตรวจรู้ และข้อมูลจากการประมวลผลภาพภายหลังการปรับแก้แต่ละชุด มาคำนวณหาความคลาดเคลื่อน ในรูปของผลรวมของค่าความคลาดเคลื่อนสัมบูรณ์ (sum absolute error, SAE) ดังสมการที่ (6-9)

ตารางที่ 6.2 ผลการแก้ไขความคลาดเคลื่อนของข้อมูลที่ได้จากการประมวลผลภาพของระบบ CPP

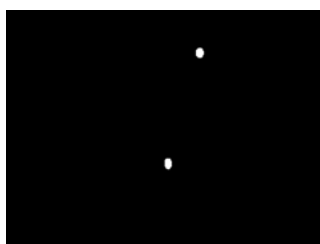
	การทดสอบ	SAE		อันดับของ Γ
		ก่อนแก้ไข	หลังแก้ไข	
ตอนที่ 1	ตำแหน่งเชิงมุมของลูกตุ้ม	17.6140	10.1458	3
	ตำแหน่งของรถ	4.3725	3.6912	2
ตอนที่ 2	ตำแหน่งเชิงมุมของลูกตุ้ม	35.1418	19.5393	3
	ตำแหน่งของรถ	6.9970	6.8119	5
ตอนที่ 3	ตำแหน่งเชิงมุมของลูกตุ้ม	52.5254	30.1909	4
	ตำแหน่งของรถ	10.0095	9.9598	4



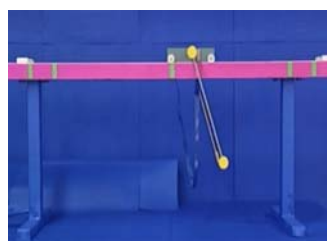
(ก)



(ข)



(ค)



(ง)

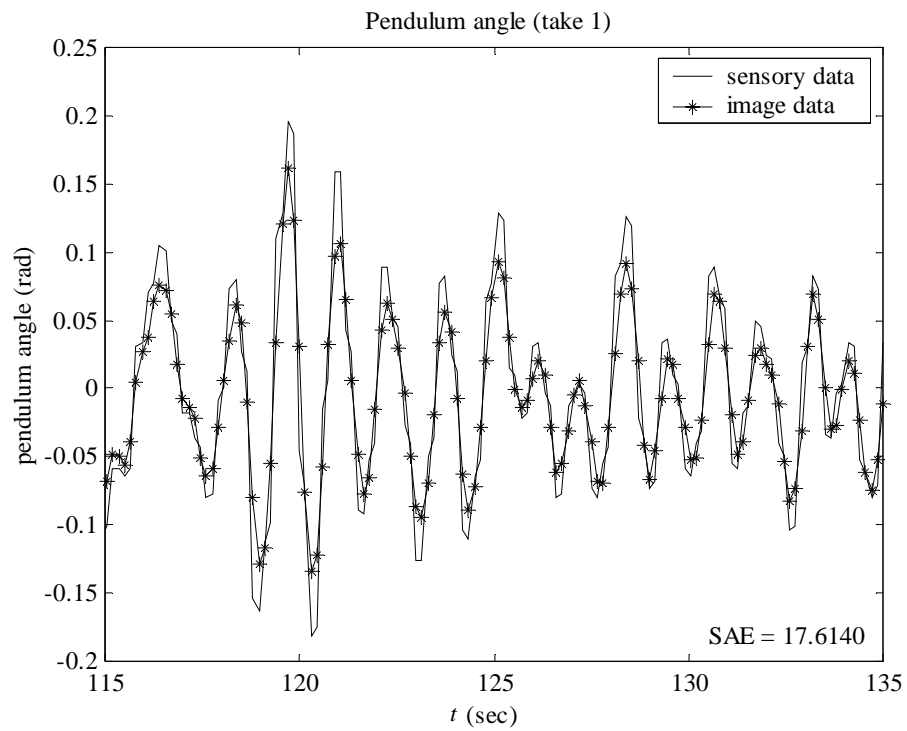
รูปที่ 6.10 ตัวอย่างการคำนวณตำแหน่งรถและตำแหน่งเชิงมุมของลูกตุ้มเมื่อมีการเคลื่อนที่

(ก) รถเคลื่อนซ้ายและลูกตุ้มเคลื่อนที่ตามเข็มนาฬิกา

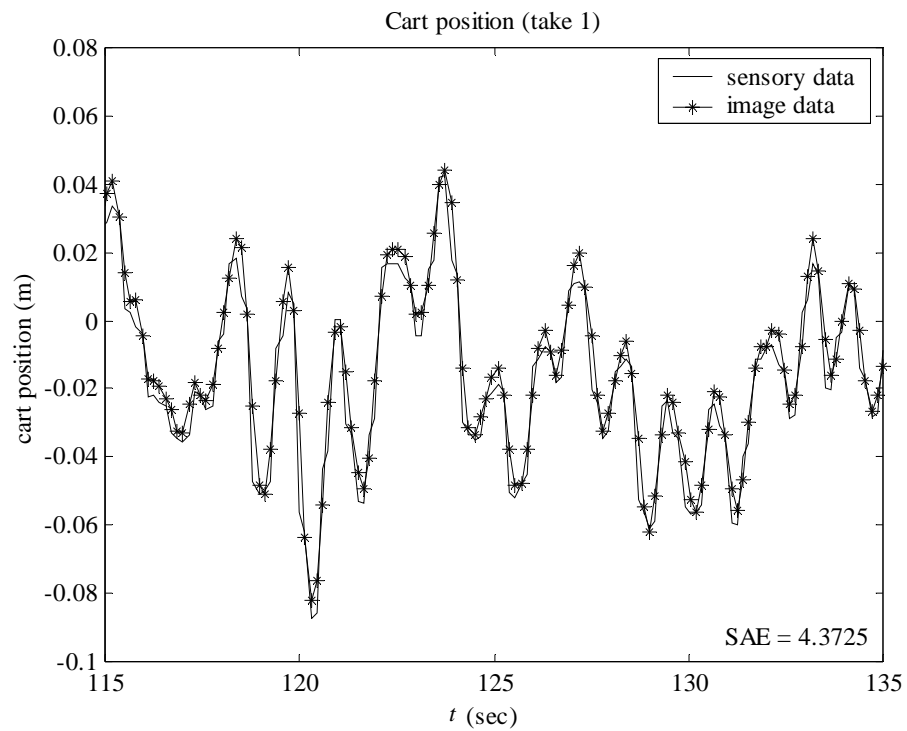
(ข) รถเคลื่อนซ้ายและลูกตุ้มเคลื่อนที่ทวนเข็มนาฬิกา

(ค) รถเคลื่อนขวาและลูกตุ้มเคลื่อนที่ตามเข็มนาฬิกา

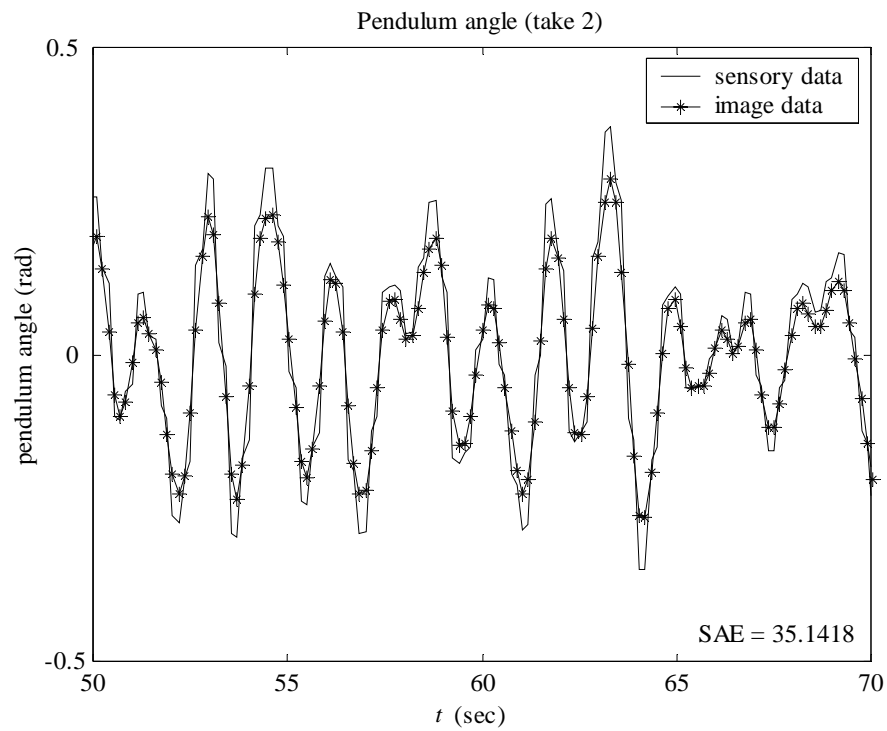
(ง) รถเคลื่อนขวาและลูกตุ้มเคลื่อนที่ทวนเข็มนาฬิกา



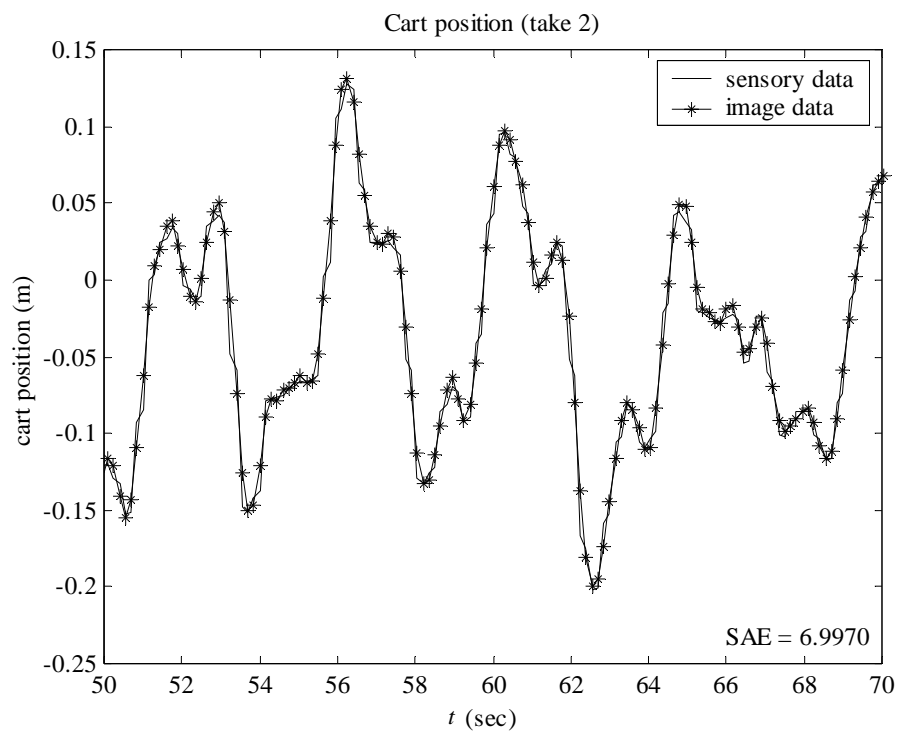
รูปที่ 6.11 ตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 1



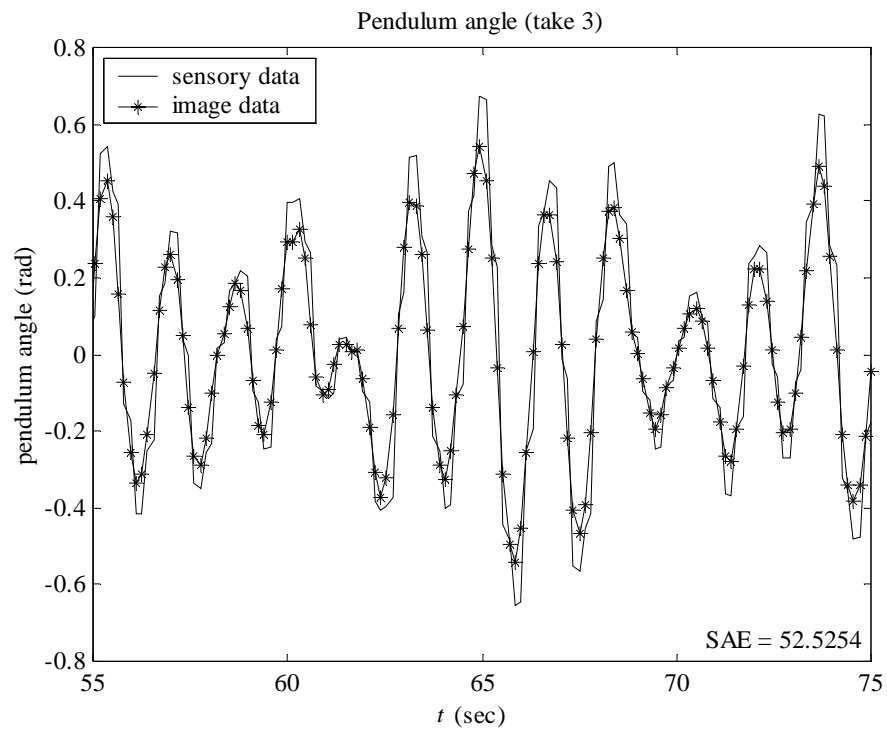
รูปที่ 6.12 ตำแหน่งของรถจากการทดสอบตอนที่ 1



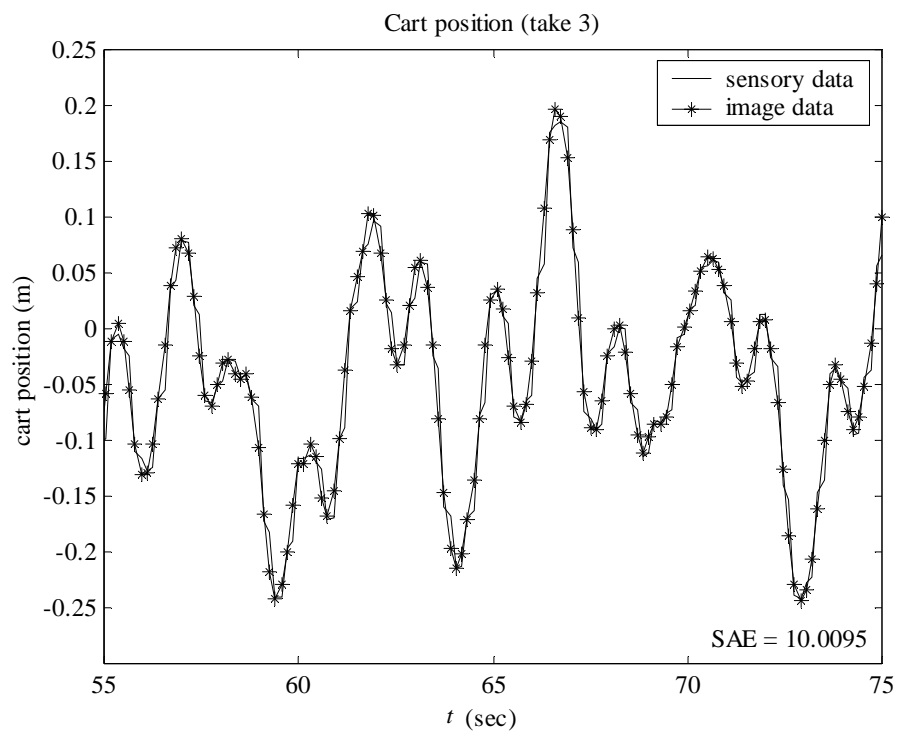
รูปที่ 6.13 ตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 2



รูปที่ 6.14 ตำแหน่งของรถจากการทดสอบตอนที่ 2



รูปที่ 6.15 ตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 3



รูปที่ 6.16 ตำแหน่งของรถจากการทดสอบตอนที่ 3

$$SAE = \sum_{i=1}^n \|S_i - \hat{I}_i\| \quad (6-9)$$

4. เลือกแบบจำลองความคลาดเคลื่อนที่เหมาะสมของข้อมูลแต่ละชุด โดยพิจารณา SAE ที่ให้ค่าน้อยที่สุดอย่างมีนัยสำคัญ

ผลการแก้ไขความคลาดเคลื่อนของข้อมูลที่ได้จากการประมวลผลภาพ โดยระเบียบวิธีกำลังสองน้อยที่สุด แสดงดังตารางที่ 6.2 ซึ่งสามารถให้การอภิปรายผลได้ดังนี้

1. ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 1 มีค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้เท่ากับ 17.6140 เมื่อทำการแก้ไขโดยใช้แบบจำลองความคลาดเคลื่อนอันดับ 3 ดังสมการที่ (6-10) ค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้ลดลงเหลือเท่ากับ 10.1458 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากการประมวลผลภาพก่อนได้รับการแก้ไขแสดงดังรูปที่ 6.17 แบบจำลองความคลาดเคลื่อนอันดับ 3 ดังสมการที่ (6-10) แสดงดังรูปที่ 6.18 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากการประมวลผลภาพภายหลังการแก้ไขแสดงดังรูปที่ 6.19 และตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้ และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อน แสดงดังรูปที่ 6.20

$$\Gamma = 1.2979 \times 10^{-1} S^3 + 1.0353 \times 10^{-3} S^2 + 2.6698 \times 10^{-1} S + 3.3443 \times 10^{-3} \quad (6-10)$$

2. ตำแหน่งของรถที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 1 มีค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้เท่ากับ 4.3725 เมื่อทำการแก้ไขโดยใช้แบบจำลองความคลาดเคลื่อนอันดับ 2 ดังสมการที่ (6-11) ค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้ลดลงเหลือเท่ากับ 3.6912 ความคลาดเคลื่อนของตำแหน่งของรถที่ได้จากการประมวลผลภาพก่อนได้รับการแก้ไขแสดงดังรูปที่ 6.21 แบบจำลองความคลาดเคลื่อนอันดับ 2 ดังสมการที่ (6-11) แสดงดังรูปที่ 6.22 ความคลาดเคลื่อนของตำแหน่งของรถที่ได้จากการประมวลผลภาพภายหลังการแก้ไขแสดงดังรูปที่ 6.23 และตำแหน่งของรถที่ได้จากตัวตรวจรู้ และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อน แสดงดังรูปที่ 6.24

$$\Gamma = -1.2790 \times 10^{-1} S^2 + 1.9086 \times 10^{-2} S - 2.7260 \times 10^{-3} \quad (6-11)$$

3. ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 2 มีค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้เท่ากับ 35.1418 เมื่อทำการแก้ไขโดยใช้แบบจำลอง

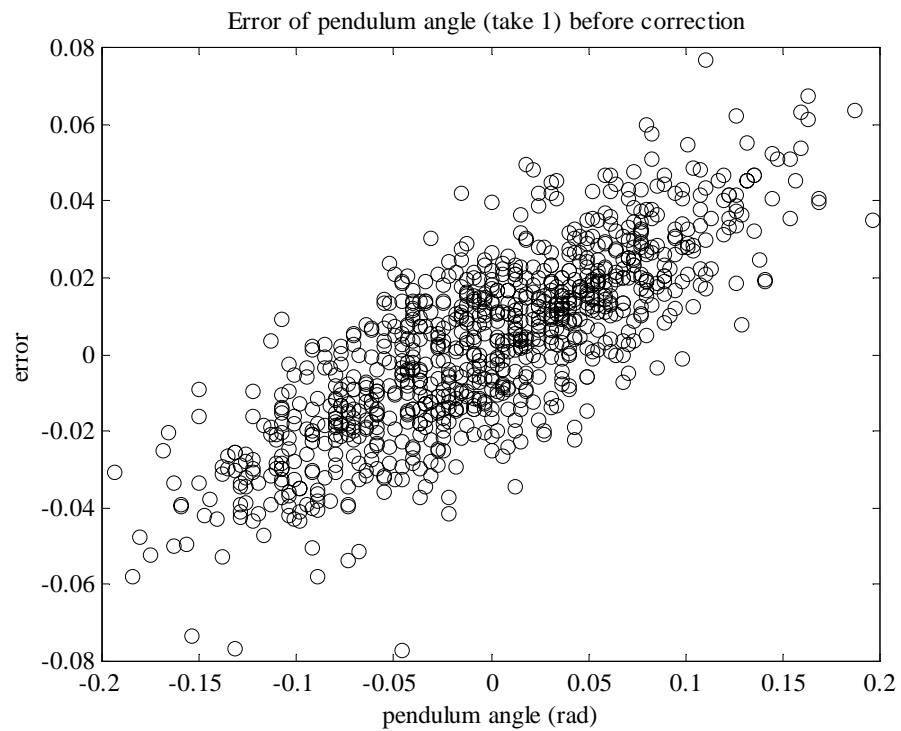
ความคลาดเคลื่อนอันดับ 3 ดังสมการที่ (6-12) ค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้ลดลงเหลือเท่ากับ 19.5393 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกค้อนที่ได้จากการประมวลผลภาพก่อนได้รับการแก้ไขแสดงดังรูปที่ 6.25 แบบจำลองความคลาดเคลื่อนอันดับ 3 ดังสมการที่ (6-12) แสดงดังรูปที่ 6.26 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกค้อนที่ได้จากการประมวลผลภาพภายหลังการแก้ไขแสดงดังรูปที่ 6.27 และตำแหน่งเชิงมุมของลูกค้อนที่ได้จากตัวตรวจรู้และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อน แสดงดังรูปที่ 6.28

$$\Gamma = -1.3616 \times 10^{-1} S^3 - 8.7487 \times 10^{-3} S^2 + 2.7361 \times 10^{-1} S + 8.2133 \times 10^{-3} \quad (6-12)$$

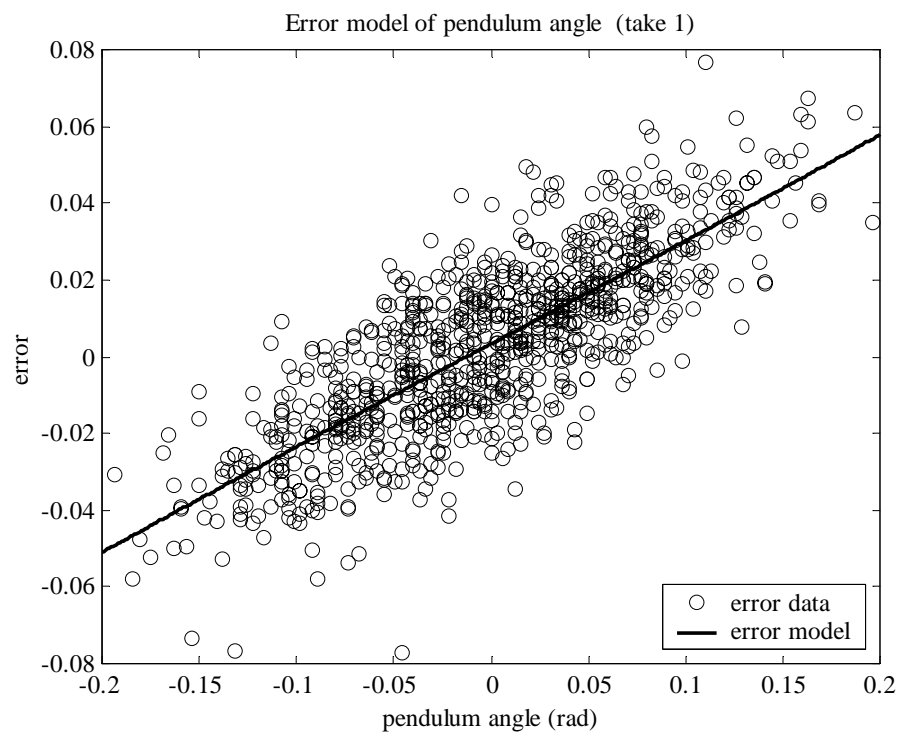
4. ตำแหน่งของรถที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 2 มีค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้เท่ากับ 6.9970 เมื่อทำการแก้ไขโดยใช้แบบจำลองความคลาดเคลื่อนอันดับ 5 ดังสมการที่ (6-13) ค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้ลดลงเหลือเท่ากับ 6.8119 ความคลาดเคลื่อนของตำแหน่งของรถที่ได้จากการประมวลผลภาพก่อนได้รับการแก้ไขแสดงดังรูปที่ 6.29 แบบจำลองความคลาดเคลื่อนอันดับ 5 ดังสมการที่ (6-13) แสดงดังรูปที่ 6.30 ความคลาดเคลื่อนของตำแหน่งของรถที่ได้จากการประมวลผลภาพภายหลังการแก้ไขแสดงดังรูปที่ 6.31 และตำแหน่งของรถที่ได้จากตัวตรวจรู้และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อน แสดงดังรูปที่ 6.32

$$\begin{aligned} \Gamma = & 2.8763 \times 10 S^5 + 9.8625 S^4 - 1.6300 \times 10^{-1} S^3 - 1.8838 \times 10^{-1} S^2 \\ & + 1.2958 \times 10^{-2} S - 1.2260 \times 10^{-3} \end{aligned} \quad (6-13)$$

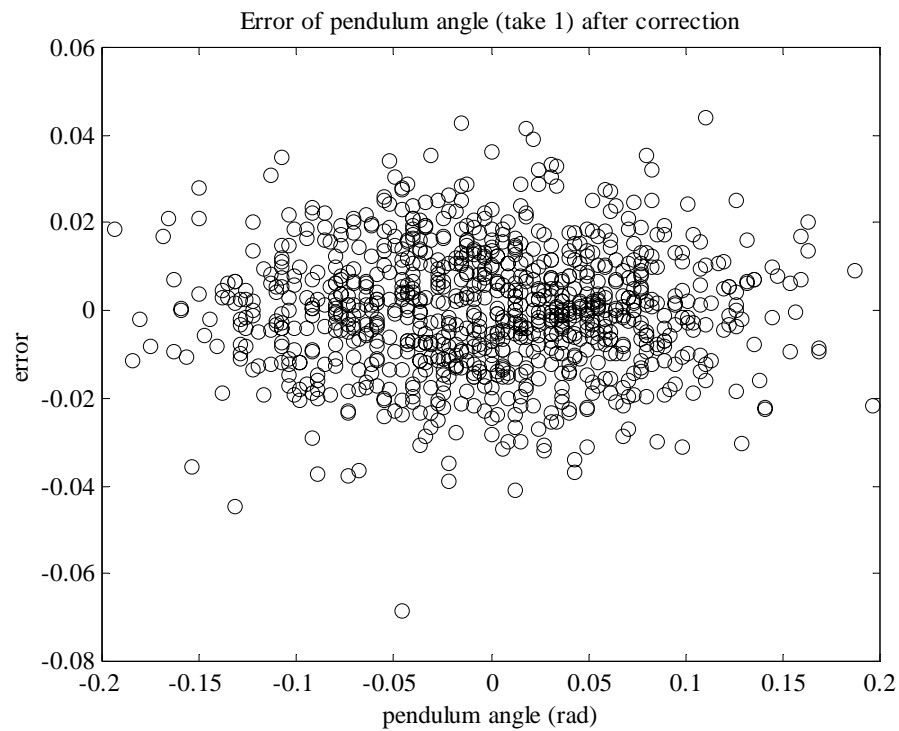
5. ตำแหน่งเชิงมุมของลูกค้อนที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 3 มีค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้เท่ากับ 52.5254 เมื่อทำการแก้ไขโดยใช้แบบจำลองความคลาดเคลื่อนอันดับ 4 ดังสมการที่ (6-14) ค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้ลดลงเหลือเท่ากับ 30.1909 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกค้อนที่ได้จากการประมวลผลภาพก่อนได้รับการแก้ไขแสดงดังรูปที่ 6.33 แบบจำลองความคลาดเคลื่อนอันดับ 4 ดังสมการที่ (6-14) แสดงดังรูปที่ 6.34 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกค้อนที่ได้จากการประมวลผลภาพภายหลังการแก้ไขแสดงดังรูปที่ 6.35 และตำแหน่งเชิงมุมของลูกค้อนที่ได้จากตัวตรวจรู้และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อน แสดงดังรูปที่ 6.36



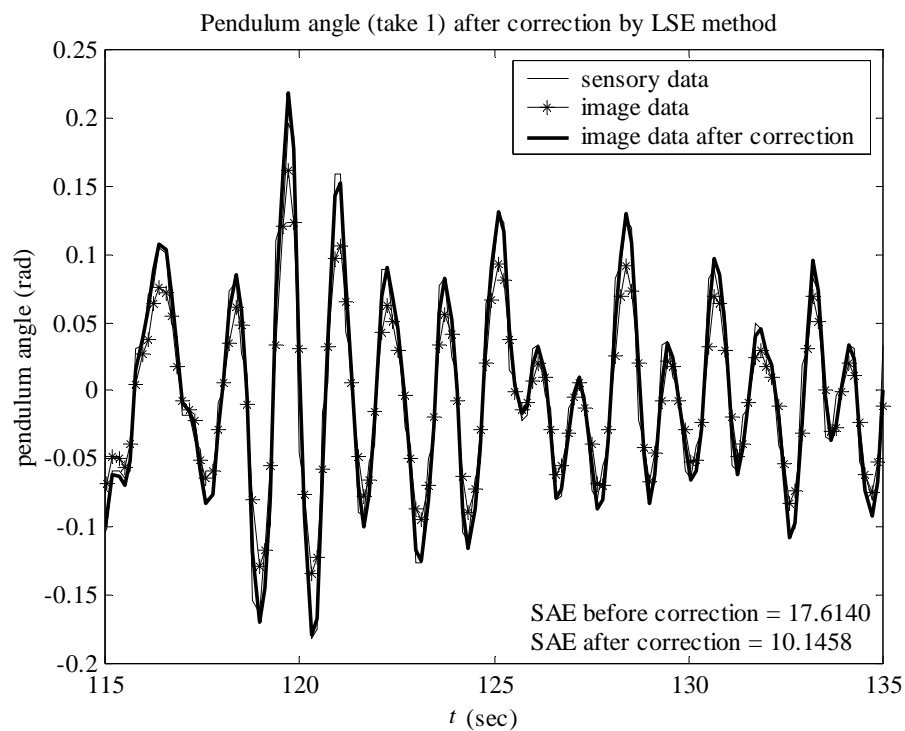
รูปที่ 6.17 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 1 ก่อนแก้ไข



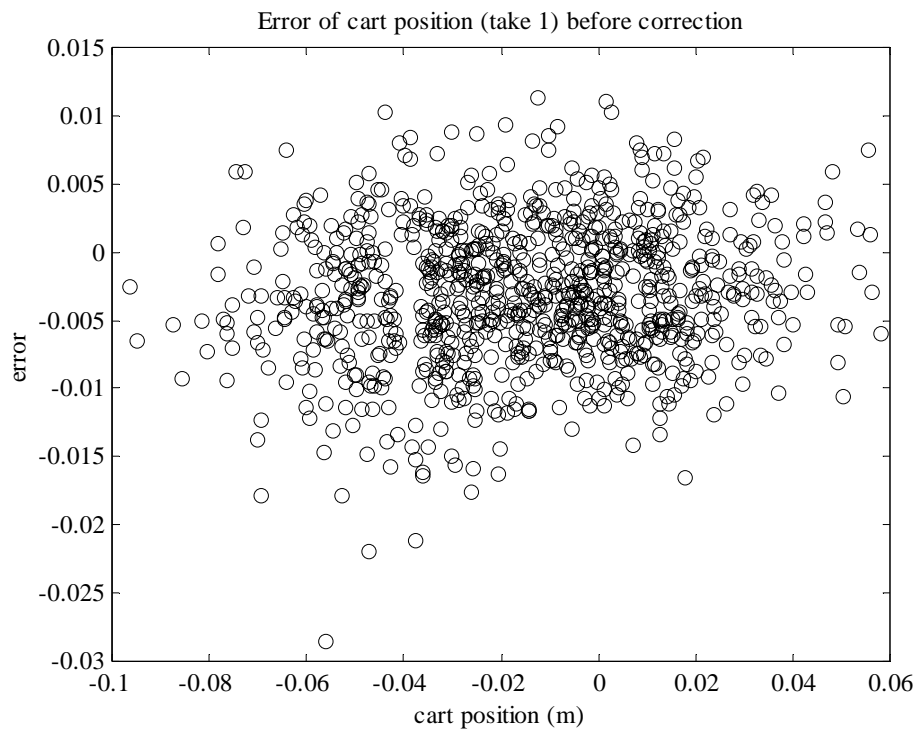
รูปที่ 6.18 แบบจำลองความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 1



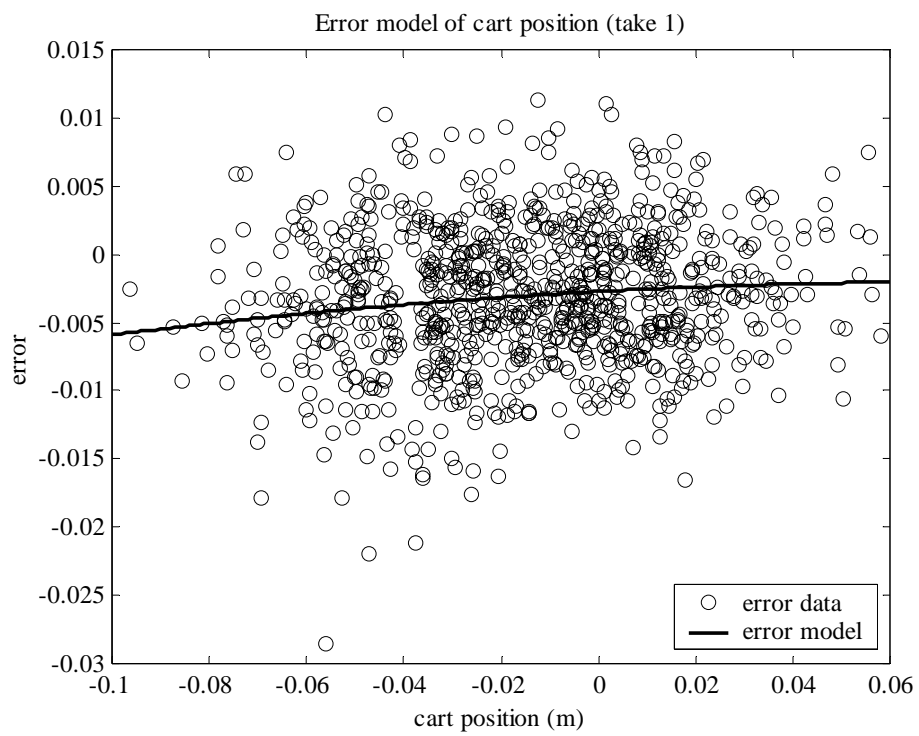
รูปที่ 6.19 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 1 หลังแก้ไข



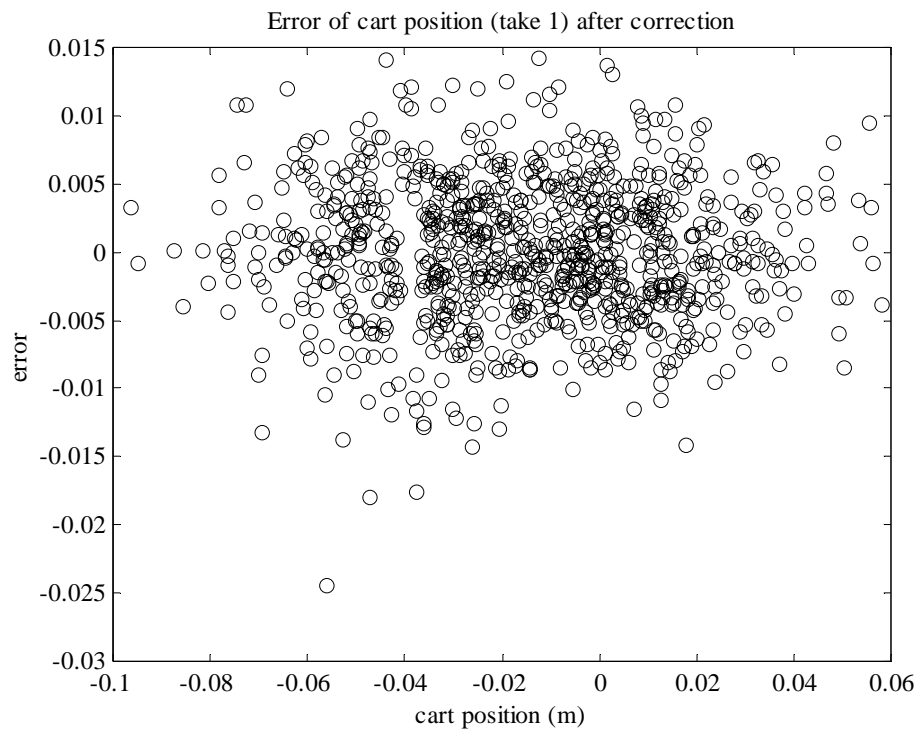
รูปที่ 6.20 ตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 1 ก่อนและหลังแก้ไข



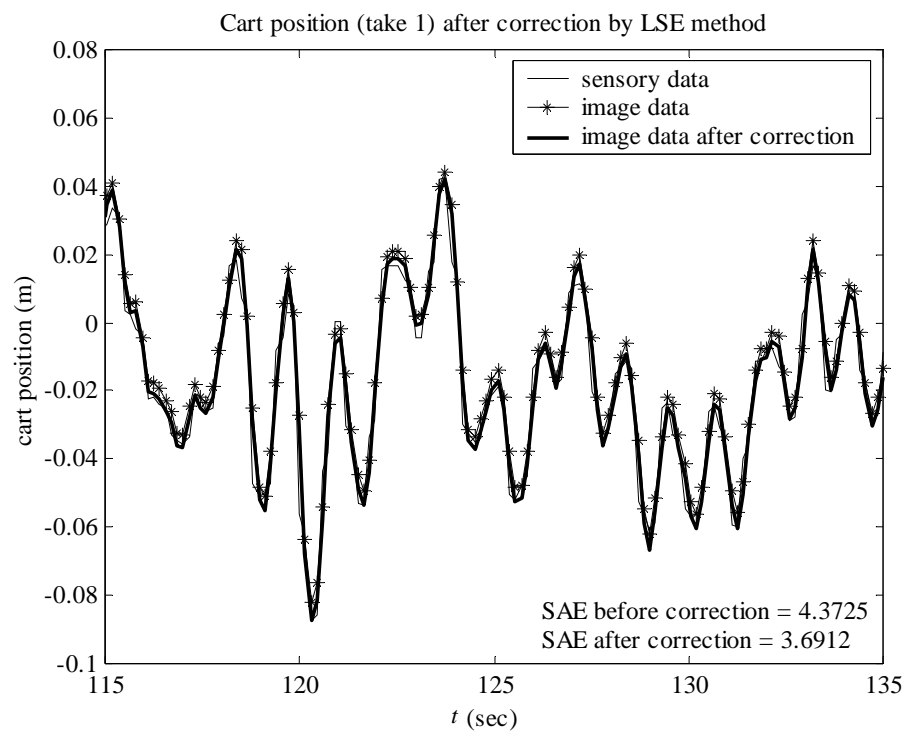
รูปที่ 6.21 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 1 ก่อนแก้ไข



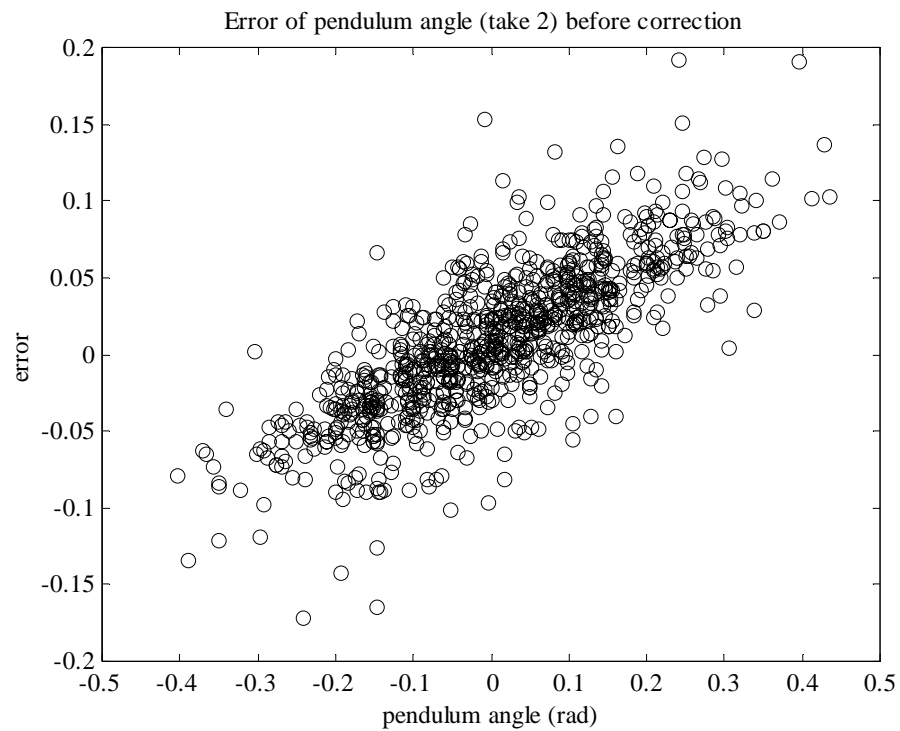
รูปที่ 6.22 แบบจำลองความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 1



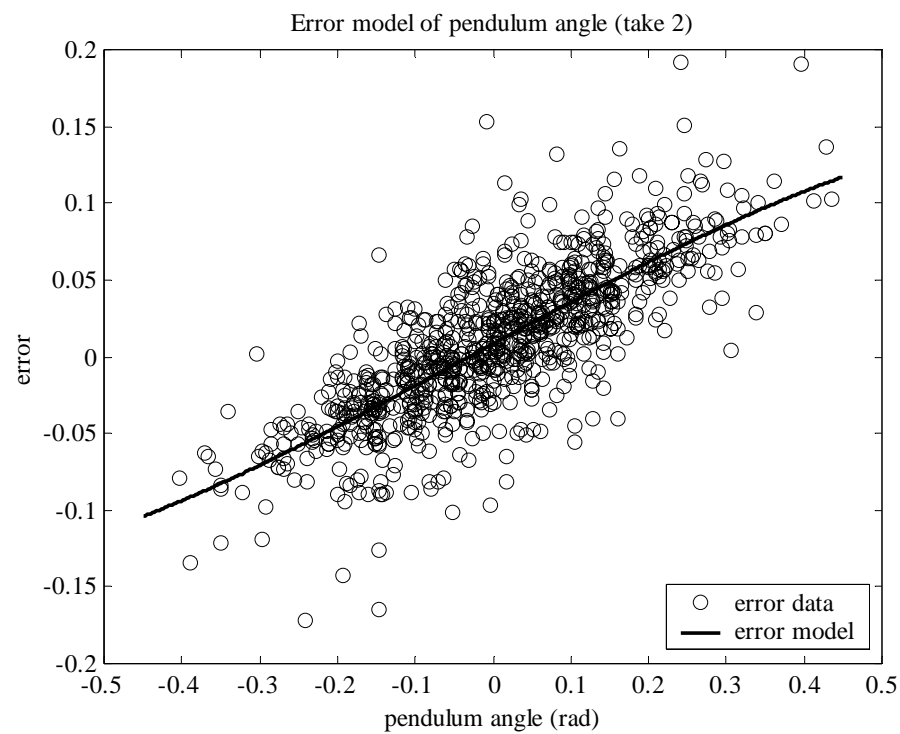
รูปที่ 6.23 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 1 หลังแก้ไข



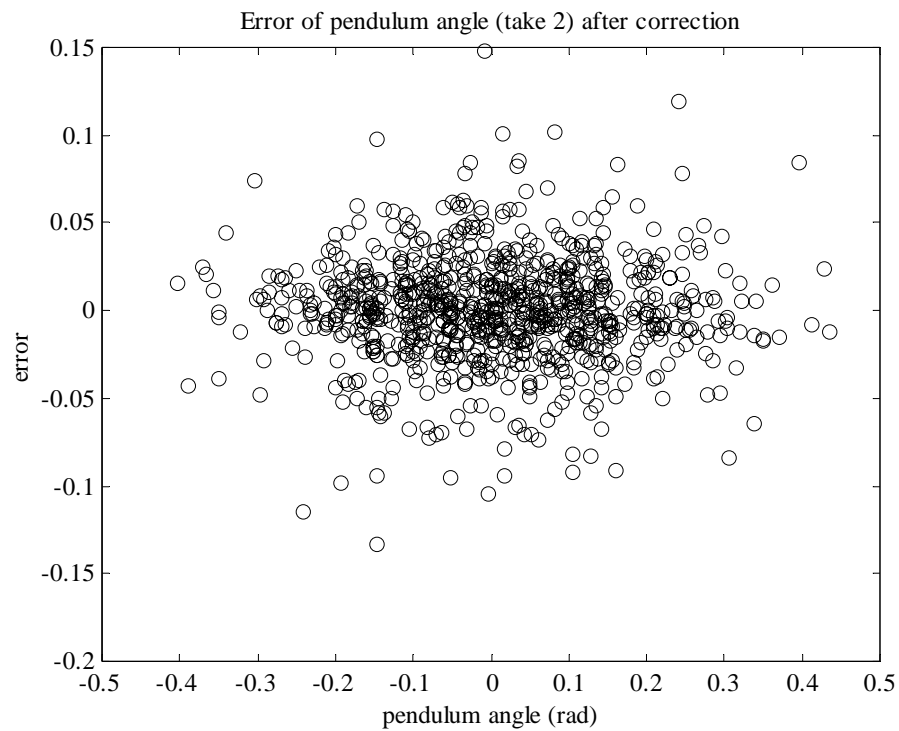
รูปที่ 6.24 ตำแหน่งของรถจากการทดสอบตอนที่ 1 ก่อนและหลังแก้ไข



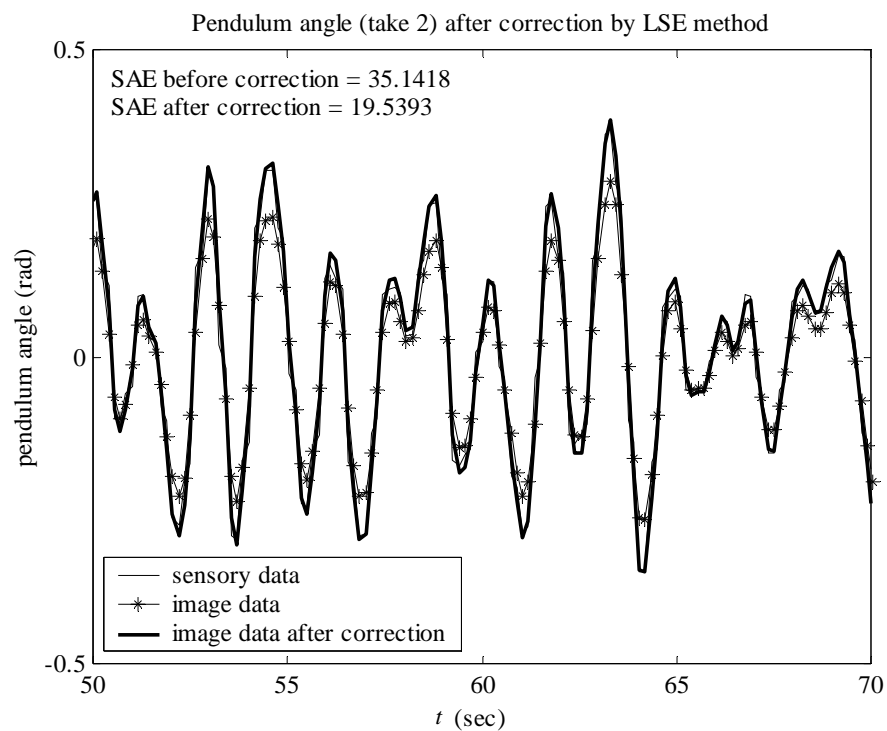
รูปที่ 6.25 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 2 ก่อนแก้ไข



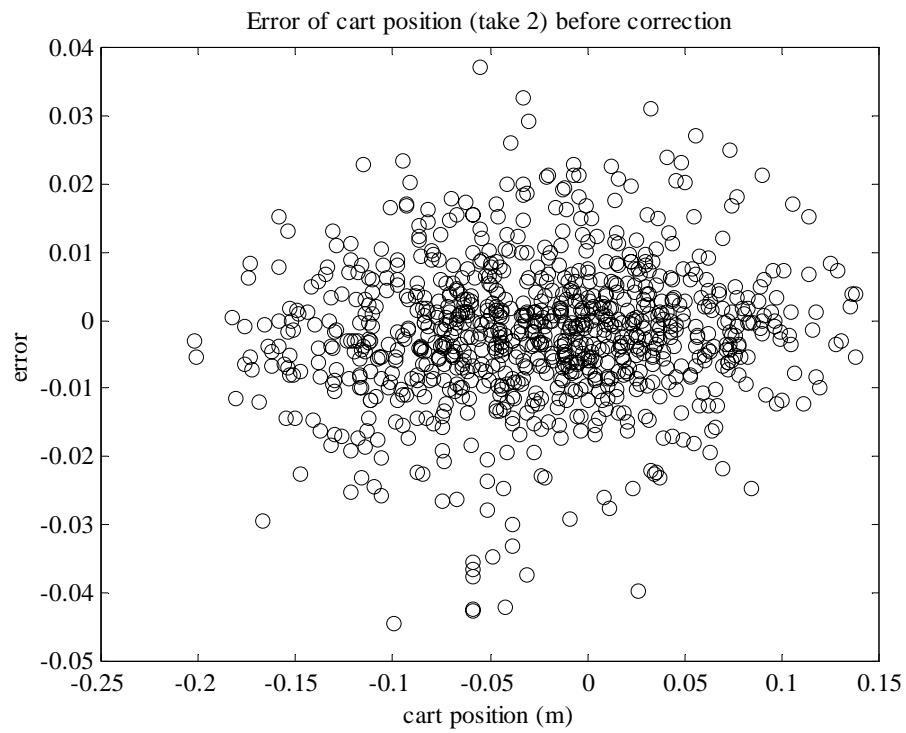
รูปที่ 6.26 แบบจำลองความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 2



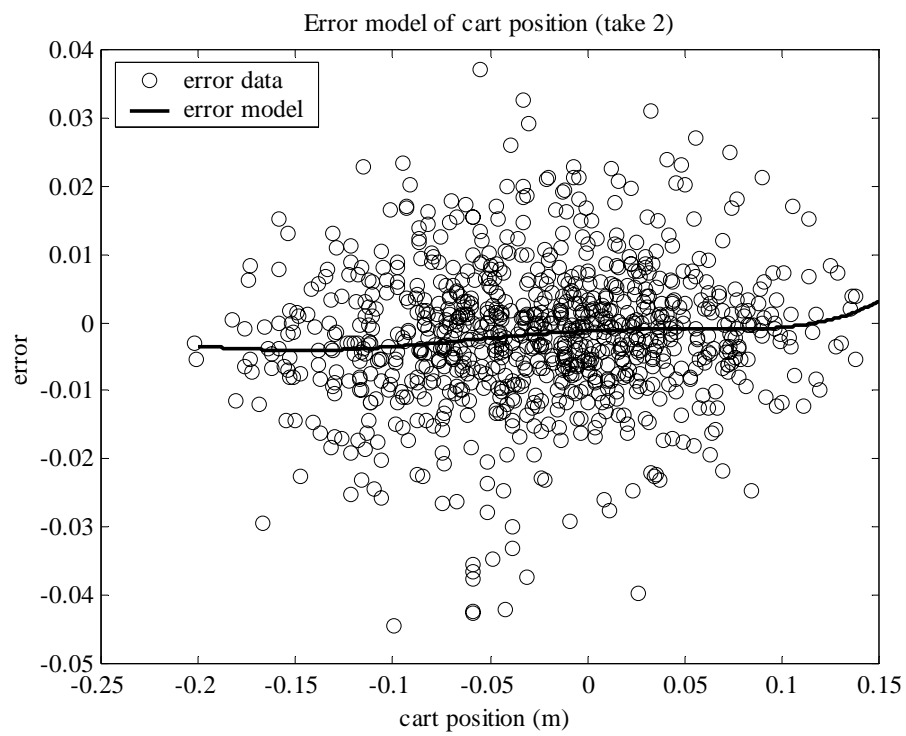
รูปที่ 6.27 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 2 หลังแก้ไข



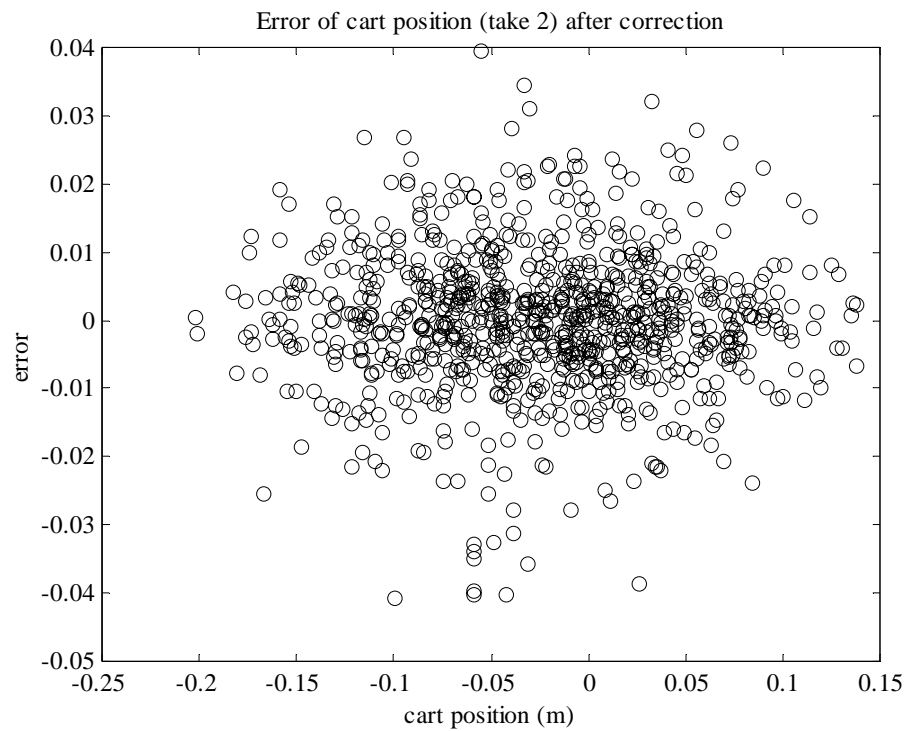
รูปที่ 6.28 ตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 2 ก่อนและหลังแก้ไข



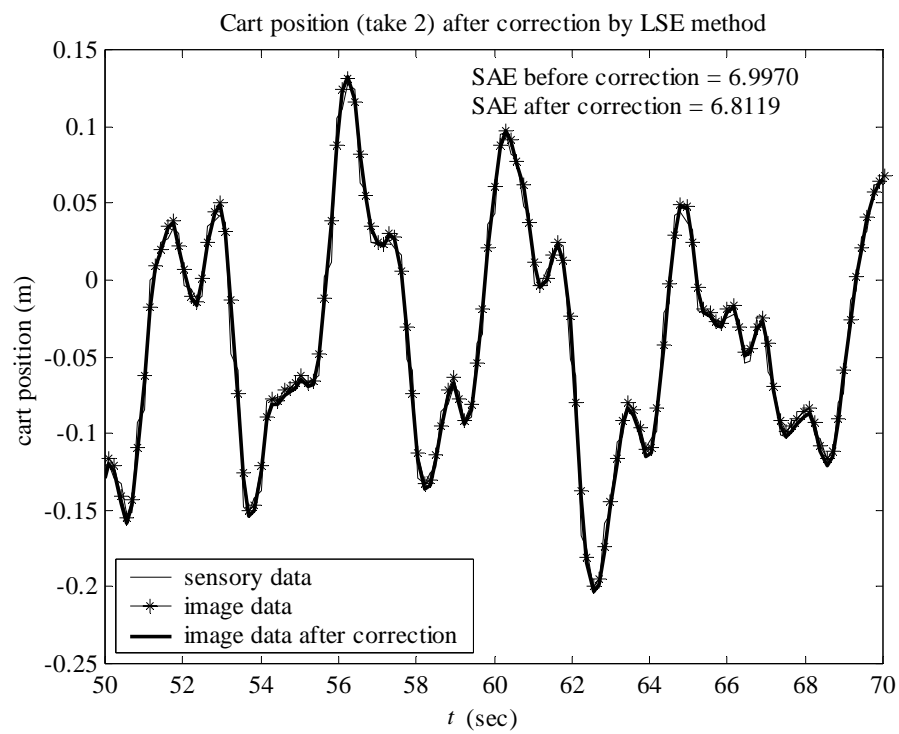
รูปที่ 6.29 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 2 ก่อนแก้ไข



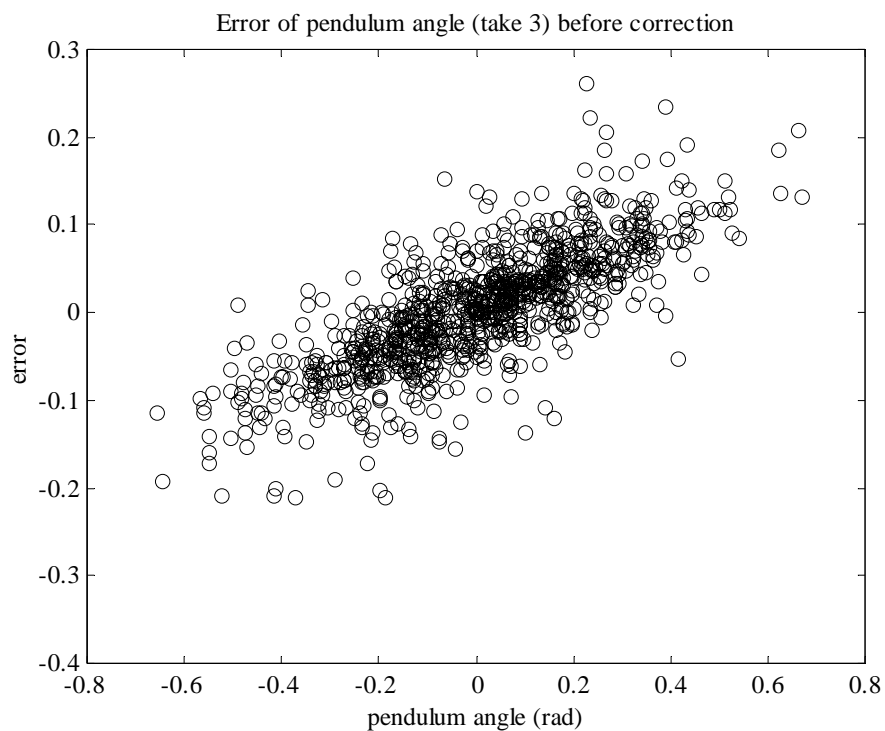
รูปที่ 6.30 แบบจำลองความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 2



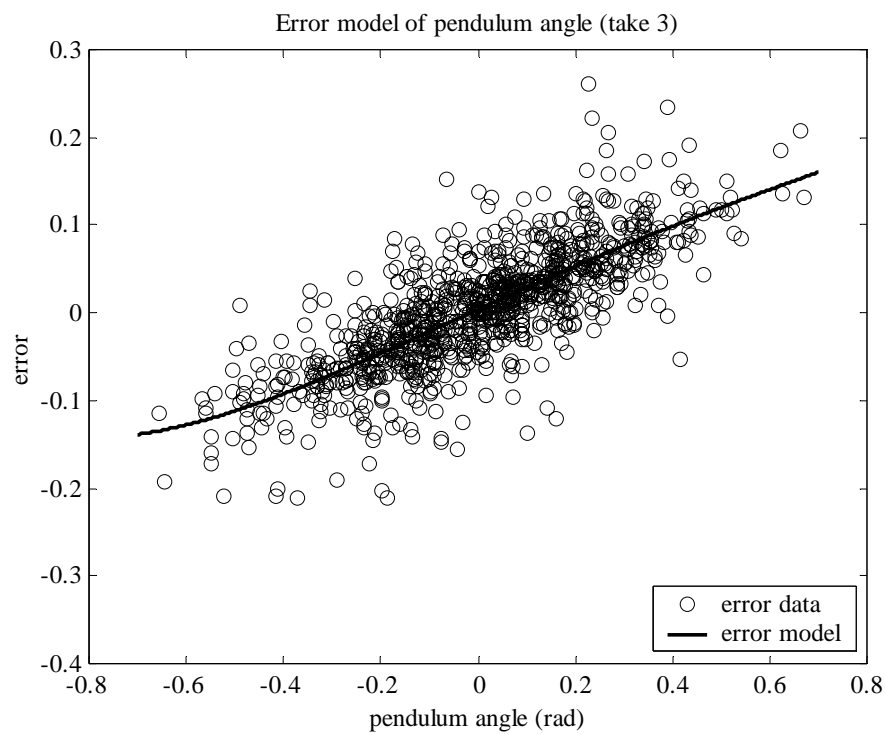
รูปที่ 6.31 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 2 หลังแก้ไข



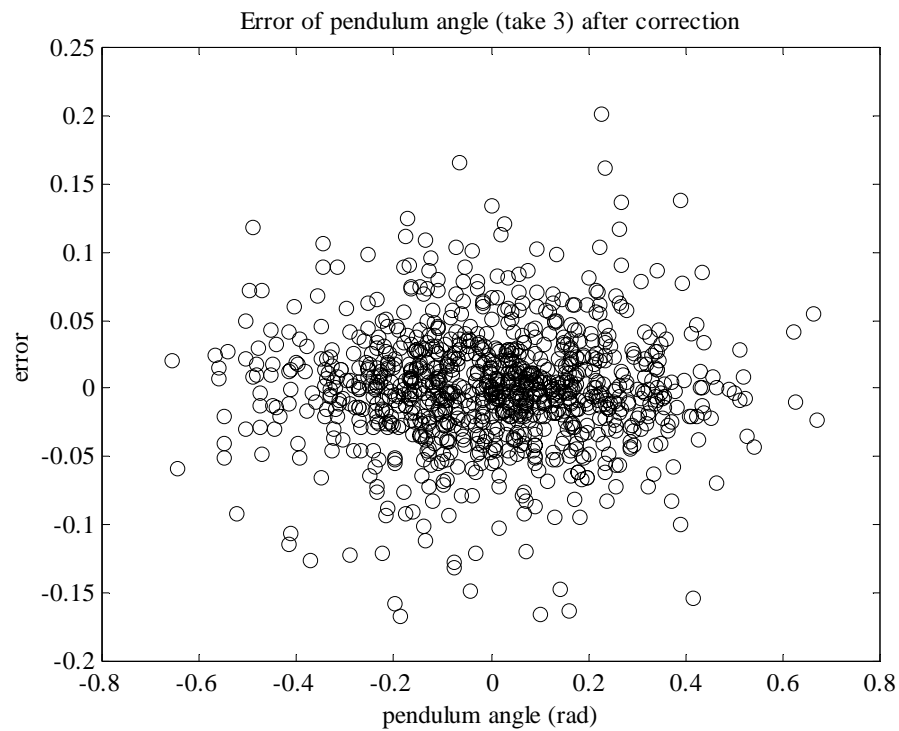
รูปที่ 6.32 ตำแหน่งของรถจากการทดสอบตอนที่ 2 ก่อนและหลังแก้ไข



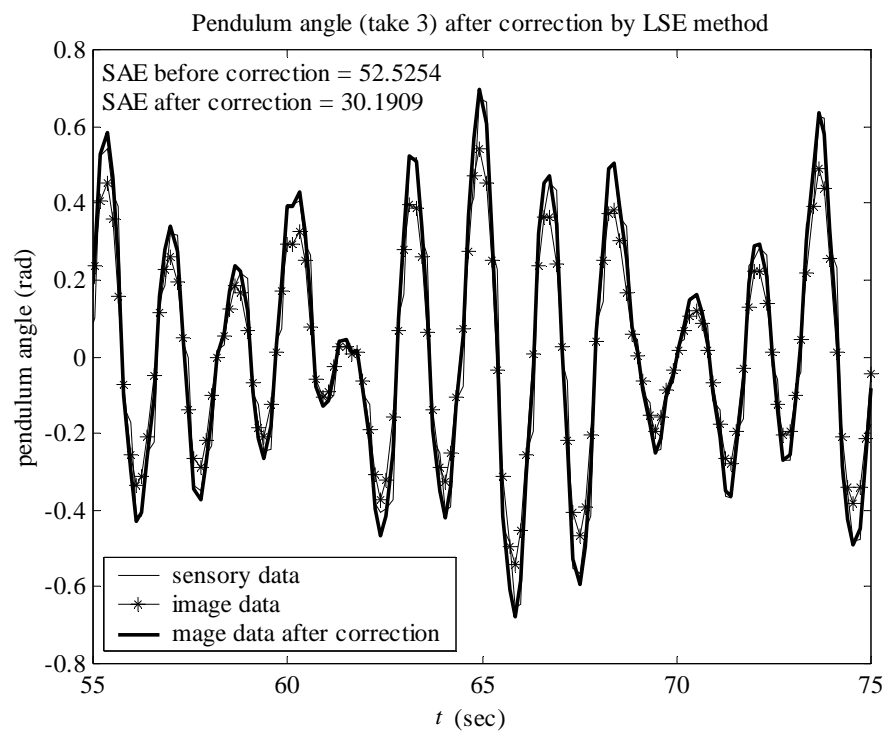
รูปที่ 6.33 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 3 ก่อนแก้ไข



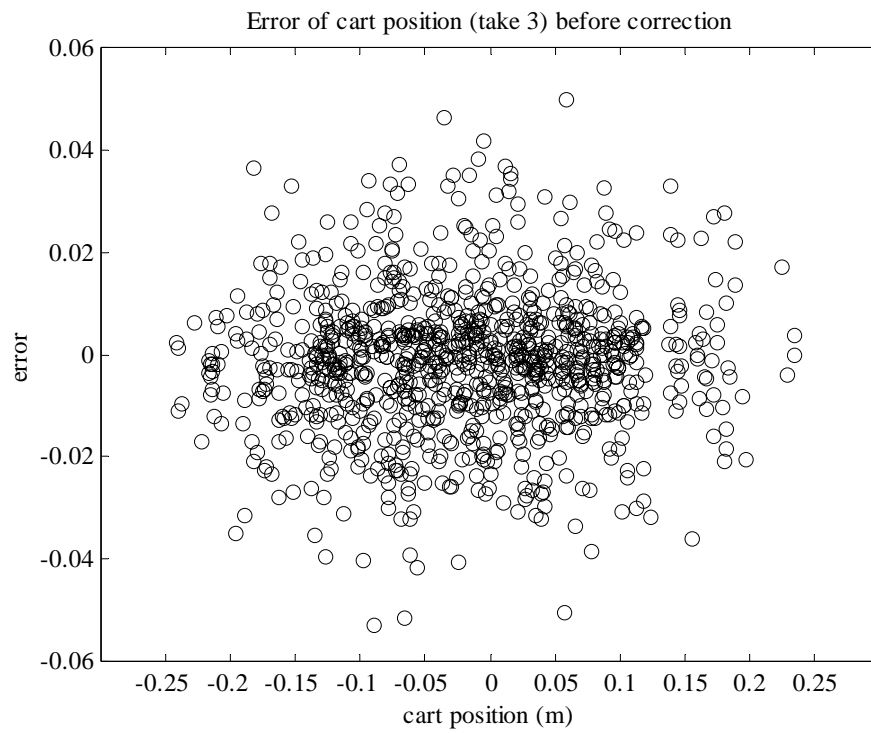
รูปที่ 6.34 แบบจำลองความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 3



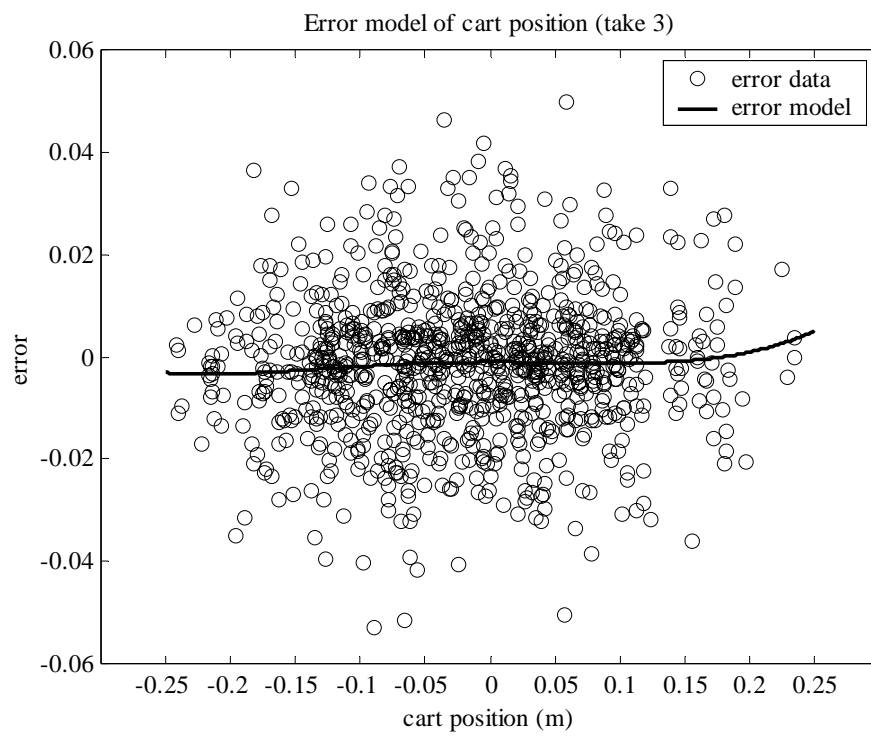
รูปที่ 6.35 ความคลาดเคลื่อนของตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 3 หลังแก้ไข



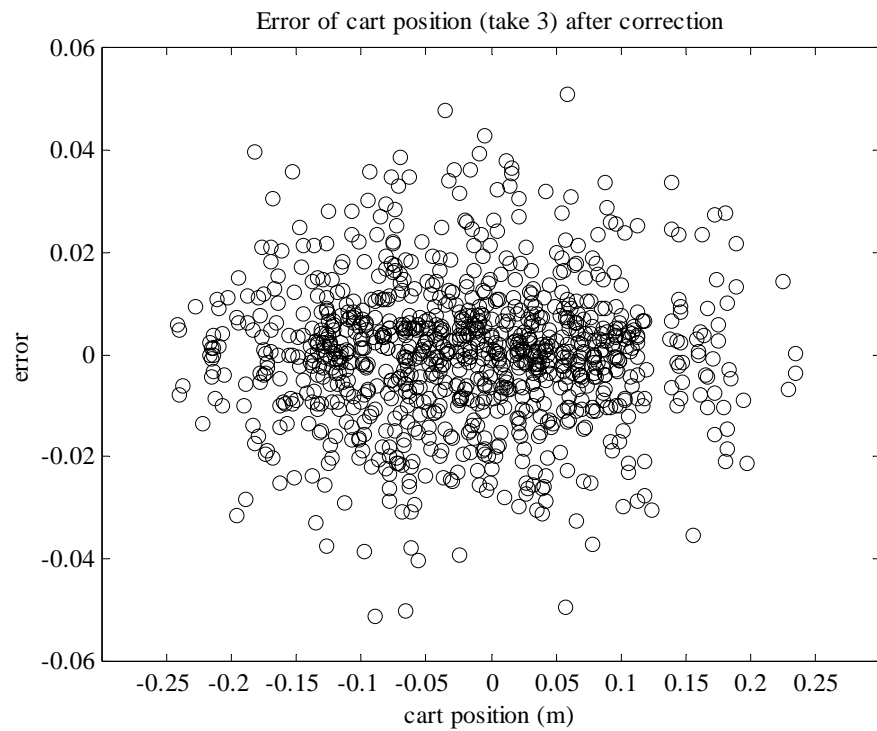
รูปที่ 6.36 ตำแหน่งเชิงมุมของลูกตุ้มจากการทดสอบตอนที่ 3 ก่อนและหลังแก้ไข



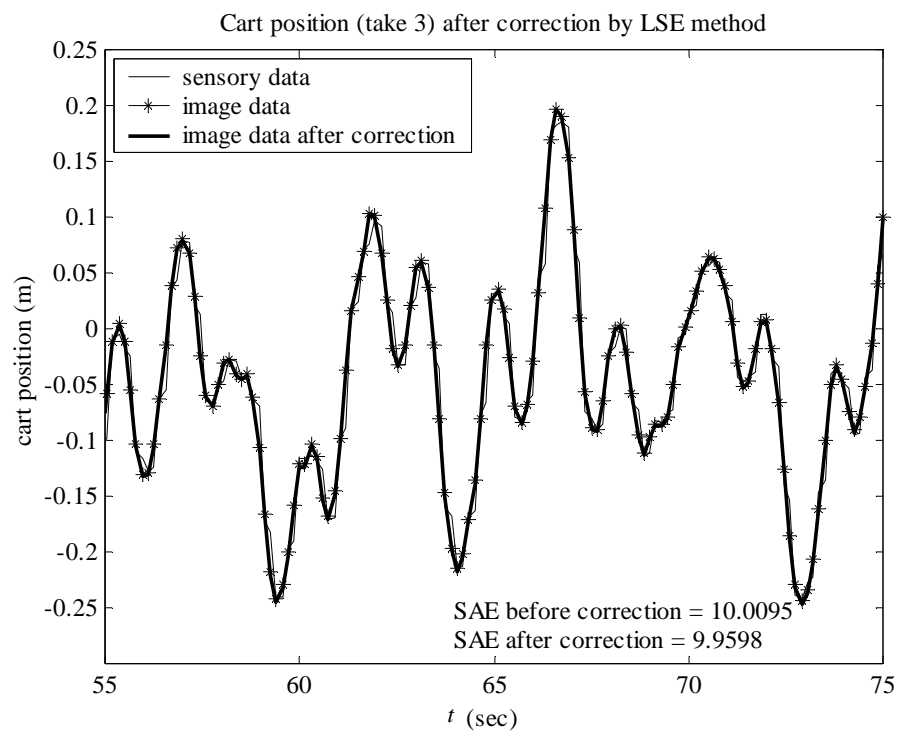
รูปที่ 6.37 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 3 ก่อนแก้ไข



รูปที่ 6.38 แบบจำลองความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 3



รูปที่ 6.39 ความคลาดเคลื่อนของตำแหน่งของรถจากการทดสอบตอนที่ 3 หลังแก้ไข



รูปที่ 6.40 ตำแหน่งของรถจากการทดสอบตอนที่ 3 ก่อนและหลังแก้ไข

$$\Gamma = 5.6419 \times 10^{-2} S^4 - 7.7038 \times 10^{-2} S^3 - 1.3319 \times 10^{-2} S^2 + 2.5129 \times 10^{-1} S + 3.6000 \times 10^{-3} \quad (6-14)$$

6. ตำแหน่งของรถที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 3 มีค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้เท่ากับ 10.0095 เมื่อทำการแก้ไขโดยใช้แบบจำลองความคลาดเคลื่อนอันดับ 4 ดังสมการที่ (6-15) ค่า SAE เมื่อเทียบกับข้อมูลจากตัวตรวจรู้ลดลงเหลือเท่ากับ 9.9598 ความคลาดเคลื่อนของตำแหน่งของรถที่ได้จากการประมวลผลภาพก่อนได้รับการแก้ไขแสดงดังรูปที่ 6.37 แบบจำลองความคลาดเคลื่อนอันดับ 4 ดังสมการที่ (6-15) แสดงดังรูปที่ 6.38 ความคลาดเคลื่อนของตำแหน่งของรถที่ได้จากการประมวลผลภาพภายหลังการแก้ไขแสดงดังรูปที่ 6.39 และตำแหน่งของรถที่ได้จากตัวตรวจรู้ และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อน แสดงดังรูปที่ 6.40

$$\Gamma = 1.6157 S^4 + 2.4871 \times 10^{-1} S^3 - 6.9427 \times 10^{-2} S^2 + 7.0822 \times 10^{-4} S - 1.0469 \times 10^{-3} \quad (6-15)$$

ในกรณีตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 1 ถึงตอนที่ 3 ค่าความคลาดเคลื่อนก่อนได้รับการแก้ไขดังแสดงในรูปที่ 6.17 6.25 และ 6.33 ตามลำดับ บ่งบอกว่าเมื่อตำแหน่งเชิงมุมของลูกตุ้มมีค่าเพิ่มมากขึ้น ค่าความคลาดเคลื่อนจะเพิ่มมากขึ้นตาม แสดงให้เห็นว่าเลนส์ที่ใช้สำหรับการบันทึกภาพในกรณีที่วัตถุเคลื่อนที่ในแนวหมุน ปรากฏความเพี้ยน (distortion) มากพอสมควร เมื่อทำการคำนวณหาแบบจำลองความคลาดเคลื่อนของข้อมูลตำแหน่งเชิงมุมของลูกตุ้มพบว่า ข้อมูลจากการทดสอบตอนที่ 1 และตอนที่ 2 ได้แบบจำลองอันดับ 3 ขณะที่ข้อมูลจากการทดสอบตอนที่ 3 ได้แบบจำลองอันดับ 4 ดังแสดงในรูปที่ 6.18 6.26 และ 6.34 ตามลำดับ ความคลาดเคลื่อนภายหลังการแก้ไขดังแสดงในรูปที่ 6.19 6.27 และ 6.35 ตามลำดับ บ่งบอกว่ามีค่าลดลงจากเดิมมาก และค่อนข้างคงที่แม้ว่าตำแหน่งเชิงมุมของลูกตุ้มจะมีค่าเพิ่มมากขึ้นก็ตาม ภาพตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้ และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อนจากการทดสอบตอนที่ 1 ถึงตอนที่ 3 ดังแสดงในรูปที่ 6.20 6.28 และ 6.36 ตามลำดับ ค่าความคลาดเคลื่อนภายหลังการแก้ไขมีค่าลดลงอย่างน่าพอใจยิ่ง

ในกรณีตำแหน่งของรถที่ได้จากการประมวลผลภาพจากการทดสอบตอนที่ 1 ถึงตอนที่ 3 ความคลาดเคลื่อนก่อนได้รับการแก้ไขดังแสดงในรูปที่ 6.21 6.29 และ 6.37 ตามลำดับ บ่งบอกว่า มี

ความคลาดเคลื่อนเล็กน้อยและค่อนข้างคงที่ ทั้งนี้อาจเป็นเพราะเลนส์ที่ใช้สำหรับการบันทึกภาพในกรณีที่วัตถุเคลื่อนที่ในแนวเส้นตรงตามแนวราบ มีความเพี้ยนไม่มากนัก เมื่อทำการคำนวณหาแบบจำลองความคลาดเคลื่อนของข้อมูลตำแหน่งของรถพบว่า ข้อมูลจากการทดสอบตอนที่ 1 ได้แบบจำลองอันดับ 2 ข้อมูลจากการทดสอบตอนที่ 2 ได้แบบจำลองอันดับ 5 และข้อมูลจากการทดสอบตอนที่ 3 ได้แบบจำลองอันดับ 4 ดังแสดงในรูปที่ 6.22 6.30 และ 6.38 ตามลำดับ ความคลาดเคลื่อนภายหลังการแก้ไขดังแสดงในรูปที่ 6.23 6.31 และ 6.39 ตามลำดับ บ่งบอกว่ามีค่าลดลงจากเดิมเล็กน้อย และค่อนข้างคงที่แม้ว่าตำแหน่งของรถจะมีค่าเพิ่มมากขึ้นก็ตาม ภาพตำแหน่งของรถที่ได้จากตัวตรวจรู้ และจากการประมวลผลภาพ ทั้งก่อนและหลังการแก้ไขความคลาดเคลื่อนจากการทดสอบตอนที่ 1 ถึงตอนที่ 3 ดังแสดงในรูปที่ 6.24 6.32 และ 6.40 ตามลำดับ ระบุว่าไม่มีความแตกต่างกันมากนัก ทั้งนี้เพราะความคลาดเคลื่อนของตำแหน่งของรถก่อนได้รับการแก้ไขมีค่าน้อยมากอยู่แล้ว

6.5 สรุป

ในบทที่ 6 นี้ ได้กล่าวถึงการทดสอบระบบ CPP เพื่อบันทึกข้อมูลอินพุตและเอาต์พุตของระบบโดยอาศัยตัวตรวจรู้ และการประมวลผลภาพ การทดสอบกำหนดให้สัญญาณอินพุตอ้างอิงเป็นสัญญาณสุ่ม การเคลื่อนที่ของระบบได้รับการบันทึกด้วยกล้องวีดิโอดิจิทัล ข้อมูลภาพได้รับการอ่านและส่งเข้าสู่คอมพิวเตอร์เพื่อทำการแปลงเป็นไฟล์วีดิโอ และทำการแยกแยะลำดับภาพเพื่อสกัดข้อมูลพลวัต โดยใช้เทคนิคการกำหนดจุดเปลี่ยนค่า เมื่อเปรียบเทียบกับข้อมูลที่ได้จากตัวตรวจรู้ ข้อมูลที่ได้จากการประมวลผลภาพมีความคลาดเคลื่อนพอสมควร เมื่อทำการแก้ไขความคลาดเคลื่อนดังกล่าวด้วยระเบียบวิธีกำลังสองน้อยที่สุด ข้อมูลที่ได้จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อนมีความถูกต้องสูงขึ้นอย่างน่าพอใจ ข้อมูลดังกล่าวจะได้รับการนำไปคำนวณเพื่อระบุเอกลักษณ์ระบบ CPP ดังรายละเอียดที่จะปรากฏในบทต่อไป

บทที่ 7

การระบุเอกลักษณ์ระบบลูกตุ้มนาฬิกาพิกานวกรถ

7.1 กล่าวนำ

ในบทที่ 7 นี้ จะกล่าวถึงการระบุเอกลักษณ์ระบบลูกตุ้มนาฬิกาพิกานวกรถ หรือระบบ CPP ระบบดังกล่าวเป็นตัวแทนของระบบพลวัตที่ใช้งานจริงในอุตสาหกรรม เช่น ก้ามปูกลยกของหนัก ในโรงงาน ที่ซึ่งก้ามปูนั้นจะห้อยลงมาจากเพดานโรงงาน และเคลื่อนที่ในระนาบ x, y เป็นต้น การระบุเอกลักษณ์ระบบ CPP แบ่งออกเป็นสองส่วนด้วยกันคือ (1) การระบุเอกลักษณ์ระบบ CPP เชิงเส้น เป็นการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม โดยใช้แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ทั้งสี่รูปแบบ ได้แก่แบบจำลอง ARX ARMAX OE และ BJ การดำเนินงานระบุเอกลักษณ์ระบบอาศัยโปรแกรม MATLAB และ System Identification Toolbox (2) การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น เป็นการระบุเอกลักษณ์ระบบโดยอาศัยการค้นหาแบบตามูเชิงปรับตัว แบบจำลองของระบบจะอยู่ในรูปของสมการอนุพันธ์ไม่เชิงเส้น การดำเนินงานระบุเอกลักษณ์ระบบอาศัยโปรแกรม C สำหรับข้อมูลเอาต์พุตของระบบ CPP ที่ใช้สำหรับการระบุเอกลักษณ์ระบบ คือตำแหน่งเชิงมุมของลูกตุ้มที่ได้รับการบันทึกด้วยตัวตรวจรู้ และการประมวลผลภาพ จากการทดสอบระบบในบทที่ 6 การระบุเอกลักษณ์ระบบ CPP ทั้งเชิงเส้นและไม่เชิงเส้น ที่ใช้ข้อมูลจากตัวตรวจรู้ และการประมวลผลภาพ ให้ผลเป็นที่น่าพอใจยิ่ง

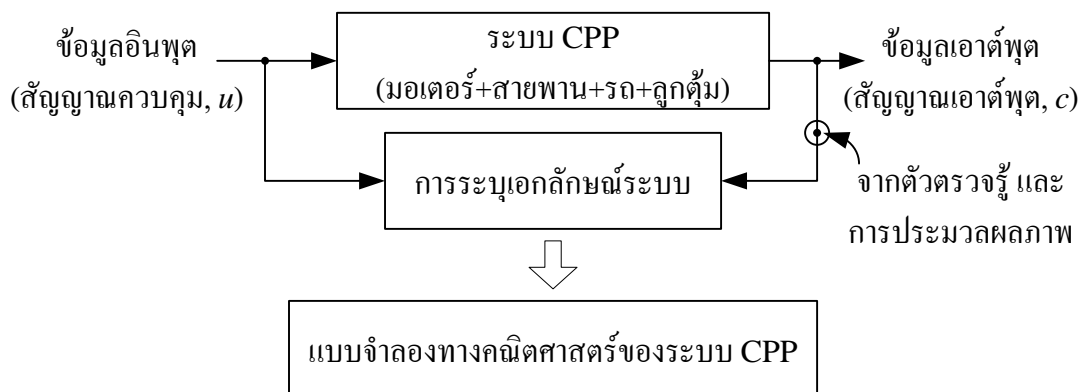
7.2 การระบุเอกลักษณ์ระบบ CPP เชิงเส้น

เมื่อพิจารณาระบบ CPP จากรูปที่ 6.1 ในบทที่ 6 พบว่าแบบจำลองทางคณิตศาสตร์เชิงเส้น ที่พัฒนาจากการวิเคราะห์ แสดงดังสมการที่ (7-1) ถึง (7-2) (รายละเอียดการพัฒนาแบบจำลอง ดังกล่าวแสดงในภาคผนวก ข.) แรง f (นิวตัน) ในสมการที่ (7-1) ถึง (7-2) จะเป็นฟังก์ชันของ สัญญาณอินพุตของมอเตอร์ u (โวลต์) ขณะที่ ϕ คือ มุมการแกว่งไกวของลูกตุ้ม (เรเดียน) y คือ ตำแหน่งของรถ (เมตร) M คือ มวลของรถ (กิโลกรัม) m คือ มวลของลูกตุ้ม (กิโลกรัม) l คือ ความยาวของก้านลูกตุ้ม (เมตร) และ g คือ ความเร่งเนื่องจากแรงโน้มถ่วง (เมตร/วินาที²)

การระบุเอกลักษณ์ระบบ CPP เชิงเส้น จะอาศัยการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม โดยใช้แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ทั้งสี่รูปแบบ คือแบบจำลอง ARX ARMAX OE และ BJ ในรูปของฟังก์ชันถ่ายโอน ซึ่งเป็นรูปลักษณะที่ใช้ประโยชน์อย่างกว้างขวาง ในวิทยาการระบบควบคุม เพราะสามารถประยุกต์กับระบบพลวัตเชิงเส้นใด ๆ ได้

$$\ddot{\phi} + \left[\frac{(M+m)g}{Ml} \right] \phi + \frac{f}{Ml} = 0 \quad (7-1)$$

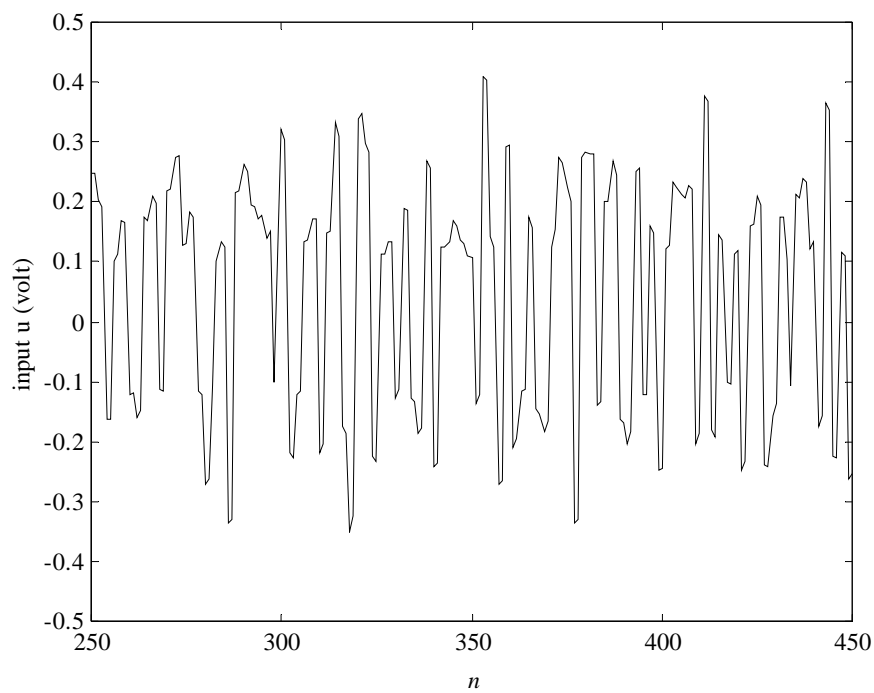
$$\ddot{y} - \left[\frac{mg}{M} \right] \phi - \frac{f}{M} = 0 \quad (7-2)$$



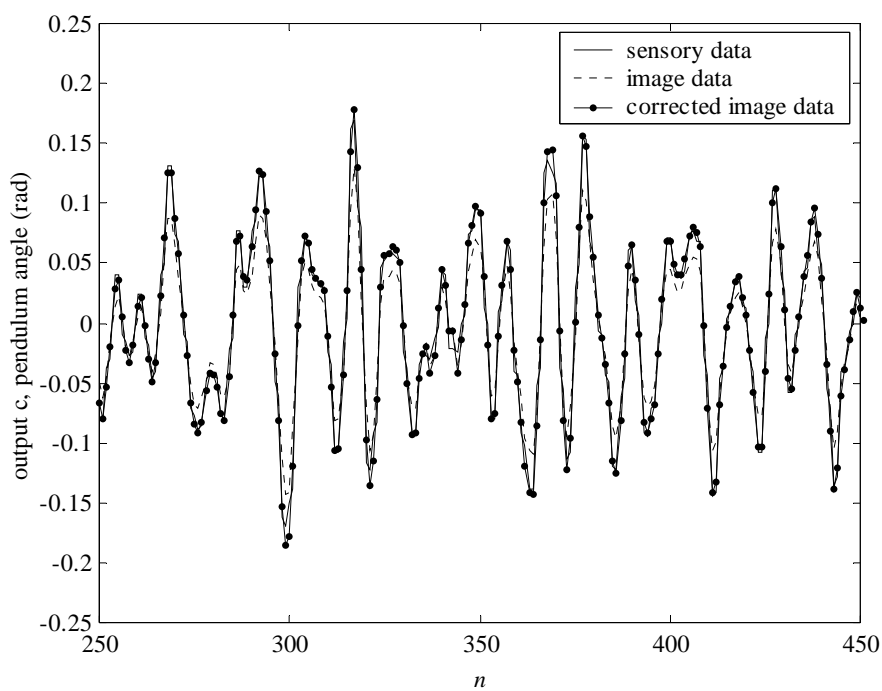
รูปที่ 7.1 การระบุเอกลักษณ์ระบบ CPP

จากรูปที่ 6.2 ในบทที่ 6 พบว่าการระบุเอกลักษณ์ระบบ CPP เป็นการระบุเอกลักษณ์ระบบย่อยโดยตรง ซึ่งอาจแสดงดังรูปที่ 7.1 ข้อมูลอินพุตและเอาต์พุตที่ใช้สำหรับการระบุเอกลักษณ์ระบบ คือสัญญาณควบคุม u และสัญญาณ c ตามลำดับ

จากผลการทดสอบระบบ CPP ในบทที่ 6 ซึ่งมีด้วยกันทั้งหมด 3 ตอน พิจารณาการทดสอบตอนที่ 1 ซึ่งลูกตุ้มแกว่งไกวด้วยมุม ± 0.2 เรเดียน เพื่อใช้ดำเนินการระบุเอกลักษณ์ระบบ CPP เชิงเส้น ข้อมูลจากการทดสอบในตอนที่ 1 ประกอบไปด้วยสัญญาณอินพุต u และสัญญาณเอาต์พุต c (ตำแหน่งเชิงมุมของลูกตุ้ม) ที่ได้จากตัวตรวจรู้ จากการประมวลผลภาพ และการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน โดยอาศัยข้อมูลจากตัวตรวจรู้เป็นข้อมูลอ้างอิง จำนวนข้อมูลของสัญญาณแต่ละสัญญาณเท่ากับ 1,000 ข้อมูล แต่จะนำมาใช้ประโยชน์ทั้งสิ้นจำนวน 900 ข้อมูล ข้อมูลส่วนต้นและส่วนท้ายของแต่ละสัญญาณจะถูกตัดทิ้ง เพราะเป็นช่วงเริ่มต้นและช่วงยุติการทดสอบระบบ ข้อมูลที่บันทึกได้ในช่วงดังกล่าวอาจมีความคลาดเคลื่อนสูง จากนั้นทำการแบ่งข้อมูลของแต่ละสัญญาณจากจำนวนทั้งสิ้น 900 ข้อมูลออกเป็นสองส่วน ข้อมูลส่วนแรกจำนวน 700 ข้อมูล เพื่อใช้ระบุเอกลักษณ์ระบบ ข้อมูลส่วนที่สองจำนวน 200 ข้อมูล เพื่อใช้ตรวจสอบความถูกต้องของแบบจำลอง ส่วนหนึ่งของข้อมูลอินพุตและเอาต์พุตได้รับการแสดงไว้ในรูปที่ 7.2 (ก) และ (ข) ตามลำดับ ทั้งนี้เพื่อความชัดเจน



(ก)



(ข)

รูปที่ 7.2 ข้อมูลอินพุตและเอาต์พุตสำหรับการระบุเอกลักษณ์ระบบ CPP เชิงเส้น

(ก) ข้อมูลอินพุต (ข) ข้อมูลเอาต์พุต (ตำแหน่งเชิงมุมของลูกตุ้ม)

การระบุเอกลักษณ์ระบบ CPP เชิงเส้น จะแบ่งออกเป็น 5 ส่วนด้วยกันคือ (1) การระบุเอกลักษณ์ระบบเพื่อพิจารณาอันดับที่เหมาะสมของแบบจำลอง (2) การระบุเอกลักษณ์ระบบโดยอาศัยข้อมูลจากตัวตรวจรู้ (3) การระบุเอกลักษณ์ระบบโดยอาศัยข้อมูลจากการประมวลผลภาพ (4) การระบุเอกลักษณ์ระบบโดยอาศัยข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน และ (5) การเปรียบเทียบผลการระบุเอกลักษณ์ระบบ ดังรายละเอียดต่อไปนี้

7.2.1 การระบุเอกลักษณ์ระบบ CPP เชิงเส้นเพื่อพิจารณาอันดับที่เหมาะสม

ในขั้นตอนนี้จะเป็นการระบุเอกลักษณ์ระบบ CPP เชิงเส้น เพื่อพิจารณาอันดับที่เหมาะสมของแบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ โดยใช้แบบจำลอง BJ ซึ่งเป็นแบบจำลองที่สมบูรณ์แบบที่สุด ดังได้รับการอธิบายไว้ในบทที่ 3 ทำการระบุเอกลักษณ์ระบบ CPP เชิงเส้น อันดับ 2 ถึง 20 ในรูปฟังก์ชันถ่ายโอนดังสมการที่ (7-3) การระบุเอกลักษณ์ระบบในขั้นตอนนี้จะอาศัยข้อมูลที่ได้จากตัวตรวจรู้เท่านั้น ดำเนินงานระบุเอกลักษณ์ระบบโดยอาศัยโปรแกรม MATLAB และ System Identification Toolbox ดังรายละเอียดในภาคผนวก ง. การพิจารณาอันดับที่เหมาะสมของแบบจำลองจะใช้เกณฑ์ค่าความคลาดเคลื่อนระหว่างข้อมูลเอาต์พุตที่ได้จากการทดสอบ (ϕ) และข้อมูลเอาต์พุตที่ได้จากแบบจำลอง ($\hat{\phi}$) ในรูปของผลรวมของค่าความคลาดเคลื่อนสัมบูรณ์ (sum absolute error, SAE) ผลรวมค่าความคลาดเคลื่อนกำลังสอง (sum square error, SSE) ค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง (mean square error, MSE) และ ค่ารากที่สองของค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง (root mean square error, RMSE) ดังสมการที่ (7-4) ถึง (7-7) ตามลำดับ

$$G(s) = \frac{b_n s^n + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}, \quad n = 2, \dots, 20 \quad (7-3)$$

$$SAE = \sum_{i=1}^n \|\phi_i - \hat{\phi}_i\| = \sum_{i=1}^n \sqrt{(\phi_i - \hat{\phi}_i)^2} \quad (7-4)$$

$$SSE = \sum_{i=1}^n (\phi_i - \hat{\phi}_i)^2 \quad (7-5)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (\phi_i - \hat{\phi}_i)^2 \quad (7-6)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\phi_i - \hat{\phi}_i)^2} \quad (7-7)$$

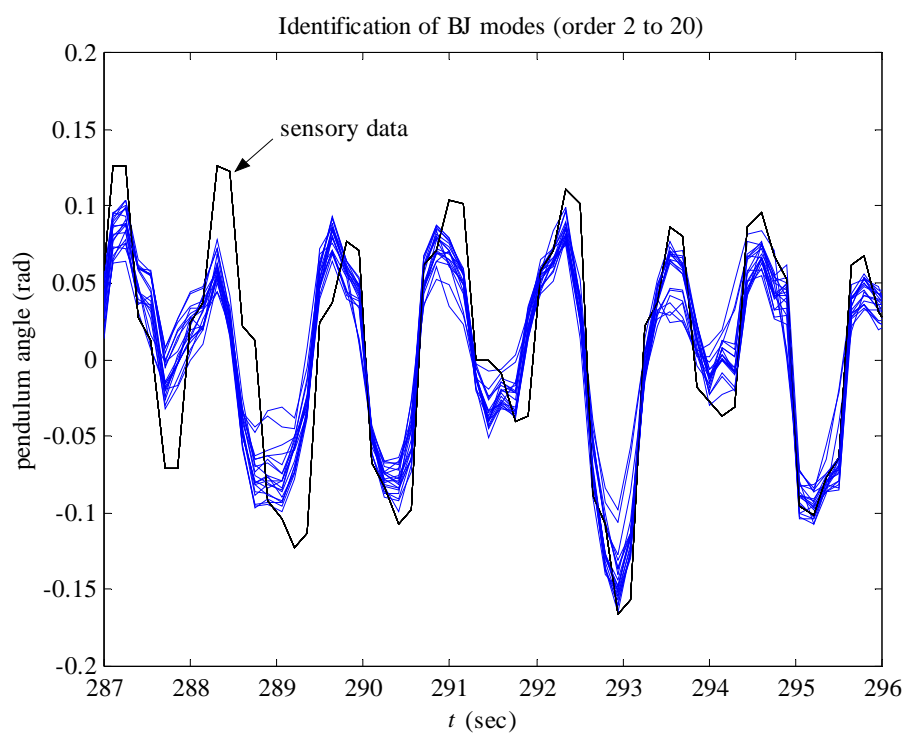
จากข้อมูลของแต่ละสัญญาณที่ได้รับการจัดสรรไว้เพื่อใช้ระบุเอกลักษณ์ระบบ จำนวนทั้งสิ้น 700 ข้อมูล จะถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลของแต่ละสัญญาณมีลักษณะเป็นรายคาบที่ยาวนานมาก ซึ่งเป็นคุณสมบัติข้อหนึ่งของสัญญาณสุ่มเทียม (Eykhoff, 1974; Ljung, 1987) ดังนั้นข้อมูลของแต่ละสัญญาณที่ใช้ระบุเอกลักษณ์ระบบจะมีจำนวนทั้งสิ้น 2,100 ข้อมูล ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้น เพื่อพิจารณาอันดับที่เหมาะสม แสดงในรูปที่ 7.3 และ 7.4 ตามลำดับ ขณะที่ค่าความคลาดเคลื่อนของแบบจำลองอันดับ 2 ถึง 20 แสดงในตารางที่ 7.1 และรูปที่ 7.5 ตามลำดับ

ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง ดังแสดงในรูปที่ 7.3 และ 7.4 สามารถสังเกตได้ถึงความสอดคล้องกันอย่างมากของกราฟทั้งหมด จนทำให้ค่อนข้างยากที่จะพิจารณาอันดับที่เหมาะสมของแบบจำลอง อย่างไรก็ตาม เมื่อพิจารณาค่าความคลาดเคลื่อนของแบบจำลองอันดับ 2 ถึง 20 จากตารางที่ 7.1 และรูปที่ 7.5 จะพบว่า SAE SSE MSE และ RMSE มีความสอดคล้องกัน กล่าวคือ เมื่ออันดับของแบบจำลองเพิ่มขึ้นจากอันดับ 2 ถึงอันดับ 10 จะสังเกตได้ว่าค่าความคลาดเคลื่อนลดลงอย่างชัดเจน แต่เมื่ออันดับของแบบจำลองเพิ่มขึ้นจากอันดับ 10 ถึงอันดับ 20 พบว่าค่าความคลาดเคลื่อนจะค่อนข้างคงที่ ดังนั้นจึงพิจารณาแบบจำลองอันดับ 10 เป็นแบบจำลองที่มีความเหมาะสมสำหรับระบบ CPP เชิงเส้น ในการนำไปใช้ประโยชน์เพื่อการระบุเอกลักษณ์ระบบในขั้นต่อไป

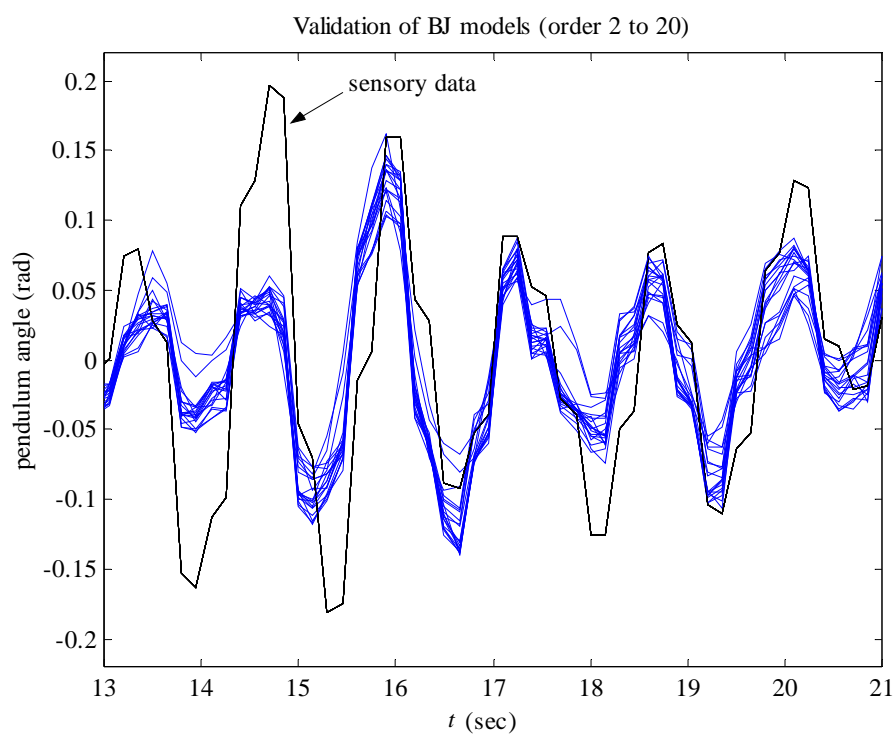
7.2.2 การระบุเอกลักษณ์ระบบ CPP เชิงเส้นโดยอาศัยข้อมูลจากตัวตรวจรู้

ในขั้นตอนนี้จะเป็นการระบุเอกลักษณ์ระบบ CPP เชิงเส้นด้วยวิธีการดั้งเดิม โดยใช้แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ทั้งสี่รูปแบบ ได้แก่แบบจำลอง ARX ARMAX OE และ BJ ในรูปของฟังก์ชันถ่ายโอนอันดับ 10 ดังสมการที่ (7-8) การระบุเอกลักษณ์ระบบในขั้นตอนนี้จะอาศัยข้อมูลที่ได้จากตัวตรวจรู้ ดำเนินงานระบุเอกลักษณ์ระบบโดยอาศัยโปรแกรม MATLAB และ System Identification Toolbox ดังรายละเอียดในภาคผนวก ง. การพิจารณาความเหมาะสมของแบบจำลองจะใช้เกณฑ์ค่าความคลาดเคลื่อนระหว่างข้อมูลเอาต์พุตที่ได้จากตัวตรวจรู้ (ϕ) และข้อมูลเอาต์พุตที่ได้จากแบบจำลอง ($\hat{\phi}$) ในรูป SAE SSE MSE และ RMSE ดังสมการที่ (7-4) ถึง (7-7) ตามลำดับ

$$G(s) = \frac{N(s)}{D(s)} = \frac{b_{10}s^{10} + \dots + b_1s + b_0}{a_{10}s^{10} + \dots + a_1s + a_0} \quad (7-8)$$



รูปที่ 7.3 ผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้นอันดับ 2 ถึง 20

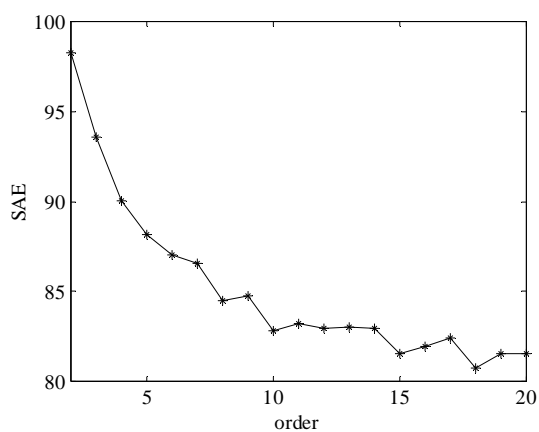


รูปที่ 7.4 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้นอันดับ 2 ถึง 20

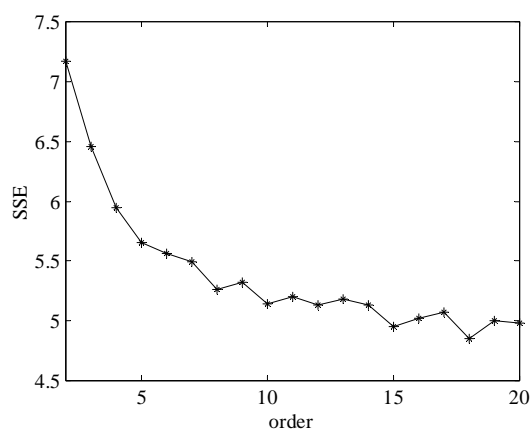
ตารางที่ 7.1 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เชิงเส้นอันดับ 2 ถึง 20

อันดับของ แบบจำลอง	ค่าความคลาดเคลื่อนจากการระบุเอกลักษณ์ระบบ CPP เชิงเส้น			
	SAE	SSE	MSE	RMSE
2	98.2772	7.1651	0.0034	0.0590
3	93.5566	6.4596	0.0030	0.0559
4	90.0225	5.9419	0.0028	0.0535
5	88.1520	5.6578	0.0026	0.0521
6	87.0427	5.5628	0.0026	0.0516
7	86.5808	5.4961	0.0026	0.0513
8	84.4732	5.2615	0.0025	0.0501
9	84.7770	5.3203	0.0025	0.0504
10	82.8163	5.1447	0.0024	0.0495
11	83.1970	5.2034	0.0024	0.0498
12	82.9194	5.1353	0.0024	0.0495
13	83.0266	5.1830	0.0024	0.0497
14	82.9196	5.1352	0.0024	0.0496
15	81.5160	4.9505	0.0023	0.0485
16	81.9612	5.0245	0.0023	0.0489
17	82.3767	5.0705	0.0024	0.0491
18	80.7105	4.8545	0.0023	0.0480
19	81.5267	4.9986	0.0023	0.0488
20	81.5240	4.9773	0.0023	0.0486

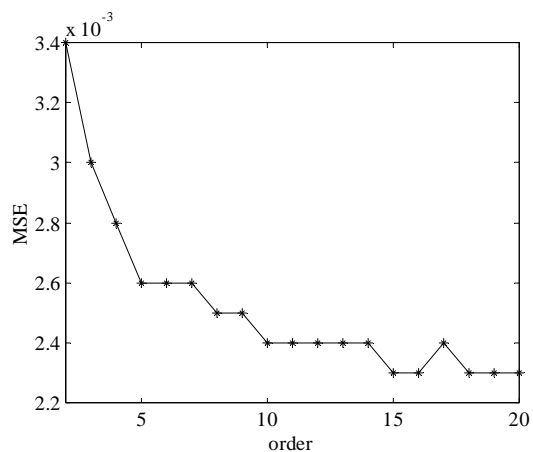
ฟังก์ชันถ่ายโอนในสมการที่ (7-8) เป็นแบบจำลองที่ได้จากการระบุเอกลักษณ์ระบบ โดยใช้แบบจำลอง ARX ARMAX OE และ BJ ของระบบ CPP เชิงเส้น ซึ่ง $N(s)$ คือ พหุนามเศษของแบบจำลอง และ $D(s)$ คือ พหุนามส่วนของแบบจำลอง โดยใช้ข้อมูลที่ได้รับการจัดสรรแบบจำลอง ARX ARMAX OE และ BJ ของระบบ CPP เชิงเส้นที่ได้จากการระบุเอกลักษณ์ระบบโดยใช้ข้อมูลจากตัวตรวจรู้ แสดงดังสมการที่ (7-9) ถึง (7-12) ตามลำดับ ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง แสดงในรูปที่ 7.6 และ 7.7 ตามลำดับ ขณะที่ความคลาดเคลื่อนจากการระบุเอกลักษณ์ระบบ แสดงในตารางที่ 7.2



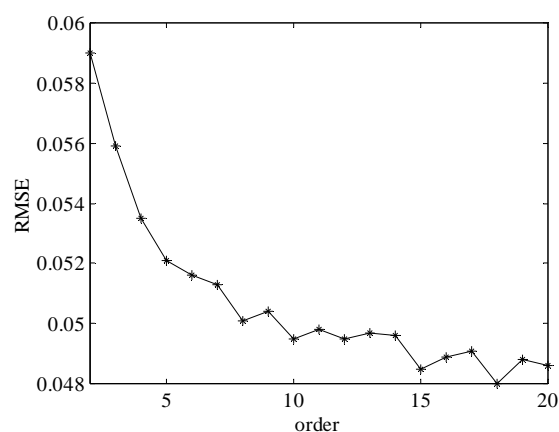
(ก)



(ข)



(ค)



(ง)

รูปที่ 7.5 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เชิงเส้นอันดับ 2 ถึง 20

(ก) SAE (ข) SSE (ค) MSE (ง) RMSE

ตารางที่ 7.2 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากตัวตรวจรู้

แบบจำลอง	ค่าความคลาดเคลื่อนจากการระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากตัวตรวจรู้			
	SAE	SSE	MSE	RMSE
ARX	89.5763	5.9105	0.0028	0.0531
ARMAX	86.3952	5.5303	0.0026	0.0513
OE	84.3785	5.2587	0.0025	0.0500
BJ	82.8163	5.1447	0.0024	0.0495

- แบบจำลอง ARX (โดยใช้ข้อมูลจากตัวตรวจรู้)

$$\begin{aligned}
 N(s) &= -0.06331s^{10} + 4.312s^9 + 41s^8 - 1643s^7 \\
 &\quad - 4.463 \times 10^4 s^6 - 7.067e005s^5 + 3.196 \times 10^6 s^4 \\
 &\quad + 1.291 \times 10^8 s^3 + 6.961 \times 10^8 s^2 + 4.308 \times 10^9 s - 3.716 \times 10^9 \\
 D(s) &= s^{10} + 68.78s^9 + 4117s^8 + 1.523 \times 10^5 s^7 \\
 &\quad + 3.509 \times 10^6 s^6 + 5.161 \times 10^7 s^5 + 4.246 \times 10^8 s^4 \\
 &\quad + 2.311 \times 10^9 s^3 + 8.586 \times 10^9 s^2 + 2.031 \times 10^{10} s + 3.538 \times 10^{10}
 \end{aligned} \tag{7-9}$$

- แบบจำลอง ARMAX (โดยใช้ข้อมูลจากตัวตรวจรู้)

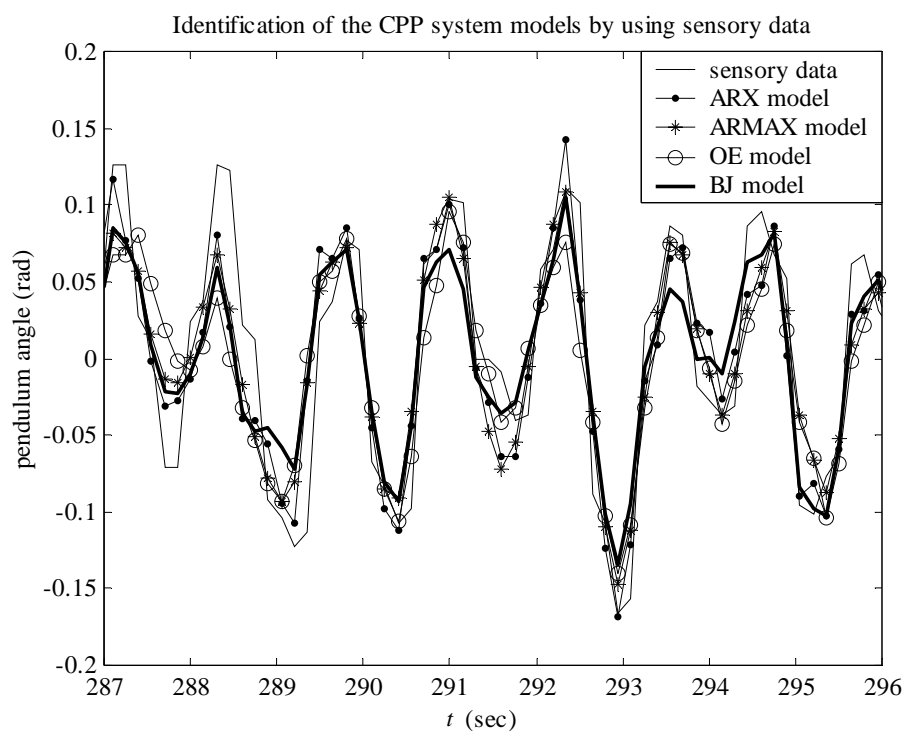
$$\begin{aligned}
 N(s) &= -0.01788s^{10} - 2.427s^9 + 689.6s^8 + 1505s^7 \\
 &\quad - 3.505 \times 10^5 s^6 + 1.721 \times 10^6 s^5 - 9.859 \times 10^7 s^4 \\
 &\quad + 5.512 \times 10^8 s^3 + 1.076 \times 10^{10} s^2 + 2.925 \times 10^{10} s - 1.5 \times 10^{10} \\
 D(s) &= s^{10} + 81.92s^9 + 1.343 \times 10^4 s^8 + 3.008 \times 10^5 s^7 \\
 &\quad + 2.147 \times 10^7 s^6 + 1.463 \times 10^8 s^5 + 5.141 \times 10^9 s^4 \\
 &\quad + 1.673 \times 10^{10} s^3 + 1.099 \times 10^{10} s^2 + 1.693 \times 10^{11} s + 3.606 \times 10^{11}
 \end{aligned} \tag{7-10}$$

- แบบจำลอง OE (โดยใช้ข้อมูลจากตัวตรวจรู้)

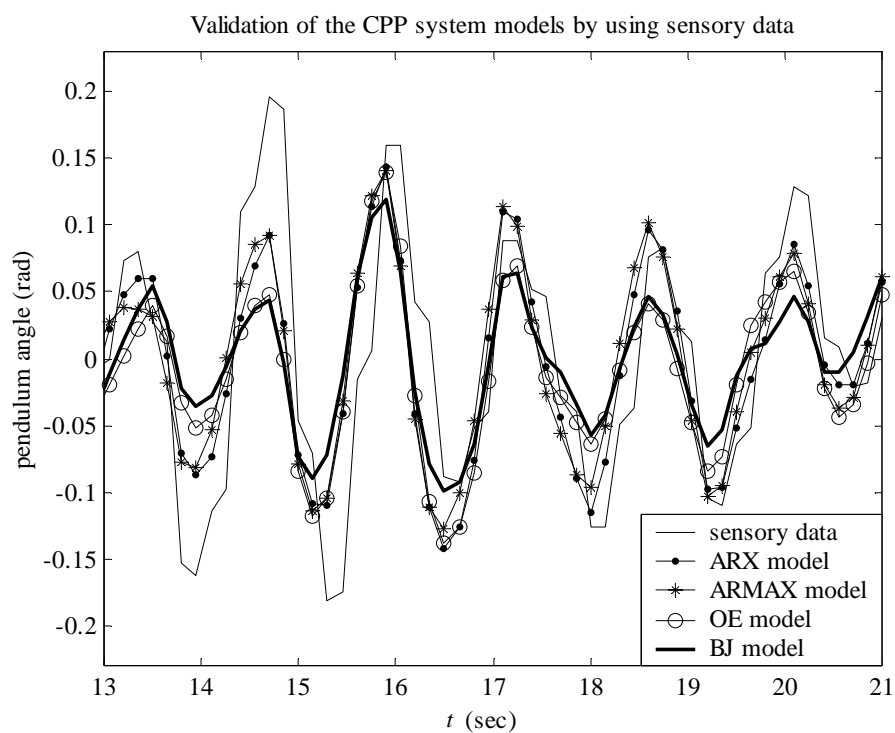
$$\begin{aligned}
 N(s) &= -0.0278s^{10} + 17.78s^9 - 157.9s^8 + 4434s^7 \\
 &\quad - 1.374 \times 10^5 s^6 + 8.046 \times 10^5 s^5 - 5.42 \times 10^6 s^4 \\
 &\quad + 8.611 \times 10^7 s^3 - 9.263 \times 10^7 s^2 + 8.338 \times 10^8 s - 8.72 \times 10^8 \\
 D(s) &= s^{10} + 74.64s^9 + 4528s^8 + 1.477 \times 10^5 s^7 \\
 &\quad + 2.475 \times 10^6 s^6 + 2.273 \times 10^7 s^5 + 1.269 \times 10^8 s^4 \\
 &\quad + 5.177 \times 10^8 s^3 + 1.575 \times 10^9 s^2 + 3.052 \times 10^9 s + 5.326 \times 10^9
 \end{aligned} \tag{7-11}$$

- แบบจำลอง BJ (โดยใช้ข้อมูลจากตัวตรวจรู้)

$$\begin{aligned}
 N(s) &= -0.05101s^{10} - 22.34s^9 + 1452s^8 - 1.515 \times 10^4 s^7 \\
 &\quad - 6.373 \times 10^4 s^6 - 3.401 \times 10^7 s^5 + 2.698 \times 10^8 s^4 \\
 &\quad + 1.611 \times 10^9 s^3 + 7.649 \times 10^9 s^2 + 7.53 \times 10^{10} s - 2.975 \times 10^{10} \\
 D(s) &= s^{10} + 216s^9 + 2.182 \times 10^4 s^8 + 1.322 \times 10^6 s^7 \\
 &\quad + 4.644 \times 10^7 s^6 + 1.096 \times 10^9 s^5 + 4.693 \times 10^9 s^4 \\
 &\quad + 4.271 \times 10^{10} s^3 + 1.167 \times 10^{11} s^2 + 3.405 \times 10^{11} s + 6.218 \times 10^{11}
 \end{aligned} \tag{7-12}$$



รูปที่ 7.6 ผลการระบุเอกลักษณ์ระบบ CPP เริงเส้น โดยใช้ข้อมูลจากตัวตรวจรู้



รูปที่ 7.7 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เริงเส้น โดยใช้ข้อมูลจากตัวตรวจรู้

- แบบจำลอง ARX (โดยใช้ข้อมูลจากการประมวลผลภาพ)

$$\begin{aligned}
 N(s) &= -0.03544s^{10} - 3.009s^9 - 79.23s^8 - 2214s^7 \\
 &\quad - 1.127e004s^6 + 6.778 \times 10^4 s^5 + 3.597 \times 10^6 s^4 \\
 &\quad + 6.578 \times 10^7 s^3 + 2.072 \times 10^8 s^2 + 1.611 \times 10^9 s - 1.762 \times 10^9 \\
 D(s) &= s^{10} + 104.4s^9 + 4935s^8 + 1.462 \times 10^5 s^7 \\
 &\quad + 2.276 \times 10^6 s^6 + 2.892 \times 10^7 s^5 + 1.976 \times 10^8 s^4 \\
 &\quad + 1.078 \times 10^9 s^3 + 3.704 \times 10^9 s^2 + 9.025 \times 10^9 s + 1.46 \times 10^{10}
 \end{aligned} \tag{7-13}$$

- แบบจำลอง ARMAX (โดยใช้ข้อมูลจากการประมวลผลภาพ)

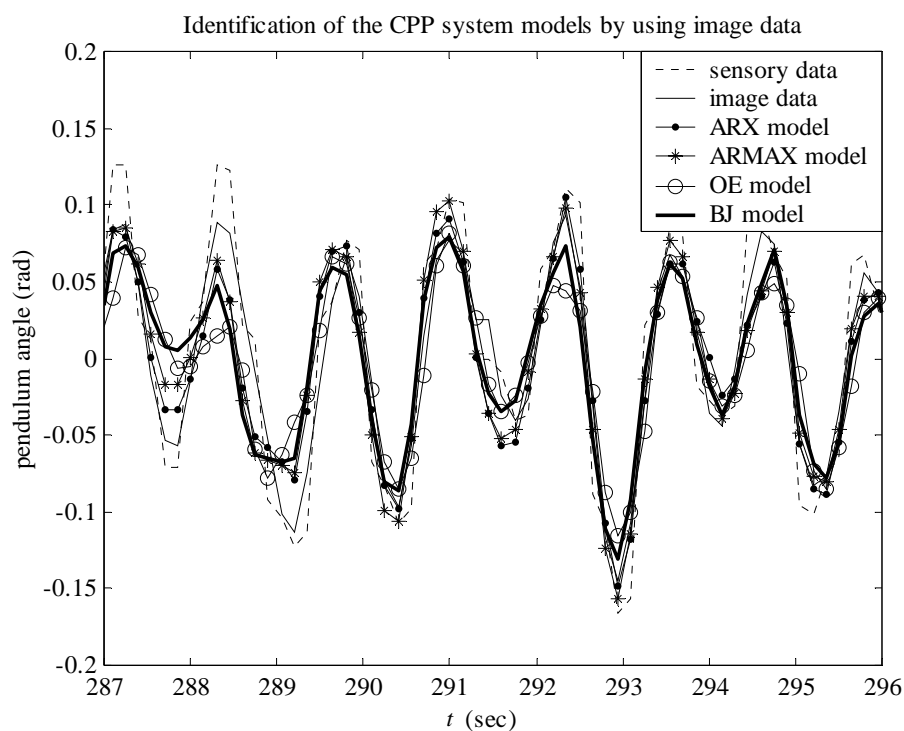
$$\begin{aligned}
 N(s) &= -0.0371s^{10} - 0.3079s^9 - 33.05s^8 + 15.48s^7 \\
 &\quad - 1756s^6 + 8.343 \times 10^4 s^5 + 3.914 \times 10^5 s^4 \\
 &\quad + 1.781 \times 10^6 s^3 + 3.357 \times 10^6 s^2 + 5.252 \times 10^6 s - 1.109 \times 10^7 \\
 D(s) &= s^{10} + 37.3s^9 + 1492s^8 + 2.527 \times 10^4 s^7 \\
 &\quad + 3.051 \times 10^5 s^6 + 1.275 \times 10^6 s^5 + 8.265 \times 10^6 s^4 \\
 &\quad + 1.759 \times 10^7 s^3 + 6.495 \times 10^7 s^2 + 7.011 \times 10^7 s + 1.125 \times 10^8
 \end{aligned} \tag{7-14}$$

- แบบจำลอง OE (โดยใช้ข้อมูลจากการประมวลผลภาพ)

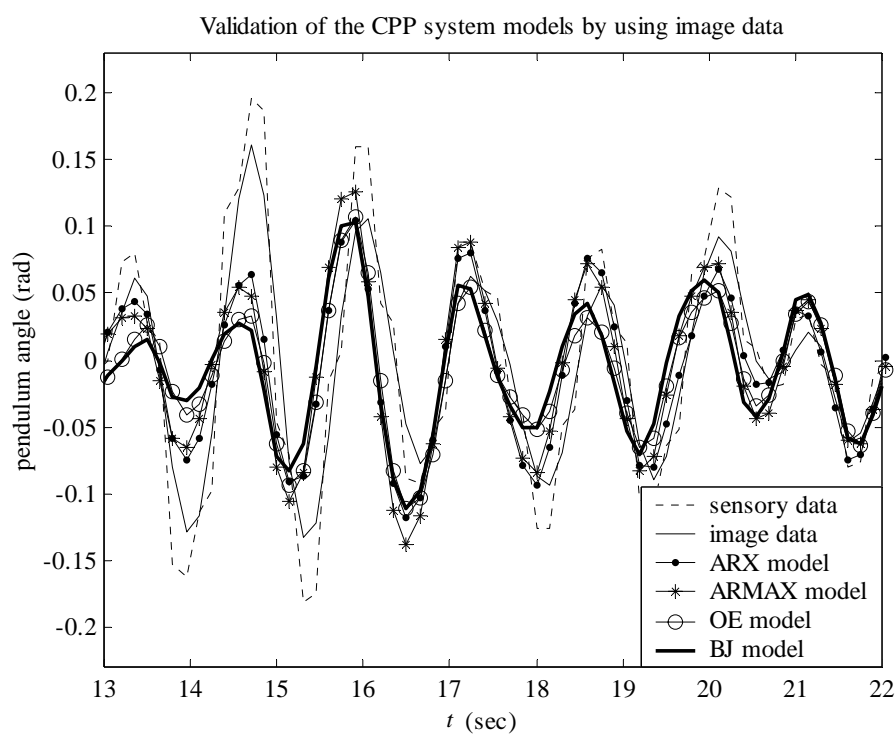
$$\begin{aligned}
 N(s) &= 0.01101s^{10} + 0.773s^9 - 51.54s^8 + 424.7s^7 \\
 &\quad - 1.619 \times 10^4 s^6 + 1.863 \times 10^5 s^5 - 3.563 \times 10^5 s^4 \\
 &\quad + 1.327 \times 10^7 s^3 - 9.983 \times 10^6 s^2 + 1.193 \times 10^8 s - 1.303 \times 10^8 \\
 D(s) &= s^{10} + 82.84s^9 + 2921s^8 + 5.713 \times 10^4 s^7 \\
 &\quad + 6.817 \times 10^5 s^6 + 5.218 \times 10^6 s^5 + 2.715 \times 10^7 s^4 \\
 &\quad + 1.044 \times 10^8 s^3 + 3.002 \times 10^8 s^2 + 5.775 \times 10^8 s + 9.117 \times 10^8
 \end{aligned} \tag{7-15}$$

- แบบจำลอง BJ (โดยใช้ข้อมูลจากการประมวลผลภาพ)

$$\begin{aligned}
 N(s) &= -0.02858s^{10} - 0.4032s^9 - 222.9s^8 - 1579s^7 \\
 &\quad - 1.672 \times 10^5 s^6 + 1.682 \times 10^6 s^5 + 1.063 \times 10^7 s^4 \\
 &\quad + 9.207 \times 10^7 s^3 + 1.153 \times 10^8 s^2 + 6.193 \times 10^8 s - 8.47 \times 10^8 \\
 D(s) &= s^{10} + 33.52s^9 + 8179s^8 + 2.078 \times 10^5 s^7 \\
 &\quad + 1.117 \times 10^7 s^6 + 3.681 \times 10^7 s^5 + 3.99 \times 10^8 s^4 \\
 &\quad + 7.733 \times 10^8 s^3 + 3.727 \times 10^9 s^2 + 4.126 \times 10^9 s + 8.168 \times 10^9
 \end{aligned} \tag{7-16}$$



รูปที่ 7.8 ผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากการประมวลผลภาพ



รูปที่ 7.9 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากการประมวลผลภาพ

ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง ARX ARMAX OE และ BJ ของระบบ CPP เชิงเส้น โดยอาศัยข้อมูลจากตัวตรวจรู้ ดังแสดงในรูปที่ 7.6 และ 7.7 เกิดค่าความคลาดเคลื่อน SAE เท่ากับ 89.5763 86.3952 84.3785 และ 82.8163 ตามลำดับ เกิดค่าความคลาดเคลื่อน SSE เท่ากับ 5.9105 5.5303 5.2587 และ 5.1447 ตามลำดับ เกิดค่าความคลาดเคลื่อน MSE เท่ากับ 0.0028 0.0026 0.0025 และ 0.0024 ตามลำดับ และ เกิดค่าความคลาดเคลื่อน RMSE เท่ากับ 0.0531 0.0513 0.0500 และ 0.0495 ตามลำดับ จากผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง สามารถสังเกตได้ถึงความสอดคล้องกันของกราฟจากข้อมูลทั้งสองส่วน เป็นที่ยืนยันถึงความถูกต้องแม่นยำของแบบจำลองทั้งสองรูปแบบ แต่เนื่องจากเกิดค่าความคลาดเคลื่อนในแบบจำลอง BJ น้อยที่สุด ดังที่ได้ระบุไว้ข้างต้น ดังนั้นแบบจำลอง BJ จึงมีความเหมาะสมมากที่สุดสำหรับระบบ CPP เชิงเส้น ในการนำไปใช้ประโยชน์ต่อไป

7.2.3 การระบุเอกลักษณ์ระบบ CPP เชิงเส้นโดยอาศัยข้อมูลจากการประมวลผลภาพ

การระบุเอกลักษณ์ระบบ CPP เชิงเส้นในขั้นตอนนี้เป็น การระบุเอกลักษณ์ระบบ ด้วยวิธีการดั้งเดิม โดยใช้แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ทั้งสองรูปแบบ ได้แก่แบบจำลอง ARX ARMAX OE และ BJ ในรูปของฟังก์ชันถ่ายโอนอันดับ 10 ดังสมการที่ (7-8) การระบุเอกลักษณ์ระบบในขั้นตอนนี้จะอาศัยข้อมูลที่ได้จากการประมวลผลภาพ ดำเนินงานระบุเอกลักษณ์ระบบโดยอาศัยโปรแกรม MATLAB และ System Identification Toolbox ดังรายละเอียดในภาคผนวก ง. การพิจารณาความเหมาะสมของแบบจำลองใช้เกณฑ์ค่าความคลาดเคลื่อนระหว่างข้อมูลเอาต์พุตที่ได้จากตัวตรวจรู้ (ข้อมูลอ้างอิง) และข้อมูลเอาต์พุตที่ได้จากแบบจำลองในรูป SAE SSE MSE และ RMSE ดังสมการที่ (7-4) ถึง (7-7) ตามลำดับ

ตารางที่ 7.3 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากการประมวลผลภาพ

แบบจำลอง	ค่าความคลาดเคลื่อนจากการระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากการประมวลผลภาพ			
	SAE	SSE	MSE	RMSE
ARX	84.6138	5.2904	0.0025	0.0502
ARMAX	80.4899	4.8697	0.0023	0.0482
OE	86.0889	5.4874	0.0026	0.0511
BJ	80.1143	4.8007	0.0023	0.0478

โดยใช้ข้อมูลของแต่ละสัญญาณที่ได้รับการจัดสรรไว้ แบบจำลอง ARX ARMAX OE และ BJ ของระบบ CPP เชิงเส้นในรูปของฟังก์ชันถ่ายโอน ที่ได้จากการระบุเอกลักษณ์ระบบ โดยใช้ข้อมูลจากการประมวลผลภาพ แสดงดังสมการที่ (7-13) ถึง (7-16) ตามลำดับ ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง แสดงในรูปที่ 7.8 และ 7.9 ตามลำดับ ขณะที่ความคลาดเคลื่อนจากการระบุเอกลักษณ์ระบบ แสดงในตารางที่ 7.3

ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง ARX ARMAX OE และ BJ ของระบบ CPP เชิงเส้นโดยอาศัยข้อมูลจากการประมวลผลภาพ ดังแสดงในรูปที่ 7.8 และ 7.9 เกิดค่าความคลาดเคลื่อน SAE เท่ากับ 84.6138 80.4899 86.0889 และ 80.1143 ตามลำดับ เกิดค่าความคลาดเคลื่อน SSE เท่ากับ 5.2904 4.8697 5.4874 และ 4.8007 ตามลำดับ เกิดค่าความคลาดเคลื่อน MSE เท่ากับ 0.0025 0.0023 0.0026 และ 0.0023 ตามลำดับ และ เกิดค่าความคลาดเคลื่อน RMSE เท่ากับ 0.0502 0.0482 0.0511 และ 0.0478 ตามลำดับ จากผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง สามารถสังเกตได้ถึงความสัมพันธ์กันของกราฟจากข้อมูลทั้งสองส่วน ซึ่งยืนยันถึงความถูกต้องแม่นยำของแบบจำลองทั้งสี่รูปแบบ แต่เนื่องจากเกิดค่าความคลาดเคลื่อนในแบบจำลอง BJ น้อยที่สุด ดังนั้นแบบจำลอง BJ จึงมีความเหมาะสมมากที่สุดสำหรับระบบ CPP เชิงเส้น ในการนำไปใช้ประโยชน์ต่อไป

7.2.4 การระบุเอกลักษณ์ระบบ CPP เชิงเส้นโดยอาศัยข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน

การระบุเอกลักษณ์ระบบ CPP เชิงเส้นในขั้นตอนี้ จะเป็นการระบุเอกลักษณ์ระบบ ด้วยวิธีการดั้งเดิมเช่นเดียวกับหัวข้อ 7.2.2 และ 7.2.3 โดยพิจารณาแบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ทั้งสี่รูปแบบ ได้แก่แบบจำลอง ARX ARMAX OE และ BJ ในรูปของฟังก์ชันถ่ายโอนอันดับ 10 ดังสมการที่ (7-8) การระบุเอกลักษณ์ระบบในขั้นตอนี้จะอาศัยข้อมูลที่ได้จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ดำเนินงานระบุเอกลักษณ์ระบบโดยอาศัยโปรแกรม MATLAB และ System Identification Toolbox ดังรายละเอียดในภาคผนวก ง. การพิจารณาความเหมาะสมของแบบจำลองจะใช้เกณฑ์ค่าความคลาดเคลื่อนระหว่างข้อมูลเอาต์พุตที่ได้จากตัวตรวจรู้ (ข้อมูลอ้างอิง) และข้อมูลเอาต์พุตที่ได้จากแบบจำลองในรูป SAE SSE MSE และ RMSE ดังสมการที่ (7-4) ถึง (7-7) ตามลำดับ

แบบจำลอง ARX ARMAX OE และ BJ ของระบบ CPP เชิงเส้นในรูปของฟังก์ชันถ่ายโอน ที่ได้จากการระบุเอกลักษณ์ระบบโดยใช้ข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน แสดงดังสมการที่ (7-17) ถึง (7-20) ตามลำดับ ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง แสดงในรูปที่ 7.10 และ 7.11 ตามลำดับ ขณะที่ความคลาดเคลื่อนจากการระบุเอกลักษณ์ระบบ แสดงในตารางที่ 7.4

- แบบจำลอง ARX (โดยใช้ข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไข)

$$\begin{aligned}
 N(s) &= -0.04395s^{10} + 0.9834s^9 - 120s^8 - 944.2s^7 \\
 &\quad - 4.226 \times 10^4 s^6 + 2.916 \times 10^5 s^5 + 6.519 \times 10^5 s^4 \\
 &\quad + 9.268 \times 10^7 s^3 + 1.862 \times 10^8 s^2 + 2.088 \times 10^9 s - 1.422 \times 10^9 \\
 D(s) &= s^{10} + 60.8s^9 + 3750s^8 + 1.129 \times 10^5 s^7 \\
 &\quad + 1.971 \times 10^6 s^6 + 2.363 \times 10^7 s^5 + 1.832 \times 10^8 s^4 \\
 &\quad + 8.765 \times 10^8 s^3 + 3.572 \times 10^9 s^2 + 7.245 \times 10^9 s + 1.494 \times 10^{10}
 \end{aligned} \tag{7-17}$$

- แบบจำลอง ARMAX (โดยใช้ข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไข)

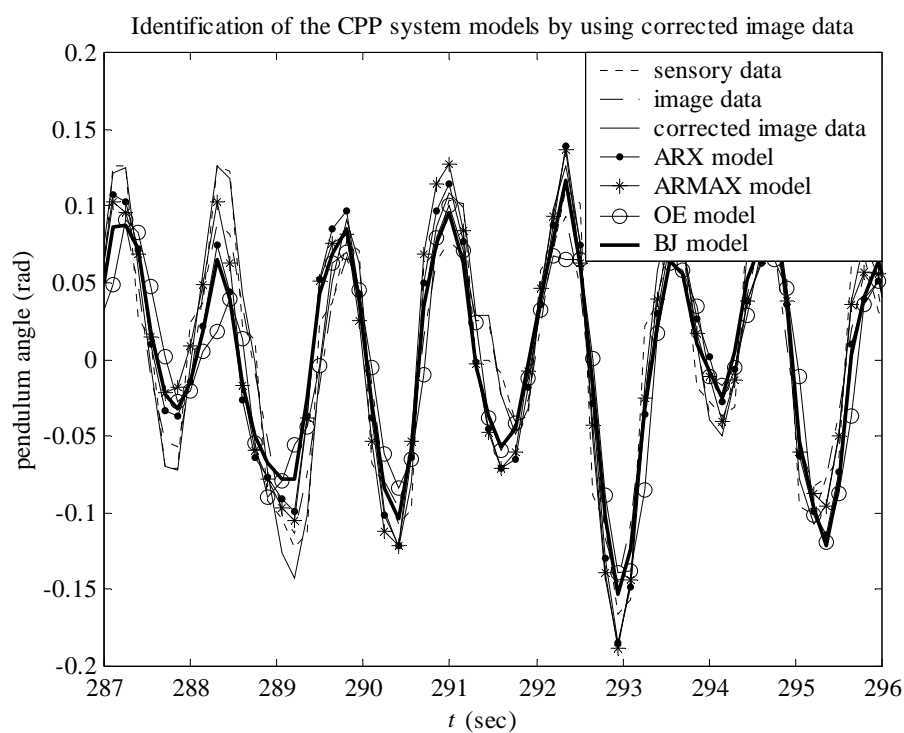
$$\begin{aligned}
 N(s) &= -0.04702s^{10} - 0.5525s^9 - 77.54s^8 - 5684s^7 \\
 &\quad - 1.145 \times 10^5 s^6 - 1.87 \times 10^5 s^5 + 1.913 \times 10^7 s^4 \\
 &\quad + 2.178 \times 10^8 s^3 + 7.035 \times 10^8 s^2 - 3.214 \times 10^8 s - 2.125 \times 10^8 \\
 D(s) &= s^{10} + 119s^9 + 6953s^8 + 2.952 \times 10^5 s^7 \\
 &\quad + 7.239 \times 10^6 s^6 + 8.114 \times 10^7 s^5 + 4.717 \times 10^8 s^4 \\
 &\quad + 1.713 \times 10^9 s^3 + 5.205 \times 10^9 s^2 + 5.87 \times 10^9 s + 5.978 \times 10^9
 \end{aligned} \tag{7-18}$$

- แบบจำลอง OE (โดยใช้ข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไข)

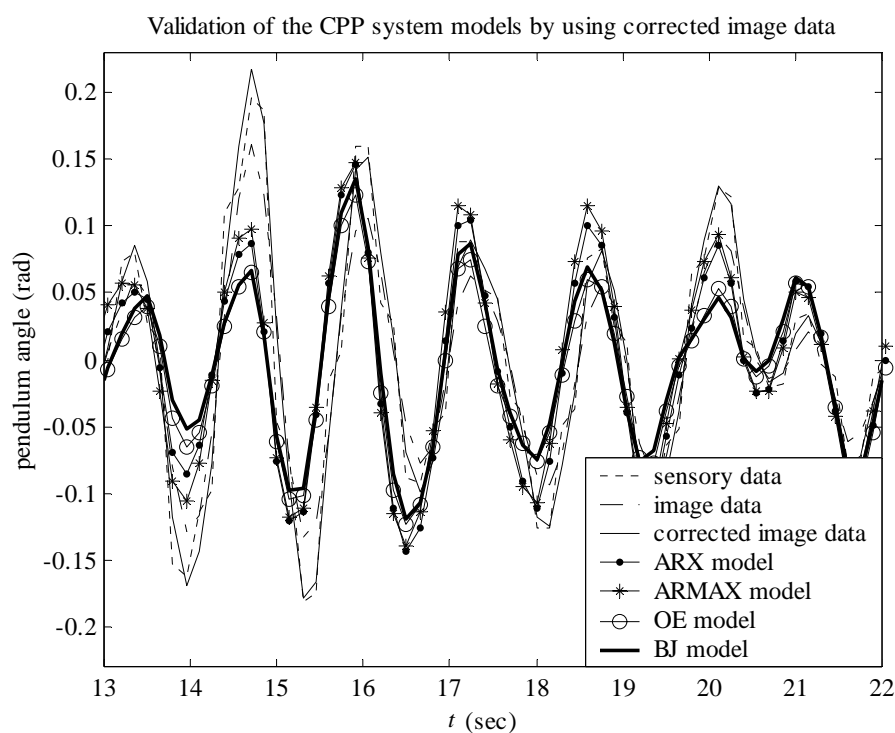
$$\begin{aligned}
 N(s) &= 0.02773s^{10} + 7.957s^9 - 165.6s^8 + 2368s^7 \\
 &\quad - 9.975 \times 10^4 s^6 + 7.557 \times 10^5 s^5 - 4.007 \times 10^6 s^4 \\
 &\quad + 1.007 \times 10^8 s^3 - 1.083 \times 10^8 s^2 + 2.116 \times 10^9 s - 1.794 \times 10^9 \\
 D(s) &= s^{10} + 113s^9 + 6012s^8 + 1.716 \times 10^5 s^7 \\
 &\quad + 2.776 \times 10^6 s^6 + 2.673 \times 10^7 s^5 + 1.686 \times 10^8 s^4 \\
 &\quad + 8.327 \times 10^8 s^3 + 3.147 \times 10^9 s^2 + 6.608 \times 10^9 s + 1.413 \times 10^{10}
 \end{aligned} \tag{7-19}$$

- แบบจำลอง BJ (โดยใช้ข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไข)

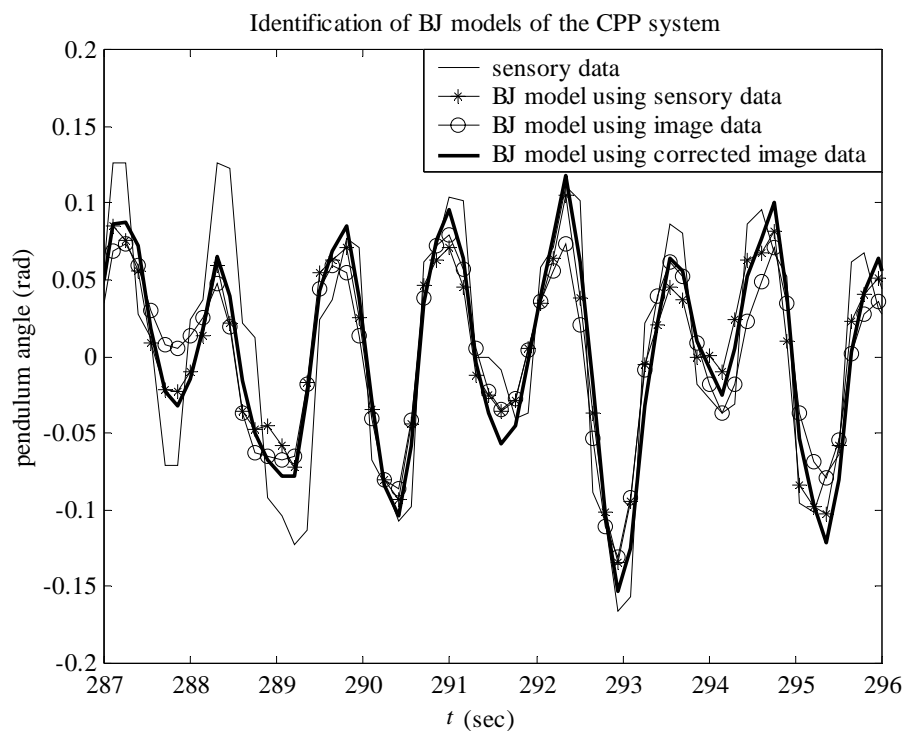
$$\begin{aligned}
 N(s) &= -0.03771s^{10} + 2.238s^9 - 156.7s^8 + 1412s^7 \\
 &\quad - 5.541 \times 10^4 s^6 + 5.608 \times 10^5 s^5 - 4.758 \times 10^6 s^4 \\
 &\quad + 1.003 \times 10^8 s^3 - 1.103 \times 10^8 s^2 + 2.68 \times 10^9 s - 8.344 \times 10^8 \\
 D(s) &= s^{10} + 46.35s^9 + 3559s^8 + 7.59 \times 10^4 s^7 \\
 &\quad + 1.331 \times 10^6 s^6 + 1.969 \times 10^7 s^5 + 1.327 \times 10^8 s^4 \\
 &\quad + 7.772 \times 10^8 s^3 + 3.524 \times 10^9 s^2 + 7.092 \times 10^9 s + 2.004 \times 10^{10}
 \end{aligned} \tag{7-20}$$



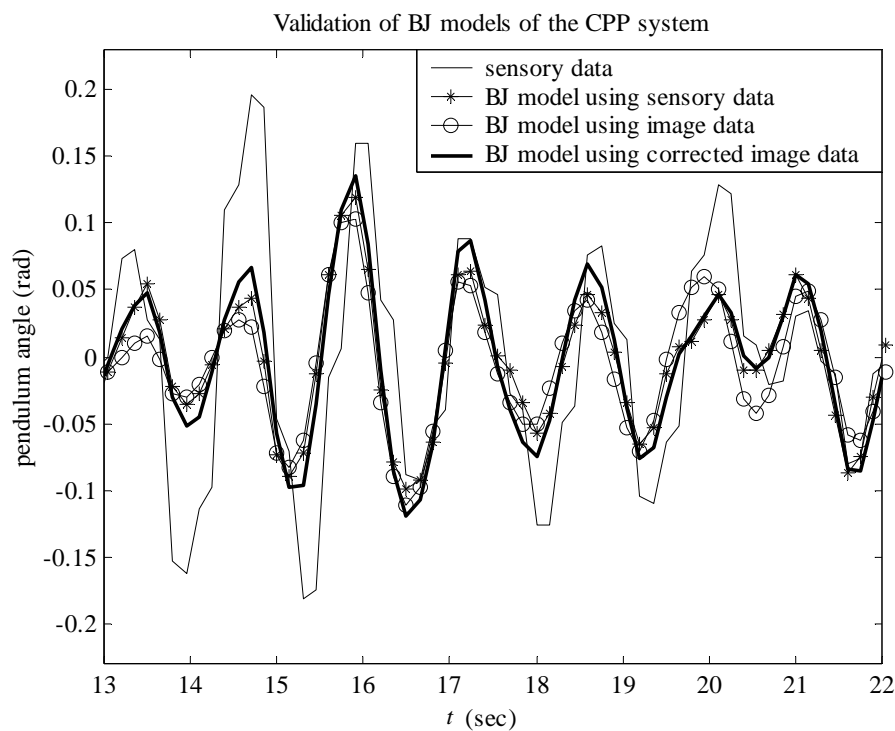
รูปที่ 7.10 ผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากการประมวลผลภาพ
ที่ได้รับการแก้ไขความคลาดเคลื่อน



รูปที่ 7.11 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP เชิงเส้น โดยใช้ข้อมูลจากการ
ประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน



รูปที่ 7.12 ผลการระบุเอกลักษณ์แบบจำลอง BJ ของระบบ CPP เชิงเส้น



รูปที่ 7.13 ผลการตรวจสอบความถูกต้องของแบบจำลอง BJ ของระบบ CPP เชิงเส้น

ตารางที่ 7.4 ค่าความคลาดเคลื่อนของแบบจำลองระบบ CPP เชิงเส้น โดยใช้ข้อมูลจาก
การประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน

แบบจำลอง	ค่าความคลาดเคลื่อนจากการระบุเอกลักษณ์ระบบ CPP เชิงเส้นโดยใช้ข้อมูล จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน			
	SAE	SSE	MSE	RMSE
ARX	85.6642	5.4506	0.0026	0.0509
ARMAX	88.4983	5.7896	0.0028	0.0525
OE	89.4028	5.9862	0.0029	0.0534
BJ	83.8216	5.3053	0.0025	0.0503

ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง ARX ARMAX OE และ BJ ของระบบ CPP เชิงเส้น โดยอาศัยข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ดังแสดงในรูปที่ 7.10 และ 7.11 เกิดค่าความคลาดเคลื่อน SAE เท่ากับ 85.6642 88.4983 89.4028 และ 83.8216 ตามลำดับ เกิดค่าความคลาดเคลื่อน SSE เท่ากับ 5.4506 5.7896 5.9862 และ 5.3053 ตามลำดับ เกิดค่าความคลาดเคลื่อน MSE เท่ากับ 0.0026 0.0028 0.0029 และ 0.0025 ตามลำดับ และเกิดค่าความคลาดเคลื่อน RMSE เท่ากับ 0.0509 0.0525 0.0534 และ 0.0503 ตามลำดับ จากผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง สามารถสังเกตได้ถึงความสัมพันธ์กันของกราฟจากข้อมูลทั้งสองส่วน ซึ่งยืนยันถึงความถูกต้องแม่นยำของแบบจำลองทั้งสองรูปแบบ แต่เนื่องจากเกิดค่าความคลาดเคลื่อนในแบบจำลอง BJ น้อยที่สุด แบบจำลอง BJ จึงมีความเหมาะสมมากที่สุดสำหรับระบบ CPP เชิงเส้นในการนำไปใช้ประโยชน์ต่อไป

7.2.5 การเปรียบเทียบผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้น

ในขั้นตอนนี้จะเป็นการเปรียบเทียบผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้นด้วยวิธีการดั้งเดิม ที่ใช้ข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน พิเคราะห์เฉพาะแบบจำลอง BJ ในรูปของฟังก์ชันถ่ายโอนอันดับ 10 เท่านั้น ทั้งนี้เนื่องจากเกิดค่าความคลาดเคลื่อนในแบบจำลอง BJ น้อยที่สุด และจากหัวข้อ 7.2.2 7.2.3 และ 7.2.4 พบว่าแบบจำลอง BJ ในรูปของฟังก์ชันถ่ายโอนอันดับ 10 ของระบบ CPP เชิงเส้นที่ได้จากการระบุเอกลักษณ์โดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน แสดงดังสมการที่ (7-12) (7-16) และ

(7-20) ตามลำดับ สมการดังกล่าวได้รับการแสดงไว้อีกครั้งเพื่อความชัดเจนในการเปรียบเทียบ
 ดังสมการที่ (7-21) ถึง (7-23) ตามลำดับ

- แบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากตัวตรวจรู้

$$\begin{aligned} N(s) = & -0.05101s^{10} - 22.34s^9 + 1452s^8 - 1.515 \times 10^4 s^7 \\ & - 6.373 \times 10^4 s^6 - 3.401 \times 10^7 s^5 + 2.698 \times 10^8 s^4 \\ & + 1.611 \times 10^9 s^3 + 7.649 \times 10^9 s^2 + 7.53 \times 10^{10} s - 2.975 \times 10^{10} \end{aligned} \quad (7-21)$$

$$\begin{aligned} D(s) = & s^{10} + 216s^9 + 2.182 \times 10^4 s^8 + 1.322 \times 10^6 s^7 \\ & + 4.644 \times 10^7 s^6 + 1.096 \times 10^9 s^5 + 4.693 \times 10^9 s^4 \\ & + 4.271 \times 10^{10} s^3 + 1.167 \times 10^{11} s^2 + 3.405 \times 10^{11} s + 6.218 \times 10^{11} \end{aligned}$$

- แบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากการประมวลผลภาพ

$$\begin{aligned} N(s) = & -0.02858s^{10} - 0.4032s^9 - 222.9s^8 - 1579s^7 \\ & - 1.672 \times 10^5 s^6 + 1.682 \times 10^6 s^5 + 1.063 \times 10^7 s^4 \\ & + 9.207 \times 10^7 s^3 + 1.153 \times 10^8 s^2 + 6.193 \times 10^8 s - 8.47 \times 10^8 \end{aligned} \quad (7-22)$$

$$\begin{aligned} D(s) = & s^{10} + 33.52s^9 + 8179s^8 + 2.078 \times 10^5 s^7 \\ & + 1.117 \times 10^7 s^6 + 3.681 \times 10^7 s^5 + 3.99 \times 10^8 s^4 \\ & + 7.733 \times 10^8 s^3 + 3.727 \times 10^9 s^2 + 4.126 \times 10^9 s + 8.168 \times 10^9 \end{aligned}$$

- แบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน

$$\begin{aligned} N(s) = & -0.03771s^{10} + 2.238s^9 - 156.7s^8 + 1412s^7 \\ & - 5.541 \times 10^4 s^6 + 5.608 \times 10^5 s^5 - 4.758 \times 10^6 s^4 \\ & + 1.003 \times 10^8 s^3 - 1.103 \times 10^8 s^2 + 2.68 \times 10^9 s - 8.344 \times 10^8 \end{aligned} \quad (7-23)$$

$$\begin{aligned} D(s) = & s^{10} + 46.35s^9 + 3559s^8 + 7.59 \times 10^4 s^7 \\ & + 1.331 \times 10^6 s^6 + 1.969 \times 10^7 s^5 + 1.327 \times 10^8 s^4 \\ & + 7.772 \times 10^8 s^3 + 3.524 \times 10^9 s^2 + 7.092 \times 10^9 s + 2.004 \times 10^{10} \end{aligned}$$

ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง BJ ของระบบ CPP เริงเส้น โดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน แสดงในรูปที่ 7.12 และ 7.13 ตามลำดับ

ขณะที่ตารางที่ 7.5 แสดงความคลาดเคลื่อนจากการระบุเอกลักษณ์แบบจำลอง BJ ของระบบ CPP เชิงเส้น

ตารางที่ 7.5 ค่าความคลาดเคลื่อนของแบบจำลอง BJ ของระบบ CPP เชิงเส้น

ข้อมูลที่ใช้	ค่าความคลาดเคลื่อนของแบบจำลอง BJ ของระบบ CPP เชิงเส้น			
	SAE	SSE	MSE	RMSE
ตัวตรวจรู้	82.8163	5.1447	0.0024	0.0495
การประมวลผลภาพ	80.1143	4.8007	0.0023	0.0478
การประมวลผลภาพที่ได้รับการแก้ไข	83.8216	5.3053	0.0025	0.0503

ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง BJ ของระบบ CPP เชิงเส้น โดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ดังแสดงในรูปที่ 7.12 และ 7.13 เกิดค่าความคลาดเคลื่อน SAE เท่ากับ 82.8163 80.1143 และ 83.8216 ตามลำดับ เกิดค่าความคลาดเคลื่อน SSE เท่ากับ 5.1447 4.8007 และ 5.3053 ตามลำดับ เกิดค่าความคลาดเคลื่อน MSE เท่ากับ 0.0024 0.0023 และ 0.0025 ตามลำดับ และเกิดค่าความคลาดเคลื่อน RMSE เท่ากับ 0.0495 0.0478 และ 0.0503 ตามลำดับ จากผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง สามารถสังเกตได้ถึงความสอดคล้องกันของกราฟจากข้อมูลทั้งสองส่วน เป็นสิ่งยืนยันถึงความถูกต้องแม่นยำตรงของแบบจำลอง BJ ของระบบ CPP เชิงเส้น โดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน โดยเฉพาะอย่างยิ่ง เมื่อเปรียบเทียบผลการระบุเอกลักษณ์ระบบ ระหว่างแบบจำลอง BJ ที่ใช้ข้อมูลจากตัวตรวจรู้ และแบบจำลอง BJ ที่ใช้ข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อนค่าความคลาดเคลื่อนที่เกิดขึ้น ดังตารางที่ 7.5 มีความใกล้เคียงกันอย่างมาก แสดงให้เห็นว่า การระบุเอกลักษณ์ระบบ โดยอาศัยเทคนิคการประมวลผลภาพที่นำเสนอ มีความถูกต้องแม่นยำเท่าเทียมกับการระบุเอกลักษณ์ระบบด้วยวิธีการแบบดั้งเดิม

เมื่อนำแบบจำลอง BJ ของระบบ CPP เชิงเส้นที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ดังสมการที่ (7-21) (7-22) และ (7-23) ตามลำดับ มาทำการพิจารณาตำแหน่งโพล (pole) และซีโร (zero) จะสามารถแสดงได้ดังสมการที่ (7-24) (7-25) และ (7-26) ตามลำดับ และรูปที่ 7.14 ถึง 7.16 แสดงตำแหน่งโพลและซีโรของแบบจำลอง BJ ของ

ระบบ CPP เชิงเส้นที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ตามลำดับ เช่นเดียวกัน

จากสมการที่ (7-24) (7-25) และ (7-26) เมื่อทำการพิจารณาโพลเด่น (dominant pole) พบว่าแบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากตัวตรวจรู้ มีโพลเด่นอยู่ที่ $-0.0720 \pm 5.4040i$ และ $-0.3228 \pm 3.1616i$ ส่วนแบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากการประมวลผลภาพ มีโพลเด่นอยู่ที่ $-0.0804 \pm 3.1873i$ และ $-0.5719 \pm 4.5499i$ และแบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน มีโพลเด่นอยู่ที่ $-0.0566 \pm 16.3109i$ และ $-0.5376 \pm 3.1462i$

เมื่อทำการพิจารณาซีโรเด่น (dominant zero) จากสมการที่ (7-24) (7-25) และ (7-26) พบว่าแบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากตัวตรวจรู้ มีซีโรเด่นอยู่ที่ 13.3333 และ 0.3792 ส่วนแบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากการประมวลผลภาพ มีซีโรเด่นอยู่ที่ 13.3333 และ 1.0070 และแบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อนมีซีโรเด่นอยู่ที่ 13.3333 และ 0.3142

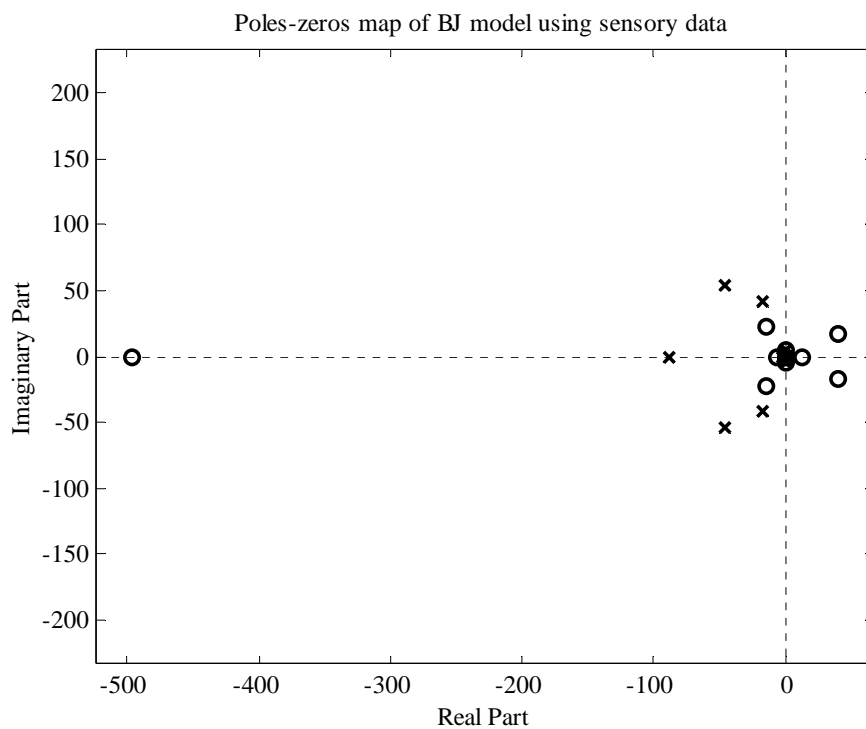
จากการอภิปรายผลเกี่ยวกับตำแหน่งโพลและซีโรที่กล่าวมาข้างต้น สามารถระบุได้อย่างชัดเจนว่าระบบ CPP เชิงเส้นเป็นระบบที่มีความเสถียรภาพ เพราะโพลทุกตัวของระบบวางตัวอยู่ทางฝั่งซ้ายมือของระนาบเชิงซ้อน ระบบดังกล่าวมีลักษณะสมบัติทางเฟสเป็น เฟสไม่ต่ำสุด (nonminimum phase characteristic) ทั้งนี้เพราะเป็นระบบที่เสถียรภาพ แต่มีซีโรบางตัวอยู่ทางฝั่งขวามือของระนาบเชิงซ้อน

แบบจำลอง BJ ของระบบ CPP เชิงเส้นที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ดังสมการที่ (7-21) (7-22) และ (7-23) ตามลำดับนั้น สามารถให้การอธิบายพลวัตของระบบ CPP เชิงเส้นได้อย่างน่าพึงพอใจ โดยเฉพาะอย่างยิ่งแบบจำลอง BJ ที่ได้จากการระบุเอกลักษณ์ระบบ โดยอาศัยข้อมูลจากตัวตรวจรู้ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ดังสมการที่ (7-21) และ (7-23) ทั้งนี้เนื่องจากเมื่อพิจารณาค่าตำแหน่งของโพลเด่นและซีโรเด่นของแบบจำลองทั้งสอง พบว่ามีความใกล้เคียงกันอย่างมาก แสดงให้เห็นว่าการระบุเอกลักษณ์ระบบ โดยอาศัยเทคนิคการประมวลผลภาพที่น่าเสนอ มีความถูกต้องแม่นยำเท่าเทียมกับการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม ที่อาศัยข้อมูลจากตัวตรวจรู้มาดำเนินการระบุเอกลักษณ์ระบบ

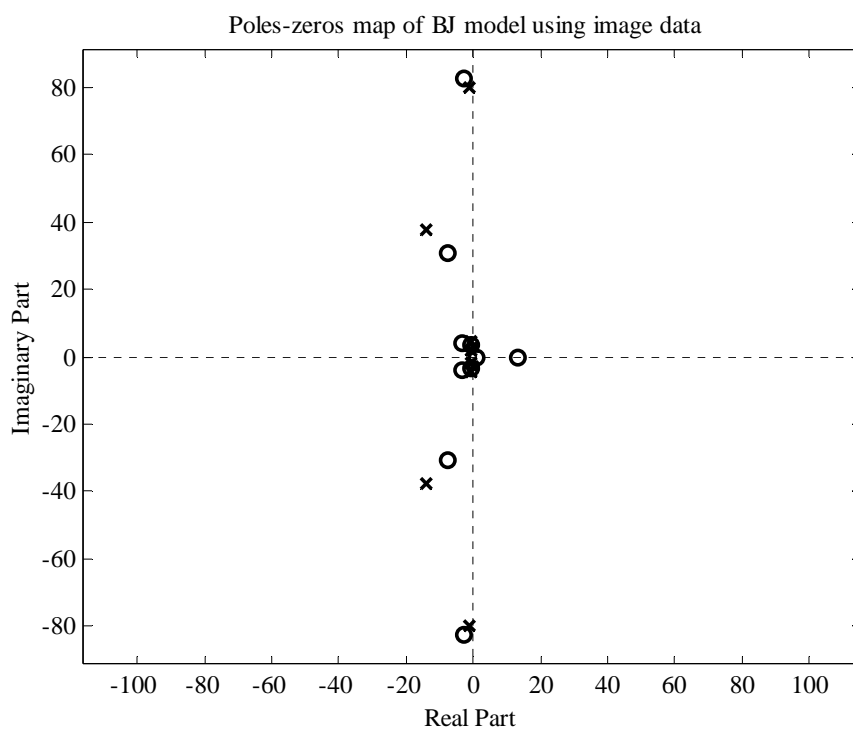
$$\begin{array}{ll}
 \text{pole} = & \text{zero} = \\
 -88.0043 & -496.5523 \\
 -45.8999+54.1008i & 40.1822+17.3938i \\
 -45.8999-54.1008i & 40.1822-17.3938i \\
 -16.5440+42.2760i & -14.7987+22.1820i \\
 -16.5440-42.2760i & -14.7987-22.1820i \\
 -0.0720+5.4040i & 13.3333 \\
 -0.0720-5.4040i & -5.7692 \\
 -0.3228+3.1616i & -0.0842+5.4340i \\
 -0.3228-3.1616i & -0.0842-5.4340i \\
 -2.3087 & 0.3792
 \end{array} \tag{7-24}$$

$$\begin{array}{ll}
 \text{pole} = & \text{zero} = \\
 -1.2057+79.7355i & -2.7778+82.9404i \\
 -1.2057-79.7355i & -2.7778-82.9404i \\
 -14.1340+37.7773i & -7.8408+30.9763i \\
 -14.1340-37.7773i & -7.8408-30.9763i \\
 -0.5719+4.5499i & 13.3333 \\
 -0.5719-4.5499i & -3.0779+4.1362i \\
 -0.0804+3.1873i & -3.0779-4.1362i \\
 -0.0804-3.1873i & -0.5265+3.3957i \\
 -0.7705+1.7607i & -0.5265-3.3957i \\
 -0.7705-1.7607i & 1.0070
 \end{array} \tag{7-25}$$

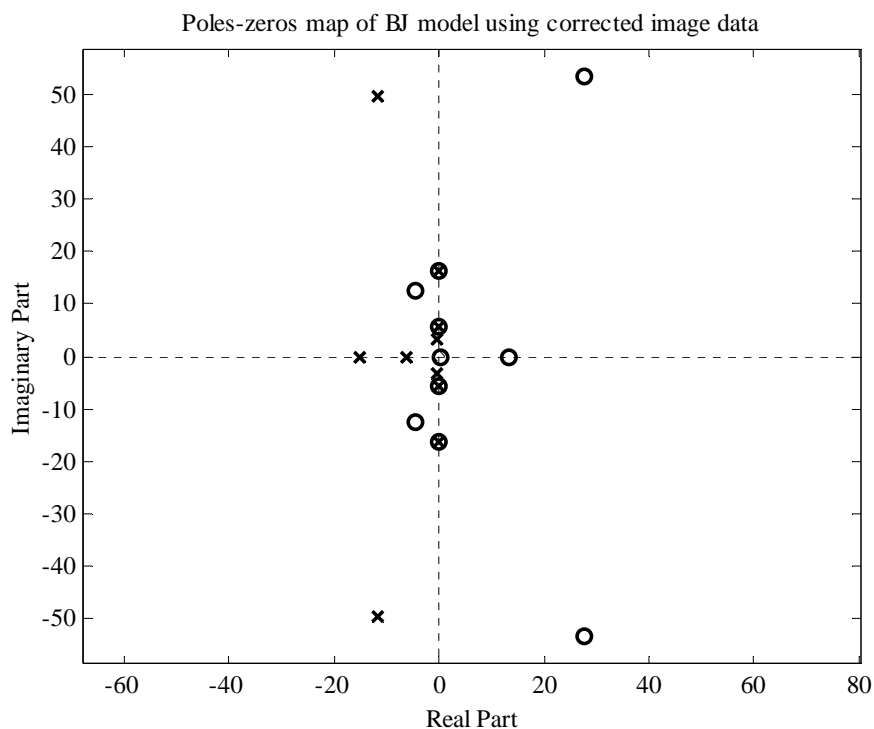
$$\begin{array}{ll}
 \text{pole} = & \text{zero} = \\
 -11.8656+49.5127i & 27.6334+53.0902i \\
 -11.8656-49.5127i & 27.6334-53.0902i \\
 -0.0566+16.3109i & 13.3333 \\
 -0.0566-16.3109i & -0.0727+16.2664i \\
 -15.0757 & -0.0727-16.2664i \\
 -6.0975 & -4.5934+12.3917i \\
 -0.1307+5.5687i & -4.5934-12.3917i \\
 -0.1307-5.5687i & -0.1187+5.6464i \\
 -0.5376+3.1462i & -0.1187-5.6464i \\
 -0.5376-3.1462i & 0.3142
 \end{array} \tag{7-26}$$



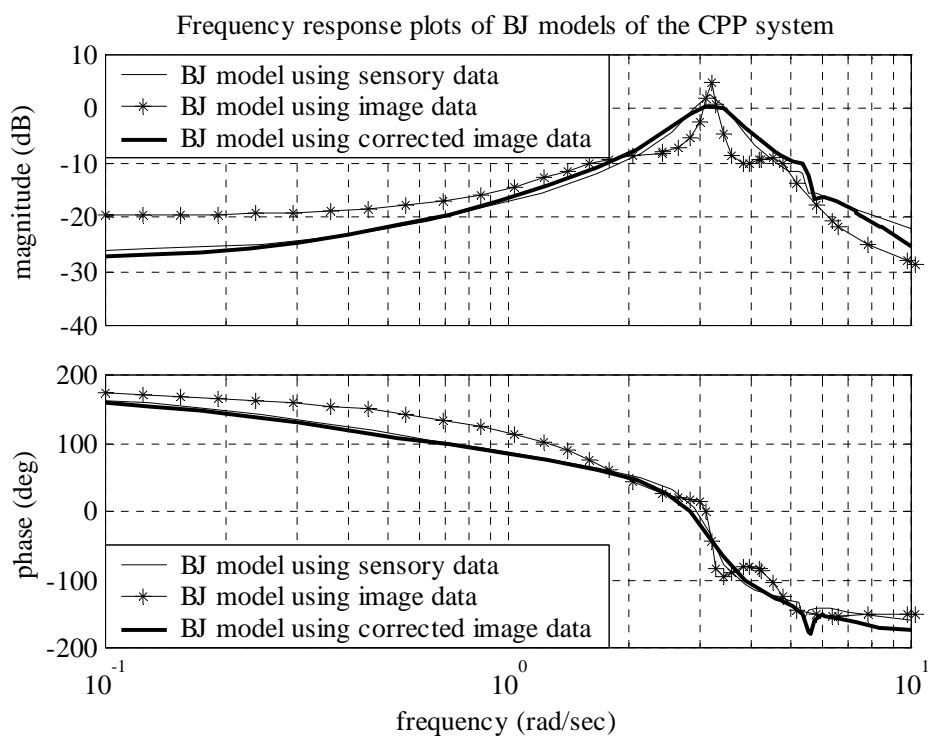
รูปที่ 7.14 ตำแหน่งโพลและซีโรของแบบจำลอง BJ ที่ใช้ข้อมูลจากตัวตรวจรู้



รูปที่ 7.15 ตำแหน่งโพลและซีโรของแบบจำลอง BJ ที่ใช้ข้อมูลจากการประมวลผลภาพ



รูปที่ 7.16 ตำแหน่งโพลและซีโรของแบบจำลอง BJ ที่ใช้ข้อมูลจากการประมวลผลภาพ
ที่ได้รับการแก้ไขความคลาดเคลื่อน



รูปที่ 7.17 ผลตอบสนองในโดเมนความถี่ของแบบจำลอง BJ ของระบบ CPP เริงเส้น

ผลตอบสนองในโดเมนความถี่ของระบบ CPP เริงเส้น ที่แสดงในรูปที่ 7.17 เป็นสิ่งยืนยันถึงความถูกต้องแม่นยำของแบบจำลอง BJ ที่ใช้ข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน โดยเฉพาะเมื่อเปรียบเทียบผลตอบสนองในโดเมนความถี่ระหว่างแบบจำลอง BJ ที่ใช้ข้อมูลจากตัวตรวจรู้ และแบบจำลอง BJ ที่ใช้ข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ผลตอบสนองในโดเมนความถี่ ทั้งในส่วนของขนาดและเฟส มีความใกล้เคียงกันอย่างมากตลอดช่วงความถี่ที่สนใจ มีค่าเรโซแนนซ์สูงสุด (resonant peak) อยู่ระหว่าง 0.5 ถึง 2.5 เดซิเบล และมีค่าความถี่เรโซแนนซ์ (resonant frequency) อยู่ระหว่าง 3.08 ถึง 3.10 เรเดียนต่อวินาที และจากผลตอบสนองในโดเมนความถี่ดังกล่าว ระบุลักษณะสมบัติทางเฟสเป็นแบบ เฟสไม่ต่ำสุดไว้อย่างชัดเจน

7.3 การระบุเอกลักษณ์ระบบ CPP ไม่เริงเส้น

เมื่อพิจารณาระบบ CPP จากรูปที่ 6.1 ในบทที่ 6 พบว่าแบบจำลองทางคณิตศาสตร์ไม่เริงเส้นที่พัฒนาจากการวิเคราะห์ แสดงดังสมการที่ (7-27) ถึง (7-28) (รายละเอียดการพัฒนาแบบจำลองแสดงในภาคผนวก ข.) ลูกตุ้มมีมวล $m = 0.251$ กิโลกรัม ก้านของลูกตุ้มซึ่งมีน้ำหนักเบา มีความยาว $l = 0.4$ เมตร และความเร่งเนื่องจากแรงโน้มถ่วง $g = 9.81$ เมตร/วินาที² ตำแหน่งของรถแสดงด้วย y (เมตร) และมุมแกว่งไกวของลูกตุ้มแสดงด้วย ϕ (เรเดียน) มวล M ของรถไม่ทราบค่า และแรง f (นิวตัน) จะเป็นฟังก์ชันของสัญญาณอินพุตของมอเตอร์ u (โวลต์) แรงดังกล่าวมีลักษณะไม่สม่ำเสมอ เพราะส่งผ่านสายพานอ่อนที่มีการสละบดอยู่ตลอดเวลา ดังนั้นจึงกำหนดให้แรง f เป็นพหุนามของอินพุต u ดังสมการที่ (7-29) ที่ซึ่ง a_i เมื่อ $i = 0, \dots, n$ คือค่าสัมประสิทธิ์ของพหุนาม ดังนั้นการระบุเอกลักษณ์ระบบ จึงเป็นการหาค่าพารามิเตอร์ M และ a_i เมื่อ $i = 0, \dots, n$

$$\ddot{\phi} = \frac{f \cos(\phi) + 0.5ml \sin(2\phi)(\dot{\phi})^2 + (M + m)g \sin(\phi)}{l[m \cos^2(\phi) - (M + m)]} \quad (7-27)$$

$$\ddot{y} = \frac{f + 0.5mg \sin(2\phi) + ml \sin(\phi)(\dot{\phi})^2}{[(M + m) - m \cos^2(\phi)]} \quad (7-28)$$

$$f = a_n u^n + \dots + a_1 u + a_0 \quad (7-29)$$

เนื่องจากการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม ไม่สามารถนำมาประยุกต์ใช้กับกรณีการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นได้ ดังนั้นการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นจะอาศัยการค้นหาแบบตาบอดเชิงปรับตัว เป็นเครื่องมือในการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง และการคำนวณผลเฉลยของแบบจำลองในรูปของสมการอนุพันธ์ไม่เชิงเส้น ดังสมการที่ (7-27) ถึง (7-28) อาศัยการคำนวณด้วยระเบียบวิธีรุงเง-คุดตา (Runge-Kutta method) ดังรายละเอียดที่ปรากฏในตำราการคำนวณเชิงตัวเลขทั่วไป เช่น ปราโมทย์ เชชะอำไพ (2544) และ Mathews และ Fink (2004) เป็นต้น

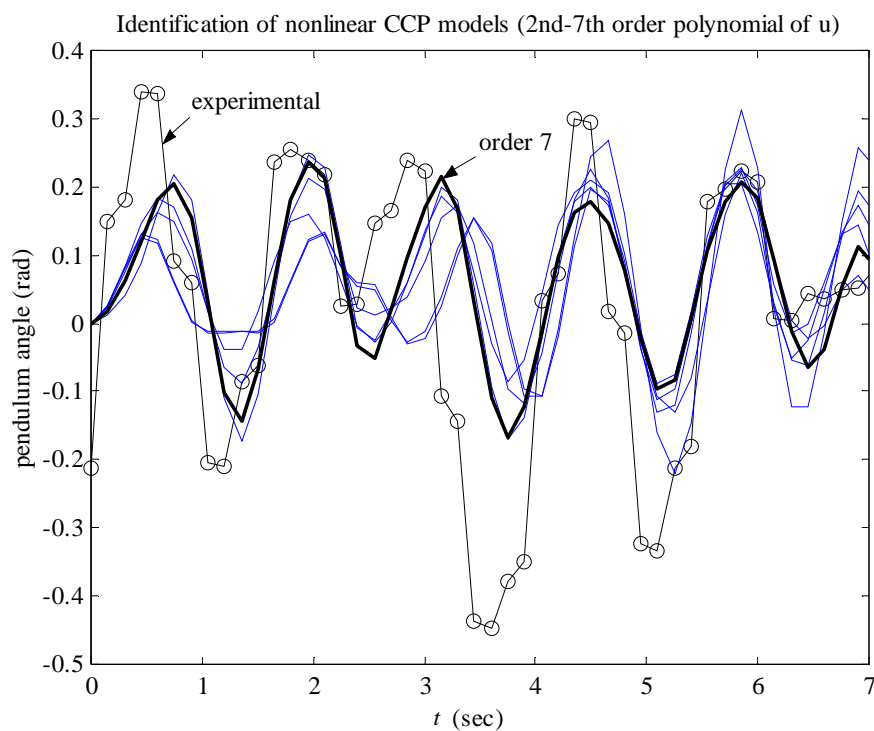
จากผลการทดสอบระบบ CPP ในบทที่ 6 พิจารณาการทดสอบตอนที่ 3 ซึ่งถูกดัดแปลงไว้ด้วยมม ± 0.6 เรเดียน เพื่อใช้ดำเนินการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น ข้อมูลจากการทดสอบประกอบไปด้วยสัญญาณอินพุต u และสัญญาณเอาต์พุต c (ตำแหน่งเชิงมุมของลูกตุ้ม) ทำการแบ่งข้อมูลของแต่ละสัญญาณออกเป็นสองส่วน ข้อมูลส่วนแรกจำนวน 150 ข้อมูล เพื่อใช้ระบุเอกลักษณ์ระบบ และข้อมูลส่วนที่สองจำนวน 50 ข้อมูล เพื่อใช้ตรวจสอบความถูกต้องของแบบจำลอง

การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น จะแบ่งออกเป็น 2 ส่วนด้วยกันคือ (1) การระบุเอกลักษณ์ระบบเพื่อพิจารณาอันดับที่เหมาะสมของแรง f ที่เป็นพหุนามของสัญญาณอินพุตของมอเตอร์ u และ (2) การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น โดยอาศัยข้อมูลจากตัวตรวจรู้จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ดังรายละเอียดต่อไปนี้

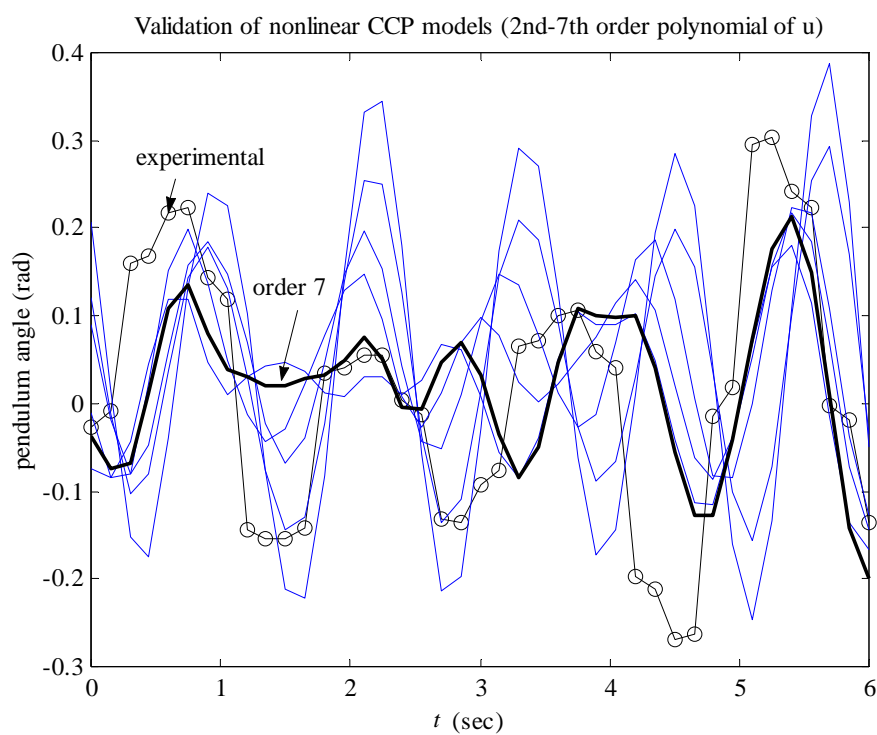
7.3.1 การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นเพื่อพิจารณาอันดับที่เหมาะสมของแรง f ที่เป็นพหุนามของ u

ในขั้นตอนนี้จะเป็นการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น เพื่อพิจารณาอันดับที่เหมาะสมของแรง f ที่เป็นพหุนามของ u จากสมการที่ (7-29) กำหนดให้แรง f เป็นพหุนามของอินพุต u อันดับ 2 ถึง 7 การระบุเอกลักษณ์ระบบในขั้นตอนนี้จะอาศัยข้อมูลที่ได้จากตัวตรวจรู้เท่านั้น เนื่องจากเป็นข้อมูลอ้างอิง การพิจารณาอันดับที่เหมาะสมจะใช้เกณฑ์ค่าความคลาดเคลื่อนระหว่างข้อมูลเอาต์พุตที่ได้จากการทดสอบ (ϕ) และข้อมูลเอาต์พุตที่ได้จากแบบจำลอง ($\hat{\phi}$) ในรูปของ SSE ดังสมการที่ (7-5) การระบุเอกลักษณ์ระบบอาศัยการค้นหาแบบตาบอดเชิงปรับตัวที่พัฒนาด้วยโปรแกรม C เป็นเครื่องมือในการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง ซึ่งมีรายละเอียดดังนี้

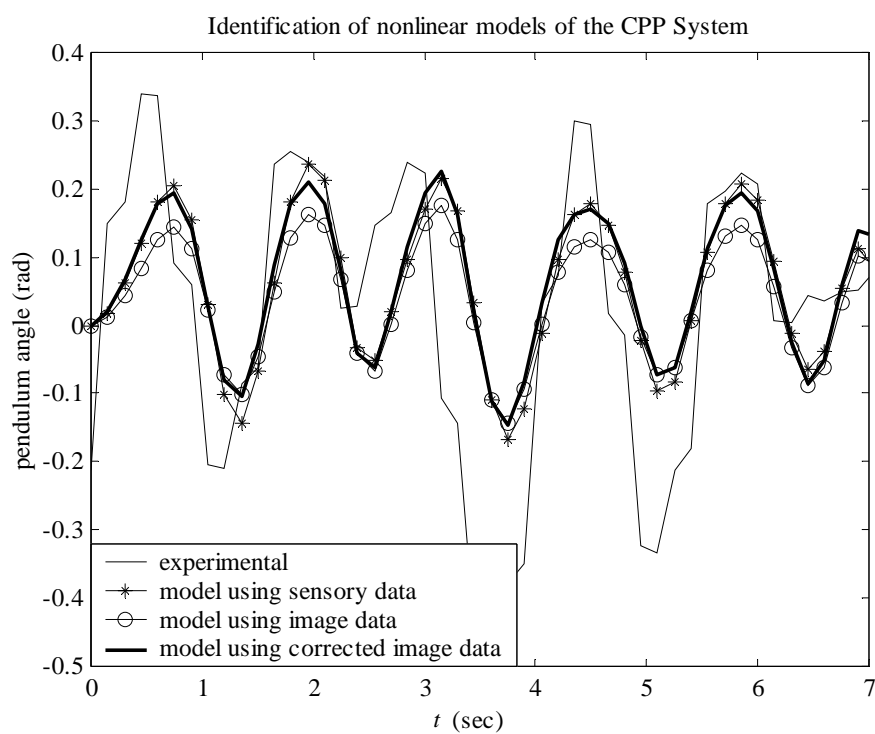
1. รัศมีการค้นหาเริ่มต้นเท่ากับ 10% ของปริภูมิการค้นหา
2. จำนวนสมาชิกค่าใกล้เคียงเท่ากับ 30 ตัว
3. จำนวนการซ้ำค่าคำตอบก่อนเรียกใช้กลไกย้อนรอยการค้นหาเท่ากับ 5 ครั้ง



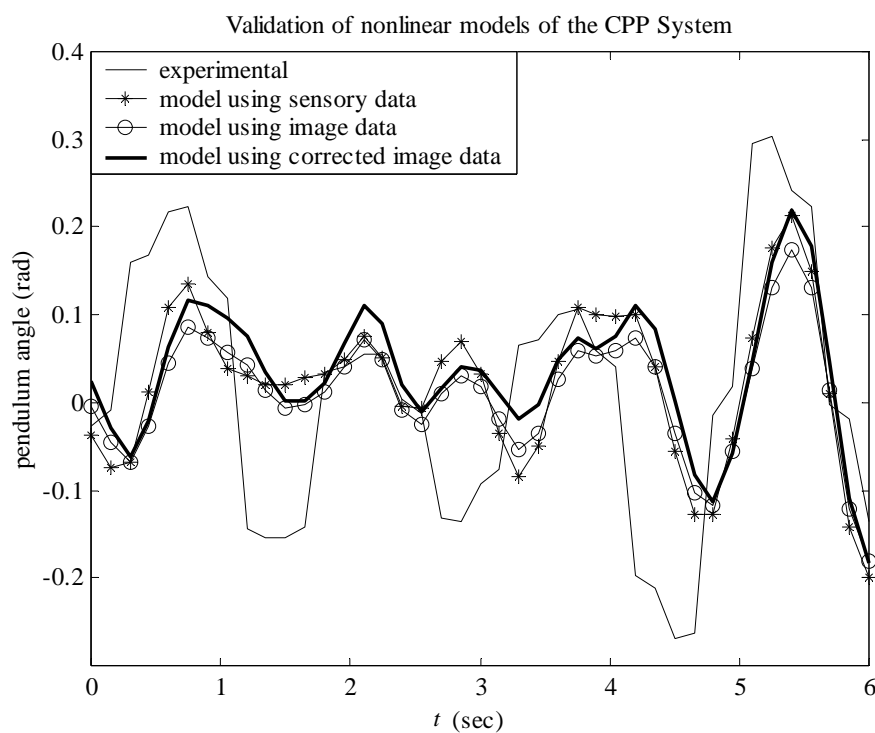
รูปที่ 7.18 ผลการระบุเอกลักษณ์ระบบเมื่อแรง f เป็นพหุนามของอินพุต u อันดับ 2 ถึง 7



รูปที่ 7.19 ผลการตรวจสอบความถูกต้องของแบบจำลองเมื่อแรง f เป็นพหุนามของอินพุต u อันดับ 2 ถึง 7



รูปที่ 7.20 ผลการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น



รูปที่ 7.21 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ CPP ไม่เชิงเส้น

ตารางที่ 7.6 ผลการระบุเอกลักษณ์ค่าพารามิเตอร์ของแบบจำลองระบบ CPP ไม่เชิงเส้น
เมื่อแรง f เป็นพหุนามของอินพุต u อันดับ 2 ถึง 7

พารามิเตอร์ที่ให้ ค่า cost น้อยที่สุด	อันดับของพหุนาม u					
	2	3	4	5	6	7
M (kg)	0.9010	0.9003	1.0824	1.0846	0.9017	0.9006
a_7	-	-	-	-	-	4.0937
a_6	-	-	-	-	6.7466	6.9857
a_5	-	-	-	-1.1121	-1.0759	-1.8275
a_4	-	-	-5.0151	-5.0322	-5.0804	-5.6048
a_3	-	-1.6827	-1.4650	-0.1482	-0.8459	-4.9867
a_2	-0.7785	-0.8527	4.8793	4.5146	-2.9384	-2.3259
a_1	0.6488	1.3533	0.9183	0.7550	1.6530	2.4923
a_0	-0.5537	-0.5308	-0.9996	-0.9988	-0.0889	-0.1482
ค่า cost น้อยที่สุด	1.5194	1.4629	1.8294	1.8355	1.3475	1.3114
ค่า cost มากที่สุด	1.5271	1.4697	1.9573	1.9570	1.4988	1.3140
ค่า cost เฉลี่ย	1.5225	1.4660	1.8489	1.8528	1.3598	1.3127
ค่า SD ของ cost	0.0021	0.0020	0.0346	0.0257	0.0328	0.0007
จำนวนรอบเฉลี่ย	1000	1000	1000	1000	1000	693.40
เวลาที่ใช้เฉลี่ย	94.6316	94.8314	95.7501	96.4392	96.7141	76.1837

- ลำดับคำตอบที่เลือกใช้เป็นค่าเริ่มต้นในกลไกย้อนรอยการค้นหาเท่ากับ -5
 - ค่ารัศมีการค้นหาหลังจากการปรับลดมีค่าเท่ากับ 20% ของค่ารัศมีการค้นหา
 - กลไกยุติการค้นหาที่กำหนดให้ค่า SSE เป็นค่า cost ซึ่งจะต้องน้อยกว่าหรือเท่ากับ 1.32 ($\text{cost} \leq 1.32$) หรือจำนวนรอบการค้นหาสูงสุดเท่ากับ 1,000 รอบ
 - ดำเนินการค้นหาทั้งหมด 20 ครั้ง (trial) เพื่อยืนยันคำตอบที่ได้จากการค้นหา
- รายละเอียดของโปรแกรมการค้นหาแบบตาบู่เชิงปรับตัว เพื่อพิจารณาอันดับที่เหมาะสมของแรง f ที่เป็นพหุนามของ u แสดงในภาคผนวก ง. ค่าพารามิเตอร์ที่จะได้รับการค้นหามีจำนวน 9 ตัวคือ M และ a_0, \dots, a_7 ปฏิบัติการค้นหาของค่าพารามิเตอร์แต่ละตัวมีขอบเขตในทางปฏิบัติดังนี้ $M = [+0.5, +1.5]$, $a_0 = [-1, +1]$, $a_1 = [-5, +5]$, $a_2 = [-5, +5]$, $a_3 = [-5, +5]$, $a_4 = [+6, +7]$, $a_5 = [-2, -1]$, $a_6 = [+6, +7]$, และ $a_7 = [+4, +5]$

การระบุเอกลักษณ์ระบบได้ดำเนินการบนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกกะเฮิร์ตซ์ ซึ่งมีหน่วยความจำ RAM ขนาด 256 เมกกะไบต์ ผลการระบุเอกลักษณ์ค่าพารามิเตอร์ของแบบจำลองระบบ CPP ไม่เชิงเส้น เมื่อแรง f เป็นพหุนามของอินพุต u อันดับ 2 ถึง 7 โดยใช้การค้นหาแบบตาบุงเชิงปรับตัว แสดงดังตารางที่ 7.6 จากการดำเนินการค้นหาทั้งหมด 20 ครั้ง ได้พิจารณาแบบจำลองที่ให้ค่า SSE น้อยที่สุด เพื่อแสดงผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง ดังแสดงในรูปที่ 7.18 และ 7.19 ซึ่งพบว่าแรง f ที่เป็นพหุนามของอินพุต u อันดับ 7 จะให้ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง ที่สอดคล้องกับผลการทดสอบมากที่สุด ทั้งนี้เนื่องจากค่า SSE (ค่า cost) ดังปรากฏในตารางที่ 7.6 มีค่าน้อยที่สุด ดังนั้นจึงพิจารณาให้แรง f เป็นพหุนามของอินพุต u อันดับ 7 ในการนำไปใช้ประโยชน์เพื่อการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นในขั้นต่อไป

7.3.2 การระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นโดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน

ในขั้นตอนนี้จะเป็นการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นโดยใช้ข้อมูลที่ได้จากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน จากแบบจำลองของระบบในสมการที่ (7-27) ถึง (7-29) การระบุเอกลักษณ์ระบบจะเป็นการค้นหาค่าพารามิเตอร์ M และ a_i เมื่อ $i = 0, \dots, 7$ โดยใช้การค้นหาแบบตาบุงเชิงปรับตัวที่พัฒนาด้วยโปรแกรม C ซึ่งรายละเอียดการกำหนดขอบเขตปริภูมิการค้นหาของพารามิเตอร์แต่ละตัว รัศมีการค้นหาเริ่มต้น จำนวนการสุ่มค่าใกล้เคียง กลไกย้อนรอยการค้นหา กลไกปรับรัศมีพื้นที่การค้นหา และกลไกยุติการค้นหา ยังคงใช้ตามที่ได้กำหนดไว้ในหัวข้อ 7.3.1 ดำเนินการค้นหาทั้งหมด 20 ครั้ง บนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกกะเฮิร์ตซ์ ซึ่งมีหน่วยความจำ RAM ขนาด 256 เมกกะไบต์ เพื่อยืนยันคำตอบที่ได้จากการค้นหา รายละเอียดของโปรแกรมการค้นหาแบบตาบุงเชิงปรับตัว เพื่อการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น โดยใช้ข้อมูลที่ได้จากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน แสดงในภาคผนวก ง.

ผลการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้นโดยอาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน โดยใช้การค้นหาแบบตาบุงเชิงปรับตัว แสดงดังตารางที่ 7.7 จากการดำเนินการค้นหาทั้งหมด 20 ครั้ง ได้พิจารณาแบบจำลองที่ให้ค่า SSE น้อยที่สุด เพื่อแสดงผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง ดังแสดงในรูปที่ 7.20 และ 7.21

จากรูปที่ 7.20 และ 7.21 ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลองเปรียบเทียบกับผลการทดสอบ มีความสอดคล้องกันพอสมควร ปรากฏ

ความคลาดเคลื่อนอยู่บ้าง ทั้งนี้อาจเป็นเพราะการพิจารณาให้แบบจำลองในส่วนของแรง f เป็นพหุนามอันดับ 7 ของอินพุต u อาจยังไม่แม่นยำพอ อย่างไรก็ตาม เมื่อพิจารณาผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง โดยใช้ข้อมูลจากตัวตรวจรู้ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน จากรูปที่ 7.20 และ 7.21 มีความใกล้เคียงกันอย่างมาก จากผลการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลองโดยการค้นหาแบบตามูเชิงปรับตัว ดังแสดงในตารางที่ 7.7 พบว่าการใช้ข้อมูลจากตัวตรวจรู้ และข้อมูลจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ให้ค่าพารามิเตอร์ที่มีความใกล้เคียงกันอย่างมาก โดยเฉพาะอย่างยิ่งค่า M , a_0 , a_1 , a_2 , a_3 , a_5 , และ a_6 ของแบบจำลองระบบ CPP ไม่เชิงเส้น แสดงให้เห็นว่าการระบุเอกลักษณ์ระบบโดยอาศัยเทคนิคการประมวลผลภาพที่นำเสนอ มีความถูกต้องแม่นยำเท่าเทียมกับการระบุเอกลักษณ์ระบบโดยอาศัยข้อมูลจากตัวตรวจรู้

7.4 สรุป

ในบทที่ 7 นี้ได้กล่าวถึงการระบุเอกลักษณ์ระบบ CPP เชิงเส้น และไม่เชิงเส้น สำหรับการระบุเอกลักษณ์ระบบ CPP เชิงเส้น เป็นการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม โดยใช้แบบจำลอง ARX ARMAX OE และ BJ ในรูปของฟังก์ชันถ่ายโอน การระบุเอกลักษณ์ระบบอาศัยโปรแกรม MATLAB และ System Identification Toolbox โดยใช้ข้อมูลจากตัวตรวจรู้จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน ซึ่งพบว่าแบบจำลอง BJ อันดับ 10 เป็นแบบจำลองที่มีความเหมาะสมสำหรับระบบ CPP เชิงเส้น ผลการระบุเอกลักษณ์แบบจำลอง BJ อันดับ 10 โดยอาศัยข้อมูลจากตัวตรวจรู้ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน มีความใกล้เคียงกันอย่างมาก ทั้งในส่วนของผลการระบุเอกลักษณ์ระบบ ผลการตรวจสอบความถูกต้องของแบบจำลอง โพลเด่น ซีโรเด่น และผลตอบสนองในโดเมนความถี่ สำหรับการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น เป็นการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง โดยอาศัยการค้นหาแบบตามูเชิงปรับตัว ที่พัฒนาขึ้นด้วยโปรแกรม C อาศัยข้อมูลจากตัวตรวจรู้ จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน เพื่อทำการระบุเอกลักษณ์ระบบ ซึ่งพบว่าผลการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น เมื่อใช้ข้อมูลจากตัวตรวจรู้ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน มีความใกล้เคียงกันอย่างมาก ทั้งในส่วนของผลการระบุเอกลักษณ์ระบบ ผลการตรวจสอบความถูกต้องของแบบจำลอง และค่าพารามิเตอร์ของแบบจำลองที่ได้จากการค้นหาจากผลการระบุเอกลักษณ์ระบบ CPP เชิงเส้นและไม่เชิงเส้น สามารถกล่าวได้ว่า การระบุเอกลักษณ์ระบบโดยอาศัยเทคนิคการประมวลผลภาพที่นำเสนอ มีความถูกต้องแม่นยำเท่าเทียมกับการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม

ตารางที่ 7.7 ผลการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น โดยใช้ข้อมูลจากตัวตรวจรู้ จากการ
ประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน

พารามิเตอร์ที่ให้ค่า cost น้อยที่สุด	ข้อมูลจาก ตัวตรวจรู้	ข้อมูลจากการ ประมวลผลภาพ	ข้อมูลจากการประมวลผลภาพที่ ได้รับการแก้ไขความคลาดเคลื่อน
M (kg)	0.9006	0.9584	0.9047
a_7	4.0937	4.1580	4.3577
a_6	6.9857	6.4098	6.9232
a_5	-1.8275	-1.9787	-1.9660
a_4	-5.6048	-5.6900	-5.2935
a_3	-4.9867	-4.6792	-4.9908
a_2	-2.3259	-2.0107	-2.5511
a_1	2.4923	2.3214	2.4656
a_0	-0.1482	-0.3462	-0.1401
ค่า cost น้อยที่สุด	1.3114	1.4018	1.3144
ค่า cost มากที่สุด	1.3140	1.7556	1.3637
ค่า cost เฉลี่ย	1.3127	1.5512	1.3377
ค่า SD ของ cost	0.0007	0.1042	0.0150
จำนวนรอบเฉลี่ย	693.40	1000	726.50
เวลาที่ใช้เฉลี่ย	76.1837	96.8401	78.0343

บทที่ 8

การทดสอบระบบกวัดแกว่งท่ออย่าง

8.1 กล่าวนำ

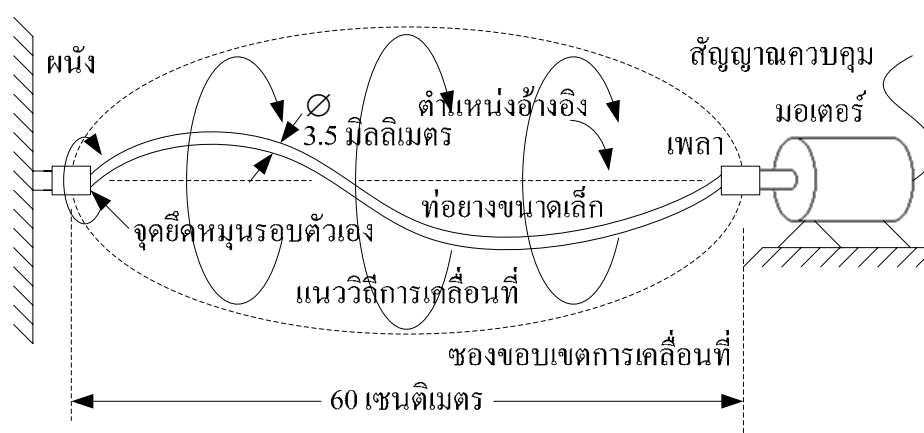
ระบบกวัดแกว่งท่ออย่าง หรือระบบ VTS (vibrating tube system) เป็นระบบพลวัตอีก ระบบหนึ่งที่ถูกใช้เพื่อการระบุเอกลักษณ์ระบบในงานวิจัยนี้ ดังนั้นเนื้อหาที่ปรากฏในบทที่ 8 จะกล่าวถึงการทดสอบระบบ VTS เพื่อบันทึกข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ โดยอาศัยการประมวลผลภาพที่พัฒนาขึ้น เนื่องจากระบบ VTS เป็นระบบที่มีการเคลื่อนที่แบบ 3 มิติ ดังนั้นเพื่อความครบถ้วนของข้อมูล การทดสอบระบบจะอาศัยกล้องวิดีโอดิจิทัลที่ผ่านการ ปรับเทียบด้วยเทคนิคที่เหมาะสมจำนวน 2 ตัว เพื่อบันทึกภาพการเคลื่อนที่ของระบบ การสกัด ข้อมูลพลวัตจากลำดับภาพที่ได้จากกล้องวิดีโอดิจิทัลทั้ง 2 ตัว จะอาศัยเทคนิคการกำหนด จุดเปลี่ยนค่า ข้อมูลอินพุตและเอาต์พุตของระบบ VTS จะได้รับการจัดสรรให้มีความถูกต้อง สมบูรณ์มากที่สุด สำหรับการระบุเอกลักษณ์ระบบต่อไป

8.2 การทดสอบระบบ VTS

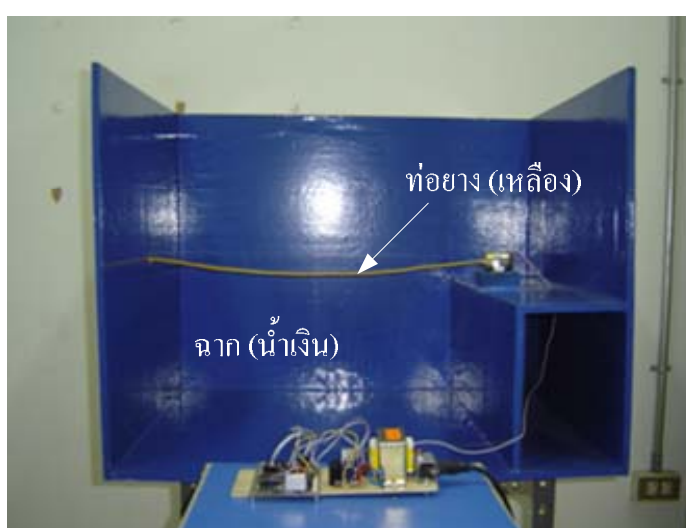
ระบบ VTS ดังแสดงในรูปที่ 8.1 เป็นระบบที่ได้รับการพัฒนาขึ้น ประกอบด้วยท่ออย่าง ขนาดความยาว 61 เซนติเมตร เส้นผ่าศูนย์กลาง 3.5 มิลลิเมตร ที่ได้รับการต่อปลายด้านหนึ่งเข้ากับ เผลาของมอเตอร์ไฟฟ้ากระแสตรง และต่อปลายอีกด้านหนึ่งเข้ากับจุดยึดซึ่งสามารถหมุนได้รอบตัว ความยาวจากเพลลาของมอเตอร์ถึงจุดยึดวัดได้ 60 เซนติเมตร การต่อท่ออย่างในลักษณะดังกล่าว ทำให้ปลายทั้งสองด้านของท่ออย่างไม่สามารถเคลื่อนที่ไปตามแนวอนได้ การเคลื่อนที่ของท่ออย่าง จะมีลักษณะการเคลื่อนที่แบบ 3 มิติที่ขึ้นกับแรงของมอเตอร์ที่ส่งผ่านมาตามเพลลา สัญญาณอินพุต ของระบบ VTS คือสัญญาณควบคุมมอเตอร์ไฟฟ้ากระแสตรง และสัญญาณเอาต์พุตของระบบ VTS คือการเคลื่อนที่ของท่ออย่าง ภาพถ่ายระบบ VTS ที่พัฒนาขึ้นแสดงดังรูปที่ 8.2

กล้องวิดีโอดิจิทัลที่ใช้บันทึกภาพการเคลื่อนที่ของระบบ VTS จำนวน 2 ตัว จะได้รับการ ติดตั้งตามตำแหน่งต่าง ๆ ที่เหมาะสม กล้องตัวที่ 1 ได้รับการติดตั้งในแนวแกน z (ด้านข้าง) เพื่อให้สามารถบันทึกการเคลื่อนที่ของระบบในระนาบ x, y และกล้องตัวที่ 2 ได้รับการติดตั้งใน แนวแกน y (ด้านบน) เพื่อให้สามารถบันทึกการเคลื่อนที่ของระบบในระนาบ z, x ดังรูปที่ 8.3 กล้องวิดีโอดิจิทัลทั้ง 2 ตัว รวมทั้งอุปกรณ์รับภาพ (frame grabber) และคอมพิวเตอร์ที่ใช้ จะต้อง มีคุณลักษณะสมบัติที่เหมือนกันทุกประการ นอกจากนี้การอ่านข้อมูลภาพผ่านอุปกรณ์รับภาพเข้าสู่

คอมพิวเตอร์ จะต้องกระทำผ่านสวิทช์ควบคุมเพียงตัวเดียว ทั้งนี้เพื่อประกันว่าภาพที่ได้จากกล้องวิดีโอจิจิตอลทั้ง 2 ตัว เป็นภาพการเคลื่อนที่ของระบบ VTS ในจังหวะเวลาเดียวกัน ดังนั้นงานวิจัยนี้จึงได้พิจารณาเลือกใช้กล้องวิดีโอจิจิตอลยี่ห้อ Sony Digital Handycam DCR-VX1000E ผนวกเลนส์ 5.9 ถึง 59 มิลลิเมตร อุปกรณ์รับภาพยี่ห้อ One Point รุ่น Cap IT2000 ซึ่งมีอัตราภาพสูงสุดเท่ากับ 25 ภาพต่อวินาที และคอมพิวเตอร์ Pentium 2 ความเร็ว 400 เมกกะเฮิร์ตซ์ ซึ่งมีหน่วยความจำ RAM ขนาด 64 เมกกะไบต์ การติดตั้งอุปกรณ์บันทึกภาพทั้ง 2 ชุด ซึ่งแต่ละชุดประกอบด้วยกล้องวิดีโอจิจิตอล อุปกรณ์รับภาพ และคอมพิวเตอร์ แสดงดังรูปที่ 8.4

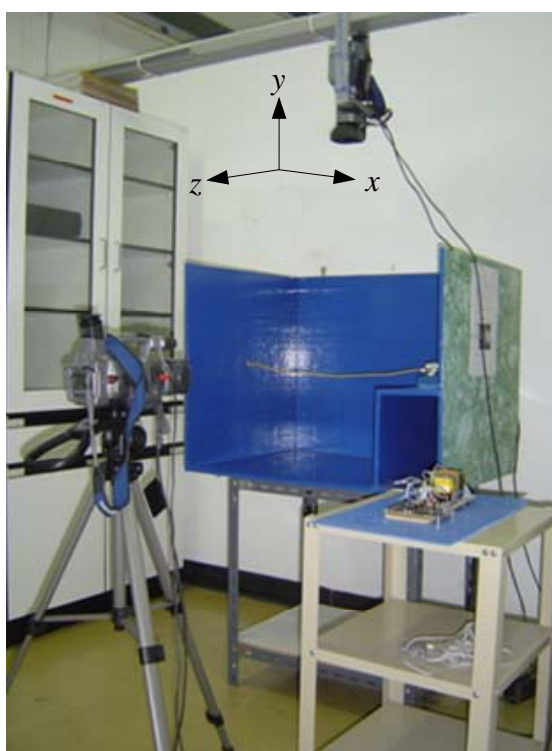


รูปที่ 8.1 ระบบ VTS

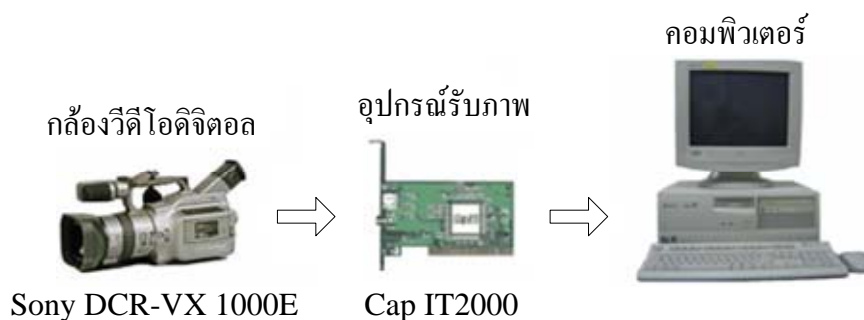


รูปที่ 8.2 ภาพถ่ายระบบ VTS

การควบคุมระบบ VTS ในงานวิจัยนี้ เป็นการควบคุมแบบวงรอบเปิด สัญญาณอินพุตที่ใช้ในการควบคุมระบบ เป็นสัญญาณทวิภาคสุ่มเทียมที่ผลิตจากไมโครคอนโทรลเลอร์ PIC16F877A ซึ่งมีขนาดความแยกซัด 8 บิต ทำงานบนอัตราสัญญาณนาฬิกา 20 เมกกะเฮิร์ตซ์ สัญญาณดังกล่าวจะได้รับการขับเร้าด้วยไอซีขับกำลัง L298 และถูกส่งไปขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง ซึ่งเป็นของบริษัท Fuji Micro รุ่น FM36E ขนาด 9,500 รอบต่อนาที 12 โวลต์ 65 มิลลิแอมแปร์ แรงจากเพลาของมอเตอร์จะส่งผ่านไปตามท่อยาง ทำให้ท่อยางเกิดการเคลื่อนที่ในปริภูมิ 3 มิติ หรืออาจเรียกว่าการเคลื่อนที่แบบ 3 มิติ (รายละเอียดของการควบคุมระบบ VTS แสดงในภาคผนวก จ.)



รูปที่ 8.3 การติดตั้งกล้องวิดีโอดิจิทัลเพื่อบันทึกภาพการเคลื่อนที่ของระบบ VTS



รูปที่ 8.4 อุปกรณ์บันทึกภาพ

ระบบ VTS จะได้รับการทดสอบเพื่อบันทึกข้อมูลอินพุตและเอาต์พุต โดยที่ข้อมูลอินพุตของระบบ VTS คือสัญญาณควบคุมมอเตอร์ไฟฟ้ากระแสตรง (สัญญาณทวิภาคสุ่มเทียม) จะได้รับการบันทึกโดยออสซิลโลสโคปแบบดิจิทัล ขณะที่ข้อมูลเอาต์พุตของระบบ VTS คือการเคลื่อนที่ของตัวอย่างแบบ 3 มิติ จะได้รับการบันทึกโดยกล้องวีดีโอดิจิทัลที่ผ่านการปรับเทียบด้วยเทคนิคที่เหมาะสมจำนวน 2 ตัว ข้อมูลภาพที่ได้จากกล้องวีดีโอดิจิทัลจะถูกอ่านผ่านอุปกรณ์รับภาพและส่งเข้าสู่คอมพิวเตอร์ในรูปแบบของไฟล์วีดีโอ (.avi) และจะได้รับการแยกแยะลำดับภาพ (.bmp) โดยอาศัยชุดคำสั่งที่พัฒนาขึ้นด้วยโปรแกรม MATLAB™ ภาพแต่ละภาพจะได้รับการคำนวณเพื่อสกัดข้อมูลพลวัต ซึ่งเป็นข้อมูลเอาต์พุตของระบบ (ตำแหน่งของตัวอย่าง) โดยใช้เทคนิคการกำหนดจุดเปลี่ยนค่า

จากรูปที่ 8.2 ระบบ VTS ได้รับการจัดทำสไลด์ให้กับองค์ประกอบต่าง ๆ ของระบบ ทั้งนี้เพื่อความสะดวกในการสกัดข้อมูลพลวัตด้วยเทคนิคการกำหนดจุดเปลี่ยนค่า ตัวอย่างซึ่งถือเป็นตัวแปรพลวัตของระบบที่สนใจได้รับการจัดทำเป็นสไลด์เลื่อน ขณะที่องค์ประกอบอื่น ๆ ของระบบได้รับการจัดทำเป็นสีน้ำเงิน

8.3 ขั้นตอนการทดสอบระบบ VTS

ขั้นตอนการทดสอบระบบ VTS เพื่อบันทึกข้อมูลอินพุตและเอาต์พุต มีรายละเอียดดังนี้

1. กำหนดให้สัญญาณอินพุตของระบบเป็นสัญญาณทวิภาคสุ่มเทียม ซึ่งมีคาบเวลาประมาณ 1.82 ชั่วโมง (รายละเอียดของการผลิตสัญญาณทวิภาคสุ่มเทียมแสดงในภาคผนวก จ.)
2. กำหนดเวลาที่ใช้ในการทดสอบมีค่าเท่ากับ 20 วินาที
3. ข้อมูลอินพุตของระบบ VTS จะได้รับการบันทึกโดยออสซิลโลสโคปแบบดิจิทัล
4. ข้อมูลเอาต์พุตของระบบ VTS จะได้รับการบันทึกโดยกล้องวีดีโอดิจิทัลที่ผ่านการปรับเทียบด้วยเทคนิคที่เหมาะสมจำนวน 2 ตัว (รายละเอียดการปรับเทียบกล้องวีดีโอดิจิทัลแสดงในภาคผนวก จ.)

การแยกแยะลำดับภาพ (.bmp) จากไฟล์วีดีโอที่ได้จากกล้องวีดีโอดิจิทัลทั้ง 2 ตัว อาศัยชุดคำสั่งที่พัฒนาขึ้นด้วยโปรแกรม MATLAB™ ดังรายละเอียดในภาคผนวก ข. และการสกัดข้อมูลพลวัตของระบบจากลำดับภาพ อาศัยเทคนิคการกำหนดจุดเปลี่ยนค่า ซึ่งมีรายละเอียดดังนี้

1. พิจารณาย่านของค่าระดับแม่สีของตัวอย่าง เพื่อกำหนดจุดเปลี่ยนค่าให้กับกลไกสกัดข้อมูลจากลำดับภาพ โดยใช้คำสั่ง `pixval` ของ MATLAB™ ซึ่งแสดงในตารางที่ 8.1
2. กำหนดจุดเปลี่ยนค่าให้กับตัวอย่างในแต่ละภาพ ที่ได้จากกล้องวีดีโอดิจิทัลแต่ละตัว โดยประยุกต์สมการที่ (4-5) และค่าระดับแม่สีจากตารางที่ 8.1 ดังสมการที่ (8-1) และ (8-2) ตามลำดับ

$$g_{tube_cam1}(x, y) = \begin{cases} 1, & \begin{cases} 115 < f_R(x, y) \leq 250 \\ 120 < f_G(x, y) \leq 255 \\ 50 < f_B(x, y) \leq 240 \end{cases} \\ 0, & otherwise \end{cases} \quad (8-1)$$

$$g_{tube_cam2}(x, y) = \begin{cases} 1, & \begin{cases} 75 < f_R(x, y) \leq 250 \\ 60 < f_G(x, y) \leq 255 \\ 20 < f_B(x, y) \leq 170 \end{cases} \\ 0, & otherwise \end{cases} \quad (8-2)$$

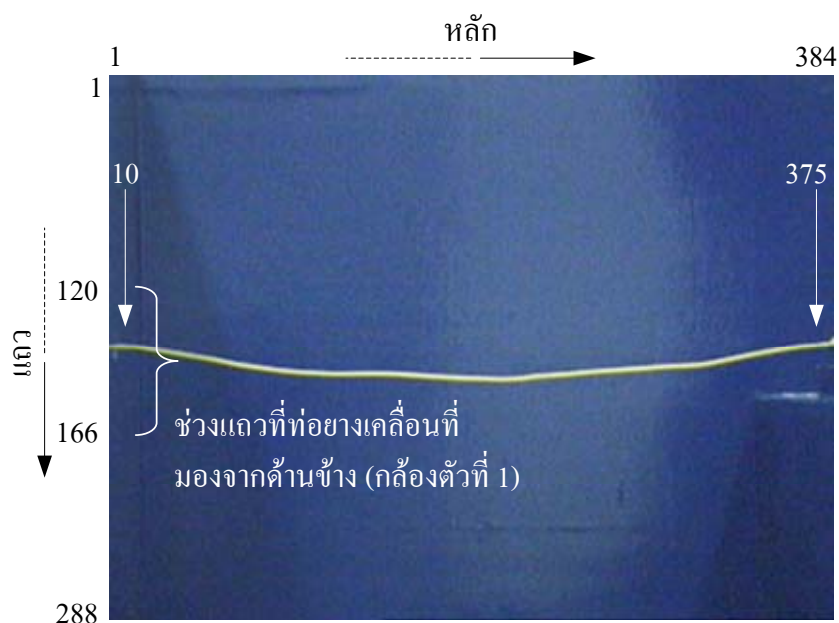
ตารางที่ 8.1 ค่าระดับแม่สีของท่อยางในระบบ VTS

วัตถุ	แถบสี	ค่าระดับแม่สี					
		กล้องวีดีโอดิจิตอลตัวที่ 1			กล้องวีดีโอดิจิตอลตัวที่ 2		
		R	G	B	R	G	B
ท่อยาง	เหลือง	115-250	120-255	50-240	75-250	60-255	20-170

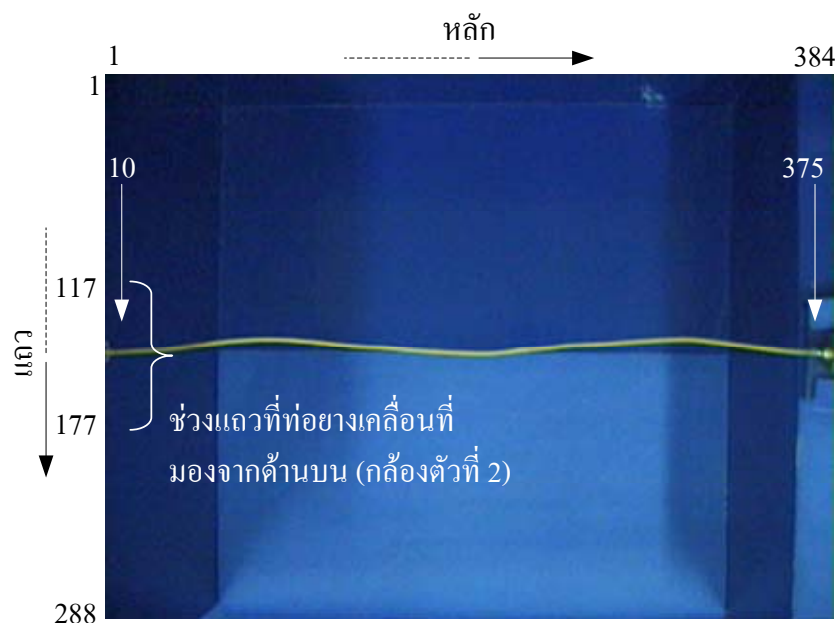
จากตารางที่ 8.1 และสมการที่ (8-1) และ (8-2) จะสังเกตได้ว่าย่านของค่าระดับแม่สีของท่อยางที่ได้จากกล้องวีดีโอดิจิตอลตัวที่ 1 และตัวที่ 2 มีค่าไม่เท่ากัน ทั้งนี้เนื่องจากกล้องวีดีโอดิจิตอลทั้ง 2 ตัว ที่ได้รับการติดตั้งอยู่ตามตำแหน่งต่าง ๆ ของระบบ ได้รับผลกระทบจากแสงภายนอก เงาม และการสะท้อนแสงที่ต่างกัน ทำให้ความสว่าง ความชัดเจน และความเปรียบต่างของภาพที่ได้จากกล้องวีดีโอดิจิตอลทั้ง 2 ตัวมีค่าไม่เท่ากัน ซึ่งสะท้อนออกมาด้วยค่าระดับแม่สีที่แตกต่างกันดังกล่าวข้างต้น

3. ขนาดของภาพสีดิจิตอลที่ใช้มีค่าเท่ากับ 288×384 จุดภาพ เมื่อพิจารณาจากลำดับภาพทั้งหมดที่ได้จากกล้องวีดีโอดิจิตอลตัวที่ 1 พบว่าท่อยางจะอยู่ในช่วงคอลัมน์ที่ 10 ถึง 375 และจะมีการเคลื่อนที่อยู่ในช่วงแถวที่ 120 ถึง 166 ดังแสดงในรูปที่ 8.5 และเมื่อพิจารณาจากลำดับภาพทั้งหมดที่ได้จากกล้องวีดีโอดิจิตอลตัวที่ 2 พบว่าท่อยางจะอยู่ในช่วงคอลัมน์ที่ 10 ถึง 375 และจะมีการเคลื่อนที่อยู่ในช่วงแถวที่ 117 ถึง 177 ดังแสดงในรูปที่ 8.6 ดังนั้นจึงกำหนดให้การสกัดข้อมูลตำแหน่งของท่อยางพิจารณาเฉพาะช่วงแถวและคอลัมน์ที่ระบุข้างต้น ทั้งนี้เพื่อการประมวลผลภาพที่รวดเร็วยิ่งขึ้น

4. ตำแหน่งอ้างอิงของท่อยาง คือ ตำแหน่งปลายของท่อยางด้านใดด้านหนึ่ง ซึ่งอาจพิจารณาจากระดับ (แถว) อ้างอิงที่ลากจากเพลลาของมอเตอร์ไปยังจุดยึด ดังแสดงในรูปที่ 8.1



รูปที่ 8.5 ช่วงแถวที่ท่อยางเคลื่อนที่เมื่อพิจารณาจากด้านข้างโดยกล้องวิดีโอดิจิทัลตัวที่ 1



รูปที่ 8.6 ช่วงแถวที่ท่อยางเคลื่อนที่เมื่อพิจารณาจากด้านบนโดยกล้องวิดีโอดิจิทัลตัวที่ 2

5. เมื่อใช้เทคนิคการกำหนดจุดเปลี่ยนค่าหากกลุ่มจุดภาพที่เป็นท่อยางได้แล้ว จะทำการคำนวณหาจุดกึ่งกลางของกลุ่มจุดภาพที่เป็นท่อยางในแต่ละคอลัมน์ ดังสมการที่ (8-3) เมื่อ r_{min} คือ ตำแหน่งแถวที่เป็นจุดสูงสุดของท่อยาง r_{max} คือ ตำแหน่งแถวที่เป็นจุดต่ำสุดของท่อยาง และ T'_j คือ จุดกึ่งกลางของกลุ่มจุดภาพที่เป็นท่อยางในแต่ละคอลัมน์

$$T'_j = \begin{cases} \frac{r_{\max} - r_{\min}}{2} + r_{\min}, & r_{\max} \neq r_{\min} \\ r_{\max}, & \text{otherwise} \end{cases}, \quad j = 1, \dots, 366 \quad (8-3)$$

6. จากการคำนวณด้วยสมการที่ (8-3) ทำให้ทราบตำแหน่งของท่อยางในภาพ จากนั้นทำการคำนวณตำแหน่งของท่อยางเปรียบเทียบกับตำแหน่งอ้างอิง ดังสมการที่ (8-4) เมื่อ T'_1 คือ ตำแหน่งของท่อยางคอลัมน์ที่ 1 ซึ่งเป็นตำแหน่งของท่อยางที่อยู่ติดกับจุดยึด และถือเป็นตำแหน่งอ้างอิง ขณะที่ T_j คือ ตำแหน่งของท่อยางเมื่อเปรียบเทียบกับตำแหน่งอ้างอิง

$$T_j = -1.6257(T'_j - T'_1), \quad j = 1, \dots, 366 \quad (8-4)$$

ค่า 1.6257 คือ ค่าคงที่ที่ได้มาจากการพิจารณาระยะห่างจากปลายของจุดยึดถึงปลายของเพลมอเตอร์ ซึ่งมีระยะทางเท่ากับ 595 มิลลิเมตร พิจารณาเป็นจำนวนจุดภาพได้เท่ากับ 366 จุดภาพ ดังนั้น 1 จุดภาพจึงมีค่าเท่ากับระยะทาง 1.6257 มิลลิเมตร

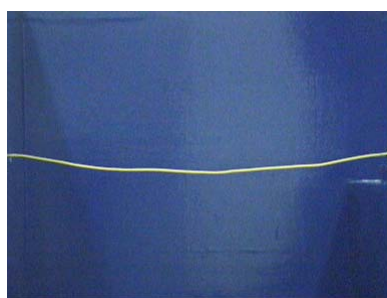
7. เนื่องจากข้อมูลตำแหน่งของท่อยางในแต่ละภาพมีจำนวนมากถึง 366 ข้อมูล (จุดภาพ) ดังนั้นจึงทำการจัดสรรข้อมูลตำแหน่งของท่อยางให้เหลือเพียง 61 ข้อมูล โดยจะพิจารณาจัดเก็บข้อมูลตำแหน่งของท่อยางในแต่ละภาพทุก ๆ 6 จุดภาพ

โปรแกรม MATLAB™ ที่ได้รับการพัฒนาขึ้นเพื่อการสกัดข้อมูลตำแหน่งของท่อยาง โดยอาศัยขั้นตอนดังกล่าวข้างต้น แสดงในภาคผนวก ข.

รูปที่ 8.7 แสดงตัวอย่างผลการคำนวณตำแหน่งของท่อยางจากภาพที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 1 โดยอาศัยโปรแกรมที่พัฒนาขึ้น จากการปรับเทียบกล้องในภาคผนวก ฉ. พบว่าภาพที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 1 จะปรากฏความเพี้ยนรูปหมอนปักเข็ม (pincushion distortion) ดังนั้นภาพระบบ VTS ที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 1 จะปรากฏความเพี้ยนดังกล่าวขึ้น ดังแสดงในรูปที่ 8.7 (ก) ภาพระบบ VTS ที่ได้รับการแก้ไขความเพี้ยนแล้ว แสดงดังรูปที่ 8.7 (ข) (รายละเอียดการปรับเทียบกล้องวิดีโอ ดิจิตอล และการแก้ไขความเพี้ยน แสดงในภาคผนวก ฉ.) ผลการสกัดข้อมูลตำแหน่งของท่อยางในช่วงแถวและคอลัมน์ที่ท่อยางเคลื่อนที่ปรากฏเป็นกลุ่มจุดภาพ ดังแสดงในรูปที่ 8.7 (ค) ผลการคำนวณข้อมูลตำแหน่งของท่อยาง แสดงในรูปที่ 8.7 (ง) และข้อมูลตำแหน่งของท่อยางที่ได้รับการจัดสรรให้เหลือเพียง 61 ข้อมูล แสดงในรูปที่ 8.7 (จ)

รูปที่ 8.8 แสดงตัวอย่างผลการคำนวณตำแหน่งของท่อยางจากภาพที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 2 โดยอาศัยโปรแกรมที่พัฒนาขึ้น และจากการปรับเทียบกล้องวิดีโอ ดิจิตอล ในภาคผนวก ฉ. พบว่าภาพที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 2 จะปรากฏความเพี้ยนรูปดัง

(barrel distortion) ดังนั้นภาพระบบ VTS ที่ได้จากกล้องวิดีโอดิจิทัลตัวที่ 2 จะปรากฏความเพี้ยนดังกล่าวขึ้น ดังแสดงในรูปที่ 8.8 (ก) ภาพระบบ VTS ที่ได้รับการแก้ไขความเพี้ยนแล้ว แสดงดังรูปที่ 8.8 (ข) ผลการสกัดข้อมูลตำแหน่งของท่อภายในช่วงแถวและคอลัมน์ที่ท่อเคลื่อนที่ปรากฏเป็นกลุ่มจุดภาพ ดังแสดงในรูปที่ 8.8 (ค) ผลการคำนวณข้อมูลตำแหน่งของท่อ แสดงในรูปที่ 8.8 (ง) และข้อมูลตำแหน่งของท่อที่ได้รับการจัดสรรให้เหลือเพียง 61 ข้อมูล แสดงในรูปที่ 8.8 (จ)



(ก)



(ข)



(ค)

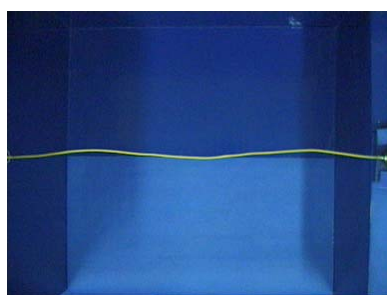


(ง)

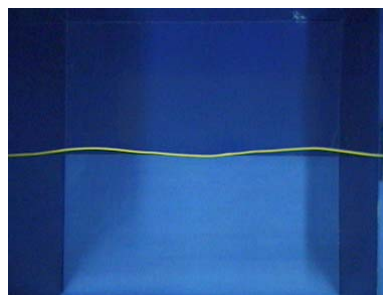


(จ)

รูปที่ 8.7 ตัวอย่างผลการคำนวณตำแหน่งของท่อจากกล้องตัวที่ 1
 (ก) ภาพปรากฏความเพี้ยน (ข) ภาพภายหลังแก้ไขความเพี้ยน
 (ค) ผลการสกัดข้อมูล (ง) ผลการคำนวณตำแหน่งของท่อ
 (จ) จัดสรรข้อมูลตำแหน่งของท่อ



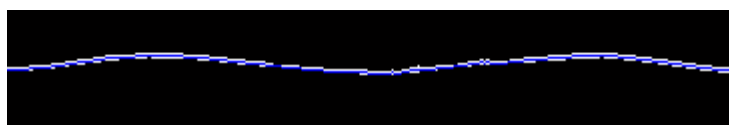
(ก)



(ข)



(ค)



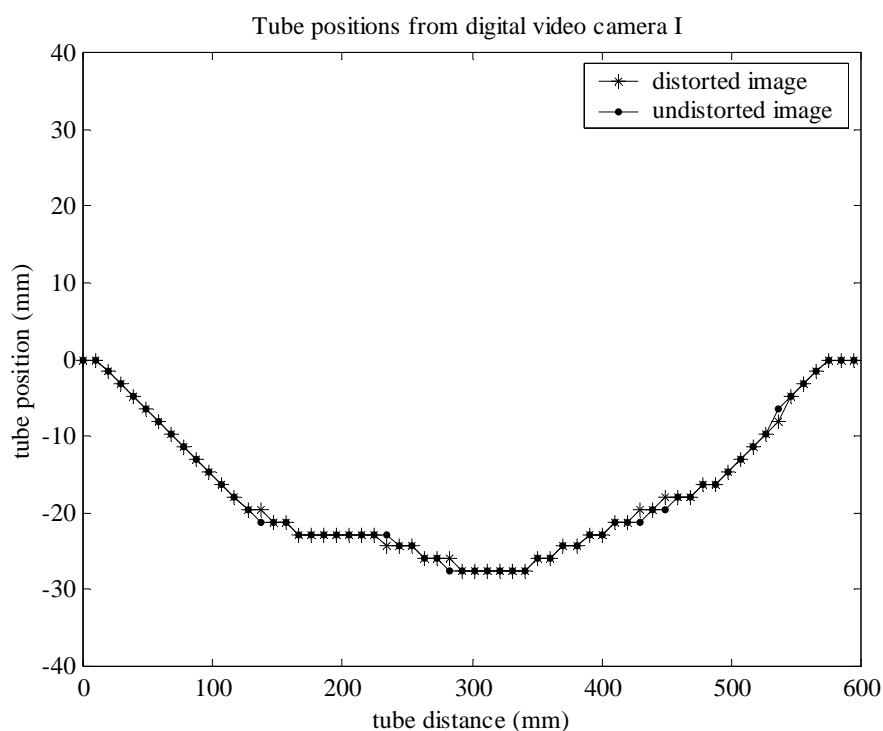
(ง)



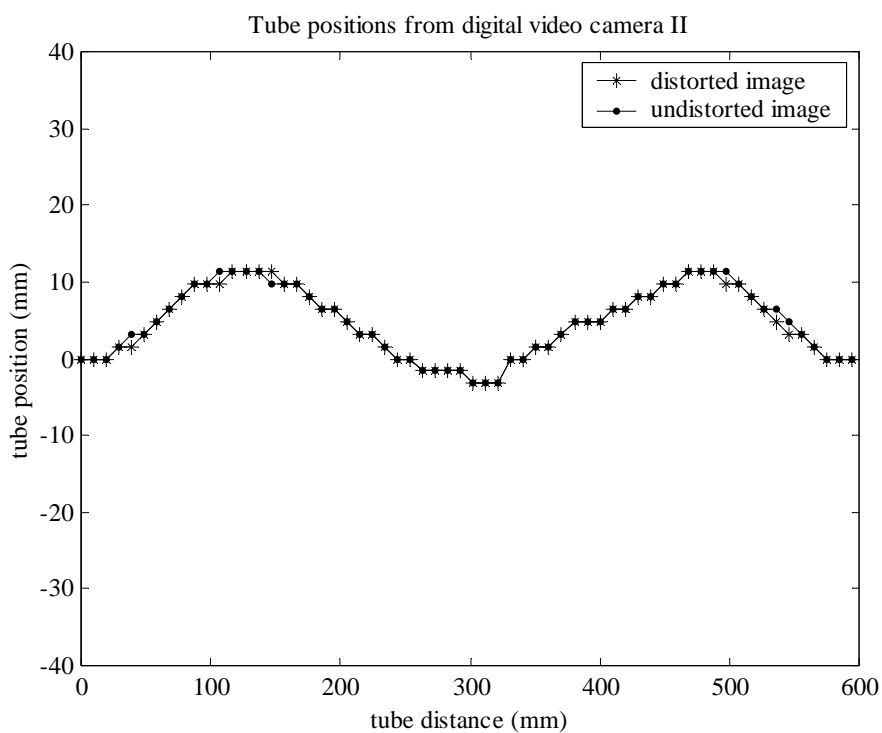
(จ)

รูปที่ 8.8 ตัวอย่างผลการคำนวณตำแหน่งของท่อยางจากกล้องตัวที่ 2
 (ก) ภาพปรากฏความเพี้ยน (ข) ภาพภายหลังแก้ไขความเพี้ยน
 (ค) ผลการสกัดข้อมูล (ง) ผลการคำนวณตำแหน่งของท่อยาง
 (จ) จัดสรรข้อมูลตำแหน่งของท่อยาง

รูปที่ 8.9 และ 8.10 แสดงผลการคำนวณตำแหน่งของท่อยางจากภาพที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 1 และตัวที่ 2 ทั้งก่อนและหลังการแก้ไขความเพี้ยน ซึ่งพบว่าข้อมูลตำแหน่งของท่อยาง มีความใกล้เคียงกันอย่างมาก แสดงให้เห็นว่าก่อนได้รับการแก้ไขความเพี้ยน กล้องวิดีโอดิจิตอล ทั้ง 2 ตัวมีความเพี้ยนปรากฏอยู่เพียงเล็กน้อย



รูปที่ 8.9 ตำแหน่งของท่อจากกล้องวิดีโอดิจิทัลตัวที่ 1 ก่อนและหลังการแก้ไขความเพี้ยน

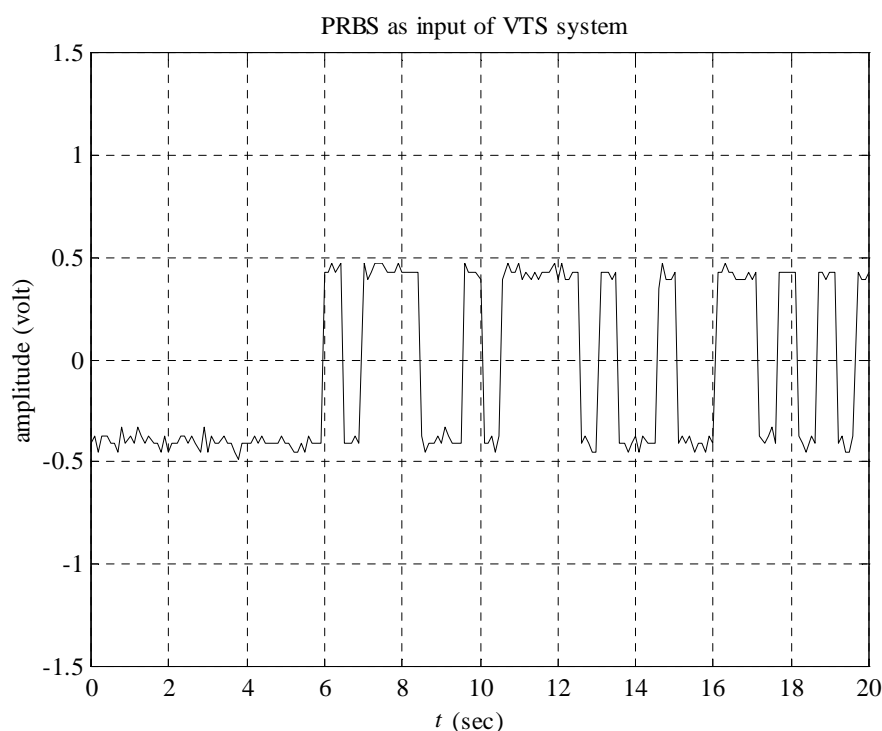


รูปที่ 8.10 ตำแหน่งของท่อจากกล้องวิดีโอดิจิทัลตัวที่ 2 ก่อนและหลังการแก้ไขความเพี้ยน

8.4 ผลการทดสอบระบบ VTS

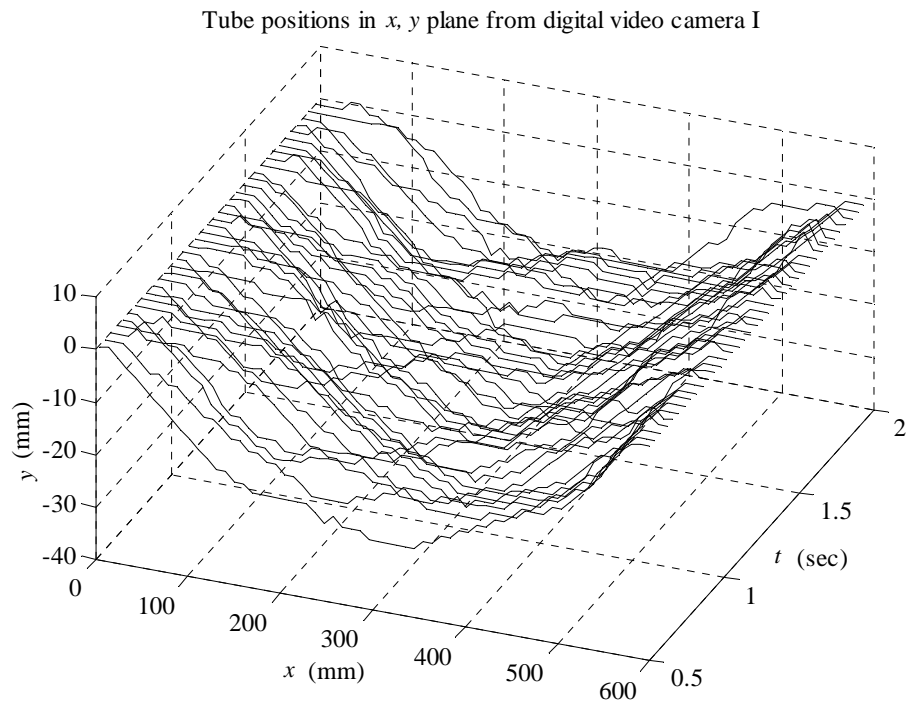
ผลการทดสอบระบบ VTS และการอภิปรายผล มีรายละเอียดดังต่อไปนี้

ข้อมูลอินพุตของระบบ VTS คือสัญญาณทวิภาคสุ่มเทียม (PRBS) ที่ผลิตจากไมโครคอนโทรลเลอร์ PIC16F877A ซึ่งถูกใช้เป็นสัญญาณอินพุตของมอเตอร์ไฟฟ้ากระแสตรง สัญญาณดังกล่าวได้รับการบันทึกด้วยออสซิลโลสโคปแบบดิจิทัล จากนั้นจึงได้นำข้อมูลอินพุตมาจัดสรรและแสดงผลด้วย MATLAB ดังรูปที่ 8.11 ซึ่งจะเห็นได้ว่าสัญญาณอินพุตของระบบมีลักษณะเป็นลำดับพัลส์สองระดับ (ประมาณ ± 0.45 โวลต์) ที่ปรากฏการสุ่มเชิงความกว้าง และมีลักษณะเป็นรายคาบที่ยาวนานมากถึง 1.82 ชั่วโมง (รายละเอียดของการผลิตสัญญาณทวิภาคสุ่มเทียมแสดงในภาคผนวก จ.)

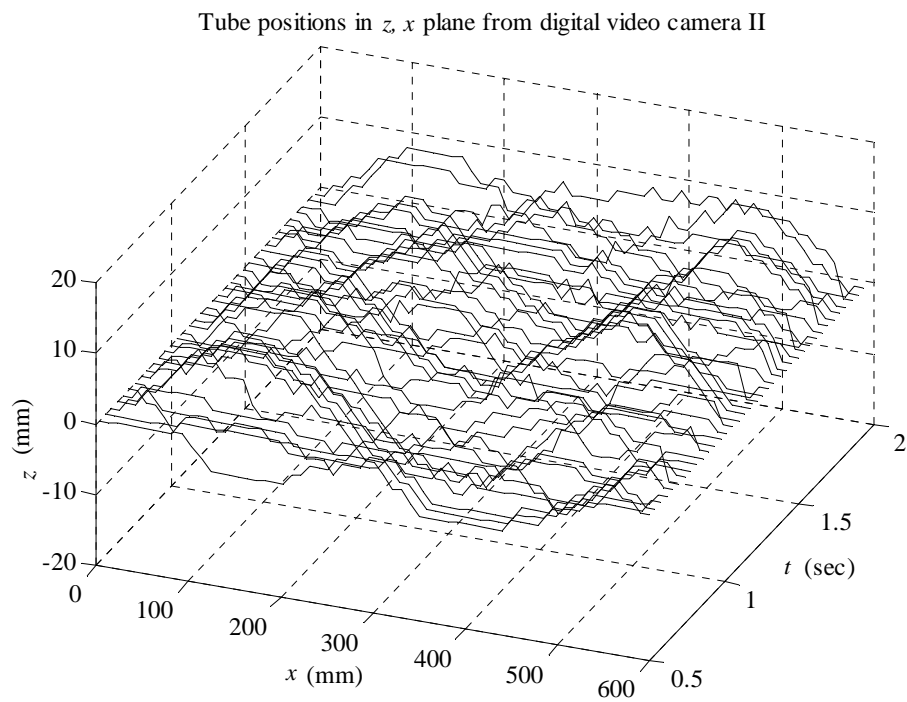


รูปที่ 8.11 สัญญาณอินพุตของระบบ VTS

ข้อมูลเอาต์พุตของระบบ VTS คือการเคลื่อนที่ของท่อยางแบบ 3 มิติ ซึ่งได้รับการบันทึกโดยกล้องวิดีโอดิจิทัลจำนวน 2 ตัวที่ผ่านการปรับเทียบด้วยเทคนิคที่เหมาะสม ข้อมูลเอาต์พุตของระบบได้รับการสกัดจากลำดับภาพโดยอาศัยเทคนิคการกำหนดจุดเปลี่ยนค่า จากผลการทดสอบ การเคลื่อนที่ของท่อยางในระนาบ x, y เทียบกับเวลา t (ข้อมูลจากกล้องวิดีโอดิจิทัลตัวที่ 1 มองจากด้านข้าง) แสดงในรูปที่ 8.12 และการเคลื่อนที่ของท่อยางในระนาบ z, x เทียบกับเวลา t (ข้อมูลจากกล้องวิดีโอดิจิทัลตัวที่ 2 มองจากด้านบน) แสดงในรูปที่ 8.13 ตามลำดับ



รูปที่ 8.12 การเคลื่อนที่ของท่ออย่างในระนาบ x, y เทียบกับเวลา t



รูปที่ 8.13 การเคลื่อนที่ของท่ออย่างในระนาบ z, x เทียบกับเวลา t

จากรูปที่ 8.12 ถึง 8.13 การเคลื่อนที่ของท่อภายในช่วงเวลา 0.5 ถึง 2 วินาที ได้รับการแสดงไว้เพื่อความชัดเจน จะเห็นได้ว่าการเคลื่อนที่ของระบบ VTS ในระนาบ x, y จะอยู่ในช่วง +5 ถึง -30 มิลลิเมตร ขณะที่การเคลื่อนที่ของระบบในระนาบ z, x จะอยู่ในช่วง ± 20 มิลลิเมตร การเคลื่อนที่ดังกล่าวได้รับการบันทึกโดยกล้องวิดีโอดิจิทัลจำนวน 2 ตัว ซึ่งมีอัตราภาพเท่ากับ 25 ภาพต่อวินาที จากการทดสอบด้วยระยะเวลา 20 วินาที ทำให้ข้อมูลเอาต์พุตที่ได้จากกล้องวิดีโอดิจิทัลแต่ละตัวมีจำนวน 30,500 ข้อมูล การพิจารณาความสัมพันธ์ระหว่างอัตราภาพของกล้องวิดีโอดิจิทัลที่ใช้ และทฤษฎีบทการชักตัวอย่าง มีรายละเอียดดังต่อไปนี้

ในกรณีที่พลวัตของระบบต่อเนื่องทางเวลามีการหมุนรอบของท่อด้วยความถี่สูงสุดเป็น f ความถี่ดังกล่าวจะได้รับการพิจารณาให้เป็นความถี่ไนควิสต์ (Nyquist frequency) f_N จากทฤษฎีบทการชักตัวอย่างของแชนนอน (Shannon's sampling frequency) (Jerri, 1977) ความถี่ของการชักตัวอย่าง (sampling frequency) f_s จะต้องมีค่าเป็น $f_s \geq 2f_N$ เพื่อป้องกันการเคลื่อนแปง (alias) ซึ่งในบางกรณี ความถี่ของการชักตัวอย่างควรมีค่าเป็น $f_s \gg 2f_N$ เพื่อประกันว่าข้อมูลที่ได้ สามารถใช้แทนพลวัตของระบบได้อย่างแท้จริง

เมื่อนำทฤษฎีบทการชักตัวอย่างมาพิจารณากับกล้องวิดีโอดิจิทัลซึ่งทำหน้าที่เป็นตัวตรวจรู้ความถี่ของการชักตัวอย่าง f_s ของกล้องกำหนดให้เป็น f_c ซึ่งจะมีค่าเท่ากับอัตราภาพ และเมื่อนำกล้องดังกล่าวไปบันทึกภาพการเคลื่อนที่ของระบบ VST ซึ่งมีพลวัตเป็นรายคาบ (ในช่วง 5 ถึง 6 วินาทีแรกที่สัญญาณอินพุตมีลักษณะเป็นสัญญาณขั้นบันได ดังรูปที่ 8.11) ปัญหาที่น่าสนใจขณะนี้คือ พลวัตของระบบ VST จะต้องมีค่าความถี่สูงสุด f_N ไม่เกินเท่าใด ที่จะทำให้กล้องวิดีโอดิจิทัลสามารถบันทึกภาพการเคลื่อนที่ของระบบได้สอดคล้องกับทฤษฎีบทการชักตัวอย่างของแชนนอน

พิจารณากำหนดความสัมพันธ์ระหว่าง f_c กับ f_N เป็นดังสมการที่ (8-5)

$$f_c = 10f_N \quad (8-5)$$

เมื่อ f_c คือ ความถี่ของการชักตัวอย่างของกล้อง และ f_N คือ ความถี่สูงสุดของระบบ VTS

จากสมการที่ (8-5) ถ้าความถี่ของการชักตัวอย่างของกล้อง f_c เท่ากับ 25 เฮิร์ตซ์ (25 ภาพต่อวินาที) ระบบ VTS จะต้องมีค่าความถี่รอบเท่ากับ 150 รอบต่อวินาที หรือจะต้องมีความถี่ f_N เท่ากับ 2.5 เฮิร์ตซ์ ซึ่งจะทำให้กล้องวิดีโอดิจิทัลสามารถบันทึกภาพการเคลื่อนที่ของระบบ VTS ได้ 10 ภาพต่อการเคลื่อนที่ครบ 1 รอบของท่อภายใน

อย่างไรก็ตาม จากการทดสอบระบบ VTS เพื่อพิจารณาความถี่รอบและความถี่ของการเคลื่อนที่ พบว่าเมื่อกำหนดให้ระบบ VTS เคลื่อนที่ด้วยความถี่รอบเท่ากับ 150 รอบต่อวินาที การเคลื่อนที่ของระบบ VTS จะไม่ราบเรียบ ทั้งนี้เนื่องจากแรงที่ส่งไปขับเคลื่อนเพื่อบังคับให้

ท่อยางเคลื่อนที่มีค่าน้อยเกินไป จากการทดสอบพบว่า ระบบ VTS จะมีการเคลื่อนที่อย่างราบเรียบ และสม่ำเสมอที่ความเร็วรอบมากกว่า 400 รอบต่อนาที ดังนั้นจึงกำหนดให้ระบบ VTS เคลื่อนที่ด้วยความเร็วรอบประมาณ 420 รอบต่อนาที หลังจากทำการทดสอบระบบ VTS จำนวน 10 ครั้ง ด้วยสัญญาณอินพุตดังรูปที่ 8.11 เพื่อหาค่าความเร็วรอบเฉลี่ย ค่าความเร็วรอบของระบบ VTS ที่บันทึกได้ในช่วง 5 ถึง 6 วินาทีแรก มีค่าเท่ากับ 416.5, 406.7, 408.3, 414.6, 414.3, 418.8, 431.2, 445.2, 457.4, และ 425.6 รอบต่อนาที เมื่อคำนวณค่าความเร็วรอบเฉลี่ยของระบบ VTS พบว่าระบบมีค่าความเร็วรอบเฉลี่ยเท่ากับ 423.86 รอบต่อนาที หรือมีค่าความถี่ของการเคลื่อนที่เฉลี่ยเท่ากับ 7.06 เฮิรตซ์

โดยอาศัยทฤษฎีบทการชักตัวอย่างของแซนนอน เมื่อระบบ VST มีความถี่ของการเคลื่อนที่เฉลี่ย f_N เท่ากับ 7.06 เฮิรตซ์ และความถี่ของการชักตัวอย่างของกล็อง f_c เท่ากับ 25 เฮิรตซ์ ความสัมพันธ์ระหว่าง f_c กับ f_N จะมีค่าเป็นดังสมการที่ (8-6) ซึ่งสอดคล้องกับทฤษฎีบทการชักตัวอย่างของแซนนอน ($f_c \geq 2f_N$) กล่าวคือ ถ้าความถี่ของการชักตัวอย่างของกล็อง f_c เท่ากับ 25 เฮิรตซ์ และระบบ VTS มีค่าความเร็วรอบเฉลี่ยเท่ากับ 423.86 รอบต่อนาที หรือมีค่าความถี่ของการเคลื่อนที่เฉลี่ย f_N เท่ากับ 7.06 เฮิรตซ์ จะส่งผลทำให้กล็องวีดิโอดิจิทัลสามารถบันทึกภาพการเคลื่อนที่ของระบบ VTS ได้ 3.54 ภาพต่อการเคลื่อนที่ครบ 1 รอบของท่อยาง

$$f_c = 3.54f_N \quad (8-6)$$

จากขั้นตอนการทดสอบ ผลการทดสอบ และการอภิปรายผลที่กล่าวมา แสดงให้เห็นว่าผลการทดสอบระบบ VTS ที่ได้มีความถูกต้องสมบูรณ์อย่างน่าพึงพอใจ

8.5 สรุป

ในบทที่ 8 นี้ ได้กล่าวถึงการทดสอบระบบ VTS เพื่อบันทึกข้อมูลอินพุตและเอาต์พุต การทดสอบกำหนดให้ข้อมูลอินพุตของระบบคือสัญญาณควบคุมมอเตอร์ไฟฟ้ากระแสตรง ซึ่งเป็นสัญญาณทวิภาคสี่เหลี่ยมที่ผลิตขึ้นจากไมโครคอนโทรลเลอร์ PIC16F877A และข้อมูลเอาต์พุตของระบบคือการเคลื่อนที่ของท่อยางแบบ 3 มิติ การเคลื่อนที่ดังกล่าวได้รับการบันทึกโดยกล็องวีดิโอดิจิทัลที่ผ่านการปรับเทียบด้วยเทคนิคที่เหมาะสมจำนวน 2 ตัว กล็องวีดิโอดิจิทัลที่ใช้มีอัตราภาพสอดคล้องกับทฤษฎีบทการชักตัวอย่างของแซนนอน การสกัดข้อมูลเอาต์พุตของระบบจากลำดับภาพที่ได้จากกล็องทั้ง 2 ตัว อาศัยเทคนิคการกำหนดจุดเปลี่ยนค่า ซึ่งให้ผลออกมาเป็นที่น่าสนใจ ข้อมูลดังกล่าวจะได้รับการนำไปคำนวณเพื่อระบุเอกลักษณ์ระบบ VTS ดังรายละเอียดที่จะปรากฏในบทต่อไป

บทที่ 9

การระบุเอกลักษณ์ระบบกวัดแกว่งท่อยาง

9.1 กล่าวนำ

ในบทที่ 9 นี้ จะกล่าวถึงการระบุเอกลักษณ์ระบบกวัดแกว่งท่อยาง หรือระบบ VTS ระบบดังกล่าวเป็นตัวแทนของระบบพลวัตที่มีการเคลื่อนที่แบบ 3 มิติ การระบุเอกลักษณ์ระบบ VTS แบ่งออกเป็นสองส่วนด้วยกันคือ การระบุเอกลักษณ์ระบบในระนาบ x, y และการระบุเอกลักษณ์ระบบในระนาบ z, x การระบุเอกลักษณ์ระบบทั้งสองส่วนอาศัยการค้นหาแบบตามเชิงปรับตัวเพื่อการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลองทางคณิตศาสตร์ ซึ่งอยู่ในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก การคำนวณผลเฉลยของแบบจำลองอาศัยการคำนวณเชิงตัวเลขตามระเบียบวิธีผลต่างสืบเนื่อง (finite difference method) ข้อมูลที่ใช้สำหรับการระบุเอกลักษณ์ระบบ VTS คือข้อมูลที่ได้จากการทดสอบระบบในบทที่ 8 ดำเนินงานระบุเอกลักษณ์ระบบโดยโปรแกรม MATLAB ขั้นตอนและการอภิปรายผลการระบุเอกลักษณ์ระบบ VTS ได้แสดงไว้ในบทที่ 9 นี้ ซึ่งพบว่าการระบุเอกลักษณ์ระบบ VTS ทั้งสองส่วนให้ผลเป็นที่น่าพอใจยิ่ง

9.2 การระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y

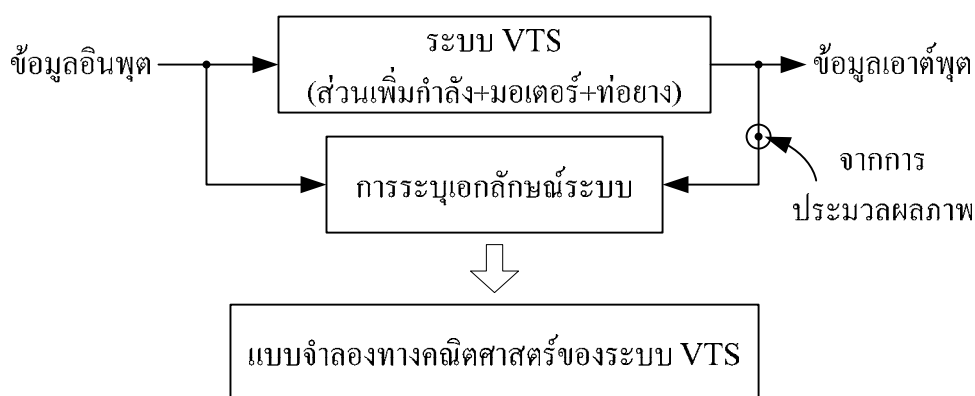
เมื่อพิจารณาระบบ VTS จากรูปที่ 8.1 ในบทที่ 8 พบว่าแบบจำลองทางคณิตศาสตร์ของระบบที่แสดงการเคลื่อนที่ของท่อยางในระนาบ x, y แสดงดังสมการที่ (9-1) แบบจำลองดังกล่าวได้รับการพัฒนาขึ้นโดยอาศัยหลักการสร้างแบบจำลองเส้นลวดสั่น (vibrating string model) ในรูปของสมการคลื่น (wave equation) ซึ่งถือเป็นสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก (hyperbolic equation) ที่พิจารณาผลของความเร่งเนื่องจากแรงโน้มถ่วง โดยกำหนดข้อตกลงเบื้องต้นให้ท่อยางไม่มีการยึดหยุ่นภายใน (มีคุณสมบัติเช่นเดียวกับเส้นลวด) ท่อยางมีเนื้อเดียวกันตลอดความยาว และไม่พิจารณาผลกระทบเนื่องจากแรงลม (การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS แสดงในภาคผนวก ข.) จากสมการที่ (9-1) ค่า μ คือ มวลของท่อยางต่อความยาวหนึ่งหน่วย (กิโลกรัม/เมตร) g คือ ความเร่งเนื่องจากแรงโน้มถ่วง (เมตร/วินาที²) และ F (นิวตัน) คือ แรงที่กระทำกับท่อยางที่ส่งผ่านมาจากมอเตอร์ไฟฟ้ากระแสตรง แรง F เป็นฟังก์ชันของสัญญาณอินพุตของมอเตอร์ไฟฟ้ากระแสตรง u (โวลต์) ที่ผ่านการเพิ่มกำลังแล้ว ดังนั้นจึงกำหนดให้แรง F เป็นดังสมการที่ (9-2) ที่ซึ่ง K_D คือ อัตราขยายที่ให้การประมาณแบบจำลองของส่วนเพิ่มกำลัง และ K_M คือ อัตราขยายที่ให้การประมาณแบบจำลองของมอเตอร์ไฟฟ้ากระแสตรง

จากสมการที่ (9-2) ส่วนเพิ่มกำลังได้รับการประมาณแบบจำลองด้วยอัตราขยายที่ไม่มีช่วงเวลาประวิง ทั้งนี้เพราะส่วนเพิ่มกำลังที่ใช้เป็นวงจรรวม (IC) ที่มีการตอบสนองอย่างรวดเร็ว ขณะที่มอเตอร์ไฟฟ้ากระแสตรงได้รับการประมาณแบบจำลองด้วยอัตราขยายที่ไม่มีช่วงเวลาประวิงเช่นเดียวกัน ทั้งนี้เพราะมอเตอร์ไฟฟ้ากระแสตรงที่ใช้เป็นมอเตอร์ขนาดเล็กและมีน้ำหนักเบา โดยมีขนาดเส้นผ่าศูนย์กลางเท่ากับ 35.74 มิลลิเมตร และมีน้ำหนักเท่ากับ 152.81 กรัม

$$\frac{\partial^2 y(x,t)}{\partial t^2} = \frac{F}{\mu} \frac{\partial^2 y(x,t)}{\partial x^2} - g \quad (9-1)$$

$$F = K_D K_M u \quad (9-2)$$

การระบุเอกลักษณ์ระบบ VTS อาจแสดงได้ดังรูปที่ 9.1 ซึ่งเป็นการหาค่าพารามิเตอร์ μ , K_D , และ K_M ของแบบจำลอง ข้อมูลอินพุตและเอาต์พุตที่ใช้สำหรับการระบุเอกลักษณ์ระบบ คือ สัญญาณอินพุตของมอเตอร์ไฟฟ้ากระแสตรง u และตำแหน่งของท่อยางในระนาบ x, y หรือ $y(x,t)$ ที่ได้จากการประมวลผลภาพ ดังแสดงไว้แล้วในรูปที่ 8.11 และ 8.12 ของบทที่ 8 ตามลำดับ



รูปที่ 9.1 การระบุเอกลักษณ์ระบบ VTS

ข้อมูลที่ได้จากการทดสอบระบบ VTS ในบทที่ 8 จะได้รับการนำมาจัดสรรเพื่อใช้ดำเนินการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y ซึ่งประกอบไปด้วยสัญญาณอินพุตและสัญญาณเอาต์พุต (ตำแหน่งของท่อยาง) ที่ได้จากกล้องวิดีโอดิจิทัลตัวที่ 1 จำนวนข้อมูลของสัญญาณเอาต์พุตเท่ากับ 30,500 ข้อมูล (500 ภาพ ๆ ละ 61 ข้อมูล) แต่จะนำมาใช้ประโยชน์ทั้งสิ้นจำนวน 4,880 ข้อมูล (80 ภาพ) ทั้งนี้เนื่องจากข้อมูลมีจำนวนมาก อีกทั้งข้อมูลส่วนต้นและส่วนท้าย

ของสัญญาณจะถูกตัดทิ้ง เพราะเป็นช่วงเริ่มต้นและช่วงยุติการทดสอบ ข้อมูลในช่วงดังกล่าวอาจมีความคลาดเคลื่อนสูง จากนั้นทำการแบ่งข้อมูลออกเป็นสองส่วน ข้อมูลส่วนแรกจำนวน 2,440 ข้อมูล (40 ภาพ) เพื่อใช้ระบุเอกลักษณ์ระบบ และข้อมูลส่วนที่สองจำนวน 2,440 ข้อมูล (40 ภาพ) เพื่อใช้ตรวจสอบความถูกต้องของแบบจำลอง

เนื่องจากการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม ไม่สามารถนำมาประยุกต์ใช้กับกรณีการระบุเอกลักษณ์ระบบ VTS ได้ ดังนั้นการระบุเอกลักษณ์ระบบจะอาศัยการค้นหาแบบตามูเชิงปรับตัว เป็นเครื่องมือในการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง การคำนวณผลเฉลี่ยของแบบจำลองในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก ตามสมการที่ (9-1) อาศัยการคำนวณเชิงตัวเลขตามระเบียบวิธีผลต่างสี่เหลี่ยม (รายละเอียดของระเบียบวิธีผลต่างสี่เหลี่ยมแสดงในภาคผนวก ฉ.) การพิจารณาความเหมาะสมของแบบจำลองอาศัยเกณฑ์ค่าความคลาดเคลื่อนระหว่างข้อมูลเอาต์พุตที่ได้จากการทดสอบ $y(x,t)$ และข้อมูลเอาต์พุตที่ได้จากแบบจำลอง $\hat{y}(x,t)$ ในรูปของ MSE ดังสมการที่ (9-3)

$$\text{MSE} = \frac{1}{n \cdot m} \sum_{j=1}^m \left(\sum_{i=1}^n \|y_i - \hat{y}_i\|^2 \right)_j, \quad i=1, \dots, n, \quad j=1, \dots, m \quad (9-3)$$

การค้นหาแบบตามูเชิงปรับตัวที่พัฒนาด้วยโปรแกรม MATLAB ได้รับการประยุกต์ใช้เพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง มีรายละเอียดดังนี้

1. ค่าพารามิเตอร์ที่จะได้รับการค้นหาจำนวน 3 ตัวคือ μ , K_D , และ K_M ปริภูมิการค้นหาของค่าพารามิเตอร์แต่ละตัวมีขอบเขตในทางปฏิบัติดังนี้ $\mu = [0, +0.1]$ และ $K_D = K_M = [0, +0.25]$ ค่า μ หรือ มวลของท่ออย่างต่อเนื่องหนึ่งหน่วยนั้นควรมีค่าไม่เกิน 0.1 กิโลกรัม/เมตร ขณะที่ค่า K_D และ K_M ต้องพิจารณากำหนดขอบเขตตามเงื่อนไขของระเบียบวิธีผลต่างสี่เหลี่ยม ดังรายละเอียดที่ปรากฏในภาคผนวก ฉ.

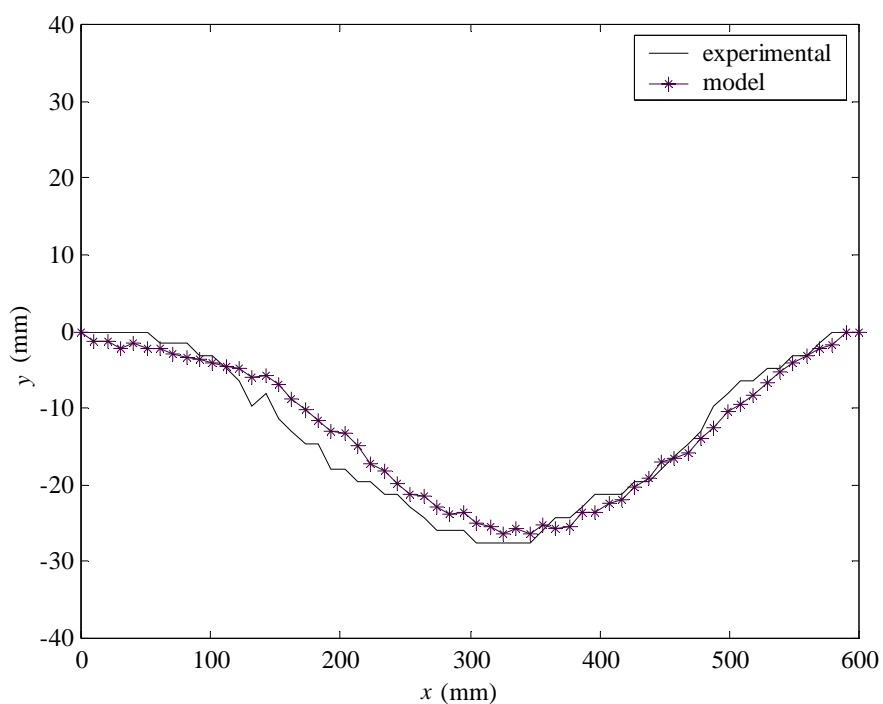
2. รัศมีการค้นหาเริ่มต้นเท่ากับ 10% ของปริภูมิการค้นหาของค่าพารามิเตอร์แต่ละตัว
3. จำนวนสมาชิกค่าใกล้เคียงเท่ากับ 30 ตัว
4. จำนวนการซ้าคำคำตอบก่อนเรียกใช้กลไกย้อนรอยการค้นหาเท่ากับ 5 ครั้ง
5. ลำดับคำตอบที่เลือกใช้เป็นค่าเริ่มต้นในกลไกย้อนรอยการค้นหาเท่ากับ -5
6. ค่ารัศมีการค้นหาหลังจากการปรับลดมีค่าเท่ากับ 20% ของค่ารัศมีการค้นหา
7. กลไกยุติการค้นหาที่กำหนดให้ค่า MSE เป็นค่า cost ซึ่งจะต้องน้อยกว่าหรือเท่ากับ 23.26 ($\text{cost} \leq 23.26$) หรือจำนวนรอบการค้นหาสูงสุดเท่ากับ 1,000 รอบ
8. ดำเนินการค้นหาทั้งหมด 1,000 ครั้ง (trial) เพื่อยืนยันคำตอบที่ได้จากการค้นหา

รายละเอียดของโปรแกรมการค้นหาแบบตาบู่เชิงปรับตัว เพื่อการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y แสดงในภาคผนวก ญ. การระบุเอกลักษณ์ระบบ VTS ได้ดำเนินการบนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกะเฮิร์ตซ์ ซึ่งมีหน่วยความจำ RAM ขนาด 256 เมกกะไบต์ ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y โดยใช้การค้นหาแบบตาบู่เชิงปรับตัว แสดงดังตารางที่ 9.1 การค้นหาสามารถพบคำตอบที่ทำให้ค่า cost เป็นไปตามกลไกยุติการค้นหา กล่าวคือมีค่า cost (MSE) เหลือเท่ากับ 23.25 ใช้เวลาในการค้นหาเฉลี่ย 8.83 วินาที และใช้จำนวนรอบในการค้นหาเฉลี่ย 10.19 รอบ ค่าพารามิเตอร์แต่ละตัวที่ได้จากการค้นหาแสดงในรูปของค่าเฉลี่ยดังนี้ K_D เท่ากับ 0.022 K_M เท่ากับ 0.024 และ μ เท่ากับ 0.0064 กิโลกรัม/เมตร จากการดำเนินการค้นหาทั้งหมด 1,000 ครั้ง ได้พิจารณาแบบจำลองที่ให้ค่า MSE น้อยที่สุด เพื่อแสดงผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง รูปที่ 9.2 ถึง 9.7 แสดงผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y และรูปที่ 9.8 ถึง 9.13 แสดงผลการตรวจสอบความถูกต้องของแบบจำลอง ทั้งนี้เพื่อความชัดเจนในการพิจารณา

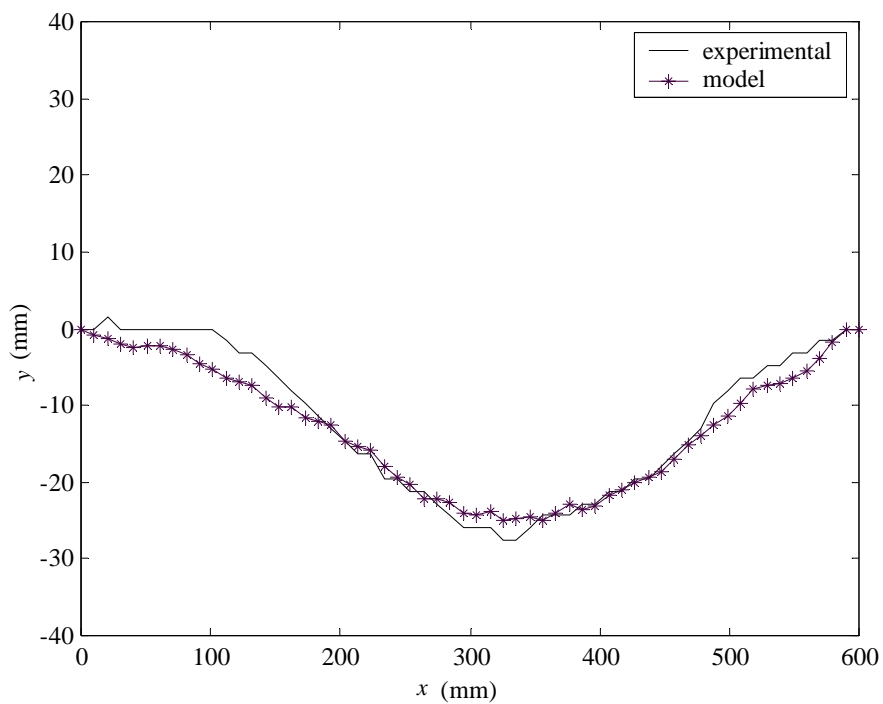
ตารางที่ 9.1 ผลการระบุเอกลักษณ์ค่าพารามิเตอร์ของแบบจำลองระบบ VTS ในระนาบ x, y

	ค่าพารามิเตอร์			ค่า cost (MSE)	เวลาที่ใช้ (sec)	จำนวนรอบ การค้นหา
	K_D	K_M	μ (kg/m)			
ค่าน้อยที่สุด	0.0015	0.0016	2.93×10^{-4}	23.2527	0.8130	1
ค่ามากที่สุด	0.0795	0.0797	0.0099	23.2527	23.5160	24
ค่าเฉลี่ย	0.0220	0.0243	0.0064	23.2527	8.8297	10.1910
ค่า SD	0.0190	0.0193	0.0023	3.03×10^{-8}	5.4170	6.0883

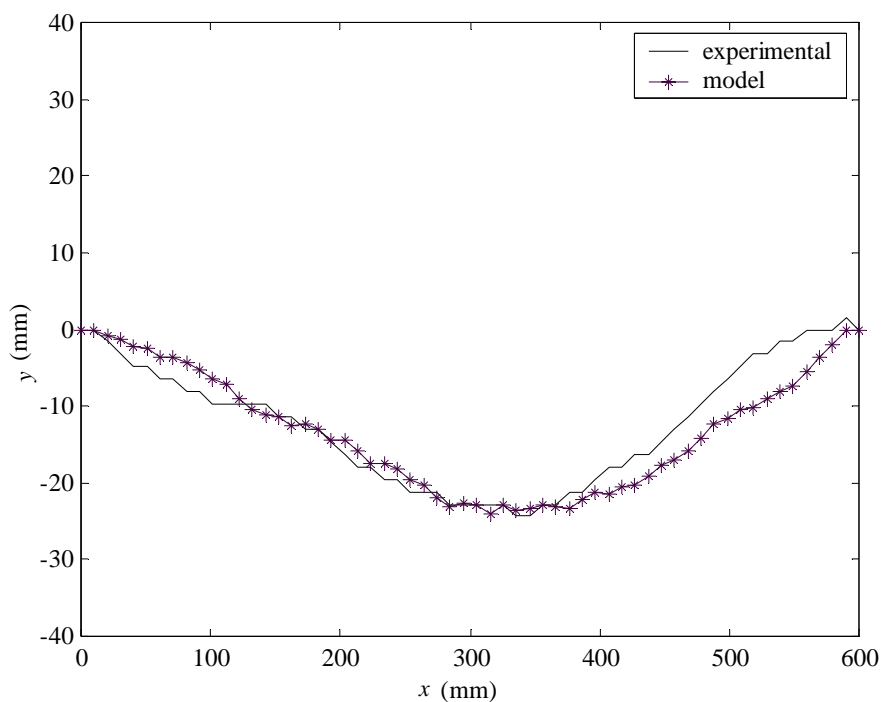
จากรูปที่ 9.2 ถึง 9.13 ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลองเปรียบเทียบกับผลการทดสอบ มีความสอดคล้องกันทั้งในส่วนของมุม (phase) และขนาด (magnitude) โดยปรากฏความคลาดเคลื่อนอยู่บ้างเล็กน้อย จากผลการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลองโดยการค้นหาแบบตาบู่เชิงปรับตัว ดังแสดงในตารางที่ 9.1 พบว่าค่า μ เฉลี่ยของแบบจำลอง ซึ่งมีค่าเท่ากับ 0.0064 กิโลกรัม/เมตร มีค่าใกล้เคียงกับค่าจริงมาก กล่าวคือมวลของท่อที่ใช้ซึ่งมีขนาดความยาว 61 เซนติเมตร ได้รับการชั่งด้วยเครื่องชั่งดิจิตอลยี่ห้อ Sartorius รุ่น AC210S มีค่าเท่ากับ 4.4504 กรัม ดังนั้นค่า μ หรือมวลของท่ออย่างต่อความยาวหนึ่งหน่วยคำนวณได้เท่ากับ 0.0073 กิโลกรัม/เมตร แสดงให้เห็นว่าการระบุเอกลักษณ์ระบบโดยอาศัยเทคนิคการประมวลผลภาพที่นำเสนอ มีความถูกต้องแม่นยำอย่างน่าพึงพอใจ



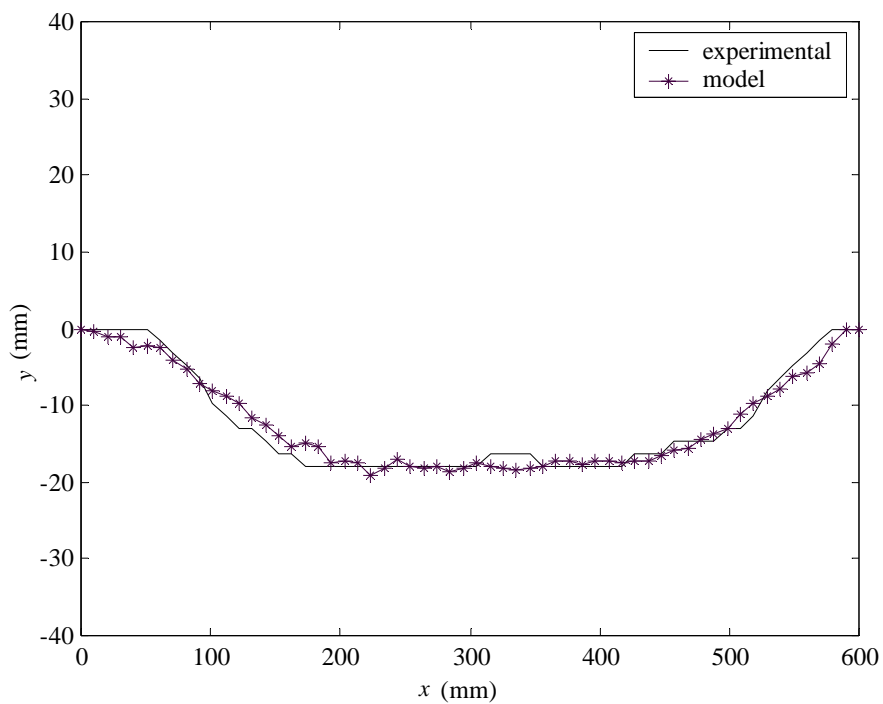
รูปที่ 9.2 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y จังหวะที่ 1



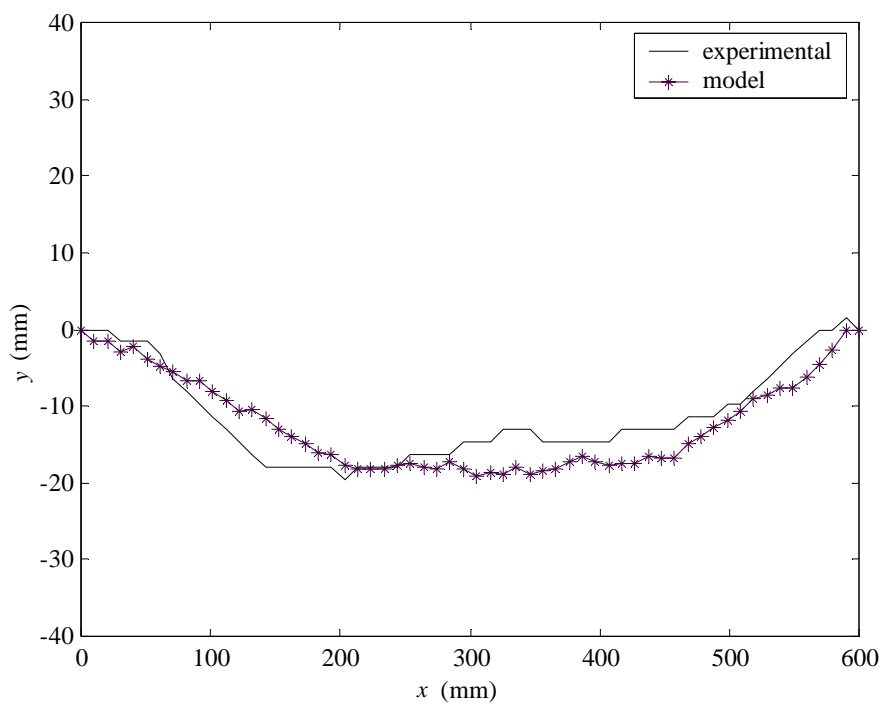
รูปที่ 9.3 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y จังหวะที่ 2



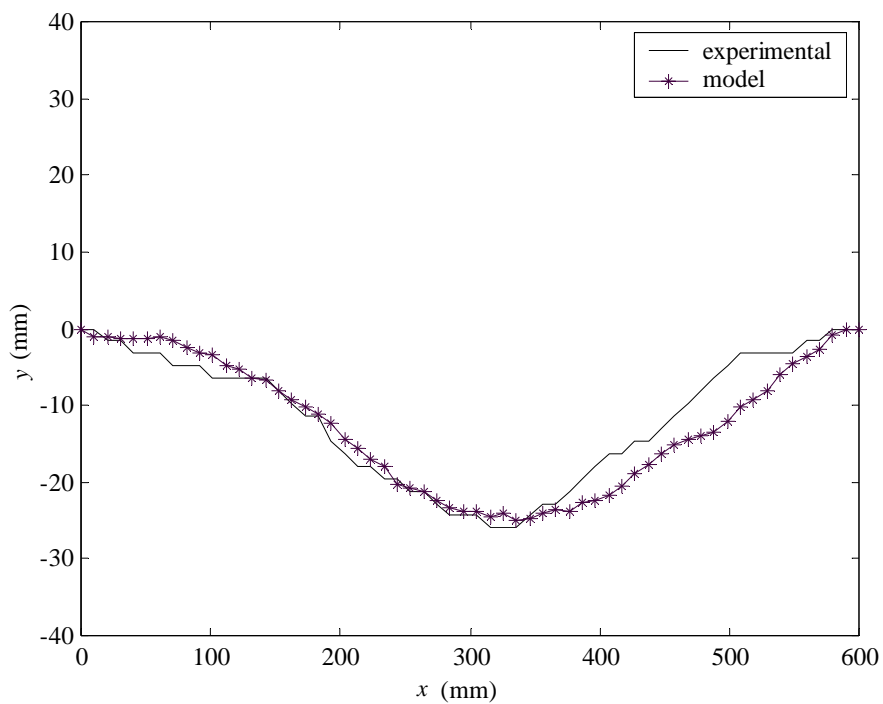
รูปที่ 9.4 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y จังหวะที่ 3



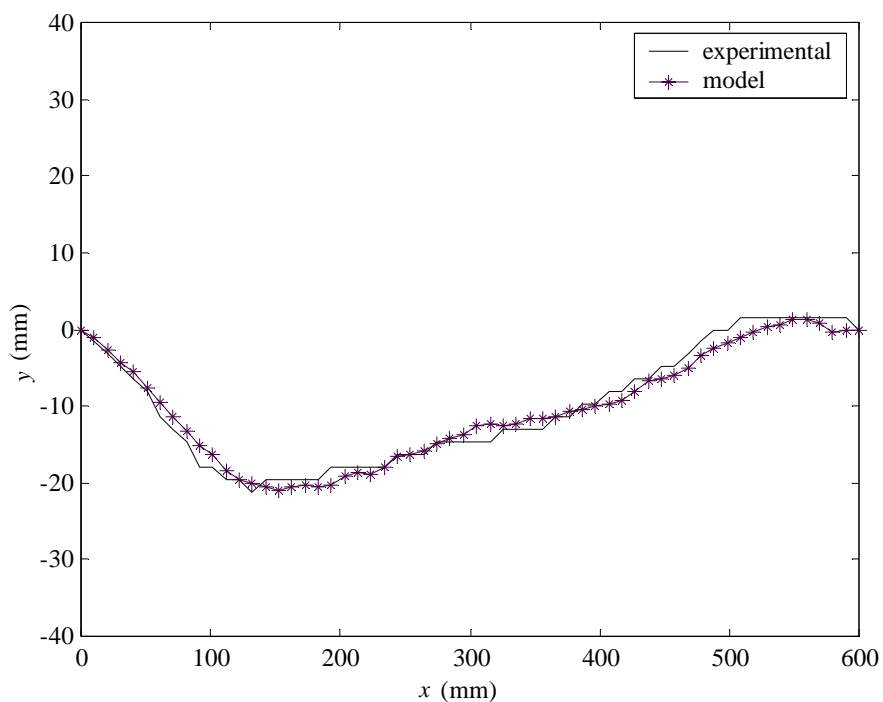
รูปที่ 9.5 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y จังหวะที่ 4



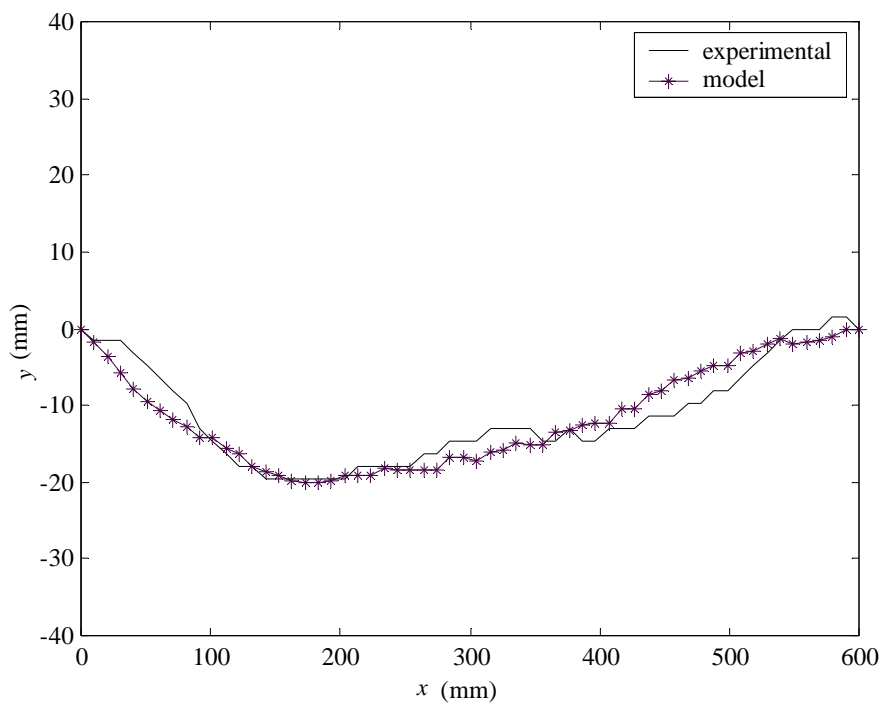
รูปที่ 9.6 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y จังหวะที่ 5



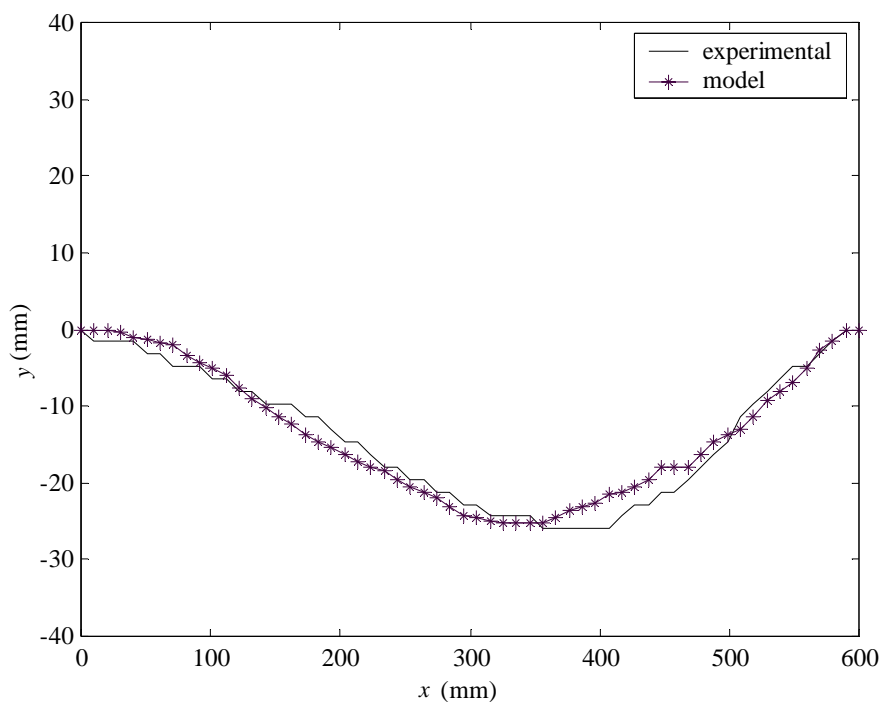
รูปที่ 9.7 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y จังหวะที่ 6



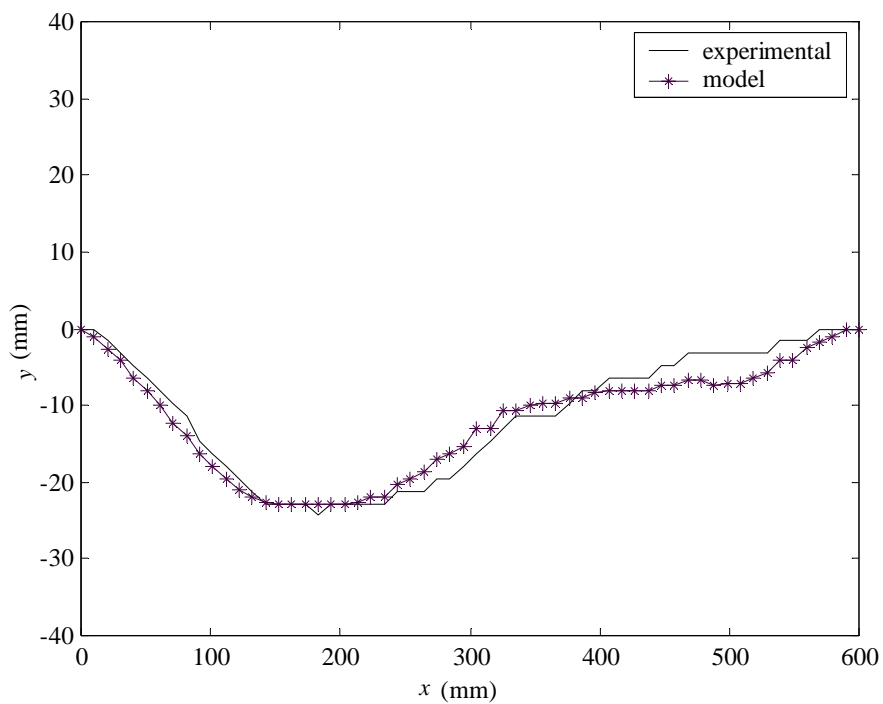
รูปที่ 9.8 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ x, y จังหวะที่ 1



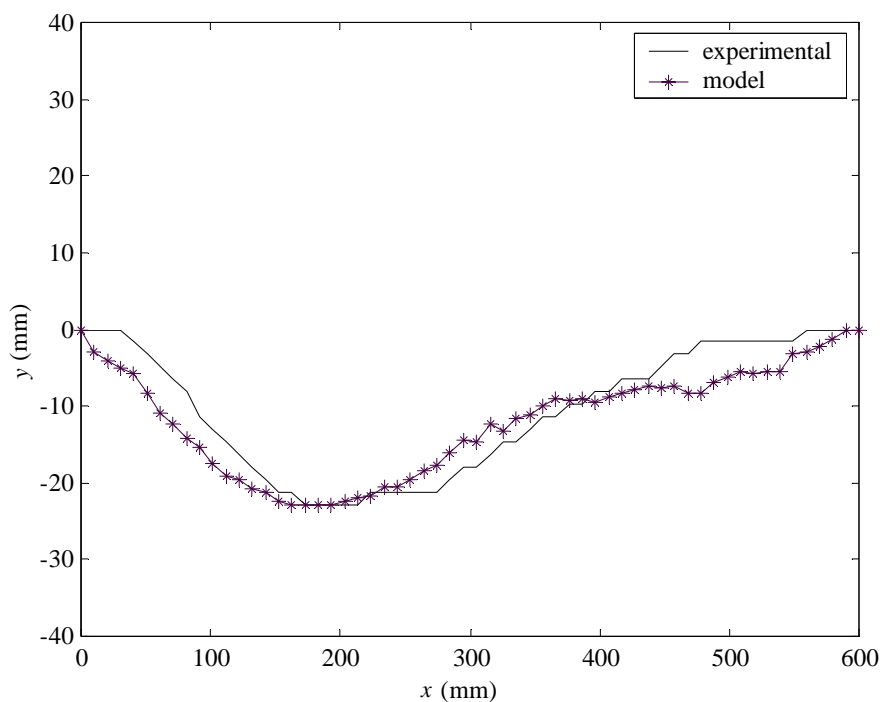
รูปที่ 9.9 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ x, y จังหวะที่ 2



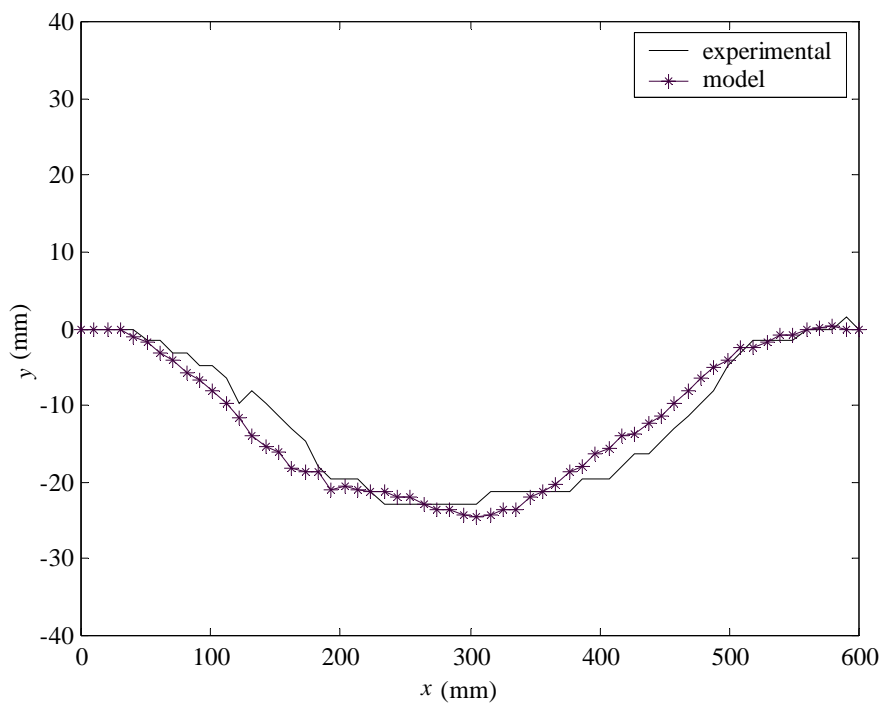
รูปที่ 9.10 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ x, y จังหวะที่ 3



รูปที่ 9.11 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ x, y จังหวะที่ 4



รูปที่ 9.12 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ x, y จังหวะที่ 5



รูปที่ 9.13 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ x, y จังหวะที่ 6

9.3 การระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x

เมื่อพิจารณาการเคลื่อนที่ของท่ออย่างในระนาบ z, x แบบจำลองทางคณิตศาสตร์ของระบบ VTS แสดงดังสมการที่ (9-4) แบบจำลองดังกล่าวได้รับการพัฒนาขึ้นโดยอาศัยหลักการเดียวกันกับแบบจำลองการเคลื่อนที่ของท่ออย่างในระนาบ x, y ที่กล่าวไว้แล้วในหัวข้อที่ 9.2 อย่างไรก็ตามในกรณีนี้จะไม่พิจารณาผลของความเร่งเนื่องจากแรงโน้มถ่วง ทั้งนี้เนื่องจากทิศทางของความเร่งเนื่องจากแรงโน้มถ่วงนั้นอยู่ในแนวแกน y ซึ่งเป็นทิศทางที่ตั้งฉากกับระนาบการเคลื่อนที่ของท่ออย่างที่กำลังพิจารณา (รายละเอียดการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS แสดงในภาคผนวก ช.) จากสมการที่ (9-4) F (นิวตัน) คือ แรงที่กระทำกับท่ออย่างที่ส่งผ่านไปตามเพลลาของมอเตอร์ไฟฟ้ากระแสตรง ซึ่งมีความสัมพันธ์ดังสมการที่ (9-2) การระบุเอกลักษณ์ระบบ VTS ในกรณีนี้เป็นการหาค่าพารามิเตอร์ μ , K_D , และ K_M ของแบบจำลอง ข้อมูลอินพุตและเอาต์พุตที่ใช้สำหรับการระบุเอกลักษณ์ระบบ คือสัญญาณอินพุตของมอเตอร์ไฟฟ้ากระแสตรง u และตำแหน่งของท่ออย่างในระนาบ z, x หรือ $z(x, t)$ ที่ได้จากการประมวลผลภาพ ดังแสดงไว้แล้วในรูปที่ 8.11 และ 8.13 ของบทที่ 8 ตามลำดับ

$$\frac{\partial^2 z(x, t)}{\partial t^2} = \frac{F}{\mu} \frac{\partial^2 z(x, t)}{\partial x^2} \quad (9-4)$$

ข้อมูลที่ได้จากการทดสอบระบบ VTS ในบทที่ 8 จะได้รับการนำมาจัดสรรเพื่อใช้ดำเนินการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x ซึ่งประกอบไปด้วยสัญญาณอินพุตและสัญญาณเอาต์พุต (ตำแหน่งของท่ออย่าง) ที่ได้จากกล้องวิดีโอดิจิทัลตัวที่ 2 จำนวนข้อมูลของสัญญาณเอาต์พุตเท่ากับ 30,500 ข้อมูล นำมาใช้ประโยชน์ทั้งสิ้นจำนวน 4,880 ข้อมูล โดยทำการแบ่งข้อมูลออกเป็นสองส่วน ข้อมูลส่วนแรกจำนวน 2,440 ข้อมูล เพื่อใช้ระบุเอกลักษณ์ระบบ และข้อมูลส่วนที่สองจำนวน 2,440 ข้อมูล เพื่อใช้ตรวจสอบความถูกต้องของแบบจำลอง

ดังได้กล่าวมาแล้วในหัวข้อที่ 9.2 ว่า การระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม ไม่สามารถนำมาประยุกต์ใช้กับกรณีการระบุเอกลักษณ์ระบบ VTS ได้ ดังนั้นการระบุเอกลักษณ์ระบบจะอาศัยการค้นหาแบบตามเชิงปรับตัว เป็นเครื่องมือในการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง การคำนวณผลเฉลยของแบบจำลองในรูปแบบของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิกตามสมการที่ (9-4) อาศัยการคำนวณเชิงตัวเลขตามระเบียบวิธีผลต่างสี่เหลี่ยม (รายละเอียดของระเบียบวิธีผลต่างสี่เหลี่ยมแสดงในภาคผนวก ฉ.) การพิจารณาความเหมาะสมของแบบจำลองอาศัยเกณฑ์ค่าความคลาดเคลื่อนระหว่างข้อมูลเอาต์พุตที่ได้จากการทดสอบ $z(x, t)$ และข้อมูลเอาต์พุตที่ได้จากแบบจำลอง $\hat{z}(x, t)$ ในรูปของ MSE ดังสมการที่ (9-5)

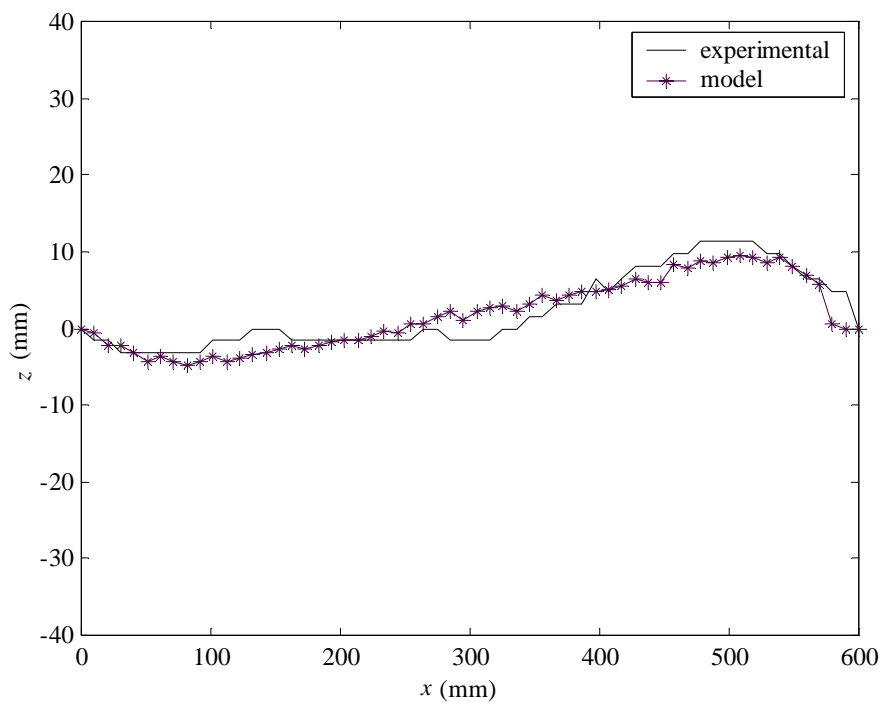
$$\text{MSE} = \frac{1}{n \cdot m} \sum_{j=1}^m \left(\sum_{i=1}^n \|z_i - \hat{z}_i\|^2 \right), \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (9-5)$$

สำหรับการค้นหาแบบตาบุงเชิงปรับตัวที่พัฒนาด้วยโปรแกรม MATLAB ได้รับการประยุกต์ใช้เพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง ซึ่งรายละเอียดการกำหนดขอบเขตปฏิบัติการค้นหาของพารามิเตอร์แต่ละตัว รัศมีการค้นหาเริ่มต้น จำนวนการสุ่มค่าใกล้เคียง กลไกย้อนรอยการค้นหา และกลไกปรับรัศมีพื้นที่การค้นหา ยังคงใช้ตามที่ได้กำหนดไว้ในหัวข้อ 9.2 สำหรับกลไกยุติการค้นหาในกรณีนี้กำหนดให้ค่า MSE เป็นค่า cost ซึ่งจะต้องน้อยกว่าหรือเท่ากับ 32.38 ($\text{cost} \leq 32.38$) หรือจำนวนรอบการค้นหาสูงสุดเท่ากับ 1,000 รอบ ดำเนินการค้นหาทั้งหมด 1,000 ครั้ง เพื่อยืนยันคำตอบที่ได้จากการค้นหา รายละเอียดของโปรแกรมการค้นหาแบบตาบุงเชิงปรับตัว เพื่อการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x แสดงในภาคผนวก ญ.

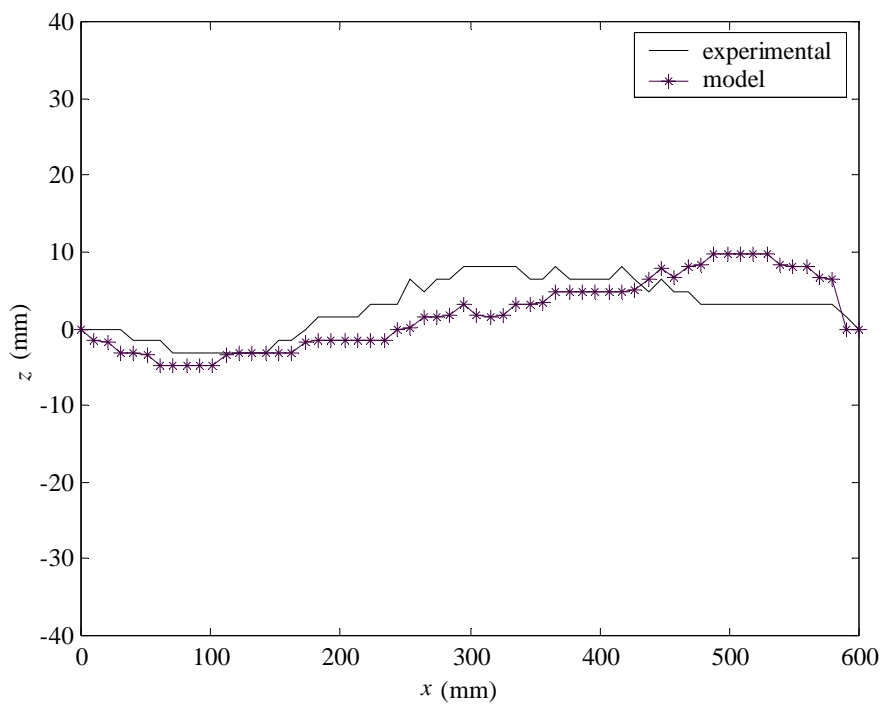
การระบุเอกลักษณ์ระบบ VTS ได้ดำเนินการบนเครื่องคอมพิวเตอร์ Pentium 4 ความเร็ว 1.6 จิกกะเฮิร์ตซ์ ซึ่งมีหน่วยความจำ RAM ขนาด 256 เมกกะไบต์ ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x โดยใช้การค้นหาแบบตาบุงเชิงปรับตัว แสดงดังตารางที่ 9.2 การค้นหาสามารถพบคำตอบที่ทำให้ค่า cost เป็นไปตามกลไกยุติการค้นหา กล่าวคือมีค่า cost (MSE) เฉลี่ยเท่ากับ 32.37 ใช้เวลาในการค้นหาเฉลี่ย 5.59 วินาที และใช้จำนวนรอบในการค้นหาเฉลี่ย 7.18 รอบ ค่าพารามิเตอร์แต่ละตัวที่ได้จากการค้นหาแสดงในรูปของค่าเฉลี่ยดังนี้ K_D เท่ากับ 0.022 K_M เท่ากับ 0.017 และ μ เท่ากับ 0.0065 กิโลกรัม/เมตร จากการดำเนินการค้นหาทั้งหมด 1,000 ครั้ง ได้พิจารณาแบบจำลองที่ให้ค่า MSE น้อยที่สุด เพื่อแสดงผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลอง รูปที่ 9.14 ถึง 9.19 แสดงผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x และรูปที่ 9.20 ถึง 9.25 แสดงผลการตรวจสอบความถูกต้องของแบบจำลอง ทั้งนี้เพื่อความชัดเจนในการพิจารณา

ตารางที่ 9.2 ผลการระบุเอกลักษณ์ค่าพารามิเตอร์ของแบบจำลองระบบ VTS ในระนาบ z, x

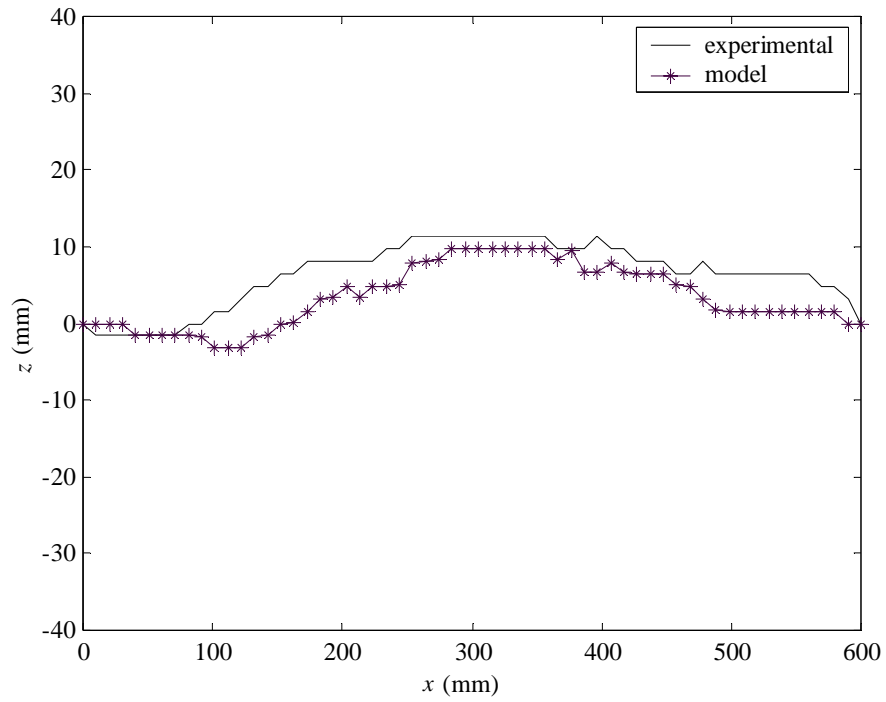
	ค่าพารามิเตอร์			ค่า cost (MSE)	เวลาที่ใช้ (sec)	จำนวนรอบ การค้นหา
	K_D	K_M	μ (kg/m)			
ค่าน้อยที่สุด	2.78×10^{-4}	1.52×10^{-4}	4.71e-004	32.3727	0.4850	1
ค่ามากที่สุด	0.0797	0.0799	0.0099	32.3727	22.9380	25
ค่าเฉลี่ย	0.0218	0.0168	0.0065	32.3727	5.5938	7.1820
ค่า SD	0.0217	0.0203	0.0023	2.24×10^{-5}	6.9001	8.6862



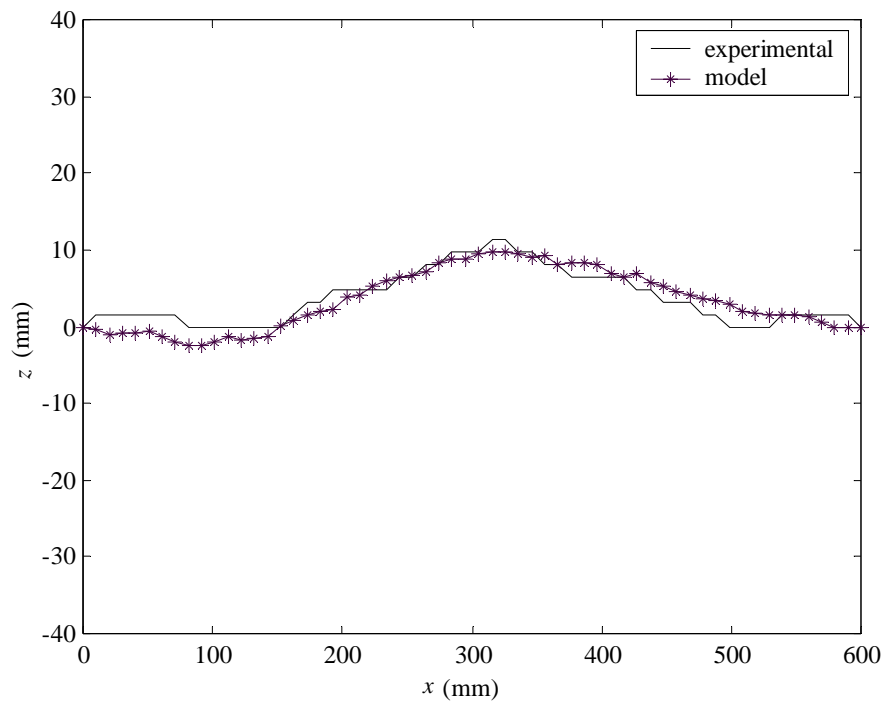
รูปที่ 9.14 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x จังหวะที่ 1



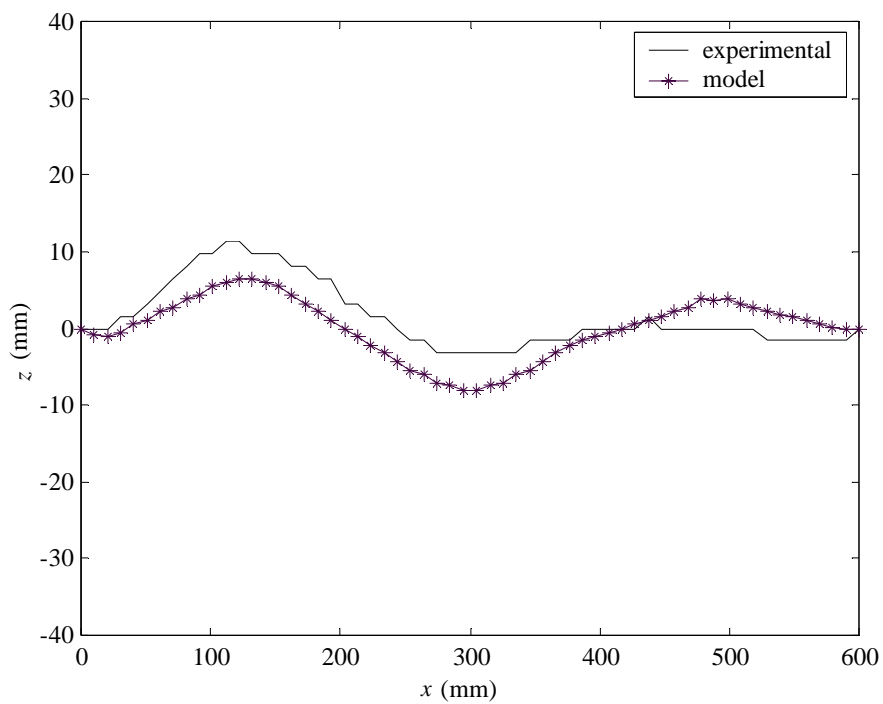
รูปที่ 9.15 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x จังหวะที่ 2



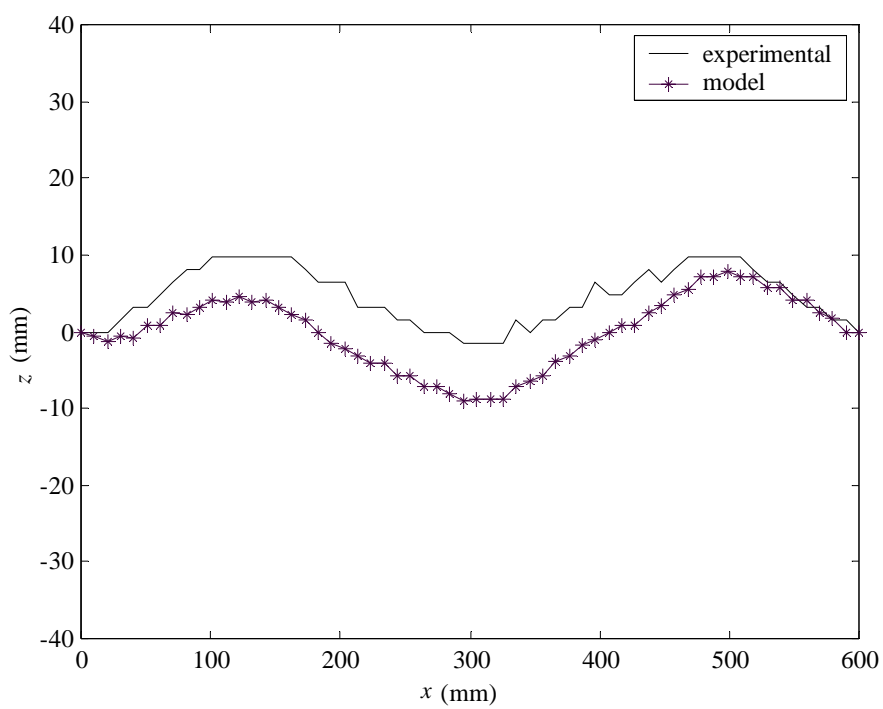
รูปที่ 9.16 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x จังหวะที่ 3



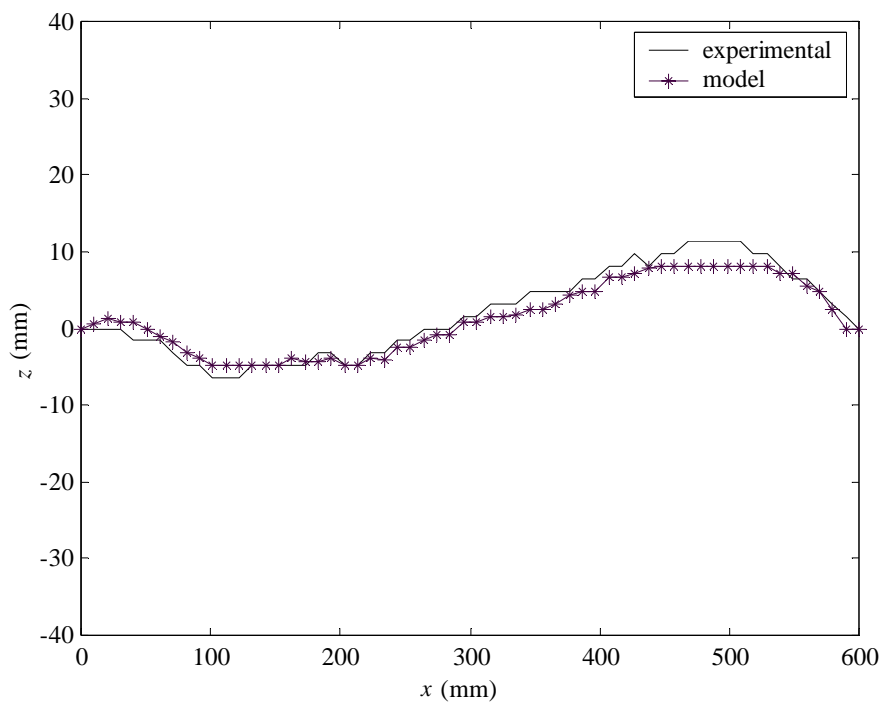
รูปที่ 9.17 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x จังหวะที่ 4



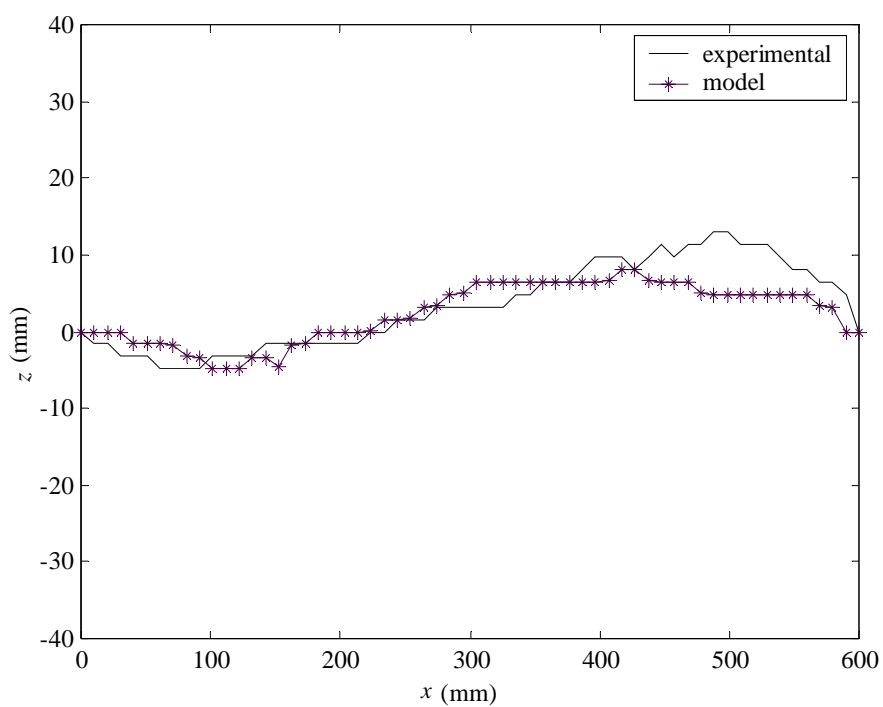
รูปที่ 9.18 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x จังหวะที่ 5



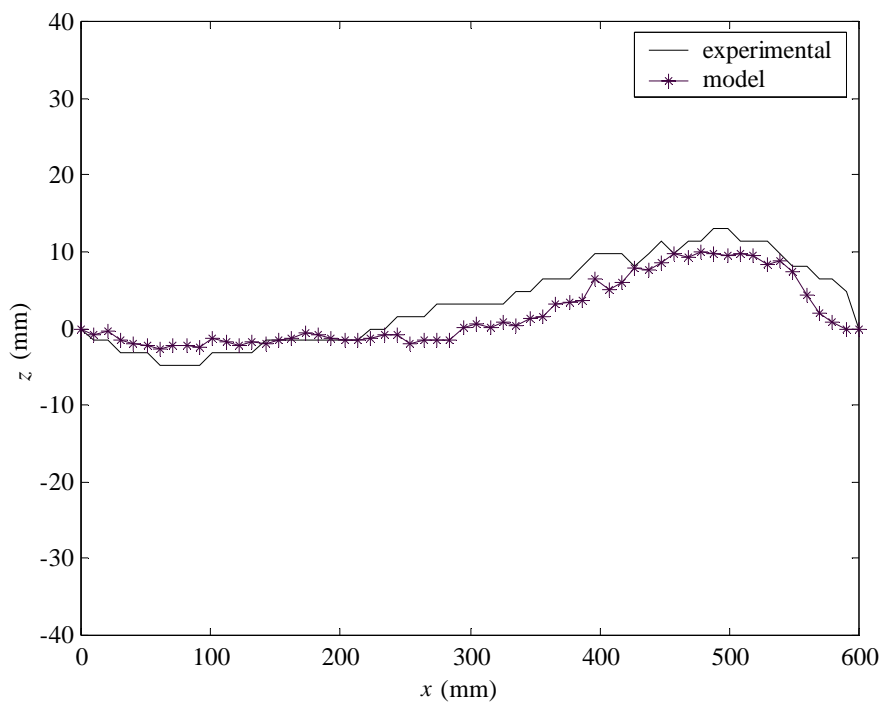
รูปที่ 9.19 ผลการระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x จังหวะที่ 6



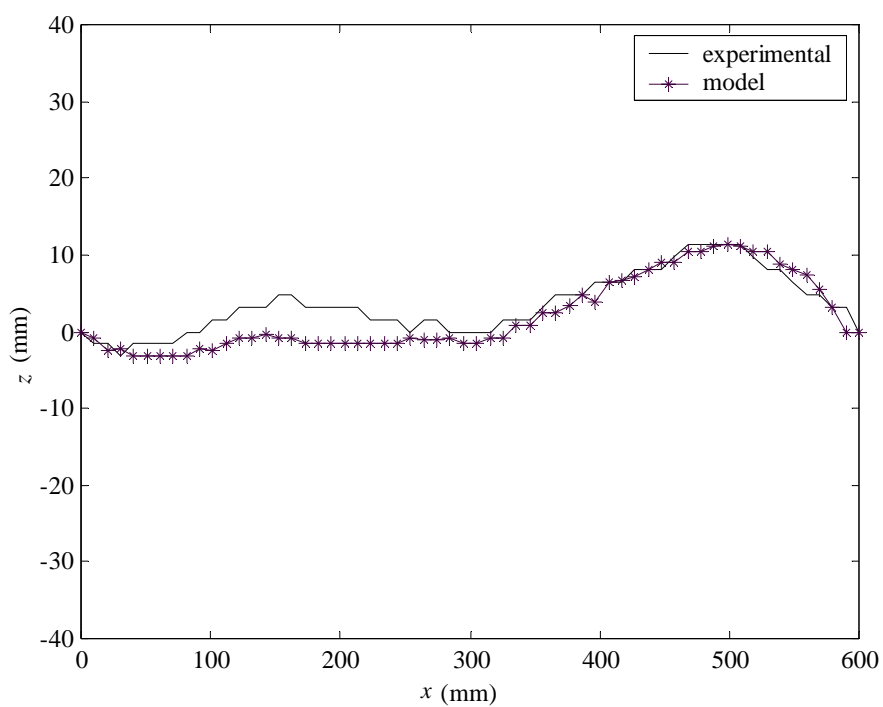
รูปที่ 9.20 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ z, x จังหวะที่ 1



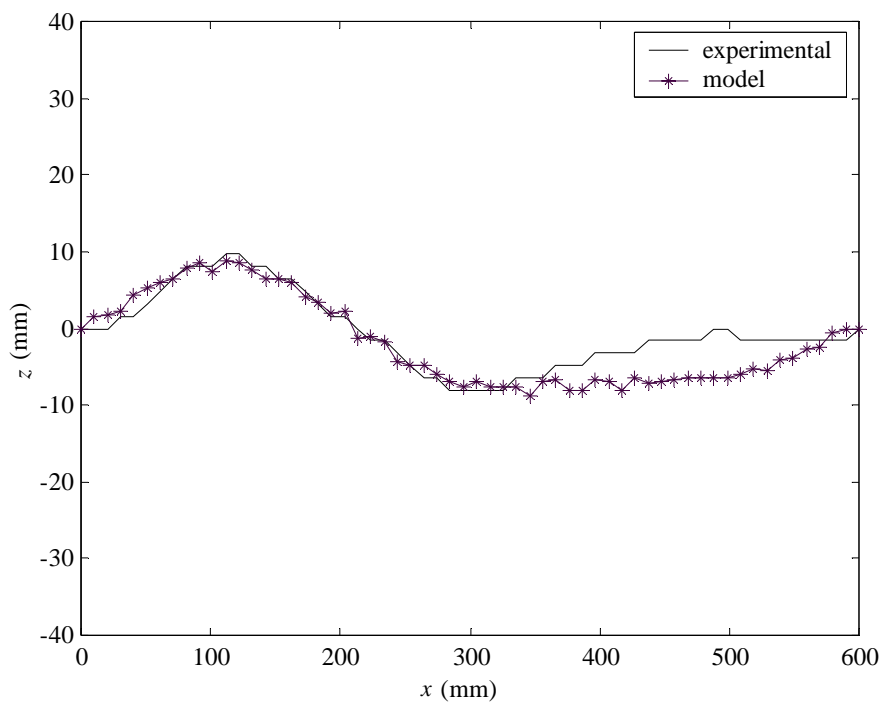
รูปที่ 9.21 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ z, x จังหวะที่ 2



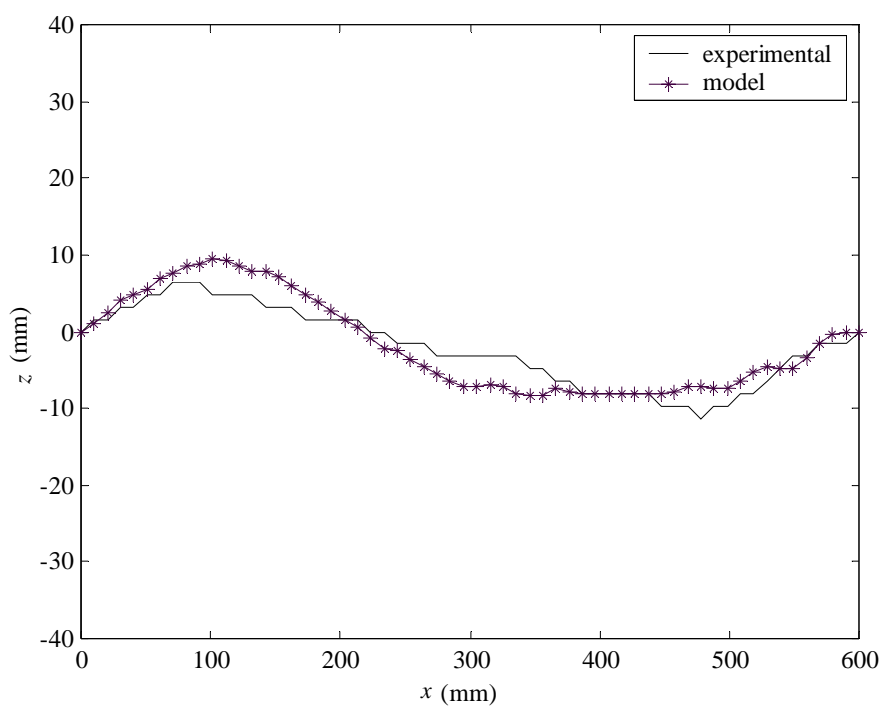
รูปที่ 9.22 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ z, x จังหวะที่ 3



รูปที่ 9.23 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ z, x จังหวะที่ 4



รูปที่ 9.24 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ z, x จังหวะที่ 5



รูปที่ 9.25 ผลการตรวจสอบความถูกต้องของแบบจำลองระบบ VTS ในระนาบ z, x จังหวะที่ 6

จากรูปที่ 9.14 ถึง 9.25 ผลการระบุเอกลักษณ์ระบบ และการตรวจสอบความถูกต้องของแบบจำลองเปรียบเทียบกับผลการทดสอบ มีความสอดคล้องกันในส่วนของมุม แต่จะปรากฏความคลาดเคลื่อนในส่วนของขนาดอยู่พอสมควร ทั้งนี้อาจเป็นผลมาจากการที่แบบจำลองทางคณิตศาสตร์ในกรณีนี้ ไม่ได้พิจารณาผลกระทบของความเร่งเนื่องจากแรงโน้มถ่วงที่กระทำกับท่ออย่าง ด้วยเพราะทิศทางของความเร่งเนื่องจากแรงโน้มถ่วงนั้นตั้งฉากกับระนาบการเคลื่อนที่ของท่ออย่าง ขณะที่ในความเป็นจริงแล้วนั้น มีความเร่งเนื่องจากแรงโน้มถ่วงกระทำกับท่ออย่าง อยู่ตลอดเวลา ดังนั้นจึงทำให้ผลที่ได้จากแบบจำลองและการทดสอบ มีความคลาดเคลื่อนปรากฏอยู่ อย่างไรก็ตาม จากผลการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลองโดยการค้นหาแบบตามูเชิงปรับตัว ดังแสดงในตารางที่ 9.2 พบว่าค่า μ เฉลี่ยของแบบจำลอง ซึ่งมีค่าเท่ากับ 0.0065 กิโลกรัม/เมตร มีค่าใกล้เคียงกับค่า μ เฉลี่ยที่ได้จากการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y ซึ่งมีค่าเท่ากับ 0.0064 กิโลกรัม/เมตร อีกทั้งยังมีค่าใกล้เคียงกับค่าจริง ซึ่งมีค่าเท่ากับ 0.0073 กิโลกรัม/เมตร อีกด้วย นอกจากนี้ค่า K_D ที่ได้จากการระบุเอกลักษณ์ระบบ VTS ทั้งสองส่วนนั้นมีค่าเท่ากัน (K_D เท่ากับ 0.022) ขณะที่ค่า K_M ที่ได้จากการระบุเอกลักษณ์ระบบ VTS ทั้งสองส่วนนั้นมีค่าใกล้เคียงกัน กล่าวคือค่า K_M ที่ได้จากการระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y และในระนาบ z, x มีค่าเท่ากับ 0.024 และ 0.017 ตามลำดับ แสดงให้เห็นว่าการระบุเอกลักษณ์ระบบ โดยอาศัยเทคนิคการประมวลผลภาพที่นำเสนอ มีความถูกต้องแม่นยำอย่างน่าพึงพอใจ

9.4 สรุป

บทที่ 9 นี้ได้กล่าวถึงการระบุเอกลักษณ์ระบบ VTS ซึ่งเป็นระบบพลวัตที่มีการเคลื่อนที่แบบ 3 มิติ และมีแบบจำลองทางคณิตศาสตร์เป็นสมการอนุพันธ์ย่อย การระบุเอกลักษณ์ระบบ VTS แบ่งออกเป็นสองส่วนคือ การระบุเอกลักษณ์ระบบในระนาบ x, y และในระนาบ z, x โดยอาศัยการค้นหาแบบตามูเชิงปรับตัวเพื่อค้นหาค่าพารามิเตอร์ของแบบจำลอง การคำนวณผลเฉลยของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก อาศัยระเบียบวิธีผลต่างสี่เหลี่ยม เนื่องจากผลการระบุเอกลักษณ์ระบบ VTS พบว่า ในระนาบ x, y ผลที่ได้จากแบบจำลองมีความสอดคล้องกับผลที่ได้จากการทดสอบทั้งในส่วนของมุมและขนาด ขณะที่ในระนาบ z, x ผลที่ได้จากแบบจำลองและการทดสอบมีความสอดคล้องกันในส่วนของมุม แต่ปรากฏความคลาดเคลื่อนในส่วนขนาดอยู่พอสมควร ทั้งนี้เนื่องจากแบบจำลองทางคณิตศาสตร์ในกรณีหลังนี้ ไม่ได้พิจารณาผลกระทบของความเร่งเนื่องจากแรงโน้มถ่วง อย่างไรก็ตาม ค่าพารามิเตอร์ของแบบจำลองที่ได้จากการระบุเอกลักษณ์ระบบทั้งสองส่วน มีค่าใกล้เคียงกับค่าจริงมาก แสดงให้เห็นว่าการระบุเอกลักษณ์ระบบ โดยอาศัยเทคนิคการประมวลผลภาพที่นำเสนอ มีความถูกต้องแม่นยำอย่างน่าพึงพอใจ

บทที่ 10

บทสรุปและข้อเสนอแนะ

10.1 ข้อสรุป

งานวิจัยวิทยานิพนธ์นี้เกี่ยวข้องกับการระบุเอกลักษณ์ระบบโดยใช้การประมวลผลภาพ เพื่อแก้ปัญหาในกรณีที่พลวัตของระบบไม่สามารถตรวจวัดได้โดยตัวตรวจรู้ทั่วไป โดยมีวัตถุประสงค์หลักคือ เพื่อพัฒนาเทคนิคการประมวลผลภาพสำหรับสกัดข้อมูลพลวัตของระบบจากลำดับภาพ เพื่อพัฒนาการค้นหาแบบตามูเชิงปรับตัวสำหรับค้นหาค่าพารามิเตอร์ในแบบจำลอง และเพื่อพัฒนาระบบกวัดแกว่งตัวอย่างต้นแบบสำหรับใช้เป็นระบบทดสอบการระบุเอกลักษณ์ระบบด้วยการประมวลผลภาพ การดำเนินงานวิจัยวิทยานิพนธ์ดังกล่าวสำเร็จคล่องตามวัตถุประสงค์ โดยสามารถสรุปผลการศึกษาวิจัยและพัฒนาทางวิศวกรรมเป็นข้อสรุปได้ดังต่อไปนี้

1. การปรับที่นัวรรณกรรมและงานวิจัยที่เกี่ยวข้อง ได้รับการรายงานไว้ในบทที่ 2 ซึ่งพบว่าการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม จะอาศัยข้อมูลความรู้เกี่ยวกับอินพุตและเอาต์พุตของระบบ การคำนวณเพื่อระบุเอกลักษณ์ระบบอาศัยคณิตศาสตร์แบบการถดถอยเชิงเส้น และระเบียบวิธีกำลังสองน้อยที่สุด เป็นแนวทางหลัก การระบุเอกลักษณ์ระบบด้วยเทคนิคทางปัญญาประดิษฐ์ มีการใช้โครงข่ายประสาทเป็นแบบจำลองของระบบ และใช้การค้นหาแบบตามูเพื่อการประมาณค่าพารามิเตอร์ในแบบจำลอง การระบุเอกลักษณ์ระบบด้วยเทคนิคการประมวลผลภาพ มีการใช้เทคนิคการประมวลผลภาพแบบดิจิทัลที่เหมาะสมเพื่อการระบุเอกลักษณ์ระบบ และใช้กล้องบันทึกภาพทำหน้าที่เป็นตัวตรวจรู้

2. รายละเอียดการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม ได้รับการศึกษาและเรียบเรียงไว้ในบทที่ 3 โดยให้ความสนใจการระบุเอกลักษณ์แบบจำลองกล่องดำเชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ ซึ่งเป็นรูปลักษณะที่ใช้ประโยชน์อย่างกว้างขวางในวิทยาการระบบควบคุมแบบจำลอง ARX ARMAX OE และ BJ จะอยู่ในรูปของฟังก์ชันถ่ายโอนโดเมน q การคำนวณเพื่อประมาณค่าพารามิเตอร์ใช้หลักการถดถอยเชิงเส้น และกำลังสองน้อยที่สุด นอกจากนี้ในบทที่ 3 ยังกล่าวถึงแนวทางการระบุเอกลักษณ์ระบบด้วยโปรแกรมสำเร็จ และการวางแผนทดสอบระบบเพื่อบันทึกข้อมูลอินพุตและเอาต์พุตอย่างถูกต้องและเพียงพอต่อการระบุเอกลักษณ์ระบบอีกด้วย

3. การพัฒนาเทคนิคการประมวลผลภาพเพื่อสกัดข้อมูลพลวัตของระบบจากลำดับภาพ ได้รับการนำเสนอในบทที่ 4 โดยอาศัยการแบ่งส่วนภาพ ด้วยเทคนิคการกำหนดจุดเปลี่ยนค่า ซึ่งตั้งอยู่บนหลักการความคล้ายคลึงกันของค่าสีในจุดภาพ

4. การระบุเอกลักษณ์ระบบไม่เชิงเส้นในงานวิจัยนี้ ใช้เทคนิคการค้นหาทางปัญญาประดิษฐ์ เพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง การค้นหาแบบตามูเชิงปรับตัวจึงได้รับการพัฒนาขึ้นเพื่อวัตถุประสงค์ดังกล่าว ดังรายละเอียดที่ปรากฏในบทที่ 5 การค้นหาแบบตามูเชิงปรับตัวได้รับการพัฒนาจากการค้นหาแบบตามู ซึ่งอาศัยหลักการค้นหาค่าใกล้เคียงแบบสุ่ม โดยเพิ่มกลไกย้อนรอยการค้นหา เพื่อแก้ปัญหาค้นหาที่ติดขัดโดยคำตอบวงแคบเฉพาะถิ่น และเพิ่มกลไกปรับรัศมีการค้นหา เพื่อเพิ่มความเร็วให้กับกระบวนการค้นหาคำตอบ (โปรแกรมการค้นหาแบบตามูเชิงปรับตัวที่พัฒนาขึ้นจาก MATLAB™ และ C แสดงในภาคผนวก ก.) ในบทที่ 5 ยังกล่าวถึงการพิสูจน์การลู่เข้าผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว โดยใช้ตรรกะทางคณิตศาสตร์ ร่วมกับตรรกะเชิงศึกษาสำนึก และทฤษฎีความน่าจะเป็น ผลการพิสูจน์ดังกล่าวพบว่า เมื่อกระบวนการค้นหาคำตอบผ่านไปด้วยระยะเวลาที่มากเพียงพออย่างสมเหตุผล ความน่าจะเป็นที่กระบวนการค้นหาแบบตามูเชิงปรับตัว จะพบคำตอบวงแคบเฉพาะถิ่นจะมีค่าเข้าใกล้หนึ่ง พบว่ากลไกปรับรัศมีการค้นหาช่วยให้การค้นหาแบบตามูเชิงปรับตัวลู่เข้าหาคำตอบด้วยเวลาที่รวดเร็วยิ่งขึ้น และพบว่ากลไกย้อนรอยการค้นหาช่วยให้การค้นหาแบบตามูเชิงปรับตัวสามารถพบคำตอบวงแคบเฉพาะถิ่นอื่น ๆ ในปริภูมิการค้นหา และหนึ่งในคำตอบวงแคบเฉพาะถิ่นที่พบคือคำตอบวงกว้าง นอกจากนี้ในบทที่ 5 ยังแสดงการวิเคราะห์สมรรถนะการค้นหาผลเฉลยของการค้นหาแบบตามูเชิงปรับตัว โดยการทดสอบเพื่อหาค่าต่ำสุดของฟังก์ชันผลรวมตรีโกณมิติแบบไม่สมมาตร ฟังก์ชันโบฮาเซฟสกี และฟังก์ชันวงกลม ซึ่งจากผลการทดสอบพบว่า การค้นหาแบบตามูเชิงปรับตัวมีสมรรถนะในการค้นหาคำตอบวงกว้างอย่างน่าพึงพอใจ

5. ระบบลูกค้อนาฬิกาผนวก หรือระบบ CPP เป็นระบบพลวัตระบบหนึ่งที่ถูกใช้เพื่อการระบุเอกลักษณ์ระบบในงานวิจัยวิทยานิพนธ์นี้ ระบบดังกล่าวได้รับการทดสอบเพื่อบันทึกข้อมูลอินพุตและเอาต์พุตอย่างถูกต้องและเพียงพอ ดังรายละเอียดที่ปรากฏในบทที่ 6 ข้อมูลอินพุตและเอาต์พุตของระบบได้รับการบันทึกโดยตัวตรวจรู้ และการประมวลผลภาพ โดยกำหนดให้ข้อมูลอินพุตเป็นสัญญาณสุ่ม การเคลื่อนที่ของระบบได้รับการบันทึกด้วยกล้องวีดีโอดิจิทัล การสกัดข้อมูลพลวัตจากลำดับภาพอาศัยเทคนิคการกำหนดจุดเปลี่ยนค่า เมื่อเปรียบเทียบกับข้อมูลจากตัวตรวจรู้ ข้อมูลจากการประมวลผลภาพปรากฏความคลาดเคลื่อนพอสมควร และเมื่อได้รับการแก้ไขด้วยระเบียบวิธีกำลังสองน้อยที่สุด ทำให้ข้อมูลมีความถูกต้องสูงขึ้นอย่างน่าพอใจ สำหรับรายละเอียดของโปรแกรมประมวลผลภาพระบบ CPP แสดงในภาคผนวก ค.

6. การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ CPP เชิงเส้นและไม่เชิงเส้น มีรายละเอียดดังปรากฏในภาคผนวก ข. การพัฒนาแบบจำลองดังกล่าวอาศัยสมการลากรางจ์แบบจำลองของระบบ CPP เชิงเส้นจะอยู่ในรูปของสมการอนุพันธ์สามัญเชิงเส้น ขณะที่แบบจำลองของระบบ CPP ไม่เชิงเส้นจะอยู่ในรูปของสมการอนุพันธ์สามัญไม่เชิงเส้น ตามลำดับ

7. การระบุเอกลักษณ์ระบบ CPP ดังรายละเอียดที่ปรากฏในบทที่ 7 แบ่งออกเป็นสองส่วน คือ ส่วนแรก เป็นการระบุเอกลักษณ์ระบบ CPP เชิงเส้น โดยใช้แบบจำลอง ARX ARMAX OE และ BJ อาศัยข้อมูลจากตัวตรวจรู้ และจากการประมวลผลภาพ เพื่อการระบุเอกลักษณ์ระบบ และตรวจสอบความถูกต้องของแบบจำลอง ซึ่งพบว่าแบบจำลอง BJ อันดับ 10 เป็นแบบจำลองที่มีความเหมาะสมสำหรับระบบ CPP เชิงเส้น ผลการระบุเอกลักษณ์ระบบโดยใช้ข้อมูลจากตัวตรวจรู้ และจากการประมวลผลภาพ มีความใกล้เคียงกันอย่างมาก ส่วนที่สอง เป็นการระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น โดยใช้การค้นหาแบบตามูเชิงปรับตัวเพื่อทำการค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลอง พบว่าผลการระบุเอกลักษณ์ระบบเมื่อใช้ข้อมูลจากตัวตรวจรู้ และจากการประมวลผลภาพ มีความใกล้เคียงกันอย่างมากเช่นเดียวกัน รายละเอียดของโปรแกรมระบุเอกลักษณ์ระบบ CPP แสดงในภาคผนวก ง. จากผลการระบุเอกลักษณ์ระบบ CPP ทั้งสองส่วน สรุปได้ว่าการระบุเอกลักษณ์ระบบโดยอาศัยเทคนิคการประมวลผลภาพที่พัฒนาขึ้น มีความถูกต้องแม่นยำเท่าเทียมกับการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิมที่อาศัยข้อมูลจากตัวตรวจรู้

8. ระบบกวัดแกว่งท่อยาง หรือระบบ VTS เป็นระบบพลวัตที่งานวิจัยวิทยานิพนธ์นี้พัฒนาขึ้นเพื่อใช้ทดสอบวิธีการระบุเอกลักษณ์ระบบด้วยการประมวลผลภาพ ระบบดังกล่าวเป็นระบบควบคุมวงรอบเปิด ดังรายละเอียดที่อธิบายไว้ในบทที่ 8 โครงสร้างของระบบ VTS ประกอบด้วยท่อยาง ที่ซึ่งปลายด้านหนึ่งได้รับการต่อเข้ากับเพลลาของมอเตอร์ไฟฟ้ากระแสตรง และปลายอีกด้านหนึ่ง ต่อเข้ากับจูดยัดซึ่งสามารถหมุนได้รอบตัว การควบคุมระบบ VTS อาศัยการกระตุ้นด้วยสัญญาณทวิภาคสุมเทียม ในฐานะสัญญาณอินพุตของระบบที่ผลิตได้จากไมโครคอนโทรลเลอร์ สัญญาณดังกล่าวถูกส่งไปยังส่วนเพิ่มกำลัง หรือตัวขับเร้า เพื่อเพิ่มกำลังให้กับสัญญาณ มอเตอร์ไฟฟ้ากระแสตรงจะถูกขับเคลื่อนด้วยสัญญาณกำลังจากตัวขับเร้า แรงจากเพลลาของมอเตอร์จะส่งผ่านไปตามท่อยาง และทำให้ท่อยางเกิดการเคลื่อนที่แบบ 3 มิติขึ้น รายละเอียดการควบคุมระบบ VTS แสดงในภาคผนวก จ.

9. ในการทดสอบระบบ VTS เพื่อบันทึกพลวัตของระบบแบบ 3 มิติ จะต้องอาศัยกล้องวิดีโอจิตอลจำนวน 2 ตัว ที่ผ่านการปรับเทียบมาก่อนแล้ว เทคนิคการปรับเทียบกล้องที่ใช้ในงานวิจัยวิทยานิพนธ์นี้ คือเทคนิคที่ได้รับการนำเสนอโดย Heikkila และ Silven ดังรายละเอียดที่ปรากฏในภาคผนวก ฉ. ซึ่งอาศัยหลักการหาค่าเหมาะที่สุดเพื่อคำนวณค่าพารามิเตอร์ของกล้อง โดยที่ค่าพารามิเตอร์เริ่มต้นของการหาค่าเหมาะที่สุดได้มาจากแบบจำลอง DLT ผลการปรับเทียบกล้องวิดีโอจิตอลด้วยเทคนิคดังกล่าว ทำให้ได้แบบจำลองกล้องที่เหมาะสม และทำให้ภาพที่ได้จากกล้องวิดีโอจิตอลทั้ง 2 ตัวได้รับการแก้ไขความเพี้ยนที่เกิดขึ้นอย่างน่าพอใจ

10. การทดสอบระบบ VTS ดังรายละเอียดที่ปรากฏในบทที่ 8 เป็นการทดสอบเพื่อบันทึกข้อมูลอินพุตและเอาต์พุตอย่างถูกต้องและเพียงพอสำหรับการระบุเอกลักษณ์ระบบ การทดสอบ

กำหนดให้ข้อมูลอินพุตเป็นสัญญาณทวิภาคสุ่มเทียม และข้อมูลเอาต์พุตเป็นการเคลื่อนที่ของทอยางแบบ 3 มิติ การเคลื่อนที่ดังกล่าวได้รับการบันทึกโดยกล้องวิดีโอดิจิทัลที่ผ่านการปรับเทียบมาก่อนแล้วจำนวน 2 ตัว การสกัดข้อมูลเอาต์พุตของระบบจากลำดับภาพอาศัยเทคนิคการกำหนดจุดเปลี่ยนค่า ซึ่งให้ผลออกมาอย่างน่าพึงพอใจ ข้อมูลอินพุตและเอาต์พุตที่ได้จากการทดสอบมีความถูกต้องและเพียงพอต่อการระบุเอกลักษณ์ระบบ สำหรับรายละเอียดของโปรแกรมประมวลผลภาพระบบ VTS แสดงในภาคผนวก ข.

11. การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS มีรายละเอียดดังปรากฏในภาคผนวก ข. โดยอาศัยหลักการสร้างแบบจำลองเส้นลาดชัน และเชื่อม แบบจำลองระบบ VTS แสดงในรูปของสมการคลื่น ซึ่งเป็นสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก แบบจำลองระบบ VTS แบบ 2 มิติ จะอยู่ในรูปของสมการอนุพันธ์ย่อยที่มีตัวแปรต้น 2 ตัว และแบบจำลองระบบ VTS แบบ 3 มิติ จะอยู่ในรูปของสมการอนุพันธ์ย่อยที่มีตัวแปรต้น 3 ตัว ตามลำดับ

12. การระบุเอกลักษณ์ระบบ VTS ดังรายละเอียดที่ปรากฏในบทที่ 9 เป็นการระบุเอกลักษณ์ระบบพลวัตที่มีการเคลื่อนที่แบบ 3 มิติ และมีแบบจำลองทางคณิตศาสตร์เป็นสมการอนุพันธ์ย่อย โดยแบ่งออกเป็นการระบุเอกลักษณ์ระบบในระนาบ x, y และในระนาบ z, x โดยอาศัยการค้นหาแบบตามูเชิงปรับตัวเพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสมของแบบจำลองการคำนวณผลเฉลยของแบบจำลองทางคณิตศาสตร์ของระบบ VTS ในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก อาศัยการคำนวณเชิงตัวเลขตามระเบียบวิธีผลต่างสืบเนื่อง ข้อมูลที่ใช้สำหรับการระบุเอกลักษณ์ระบบ VTS คือข้อมูลที่ได้จากการประมวลผลภาพที่ได้จากกล้องวิดีโอดิจิทัลจากการทดสอบระบบในบทที่ 8 ผลการระบุเอกลักษณ์ระบบ VTS พบว่า ในระนาบ x, y ผลที่ได้จากแบบจำลองมีความสอดคล้องกับผลที่ได้จากการทดสอบ ทั้งในส่วนของมุมและขนาด ขณะที่ในระนาบ z, x ผลที่ได้จากแบบจำลองและจากการทดสอบมีความสอดคล้องกันในส่วนของมุม แต่ปรากฏความคลาดเคลื่อนในส่วนองขนาดอยู่พอสมควร ค่าพารามิเตอร์ของแบบจำลองที่ได้จากการระบุเอกลักษณ์ระบบมีความใกล้เคียงกับค่าจริงมาก จากผลการระบุเอกลักษณ์ระบบ VTS สรุปได้ว่า การระบุเอกลักษณ์ระบบโดยอาศัยเทคนิคการประมวลผลภาพที่นำเสนอมีความถูกต้องแม่นยำอย่างน่าพึงพอใจ สำหรับรายละเอียดการคำนวณเชิงตัวเลขตามระเบียบวิธีผลต่างสืบเนื่องเพื่อคำนวณผลเฉลยของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก แสดงในภาคผนวก ฉ. ขณะที่รายละเอียดของโปรแกรมระบุเอกลักษณ์ระบบ VTS แสดงในภาคผนวก จ.

13. จากผลการดำเนินงานวิจัยวิทยานิพนธ์ที่ผ่านมา ก่อให้เกิดผลงานวิจัยตีพิมพ์และเผยแพร่ในรูปแบบต่าง ๆ ดังนี้ บทความวิจัยได้รับการตีพิมพ์ในวารสารวิชาการนานาชาติจำนวน 3 บทความ บทความวิจัยได้รับการตีพิมพ์ในการประชุมวิชาการนานาชาติจำนวน 3 บทความ และลิขสิทธิ์โปรแกรมการค้นหาแบบตามูเชิงปรับตัว ดังรายละเอียดที่ปรากฏในภาคผนวก ฉ.

10.2 ข้อเสนอแนะ

จากการดำเนินงานวิจัยวิทยานิพนธ์ที่ผ่านมาและผลที่ได้ ทำให้เกิดแนวคิดและข้อเสนอแนะในการดำเนินงานวิจัยต่อไปในอนาคต ดังต่อไปนี้

1. ควรมีการศึกษาวิจัยเกี่ยวกับการวิเคราะห์หาอันดับที่เหมาะสมของแบบจำลองกล่องดำเชิงเส้นที่สมมูลกับพลวัตของระบบ โดยอาจใช้แนวทางการประสาน หรือสังวัตนาการ (convolution) ระหว่างข้อมูลอินพุตและเอาต์พุตของระบบที่ได้จากการทดสอบ หรือแนวทางการค้นหาด้วยเทคนิคทางปัญญาประดิษฐ์ แทนการปรับอันดับของแบบจำลองจากน้อยไปสู่มาก (จากอันดับต่ำไปสู่อันดับสูง) เพื่อหาอันดับที่เหมาะสม

2. ในวิทยาการระบบควบคุมและในทางปฏิบัติ ระบบไม่เชิงเส้นได้รับการประยุกต์ใช้งานอย่างกว้างขวางเช่นเดียวกับระบบเชิงเส้น ดังนั้นแนวทางการระบุเอกลักษณ์แบบจำลองกล่องดำไม่เชิงเส้นชนิดที่ปรากฏค่าพารามิเตอร์ จึงควรได้รับการศึกษาวิจัยเพิ่มเติม โดยอาจเริ่มต้นจากการระบุเอกลักษณ์ระบบไม่เชิงเส้นแบบดั้งเดิม เช่นอาศัยแบบจำลอง NARMAX รวมทั้งการประยุกต์ใช้เทคนิคทางปัญญาประดิษฐ์เพื่อการค้นหาพารามิเตอร์ที่เหมาะสมของแบบจำลอง

3. การพิสูจน์การลู่เข้าหาผลเฉลยของ ATS ควรได้รับการศึกษาวิจัยเพิ่มเติมในแง่มุมต่าง ๆ เช่น อัตราการลู่เข้าหาผลเฉลย (convergence rate) ซึ่งจะนำเอาเวลา และความถี่ในการค้นพบคำตอบที่เหมาะสมเข้ามาเกี่ยวข้องกับการพิสูจน์ด้วย หรืออาจให้ความสนใจศึกษาถึงความสมมูลกันระหว่าง ATS กับทฤษฎีลูกโซ่มาร์คอฟ (Markov chains) ในบางแง่มุม รวมทั้งการพิสูจน์การลู่เข้าหาผลเฉลยของ ATS ตามทฤษฎีลูกโซ่มาร์คอฟ เป็นต้น

4. การค้นหาแบบตามูเชิงปรับตัว หรือ ATS ควรได้รับการพัฒนาให้มีสมรรถนะในการค้นหาคำตอบที่ดีขึ้น แนวทางหนึ่งคือ ภายในปริภูมิการค้นหาเดียวกัน อาจกำหนดให้มี ATS มากกว่าหนึ่งกระบวนการดำเนินการค้นหาไปพร้อม ๆ กัน ATS แต่ละกระบวนการซึ่งมีค่าคำตอบเริ่มต้นที่ต่างกัน จะดำเนินการค้นหาคำตอบที่เหมาะสมบนปริภูมิการค้นหาเดียวกัน โดยไม่แบ่งเป็นปริภูมิย่อยเพื่อแบ่งส่วนพื้นที่การค้นหา รายชื่อตามู หรือ TL ของ ATS แต่ละกระบวนการจะได้รับการเปรียบเทียบกันเป็นระยะ ๆ เพื่อกำจัด ATS บางกระบวนการที่มีพฤติกรรมการค้นหาคำตอบแบบวนเวียน ล่าช้า หรือเกิดการลือกโดยคำตอบวงแคบเฉพาะถิ่น และเมื่อ ATS กระบวนการใดพบคำตอบที่ทำให้ค่าฟังก์ชันวัตถุประสงค์สอดคล้องกับเกณฑ์ยุติการค้นหา กระบวนการค้นหาจะยุติลง จากแนวทางดังกล่าว อาจนำไปสู่การพัฒนาเทคนิคการค้นหาทางปัญญาประดิษฐ์แบบใหม่นามว่า MATS (multiple-adaptive tabu search) ซึ่งเป็นคนละแนวทางกับการค้นหาแบบตามูเชิงขนาน (parallel tabu search) ที่ซึ่ง TS แต่ละตัวจะทำการค้นหาคำตอบบนปริภูมิการค้นหาคนละปริภูมิกัน รวมทั้งมีการใช้โพรเซสเซอร์แบบขนานในการประมวลผลอีกด้วย นอกจากนี้ ควรมีการพิสูจน์การลู่เข้าหาผลเฉลย และอัตราการลู่เข้าของ MATS ด้วย

5. การใช้งานอุปกรณ์บันทึกภาพ ซึ่งประกอบด้วยกล้องวิดีโอดิจิทัล และอุปกรณ์รับภาพ ในฐานะตัวตรวจรู้ของระบบ ควรพิจารณาเลือกใช้อุปกรณ์ที่มีอัตราภาพที่สูงเพียงพอ ทั้งนี้เพื่อประกันว่าข้อมูลที่ได้จะมีความครบถ้วนสมบูรณ์ และสามารถใช้แทนพลวัตของระบบได้อย่างแท้จริงโดยไม่ปรากฏการเคลื่อนไหว โดยจะต้องพิจารณาถึงความสมมูลกันระหว่างอัตราภาพของอุปกรณ์บันทึกภาพและทฤษฎีบทการชักตัวอย่างของแซนอนประกอบด้วย ในอุปกรณ์รับภาพที่มีราคาถูก ถึงแม้จะมีอัตราภาพที่สูง แต่อาจประสบปัญหาการตกหายของภาพ (frame dropped) กรณีดังกล่าวอาจนำมาซึ่งการเคลื่อนไหวของข้อมูลได้เช่นเดียวกัน

6. กล้องวิดีโอดิจิทัลที่ได้รับการประยุกต์ใช้เป็นตัวตรวจรู้ของระบบ จะต้องได้รับการปรับเทียบด้วยเทคนิคที่เหมาะสมเพื่อแก้ไขความเพี้ยน ในงานบางประเภทที่ต้องการความถูกต้องเกี่ยวกับตำแหน่งและท่าทางของกล้องที่สัมพันธ์กับวัตถุ จำเป็นต้องให้ความสนใจเกี่ยวกับการแก้ไขความเพี้ยนเชิงเรขาคณิต (geometrical distortion) อันประกอบไปด้วย ความเพี้ยนเชิงรัศมี (radial distortion) ความเพี้ยนเชิงผิวสัมผัส (tangential distortion) ความเพี้ยนเยื้องศูนย์กลาง (decentering distortion) และความเพี้ยนปริซึมบาง (thin prism distortion) และในงานบางประเภทที่ต้องการความถูกต้องเกี่ยวกับความสว่าง ความเปรียบต่าง และสีที่แท้จริงของวัตถุ จะต้องให้ความสนใจเกี่ยวกับการแก้ไขความเพี้ยนเชิงรังสี (radiometrical distortion) ซึ่งเป็นความเพี้ยนอันเนื่องมาจากผลกระทบจากความสว่างของแสงภายนอก เงา และการสะท้อนแสงของวัตถุ นอกจากนี้การใช้งานกล้องวิดีโอดิจิทัลมากกว่าหนึ่งตัวในฐานะตัวตรวจรู้ของระบบ อาจจำเป็นต้องได้รับการปรับเทียบไปพร้อม ๆ กันด้วยเทคนิคการปรับเทียบกล้องแบบสเตอริโอ (stereo calibration)

7. ควรมีการศึกษาวิจัยเกี่ยวกับการออกแบบและอนุวัตอุปกรณ์ที่เรียกว่า Video Tracker เพื่อทำหน้าที่เป็นตัวตรวจรู้สำหรับการป้อนกลับด้วยภาพ (visual feedback) อุปกรณ์ดังกล่าวจะเป็นประโยชน์อย่างยิ่งต่อระบบควบคุมที่ไม่สามารถติดตั้งตัวตรวจรู้แบบทั่วไปได้ ทั้งนี้อาจเริ่มต้นจากการประยุกต์ใช้กล้องวิดีโอดิจิทัล และอุปกรณ์รับภาพที่สามารถหาซื้อได้ในท้องตลาด พร้อมทั้งการประดิษฐ์โปรแกรมประมวลผลภาพที่เหมาะสมเพื่อสกัดข้อมูลพลวัตของระบบ Video Tracker ที่พัฒนาขึ้นจะต้องได้รับการพัฒนาแบบจำลองทางคณิตศาสตร์อย่างถูกต้อง ซึ่งอาจปรากฏช่วงเวลาประวิง จะต้องได้รับการคำนวณและทดสอบหาอัตราการชักตัวอย่างหรืออัตราภาพที่แม่นยำ รวมทั้งการคำนวณเปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ยของข้อมูลพลวัตที่ได้ การออกแบบและอนุวัต Video Tracker จะเป็นการบูรณาการความรู้ด้านการพัฒนาแบบจำลองทางคณิตศาสตร์ การปรับเทียบกล้อง และการประมวลผลภาพอีกด้วย

8. การควบคุมระบบ VTS ควรได้รับการศึกษาวิจัยต่อไป ทั้งนี้เพื่อแสวงหาแนวทางการควบคุมระบบที่เคลื่อนไหวและอ่อนไหวอยู่ตลอดเวลา ให้สามารถเคลื่อนที่ตามลักษณะท่าทาง หรือ

ตามโหมคที่ต้องการได้ โดยอาจเริ่มต้นทำการดัดแปลงระบบให้สามารถควบคุมได้ง่ายขึ้น เช่นการเปลี่ยนมาใช้เชือกแทนท่อยาง หรือการปลดปลายของท่อทางด้านที่ต่อเข้ากับจุดยึดเพื่อให้สามารถเคลื่อนที่ได้อย่างอิสระ ความพยายามควบคุมระบบควรเริ่มต้นจากการควบคุมวงรอบเปิด เพื่อศึกษาพฤติกรรมทางพลวัตในโหมคต่าง ๆ โดยใช้อุปกรณ์บันทึกภาพทำหน้าที่เป็นตัวตรวจรู้ของระบบ เมื่อได้ศึกษาและวิเคราะห์พลวัตของระบบวงรอบเปิดอย่างท่วงแท้แล้ว จึงพัฒนาให้เป็นการควบคุมแบบวงรอบปิดที่อาศัยการป้อนกลับด้วยภาพ เพื่อควบคุมให้ระบบเคลื่อนที่ในโหมคต่าง ๆ ตามที่ต้องการ ทั้งนี้การออกแบบระบบอาจจำเป็นต้องพิจารณาภายใต้บริบทการควบคุมแบบคงทน เนื่องจากการป้อนกลับด้วยภาพจะปรากฏช่วงเวลาประวิง และความไม่แน่นอนขึ้นในแบบจำลอง จากนั้นจึงทำการเปลี่ยนจากเชือกกลับมาเป็นท่อยาง และทำการควบคุมระบบตามลำดับ ระบบควบคุม VTS ควรได้รับการออกแบบทั้งฮาร์ดแวร์และซอฟต์แวร์ และพัฒนาเป็นระบบต้นแบบเพื่อเป็นตัวอย่างในการประยุกต์ใช้งานจริง ทั้งนี้เพราะผู้วิจัยมีความเชื่อมั่นว่าในอนาคตจะมีการประยุกต์ใช้ประโยชน์ระบบควบคุมในลักษณะดังกล่าวอย่างแน่นอน

9. สืบเนื่องจากข้อ 8 ในกรณีที่ระบบ VTS เคลื่อนที่ด้วยความเร็วค่อนข้างสูง จนกระทั่งอุปกรณ์บันทึกภาพซึ่งทำหน้าที่เป็นตัวตรวจรู้ของระบบ ไม่สามารถตอบสนองได้ทันการ การเคลื่อนที่ของข้อมเกิดขึ้นกับข้อมูลอย่างเล็งไม่พ้น สำหรับแนวทางการแก้ปัญหาดังกล่าวข้างต้น อาจจำเป็นต้องใช้อุปกรณ์บันทึกภาพมากกว่าหนึ่งตัว ที่ได้รับการติดตั้งอยู่ตามตำแหน่งต่าง ๆ ที่เหมาะสม เพื่อบันทึกพลวัตของระบบในเวลาที่เหมาะสมไป ภาพจากอุปกรณ์บันทึกภาพแต่ละตัว จะได้รับการประมวลผลตามลำดับก่อนหลัง ข้อมูลพลวัตที่ได้จะถูกจัดเก็บไว้เป็นชุด และระบบควบคุมจะนำข้อมูลในอดีตที่เก็บมาได้เป็นชุด ๆ ไปดำเนินการกระบวนการตามลำดับ แนวทางดังกล่าวจะเอื้อประโยชน์ต่อการควบคุมระบบป้อนกลับด้วยภาพเป็นอย่างมาก ในกรณีที่เทคโนโลยีในการผลิตให้กล้องวีดีโอดิจิทัลและอุปกรณ์รับภาพมีอัตราภาพที่สูงมาก ๆ ยังไปไม่ถึง หรืออุปกรณ์มีราคาแพงมาก แนวทางที่กล่าวมา จะนำไปสู่การศึกษาและวิเคราะห์ระบบควบคุมป้อนกลับด้วยภาพที่ตั้งอยู่บนบริบทการควบคุมแบบคงทน และการควบคุมเชิงพยากรณ์ (predictive control)

10. ควรมีการศึกษาและพัฒนาเทคนิคการระบุเอกลักษณ์ระบบด้วยการประมวลผลภาพสำหรับระบบ VTS เพื่อให้ได้แบบจำลองทางคณิตศาสตร์แบบ 3 มิติ โดยให้ความสนใจศึกษาวิเคราะห์ในแง่มุมของการติดตั้งกล้องตามตำแหน่งต่าง ๆ ที่เหมาะสม รวมทั้งการจัดวางท่อยางในระนาบที่เหมาะสม เช่น ในระนาบตั้งฉาก และในระนาบเฉียง เป็นต้น

11. การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ CPP ไม่เชิงเส้น โดยเฉพาะแบบจำลองในส่วนองแรงที่ส่งผ่านไปตามสายพานอ่อน ควรได้รับการศึกษาและพัฒนาอย่างละเอียด โดยอาจพิจารณาให้แบบจำลองในส่วนดังกล่าวอยู่ในรูปของสมการอนุพันธ์ย่อย เพื่อให้สามารถอธิบายพลวัตของระบบได้อย่างแม่นยำมากที่สุด

รายการอ้างอิง

- กองพัน อารีรักษ์. (2544). การระบุเอกลักษณ์ไม่เป็นเชิงเส้นและการตรวจสอบเสถียรภาพของระบบ
สองมวลความเฉื่อยที่ปรากฏการกำทอนเชิงกล. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตร
มหาบัณฑิต, มหาวิทยาลัยเทคโนโลยีสุรนารี, นครราชสีมา.
- ปราโมทย์ เดชะอำไพ. (2544). ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม. จุฬาลงกรณ์มหาวิทยาลัย.
- วัชรพงษ์ โขวิฑูรกิจ. (2546). คณิตศาสตร์วิศวกรรมไฟฟ้าขั้นสูง. จุฬาลงกรณ์มหาวิทยาลัย.
- Abdel-Aziz, Y.I., and Karara, H.M. (1971). Direct linear transformation into object
space coordinates in close-range photogrammetry. In **Proceeding of the
Symposium on Close-Range Photogrammetry**. (pp.1-18). USA, Urbana:
Illinois.
- Aggarwal, J.K., and Nandhakumar, N. (1988). On the computation of motion from
sequences of images-a reviews. In **Proceeding of IEEE**. 76: 917-935.
- AI, B. (2000). **Handbook of Image and Video Processing**. Academic Press.
- Bakstein, H. (1999). **A complete DLT-based camera calibration with a virtual 3D
calibration object**. Diploma thesis, Charles University, Prague.
- Battiti, R. (1996). Reactive search: toward self-tuning heuristics. **Modern heuristic
search method**. John Wiley & Sons.
- Battiti, R., and Tecchiolli, G. (1994). The reactive Tabu search. **ORSA Journal on
Computing**. 6(2): 126-140.
- Bloch, A.M., Leonard, N.E., and Marsden, J.E. (2000). Controlled lagrangians and the
stabilization of mechanical system I: the first matching theorem. **IEEE
Transaction on Automatic Control**. 45(12): 2253-2270.

- Boaghe, O.M., Billings, S.A., Li, L.M., Fleming, P.J., and Liu, J. (2002). Time and frequency domain identification and analysis of a gas turbine engine. **Control Engineering Practice**. 10: 1347-1356.
- Bohachevsky, I.O., Johnson, M.E., and Stein, M.L. (1986). Generalized simulated annealing for function optimization. **Technometrics**. 28(3): 209-218.
- Cannon, R.H. (1967). **Dynamic of physical systems**. McGraw-Hill.
- Chen, X., Kano, H., and Fukuda, T. (2000). A new robust observer for the perspective system. In **Proceeding of the 39th Conference on Decision and Control**. (pp.5163-5168). Australia.
- Chong, E.K.P., and Zak, S.H. (2001). **An Introduction of Optimization**. Wiley Interscience.
- Chung, H.-C., Liang, J., Kushiyama, S., and Shinozuka, M. (2004). Digital image processing for non-linear system identification. **Non-Linear Mechanics**. 39: 691-707.
- Cordeau, J-F., and Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. **Transportation Research Part B: Methodological**. 37(6): 579-594.
- Davies, W.D.T. (1970). **System identification for self-adaptive control**. John Wiley & Sons.
- Eykhoff, P. (1974). **System identification: Parameter and state estimation**. John Wiley & Sons.
- Feedback Instruments Ltd. Digital pendulum control system.
- Forssell, U., and Ljung, L. (1998). Identification of unstable systems using Output Error and Box-Jenkins model structures. In **Proceeding of the 37th Conference on Decision and Control**. (pp.3932-3937). USA.

- Forssell, U., and Ljung, L. (2000). A projection method for closed-loop identification. **IEEE Transaction on Automatic Control**. 45(11): 2101-2106.
- Forssell, U., and Ljung, L. (2000). Identification of unstable systems using Output Error and Box-Jenkins model structures. **IEEE Transaction on Automatic Control**. 45(1): 137-141.
- Forstor, R.R., Jezek, K.C., and Sohn, H.G. (1998). Analysis of glacier flow dynamics from preliminary RADARSAT InSAR data. In **Proceeding of IEEE International Symposium on Geoscience and Remote Sensing**. (pp.2225-2227). USA, WA: Seattle.
- French, A.P. (1992). **Vibrations and waves**. M.I.T Introductory Physics Series, Chapman & hall.
- Fukuyama, Y. (2000). Reactive tabu search for distribution load transfer operation. In **IEEE Power Engineering Society**. Winter Meeting, (pp.1-6). Singapore.
- Ghosh, B.K., and Loucks, E.P. (1996). A realization theory for perspective systems with applications to parameter estimation problems in machine vision. **IEEE Transaction on Automatic Control**. 41(12): 1706-1722.
- Ghosh, B.K., Inaba, H., and Takahashi, S. (2000). Identification of riccati dynamic under perspective and orthographic observations. **IEEE Transaction on Automatic Control**. 45(7): 1267-1278.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. **Computers and Operations Research**. 13: 533-549.
- Glover, F. (1989). Tabu search – Part I. **ORSA Journal on Computing**. 1: 190-206.
- Glover, F. (1990). Tabu search – Part II. **ORSA Journal on Computing**. 2: 4-32.
- Glover, F. (1994). Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). **Discrete Applied Mathematics**. 49: 231-255.

- Glover, F., and Hanafi, S. (2001). Finite convergence of Tabu Search. In **MIC'2001-4th Metaheuristics**. (pp.333-336).
- Glover, F., and Laguna, M. (1997). **Tabu Search**. Kluwer Academic Publishers. Norwell.
- Glover, F., Mulvey, J.M., and Hoyland, K. (1994). Solving dynamic stochastic control problems in finance using tabu search with variable scaling. **Statistics and Operations Research Technical Report SOR-94-13**. Princeton University. USA.
- Guerrero, J.J., and Sagues, C. (2001). Estimating the motion direction from brightness gradient on lines. **IEEE Transaction on System, Man, and Cybernetics, part C: Applications and Reviews**. 31(3): 419-426.
- Hanafi, S. (2000). On the convergence of Tabu search. **Heuristics**. 7: 47-58.
- Heikkila, J., and Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In **IEEE International Conference on Computer Vision and Pattern Recognition**. (pp.1106-1112).
- Hertz, A. (1992). Finding a feasible course schedule using tabu search. **Discrete Applied Mathematics**. 35: 255-270.
- Hertz, A., and De Werra, D. (1987). Using tabu search technique for graph coloring. **Computing**. 39: 345-351.
- Jerri, A.J. (1977). The Shannon sampling theorem-its various extensions and applications: A tutorial reviews. In **Proceeding of the IEEE**. 65(11): 1565-1596.
- Johansson, R. (1993). **System modeling and identification**. Prentice-Hall.

- Kano, H., and Ghosh, B.K. (2000). Identification of relative position and orientation of two cameras from motion and shape parameters of moving rigid body. In **Proceeding of the 39th Conference on Decision and Control**. (pp.5169-5174). Australia.
- Karaboga, D., and Kalinli, A. (1997). Training recurrent neural network for dynamic system identification using parallel tabu search algorithm. In **Proceeding of the 12th IEEE International Symposium of Intelligent Control**. (pp.113-118). Turkey.
- Kochetov, Y.A., and Goncharov, E.N. (2000). Probabilistic Tabu search algorithm for multi-stage uncapacitated facility location problem. In **Operations Research Proceeding. Springer**. (pp.65-70).
- Kreyszig, E. (1978). **Introductory Functional Analysis with Applications**. John Wiley & Sons.
- Kulworawanichpong, T., and Sujitjorn, S. (2002). Optimal power flow using Tabu search. **IEEE Power Engineering Review**. 22(6): 37-40.
- Landau, I.D. (1990). **System identification and control design using P.I.M. + software**. Prentice-Hall.
- Lee, G.D., Jun, S., and Kim, S.W. (2000). Guided neural network and its application to longitudinal dynamic identification of a vehicle. **IEICE Transaction on Fundamentals**. E83-A(7): 1467-1472.
- Lenz, R.K., and Tsai, R.Y. (1988). Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology. **IEEE Transaction on Pattern Analysis and Machine Intelligent (PAMI)**. 10(5): 713-720.

- Lin, B., and Miller, D.C. (2000). Application of Tabu search to model identification. In **AICHE Annual Meeting**. Los Angeles.
- Ljung, L. (1987). **System identification: Theory for the user**. Prentice-Hall.
- Ljung, L. (1991). A simple start-up procedure for canonical form state-space identification, based on subspace approximation. In **Proceeding of the 30th Conference on Decision and Control**. (pp.1333-1336). England.
- Ljung, L. (1994). From data to model: A guided tour. In **International Conference on Control**. 1: (203-208). UK.: Coventry.
- Ljung, L. (1995). **System identification toolbox: User's guide**. The Math Works, Inc.
- Ljung, L. (2001). Black-box models from input-output measurements. In **IEEE International Conference on Instrument and Measurement Technology**. (pp.138-146). Hungary.
- Ljung, L., and Forssell, U. (1999). An alternative motivation for the indirect approach to closed-loop identification. **IEEE Transaction on Automatic Control**. 44(11): 2206-2209.
- Ljung, L., and Glad, T. (1994). **Modeling of dynamic systems**. PRT Prentice-Hall.
- Ljung, L., and Sjoberg, J. (1992). A system identification perspective on neural nets. In **Proceeding of IEEE-SP Workshop**. (pp.423-435). Denmark: Helsingoer.
- Mantawy, A.H., Abdel-Magid, Y.L., and Selim, S.Z. (1998). Unit commitment by tabu search. **IEE Generation Transmission and Distribution**. 1(45): 56-64.
- Mathews, J.H., and Fink, K.D. (2004). **Numerical methods using MATLAB**. PRT Prentice-Hall.

- Matsuo, T., Yoshino, R., Suemitsu, H., and Nakano, K. (2004). Nominal performance recovery by PID+Q controller and its application to antisway control of crane lifter with visual feedback. **IEEE Control System Technology**. 12(1): 156-166.
- McKelvey, T., Akcay, H., and Ljung, L. (1996). Subspace-based multivariable system identification from frequency response data. **IEEE Transaction on Automatic Control**. 41(7): 960-979.
- Miller, S.D., Marchetto, J., Airaghi, S., Koumoutsakos, P. (2002). Optimization based on bacterial chemotaxis. **IEEE Evolutionary Computation**. 6: 16-29.
- Nalwa, V.S. (1993). **A guided tour of computer vision**. Addison-Wesley.
- Nara, K., Hayashi, Y., Ikeda, K., and Ashizawa, T. (2001). Application of Tabu search to optimal placement of distributed generators. In **IEEE International Conference on Power Engineering Society Winter Meeting**. (pp.928-932).
- Niu, S.S., Ljung, L., and BJORCK, A. (1996). Decomposition methods for solving least-squares parameter estimation. **IEEE Transaction on Signal Processing**. 44(11): 2847-2852.
- Nowicki, E., and Smutnicki, C. (1996). A fast tabu search algorithm for the flow shop problem. **European Journal of Operational Research**. 91: 160-175.
- Rafael, C.G., and Richard, E. W. (1992). **Digital Image Processing**. Addison-Wesley Publishing Company.
- Roach, J.W., and Aggarwal, J.K. (1980). Determining the movement of objects from a sequence of images. **IEEE Transaction on Pattern Analysis and Machine Intelligence**. PAMI-2(6): 554-562.

- Rousseau, P., Desrochers, A., and Krouglicof, N. (2001). Machine vision system for the automatic identification of robot kinematic parameters. **IEEE Transaction on Robotics and Automation**. 17(6): 972-978.
- Silva, E.L.D., Areiza, J.M.O., Oliveira, G.C.D., Binato, B. (2001). Transmission network expansion planning under a tabu search approach. **IEEE Power Systems**. 16(1): 62-68.
- Stiller, C., and Konrad, J. (1999). Estimating motion in image sequences. **IEEE Signal Processing Magazine**. : 70-91.
- Sun, L., Liu, W., and Sano, A. (1997). Over-sampling approach to closed-loop identification. In **Proceeding of the 36th Conference on Decision and Control**. (pp.1253-1258). USA.
- Toune, S., Fudo, H., Genji, T., Fukuyama, Y., and Nakanishi, Y. (1998). A reactive Tabu search for service restoration in electric power distribution systems. In **IEEE Evolutionary Computation**. : 1-7. Alaska.
- Tsai, R.Y. (1987). A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. **IEEE Journal of Robotics and Automation** 3(4): 323-344.
- Van Den Hof, P.M.J. (1991). Approximate identification with linear regression models. In **Proceeding of the 30th Conference on Decision and Control**. (pp.3108-3113). England.
- Van Den Hof, P.M.J., Heuberger, P.S.C., and Bokor, J. (1994). System identification with generalized orthonormal basis functions. In **Proceeding of the 33rd Conference on Decision and Control**. (pp.3382-3387). USA: FL.

- Van Den Hof, P.M.J., Schrama, R.J.P., and Bosgar, O.H. (1992). An indirect method for transfer function estimation from closed loop data. In **Proceeding of the 31st Conference on Decision and Control**. (pp.1702-1706). USA: Arizona.
- Viberg, M., Ottensten, B., Wahlberg, B., and Ljung, L. (1991). A statistical perspective on state-space modeling using subspace methods. In **Proceeding of the 30th Conference on Decision and Control**. (pp.1337-1342). England.
- Vries, D.K., and Van Den Hof, P.M.J. (1995). Frequency domain identification with generalized orthonormal basis functions. In **Proceeding of the 34th Conference on Decision and Control**. (pp.1240-1245). USA: LA.
- Wallin, R., Isaksson, A.J., and Ljung, L. (2000). An indirect method for identification of ARX models from incomplete data. In **Proceeding of the 39th Conference on Decision and Control**. (pp.203-208). Australia.
- Webster, R. (1994). **Convexity**. Oxford Science Publication.
- Weng, J., Huang, T.S., and Ahuja, N. (1987). 3-D motion estimation, understanding, and prediction from noisy image sequences. **IEEE Transaction on Pattern Analysis and Machine Intelligence**. PAMI-9(3): 370-389.
- Yao, X., Karady, G.G., Farmer, R.G., and Agrawal, B.L. (2001). Estimation of generator excitation-system parameters by tabu search. In **IEEE Power Engineering Society**. Winter Meeting. (3): 1185 – 1190.
- Yeung, L.F., and Li, X.W. (1996). Multi-input system identification and its applications using wavelet constructive method. In **Proceeding of the 35th Conference on Decision and Control**. (pp.3230-3235). Japan.
- Zhang, G., Habenicht, W., and SpieB, W.E.L. (2003). Improving the structure of deep frozen and chilled food chain with tabu search procedure. **Food Engineering**. 60(1): 67-79.

ภาคผนวก ก

โปรแกรมการค้นหาแบบตามูเชิงปรับตัว

โปรแกรมการค้นหาแบบตามูเชิงปรับตัว

โปรแกรมการค้นหาแบบตามูเชิงปรับตัว หรือโปรแกรม ATS ในงานวิจัยวิทยานิพนธ์นี้ ได้พัฒนาขึ้นโดยอาศัยโปรแกรม MATLAB™ และ C ดังรายละเอียดต่อไปนี้

โปรแกรม ATS ที่พัฒนาด้วย MATLAB™

```
% โปรแกรม ATS ที่พัฒนาด้วย MATLAB™
% เริ่มต้นประจักษ์โดยกองพัน อารีรักษ์ (2002)
% พัฒนาและปรับปรุงโดยเฉชา พวงดาวเรือง (2003)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี
```

โปรแกรม ATS.m

```
function [best_neighbor,best_error,time,radius,count,local]=ATS(N)

% โปรแกรมหลัก มี N เป็นอินพุตของโปรแกรม
% N คือจำนวนพารามิเตอร์ที่ต้องการค้นหา (N = 2 เพื่อเป็นตัวอย่าง)
% เอาต์พุตของโปรแกรมคือ
% ค่าพารามิเตอร์ที่ดีที่สุดหรือค่าคำตอบวงกว้าง (best_neighbor)
% ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด (best_error)
% ค่าเวลาที่ใช้ในการค้นหา (time)
% ค่ารัศมีการค้นหา (radius)
% ค่าจำนวนรอบการค้นหาทั้งหมด (count)
% ค่าคำตอบวงแคบเฉพาะถิ่น (local)

xlimit=[1 1;-1 -1];
% กำหนดปริภูมิการค้นหาของพารามิเตอร์
% แถวที่ 1 ของ xlimit คือขอบเขตบนของพารามิเตอร์
% แถวที่ 2 ของ xlimit คือขอบเขตล่างของพารามิเตอร์

for r=1:5
    S(r,:)=(xlimit(1,:)-xlimit(2,:)).*rand(1,N)+xlimit(2,:);
end
% สุ่มคำตอบเริ่มต้นภายในปริภูมิการค้นหาของพารามิเตอร์แต่ละตัวจำนวน 5 ชุด
% โดยอาศัยความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 

for k=1:size(S,1)
    ysim=obj(S(k,1),S(k,2));
    costvalue(k,1)=ysim;
end
% ประเมินคำตอบเริ่มต้นด้วยฟังก์ชันวัตถุประสงค์ใน โปรแกรม obj
% อินพุตของ โปรแกรม obj คือคำตอบเริ่มต้นที่สุ่มได้จำนวน 5 ชุด
% เอาต์พุตคือค่าฟังก์ชันวัตถุประสงค์ที่ได้จากคำตอบแต่ละชุด
```

```

% เก็บค่าฟังก์ชันวัตถุประสงค์ทั้ง 5 ค่าไว้ใน costvalue
[best_error,index]=min(costvalue); % หาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดใน costvalue
% เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดไว้ใน best_error
S0=S(index,:); % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุดไว้ใน S0
max_count=10000; % กำหนดจำนวนรอบสูงสุดในการค้นหา
% เพื่อใช้เป็นเกณฑ์ยุติการค้นหา (10,000 เพื่อเป็นตัวอย่าง)
best_neighbor=S0; % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
% ไว้ใน best_neighbor
neighbor_list=zeros(5,N); % รีเซ็ตค่า neighbor_list
radius=0.2; % กำหนดรัศมีการค้นหาเริ่มต้น (0.2 เพื่อเป็นตัวอย่าง)
Number_neighb=30; % กำหนดจำนวนการสุ่มค่าใกล้เคียง (30 เพื่อเป็นตัวอย่าง)
overall_best_error=best_error; % ปรับค่า overall_best_error = best_error
overall_neighbor=best_neighbor; % ปรับค่า overall_neighbor = best_neighbor
n=0; % กำหนดค่าเริ่มต้นของจำนวนการซ้ำของค่าคำตอบ
t=0; % กำหนดค่าตัวเลื่อนสำหรับ local
tt=0; % กำหนดค่าตัวเลื่อนสำหรับ tabu_list
ttt=0; % กำหนดค่าตัวเลื่อนสำหรับ best_error_list
count=0; % กำหนดค่าเริ่มต้นของจำนวนรอบการค้นหา
n_back_tracking=0; % กำหนดค่าเริ่มต้นของจำนวนการเรียกใช้กลไกย้อนรอยการค้นหา
tic; % เริ่มต้นคำนวณเวลาที่ใช้ในการค้นหา
% Start ATS
t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
ttt=ttt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ best_error_list
disp([count best_error overall_best_error])
% แสดงค่า count, best_error, และ overall_best_error
local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
local(t,2:3)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 และ 3 ของ local
local(t,4)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 4 ของ local
tabu_list(tt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ tabu_list
tabu_list(tt,2:3)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 และ 3 ของ tabu_list
tabu_list(tt,4)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 4 ของ tabu_list
best_error_list(ttt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ best_error_list
best_error_list(ttt,2:3)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 และ 3 ของ best_error_list
best_error_list(ttt,4)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 4 ของ best_error_list
for count=1:max_count % วนรอบการค้นหา
S1=random_neigh(Number_neighb,radius,xlimit,S0);
% เรียกใช้โปรแกรม random_neigh เพื่อสุ่มค่าใกล้เคียงรอบคำตอบ S0
% ภายในปริภูมิการค้นหาปัจจุบัน
% อินพุตของโปรแกรมนี้คือ Number_neighb, radius, xlimit, และ S0
% เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ Number_neighb (30 ตัว)

```



```

[best_error1,best_neighbor1,best_error,best_neighbor]=objective(S1,best_error,S0);
    % เรียกใช้โปรแกรม objective เพื่อประเมินค่าใกล้เคียงที่สุดที่ได้
    % อินพุตของ โปรแกรมนี้คือ S1, best_error, และ S0
    % เอาต์พุตคือ best_error1, best_neighbor1,
    % best_error, และ best_neighbor
neighbor_list(k,1:size(S0,2))=[best_neighbor1];
    % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
    % ภายในปริภูมิการค้นหาย่อยปัจจุบันไว้ในคอลัมน์ที่ 1 และ 2 ของ neighbor_list
neighbor_list(k,size(S0,2)+1)=best_error1;
    % เก็บค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
    % ภายในปริภูมิการค้นหาย่อยปัจจุบันไว้ในคอลัมน์ที่ 3 ของ neighbor_list
if (best_error1-best_error)>1e-18
    n=n+1;
else
    n=0;
end
    % ตรวจสอบการซ้ำของค่าคำตอบโดยการเปรียบเทียบระหว่าง
    % best_error1 และ best_error ถ้ามีการซ้ำของค่าคำตอบ ให้ปรับเพิ่มค่า n
tt=tt+1;
    % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
tabu_list(tt,1)=count;
    % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ tabu_list
tabu_list(tt,2:3)=best_neighbor1;
    % เก็บค่า best_neighbor1 ไว้ในคอลัมน์ที่ 2 และ 3 ของ tabu_list
tabu_list(tt,4)=best_error1;
    % เก็บค่า best_error1 ไว้ในคอลัมน์ที่ 4 ของ tabu_list
ttt= ttt+1;
    % ปรับเพิ่มค่าตัวเลื่อนสำหรับ best_error_list
best_error_list(ttt,1)=count;
    % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ best_error_list
best_error_list(ttt,2:3)=best_neighbor;
    % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 และ 3 ของ best_error_list
best_error_list(ttt,4)=best_error;
    % เก็บค่า best_error ไว้ในคอลัมน์ที่ 4 ของ best_error_list
disp([count best_error overall_best_error])
    % แสดงค่า count, best_error, และ overall_best_error

% Start AR mechanism
    % (กำหนดให้มี 4 เงื่อนไขเพื่อเป็นตัวอย่าง)
if best_error<=1e-1
    % เงื่อนไขที่ 1) ถ้า best_error  $\leq 1 \times 10^{-1}$  แล้ว radius =  $2 \times 10^{-3}$ 
    radius=2e-3;
end
if best_error<=1e-2
    % เงื่อนไขที่ 2) ถ้า best_error  $\leq 1 \times 10^{-2}$  แล้ว radius =  $2 \times 10^{-5}$ 
    radius=2e-5;
end
if best_error<=1e-3
    % เงื่อนไขที่ 3) ถ้า best_error  $\leq 1 \times 10^{-3}$  แล้ว radius =  $2 \times 10^{-7}$ 
    radius=2e-7;
end
if best_error<=1e-4
    % เงื่อนไขที่ 4) ถ้า best_error  $\leq 1 \times 10^{-4}$  แล้ว radius =  $2 \times 10^{-9}$ 
    radius=2e-9;
end

```

```

end
if (best_error<1e-5)          % ตรวจสอบเกณฑ์ยุติการค้นหา เมื่อ best_error < 1×10-5
    t=t+1;                    % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
    tt=tt+1;                  % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
    disp([count best_error overall_best_error])
                                % แสดงค่า count, best_error, และ overall_best_error
    local(t,1)=count;         % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
    local(t,2:3)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 และ 3 ของ local
    local(t,4)=best_error;    % เก็บค่า best_error ไว้ในคอลัมน์ที่ 4 ของ local
    break;                    % ยุติการค้นหา
end
% Start BT mechanism
                                % เมื่อจำนวนการซ้ำของค่าคำตอบเท่ากับ 5 (เพื่อเป็นตัวอย่าง)
if n>=5
    n_back_tracking=n_back_tracking+1;
                                % เพิ่มค่า n_back_tracking
    TEMP=tabu_list(count-3:count+1,:);
                                % ทำการจัดลำดับ (ranking) ค่าฟังก์ชันวัตถุประสงค์
                                % และค่าพารามิเตอร์ 5 ชุดสุดท้ายก่อนมีการเรียกใช้กลไกย้อนรอยการค้นหา
                                % TEMP จะเก็บค่าฟังก์ชันวัตถุประสงค์ และค่าพารามิเตอร์ 5 ชุดสุดท้าย
                                % ก่อนทำการจัดลำดับ
    [MAX,INDEX] = max(TEMP(:,4));
                                % หาค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
    RANK(5,:) = TEMP(INDEX,:);
                                % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
                                % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 5 ใน RANK
    TEMP(INDEX,4)=0;           % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
                                % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 4 ค่า
    [MAX,INDEX] = max(TEMP(:,4));
                                % หาค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
    RANK(4,:) = TEMP(INDEX,:);
                                % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
                                % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 4 ใน RANK
    TEMP(INDEX,4)=0;           % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
                                % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 3 ค่า
    [MAX,INDEX] = max(TEMP(:,4));
                                % หาค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
    RANK(3,:) = TEMP(INDEX,:);
                                % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
                                % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 3 ใน RANK
    TEMP(INDEX,4)=0;           % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
                                % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 2 ค่า

```

```

[MAX,INDEX] = max(TEMP(:,4));
    % หาค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(2,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 2 ใน RANK
TEMP(INDEX,4)= 0; % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 1 ค่า
[MAX,INDEX] = max(TEMP(:,4));
    % หาค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(1,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 1 ใน RANK
TEMP(INDEX,4)= 0; % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะไม่มีค่าฟังก์ชันวัตถุประสงค์ให้ทำการจัดลำดับอีกต่อไป
neighbor=RANK(5,2:3); % แทนค่า neighbor ด้วยค่าพารามิเตอร์ลำดับที่ 5 ใน RANK
    % ซึ่งเป็นค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์แตกต่างจาก
    % ค่าคำตอบที่ซ้ำมากที่สุดเมื่อเทียบกับอีก 4 ลำดับที่เหลือใน RANK
S0=neighbor; % ปรับค่า S0 = neighbor
if best_error<overall_best_error
    overall_best_error=best_error;
    overall_best_neighbor=best_neighbor;
    % ปรับค่า overall_best_error
    % ถ้า best_error < overall_best_error
    % ให้แทนค่า overall_best_error ด้วย best_error
    % และแทนค่าพารามิเตอร์ overall_best_neighbor ด้วย best_neighbor
t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
local(t,2:3)=best_neighbor;
    % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 และ 3 ของ local
local(t,4)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 4 ของ local
end
best_error=RANK(5,4); % แทนค่า best_error ด้วยค่าฟังก์ชันวัตถุประสงค์ลำดับที่ 5 ใน RANK
    % ซึ่งเป็นค่าฟังก์ชันวัตถุประสงค์ที่แตกต่างจาก
    % ค่าคำตอบที่ซ้ำมากที่สุดเมื่อเทียบกับอีก 4 ลำดับที่เหลือใน RANK
n=0; % ปรับค่า n = 0
else % ไม่เช่นนั้นแล้ว
    S0=best_neighbor; % แทนค่า S0 ด้วย best_neighbor
    best_error=best_error;% แทนค่า best_error ด้วย best_error
end
end
if overall_best_error<best_error

```

```

best_error=overall_best_error;
best_neighbor=overall_best_neighbor;

% ปรับค่า best_error
% ถ้า overall_best_error < best_error
% ให้แทนค่า best_error ด้วย overall_best_error
% และแทนค่าพารามิเตอร์ best_neighbor ด้วย overall_best_neighbor

end

time=toc; % ผู้คิดการคำนวณเวลาการค้นหา

return

```

โปรแกรม obj.m

```

function y=obj(x,y) % โปรแกรม obj คือ โปรแกรมคำนวณค่าฟังก์ชันวัตถุประสงค์
% กำหนดให้เป็นฟังก์ชัน โบฮาเชฟสกี (เพื่อเป็นตัวอย่าง)
% อินพุตของ โปรแกรมคือคำตอบเริ่มต้นที่สุ่มได้ (x,y)
% เอาต์พุตคือค่าฟังก์ชันวัตถุประสงค์ที่ได้จากคำตอบ x,y

y=(x.^2)+(2*y.^2)-(0.3*cos(3*pi*x))-(0.4*cos(4*pi*y))+0.7;

return

```

โปรแกรม objective.m

```

function [best_error1,best_neighbor1,best_error,best_neighbor]=objective(S1,best_error,S0)
% โปรแกรม objective เป็น โปรแกรมประเมินค่าใกล้เคียงที่สุ่มได้รอบ S0
% อินพุตของ โปรแกรมนี้คือ S1, best_error, และ S0
% เอาต์พุตคือ best_error1, best_neighbor1,
% best_error, และ best_neighbor

error=[]; % ตัวแปร error สำหรับรองรับค่าฟังก์ชันวัตถุประสงค์
% ของค่าใกล้เคียงที่ได้จากการสุ่ม

for k=1:size(S1,1)
    ysim=obj(S1(k,1),S1(k,2)); % ประเมินค่าใกล้เคียงใน S1 ด้วยฟังก์ชันวัตถุประสงค์ในโปรแกรม obj
    error(k,1)=ysim; % เก็บค่าฟังก์ชันวัตถุประสงค์ของค่าใกล้เคียงทั้งหมด
end

[best_error1,index]=min(error); % หาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในบรรดาค่าใกล้เคียง
% และเก็บไว้ใน best_error1

best_neighbor1=S1(index,:); % เก็บค่าพารามิเตอร์ของค่าใกล้เคียงที่ทำให้
% ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุดไว้ใน best_neighbor1

if best_error1<best_error % ปรับค่า best_error
    best_error=best_error1; % ถ้า best_error1 < best_error
% ให้แทนค่า best_error ด้วย best_error1

    best_neighbor1=S1(index,:); % และแทนค่าพารามิเตอร์ best_neighbor ด้วย S1(index,:)
else % ไม่เช่นนั้นแล้ว

```

```

        best_neighbor=S0;          % แทนค่า best_neighbor ด้วย S0
    end
return

```

โปรแกรม rand1.m

```

function x=rand1(a,b)           % โปรแกรม rand1 เป็นโปรแกรมสุ่มค่าพารามิเตอร์ในช่วงที่กำหนด
                                % อินพุตของโปรแกรมคือขอบเขตบนของพารามิเตอร์ (b) และ
                                % ขอบเขตล่างของพารามิเตอร์ (a)
                                % โดยอาศัยความสัมพันธ์  $(b-a)\text{rand} + a$ 
                                % เอาต์พุตของโปรแกรมคือผลของการสุ่มค่าพารามิเตอร์ในช่วง a ถึง b

    x=a+rand*(b-a);
return

```

โปรแกรม random_neigh.m

```

function S1=random_neigh(Number_neighb,radius,xlimit,S0)
                                % โปรแกรม random_neigh เป็นโปรแกรมสุ่มค่าใกล้เคียงรอบค่าตอบ S0
                                % อินพุตของโปรแกรมนี้คือ Number_neighb, radius, xlimit, และ S0
                                % เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ Number_neighb

    for u=1:Number_neighb
        for k=1:size(xlimit,2)
            S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
                                % ทำการสุ่มค่าใกล้เคียงโดยอาศัยความสัมพันธ์
                                %  $S_{1,i} = S_0 + (\text{radius} \cdot \text{rand1}_i(-1,1) \cdot (x_{n,i} - x_{1,i}))$ ,
                                % for  $i = 1, \dots, \text{Number\_neighb}$ 
                                % เรียกใช้โปรแกรม rand1 โดยให้ทำการสุ่มค่าในช่วง -1 ถึง 1
                                % นั่นหมายความว่า การสุ่มมีพื้นที่อยู่ในช่วง  $S_0 \pm (\text{radius} \cdot (x_{n,i} - x_{1,i}))$ 

            while ( S1(u,k)>xlimit(1,k) | S1(u,k)<xlimit(2,k) )
                % ค่าใกล้เคียงที่สุ่มได้จะต้องอยู่ในขอบเขตบนและขอบเขตล่าง
                % ของค่าพารามิเตอร์ ถ้าค่าใกล้เคียงที่สุ่มได้
                % มีค่าเกินขอบเขตบนและขอบเขตล่างที่กำหนด
                % ให้ทำการสุ่มค่าใกล้เคียงใหม่

                S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
            end
        end
    end
return

```

โปรแกรม ATS ที่พัฒนาด้วย C

```
// โปรแกรม ATS ที่พัฒนาด้วย C
// พัฒนาโดยเดชา พวงดาวเรือง (2002)
// สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#define pi 3.14159 // กำหนดค่าคงที่  $\pi = 3.14159$ 
#define para 2 // กำหนดจำนวนพารามิเตอร์ที่ต้องการค้นหาเท่ากับ 2 (เพื่อเป็นตัวอย่าง)
#define row 5 // กำหนดจำนวนการสุ่มค่าตอบเริ่มต้นเท่ากับ 5 ชุด (เพื่อเป็นตัวอย่าง)
#define n_ne 30 // กำหนดจำนวนการสุ่มค่าใกล้เคียงเท่ากับ 30 (เพื่อเป็นตัวอย่าง)
#define tb_list 30 // กำหนดความยาวของ TL (tabu list) เท่ากับ 30 (เพื่อเป็นตัวอย่าง)
float obj(float x1,float x2); // กำหนดฟังก์ชันวัตถุประสงค์ obj มีอินพุต 2 ตัวคือ x1 และ x2 (เพื่อเป็นตัวอย่าง)
void main()
{
/*****
// ประกาศตัวแปรใช้งาน
*****/
clock_t start,end; // ประกาศตัวแปรเพื่อคำนวณเวลาการค้นหา
int st_ini; // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int i,i_rank,i_tb_list,ii,j,ct,ct_0,o,ct_0_ini,i_ini,j_ini,ct_ini,ct_tb,ct_tb_0,count;
// ตัวแปรใช้งานเพื่อการนับค่าและวนรอบ
int st; // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int n; // ตัวนับจำนวนการซ้ำของค่าคำตอบ
int n_back; // ตัวนับจำนวนการใช้กลไกย้อนรอยการค้นหา
int ct_max; // จำนวนรอบการค้นหาสูงสุด
float p[para][2]; // ตัวแปร p คือตัวแปรกำหนดขอบเขตของพารามิเตอร์แต่ละตัว
// มีขนาดเท่ากับจำนวนพารามิเตอร์  $\times 2$  (ขอบบน,ขอบล่าง)
float f_ini,f_min_ini; // f_ini คือค่าฟังก์ชันวัตถุประสงค์เริ่มต้นในแต่ละรอบการค้นหา
// f_min_ini คือค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
float s0_ini[row][para]; // ค่าพารามิเตอร์เริ่มต้นจำนวน  $5 \times 2$  ชุด (เพื่อเป็นตัวอย่าง)
float s; // ตัวแปรชั่วคราวทำหน้าที่รับส่งค่าพารามิเตอร์
float f_min_best_ini[1]; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
float s0_best_ini[1][para]; // ค่าพารามิเตอร์ของ f_min_best_ini
float div,rd,rd_0; // ตัวแปรเกี่ยวกับการสุ่มค่า
// div = 32767 เป็นตัวหารค่าที่สุ่มได้เพื่อทำให้มีค่าสูงสุดเท่ากับ 1
// rd คือค่าที่สุ่มได้จากคำสั่ง rand() มีค่าบวกสูงสุดเท่ากับ 32767
// rd_0 = rd/div คือค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1
float f_tb[tb_list]; // ค่าฟังก์ชันวัตถุประสงค์ที่เก็บใน TL
```

```

float f_tb_rank[tb_list];           // ค่าฟังก์ชันวัตถุประสงค์ที่เรียงลำดับแล้ว
float f_tb1;                       // ค่าฟังก์ชันวัตถุประสงค์เพื่อใช้ในการเปรียบเทียบ
float s_tb[tb_list][para];         // ค่าพารามิเตอร์ของ f_tb
float s_tb_rank[tb_list][para];    // ค่าพารามิเตอร์ของ f_tb_rank
float a,b;                         // ขอบเขตของค่าสุ่ม a = -1 (ขอบล่าง) b = 1 (ขอบบน)
float c;                           // ค่าที่สุ่มได้จากคำสั่ง rand() มีค่าบวกสูงสุดเท่ากับ 32767
float r0;                          // rd_0 = rd/div คือค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1
float r;                           // ค่าสุ่ม r = a+[r0*(b-a)]
float rad;                         // ตัวแปรสุ่มในการค้นหา
float f_min0[1];                   // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุดมีค่าเท่ากับ f_min_best_ini
float s0[1][para];                 // ค่าพารามิเตอร์ของ f_min0 มีค่าเท่ากับ s0_best_ini
float s1[n_ne][para];             // ค่าพารามิเตอร์ของค่าใกล้เคียงในแต่ละรอบการค้นหา
float del[1][para];               // ช่วงของค่าพารามิเตอร์แต่ละตัว
float f_min1_old;                  // ค่าฟังก์ชันวัตถุประสงค์เพื่อการเปรียบเทียบหาค่าที่ดีกว่า
float f_min1[n_ne];                // คำตอบทั้งหมดในแต่ละรอบการค้นหา
float f_min1_best[1];              // คำตอบที่ดีที่สุดในแต่ละรอบการค้นหา
float s1_best[1][para];           // ค่าพารามิเตอร์ของคำตอบที่ดีที่สุดในแต่ละรอบการค้นหา
float f_min0_old,f_min0_new;       // f_min0_old คือคำตอบที่ดีที่สุดในอดีต
                                   // f_min0_new คือคำตอบที่ดีที่สุดในปัจจุบัน

float local_f[1];                  // คำตอบวงแคบเฉพาะถิ่น
float local_ne[1][para];          // ค่าพารามิเตอร์ของ local_f
float global_f[1];                 // คำตอบวงกว้าง
float global_ne[1][para];         // ค่าพารามิเตอร์ของ global_f

/*****
// สุ่มคำตอบเริ่มต้นภายในปริภูมิการค้นหา
*****/

start=clock();                    // เริ่มต้นคำนวณเวลาการค้นหา
ct=0;                             // รีเซต ct = 0
ct_0_ini=0;                       // รีเซต ct_0_ini = 0
count=0;                          // รีเซต count = 0
ct_tb=0;                          // รีเซต ct_tb = 0
ct_tb_0=0;                        // รีเซต ct_tb_0 = 0
i_tb_list=0;                      // รีเซต i_tb_list = 0
f_min_ini=32767;                   // กำหนด f_min_ini = 32767 ไว้เพื่อการเปรียบเทียบ (เพื่อเป็นตัวอย่าง)
div=32767;                        // กำหนด div = 32767
p[0][0]=-1;                       // กำหนดขอบเขตล่างของพารามิเตอร์ x1 = -1 (เพื่อเป็นตัวอย่าง)
p[0][1]=1;                        // กำหนดขอบเขตบนของพารามิเตอร์ x1 = 1 (เพื่อเป็นตัวอย่าง)
p[1][0]=-1;                       // กำหนดขอบเขตล่างของพารามิเตอร์ x2 = -1 (เพื่อเป็นตัวอย่าง)
p[1][1]=1;                        // กำหนดขอบเขตบนของพารามิเตอร์ x2 = 1 (เพื่อเป็นตัวอย่าง)
printf("Enter seed of random no.\nfor finding initial feasible solution = ");
scanf("%d",&st_ini);              // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด

```

```

srand(st_ini); // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
for(i_ini=0;i_ini<row;i_ini++)
{
    j_ini=0; // ตัวชี้ค่าพารามิเตอร์ x1
    rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
    rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
    s=((p[0][1]-p[0][0])*rd_0)+p[0][0]; // สุ่มค่าพารามิเตอร์ x1 จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
    // เก็บไว้ใน s โดยใช้ความสัมพันธ (xh,i - xl,i)rand + xl,i, for i = 1, ..., 5
    s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ x1 ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
    j_ini=1; // ตัวชี้ค่าพารามิเตอร์ x2
    rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
    rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
    s=((p[1][1]-p[1][0])*rd_0)+p[1][0]; // สุ่มค่าพารามิเตอร์ x2 จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
    // เก็บไว้ใน s โดยใช้ความสัมพันธ (xh,i - xl,i)rand + xl,i, for i = 1, ..., 5
    s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ x2 ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
/*****
// ประเมินค่าพารามิเตอร์ที่สุ่มได้โดยฟังก์ชันวัตถุประสงค์ obj
*****/
for(i_ini=0;i_ini<row;i_ini++)
{
    f_ini=obj(s0_ini[i_ini][0],s0_ini[i_ini][1]);
    if(f_ini<=f_min_ini) // เปรียบเทียบ f_ini และ f_min_ini เพื่อหาค่าน้อยที่สุด
    {
        f_min_ini=f_ini; // ถ้า f_ini ≤ f_min_ini แล้วให้ f_min_ini = f_ini
        ct_ini=ct_0_ini; // ปรับตัวนับ ct_ini = ct_0_ini
    }
    ct_0_ini=ct_0_ini+1; // เพิ่มค่า ct_0_ini ขึ้น 1
}
f_min_best_ini[0]=f_min_ini; // ให้ f_min_best_ini = f_min_ini
f_tb[i_tb_list]=f_min_best_ini[0]; // เก็บค่า f_min_best_ini ไว้ใน f_tb
printf("f_min_best_ini[0] = %.4e\n",f_min_best_ini[0]); // พิมพ์ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่น้อยที่สุด
for(j=0;j<para;j++)
{
    s0_best_ini[0][j]=s0_ini[ct_ini][j]; // เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นมีค่าน้อยที่สุด
    s_tb[i_tb_list][j]=s0_ini[ct_ini][j]; // ไว้ใน s0_best_ini และ s_tb
}
i_tb_list=i_tb_list+1; // เพิ่มค่า i_tb_list ขึ้น 1

```



```

/*****
// เริ่มต้นการค้นหาแบบตาข่ายปรับตัว
*****/

n_back=0; // รีเซต n_back = 0
del[0][0]=2; // กำหนดช่วงพารามิเตอร์ x1 (x1 = p[0][1]-p[0][0])
del[0][1]=2; // กำหนดช่วงพารามิเตอร์ x2 (x2 = p[1][1]-p[1][0])
s0[0][0]=s0_best_ini[0][0];
s0[0][1]=s0_best_ini[0][1]; // เก็บ s0_best_ini ไว้ใน s0
f_min0[0]=f_min_best_ini[0]; // เก็บ f_min_best_ini ไว้ใน f_min0
f_min0_old=f_min0[0]; // เก็บ f_min0 ไว้ใน f_min0_old เพื่อเปรียบเทียบ
ct_0=0; // รีเซต ct_0 = 0
rad=0.2; // กำหนดรัศมีการค้นหาเริ่มต้น = 0.2 (เพื่อเป็นตัวอย่าง)
a=-1; // ขอบเขตล่างของค่าสุ่ม a = -1
b=1; // ขอบเขตบนของค่าสุ่ม b = 1
n=0; // รีเซต n = 0
local_f[0]=f_min0[0]; // เก็บ f_min0 ไว้ใน local_f
local_ne[0][0]=s0[0][0];
local_ne[0][1]=s0[0][1]; // เก็บ s0 ไว้ใน local_ne
global_f[0]=local_f[0]; // เก็บ local_f ไว้ใน global_f
global_ne[0][0]=local_ne[0][0];
global_ne[0][1]=local_ne[0][1]; // เก็บ local_ne ไว้ใน global_ne
printf("Enter seed of random no. for ATS = ");
scanf("%d",&st); // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด
srand(st); // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
printf("Enter number of ct_max = ");
scanf("%d",&ct_max); // รับค่าจำนวนรอบการค้นหาสูงสุดผ่านคีย์บอร์ด
x2: for(o=0;o<ct_max;o++)
{
/*****
// กลไกปรับรัศมีการค้นหา (กำหนดให้มี 3 เงื่อนไขเพื่อเป็นตัวอย่าง)
*****/

if(global_f[0]>1e-3)
{
rad=0.2; // เงื่อนไขที่ 1) ถ้า global_f[0] > 1×10-3 แล้ว radius = 0.2
}
else if((global_f[0]>1e-4)&&(global_f[0] <=1e-3))
{
rad=0.15; // เงื่อนไขที่ 2) ถ้า 1×10-4 < global_f ≤ 1×10-3 แล้ว radius = 0.15
}
else if((global_f[0]>1e-5)&&(global_f[0]<=1e-4))
{

```

```

rad=0.1; // เงื่อนไขที่ 3) ถ้า  $1 \times 10^{-5} < \text{global\_f} \leq 1 \times 10^{-4}$  แล้ว radius = 0.1
}

/*****
// คุ่มค่าใกล้เคียงจำนวน n_ne ชุด แล้วประเมินด้วยฟังก์ชันวัตถุประสงค์
*****/

for(i=0;i<n_ne;i++)
{
for(j=0;j<para;j++)
{
c=rand(); // ค่า c จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
r0=c/div; // ค่า r0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
r=a+r0*(b-a); // ค่า r = a+[r0*(b-a)]
s1[i][j]=s0[0][j]+r*rad*del[0][j]; // คำนวณค่าใกล้เคียง
if((s1[i][j]<p[j][0])||(s1[i][j]>p[j][1]))
{
while((s1[i][j]<p[j][0])||(s1[i][j]>p[j][1]))
{
c=rand(); // ค่าใกล้เคียงที่ได้จะต้องอยู่ในขอบเขตบนและขอบเขตล่าง
r0=c/div; // ของค่าพารามิเตอร์ ถ้าค่าใกล้เคียงที่สุ่มได้มีค่าเกินขอบเขตบนและขอบเขตล่าง
r=a+r0*(b-a); // ที่กำหนด ให้ทำการสุ่มค่าใกล้เคียงใหม่
s1[i][j]=s0[0][j]+r*rad*del[0][j];
}
}
}
}

/*****
// ประเมินค่าพารามิเตอร์ของค่าใกล้เคียงที่สุ่มได้โดยฟังก์ชันวัตถุประสงค์ obj
*****/

for(i=0;i<n_ne;i++)
{
f_min1[i]=obj(s1[i][0],s1[i][1]);
}

// เก็บค่าฟังก์ชันวัตถุประสงค์จากค่าใกล้เคียงแต่ละค่าไว้ใน f_min1
f_min1_old=f_min1[0]; // เก็บ f_min1 ตัวแรก (f_min1[0]) ไว้ใน f_min1_old เพื่อเปรียบเทียบ
for(i=0;i<n_ne;i++)
{
if(f_min1[i]<f_min1_old) // ถ้า f_min1 < f_min1_old
{
f_min1_old=f_min1[i]; // ให้ f_min1_old = f_min1
ct=ct_0; // ปรับค่า ct = ct_0
}
}
ct_0=ct_0+1; // เพิ่มค่า ct_0 ขึ้น 1

```



```

/*****
// ปรับค่าค่าตอบและพารามิเตอร์
/*****

if(f_min1_best[0]<f_min0[0])      // ถ้า f_min1_best < f_min0
{
    f_min0[0]=f_min1_best[0];    // เก็บ f_min1_best ไว้ใน f_min0
    for(j=0;j<para;j++)
    {
        s0[0][j]=s1_best[0][j];  // เก็บค่าพารามิเตอร์ s1_best ไว้ใน s0
    }
}

/*****
// ตรวจสอบจำนวนรอบการค้นหา
/*****

if(count>=ct_max)              // ถ้า count ≥ ct_max
{
    goto x3;                    // กระโดดไปทำงานที่ลาเบล x3
}
count=count+1 ;                // เพิ่ม count ขึ้น 1
}

/*****
// เปรียบเทียบค่าตอบวงแคบเฉพาะถิ่นและค่าตอบวงกว้าง
/*****

x1: if(local_f[0]<global_f[0])   // ถ้า local_f < global_f
{
    global_f[0]=local_f[0];     // เก็บ local_f ไว้ใน global_f
    for(j=0;j<para;j++)
    {
        global_ne[0][j]=local_ne[0][j];
    }                            // เก็บค่าพารามิเตอร์ local_ne ไว้ใน global_ne
    printf("%d\t%.1e\t%.4e\n",count,rad,global_f[0]);
}                                // พิมพ์ค่า count, rad, และ global_f
if(global_f[0]<1e-6)            // ตรวจสอบเกณฑ์ยุติการค้นหา เมื่อ global_f < 1×10-6
{
    goto x3;                    // กระโดดไปทำงานที่ลาเบล x3
}

/*****
// จัดเรียงลำดับ (ranking) ค่าตอบที่บรรจุใน TL
/*****

i_rank=0;                      // รีเซ็ตค่า i_rank
for(ii=0;ii<i_tb_list;ii++)
{

```

```

f_tb1=f_tb[0]; // เก็บค่า f_tb ตัวแรก (f_tb[0]) ไว้ใน f_tb1 เพื่อเปรียบเทียบ
for(i=0;i<i_tb_list;i++)
{
    if(f_tb[i]<f_tb1) // ถ้า f_tb < f_tb1
    {
        f_tb1=f_tb[i]; // เก็บ f_tb ไว้ใน f_tb1
        ct_tb=ct_tb_0; // ให้ ct_tb = ct_tb_0
    }
    ct_tb_0=ct_tb_0+1; // เพิ่มค่า ct_tb_0 ขึ้น 1
}
f_tb[ct_tb]=1e7; // แทนค่า f_tb ที่พบว่ามีความน้อยกว่า f_tb1 ด้วยค่ามาก ๆ
// (1×107 เพื่อเป็นตัวอย่าง) ทั้งนี้เพื่อให้การหาค่าน้อยที่สุดในรอบต่อไป
// จะไม่นำเอาค่า f_tb ที่พบแล้วมาพิจารณาซ้ำอีก
f_tb_rank[i_rank]=f_tb1; // เก็บ f_tb1 ไว้ใน f_tb_rank
// เมื่อโปรแกรมในส่วนนี้สิ้นสุดลง ค่าที่ถูเก็บใน f_tb_rank
// จะเรียงจากน้อยสุดไปมากที่สุด

for(j=0;j<para;j++)
{
    s_tb_rank[i_rank][j]=s_tb[ct_tb][j];
} // เก็บค่าพารามิเตอร์ของ f_tb_rank ไว้ใน s_tb_rank
ct_tb=0; // รีเซ็ต ct_tb = 0
ct_tb_0=0; // รีเซ็ต ct_tb_0 = 0
i_rank=i_rank+1; // เพิ่มค่า i_rank ขึ้น 1
}

/*****
// แทนค่าตอบเริ่มต้นใหม่จากคำตอบที่เรียงลำดับแล้ว
*****/

i_tb_list=0; // รีเซ็ต i_tb_list = 0
f_min0[0]=f_tb_rank[4]; // แทนค่า f_min0 ด้วยค่าลำดับที่ 5 ใน f_tb_rank (f_tb_rank[4])
for(j=0;j<para;j++)
{
    s0[0][j]=s_tb_rank[4][j]; // แทนค่าพารามิเตอร์ s0 ด้วยค่าพารามิเตอร์ชุดที่ 5 ใน s_tb_rank
}
f_min0_old=f_min0[0]; // แทน f_min0_old ด้วย f_min0
n=0; // รีเซ็ต n = 0
count=count+1; // เพิ่มค่า count ขึ้น 1
goto x2; // กระโดดไปทำงานที่ลาเบล x2

/*****
// แสดงผลการค้นหา
*****/

x3: end=clock(); // ยุติการค้นหา และยุติการคำนวณเวลาการค้นหา
printf("global_f = %.4e\n",global_f[0]); // พิมพ์คำตอบวงกว้าง global_f

```

```

printf("Execution time = %f second\n", (end-start)/CLK_TCK);
// พิมพ์เวลาที่ทั้งหมดที่ใช้ในการค้นหา

printf("x1 = %.4e\nx2 = %.4e\n", global_ne[0][0], global_ne[0][1]);
// พิมพ์ค่าพารามิเตอร์ของ global_f

printf("End of program"); // พิมพ์โปรแกรมสิ้นสุด

getch();
}

/*****
// ฟังก์ชันวัตถุประสงค obj กำหนดให้เป็นฟังก์ชัน โบฮาเซฟสกี (เพื่อเป็นตัวอย่าง) โดยมีตัวแปรอินพุต 2 ค่าคือ x1 และ x2
*****/

float obj(float x1, float x2)
{
    float output; // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค
    output = (x1*x1) + 2*(x2*x2) - (0.3*cos(3*pi*x1)) - (0.4*cos(4*pi*x2)) + 0.7;
    return(output);
}

/*****/

```

ภาคผนวก ข

แบบจำลองทางคณิตศาสตร์ของระบบ CPP

แบบจำลองทางคณิตศาสตร์ของระบบ CPP

ระบบลูกตุ้มนาฬิกาพวงกรวด หรือระบบ CPP ดังแสดงในรูปที่ 6.1 เป็นระบบที่ประกอบด้วยลูกตุ้ม (pendulum) ที่ต่อเข้ากับรถ (cart) ลูกตุ้มจะแกว่งไกวอยู่ในระนาบ x, y อย่างอิสระตามการเคลื่อนที่ของรถที่ถูกขับด้วยแรง f (นิวตัน) แรงดังกล่าวกำเนิดขึ้นจากการเคลื่อนที่ของมอเตอร์และส่งผ่านตามสายพาน โดยที่มุม ϕ คือ มุมการแกว่งไกวของลูกตุ้ม (เรเดียน) u คือ สัญญาณอินพุตของมอเตอร์ (โวลต์) y คือ ตำแหน่งของรถ (เมตร) M คือ มวลของรถ (กิโลกรัม) m คือ มวลของลูกตุ้ม (กิโลกรัม) l คือ ความยาวของก้านลูกตุ้ม (เมตร) และ g คือ ความเร่งเนื่องจากแรงโน้มถ่วง (เมตร/วินาที²)

การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ CPP อาจดำเนินการได้หลายแนวทาง เช่น อาศัยกฎการเคลื่อนที่ของนิวตัน หรือสมการการเคลื่อนที่ของลากรานจ์ (Lagrange's equation of motion) เป็นต้น สำหรับแบบจำลองทางคณิตศาสตร์ของระบบ CPP ในงานวิจัยนี้ แบ่งออกเป็นสองรูปแบบคือ แบบจำลองไม่เชิงเส้น และแบบจำลองเชิงเส้น ดังรายละเอียดต่อไปนี้

แบบจำลองระบบ CPP ไม่เชิงเส้น

การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ CPP ไม่เชิงเส้นในงานวิจัยนี้ จะอาศัยสมการการเคลื่อนที่ของลากรานจ์ ซึ่งฟังก์ชันลากรานจ์ถูกกำหนดให้เท่ากับผลต่างระหว่างพลังงานจลน์ (kinetic energy) และพลังงานศักย์ (potential energy) ของระบบ (Cannon, 1967; Bloch, Leonard, and Marsden, 2000) ดังแสดงในสมการที่ (ข-1) โดยกำหนดให้ก้านของลูกตุ้มปราศจากน้ำหนัก และแรง f จะเป็นฟังก์ชันของสัญญาณอินพุต u

$$L(\phi, y, \dot{\phi}, \dot{y}) \triangleq K(\phi, y, \dot{\phi}, \dot{y}) - V(\phi) \quad (\text{ข-1})$$

เมื่อ $L(\cdot)$ คือ ฟังก์ชันลากรานจ์ $K(\cdot)$ คือ สมการพลังงานจลน์ $V(\cdot)$ คือ สมการพลังงานศักย์ $\dot{\phi}$ คือ $\frac{d\phi}{dt}$ และ \dot{y} คือ $\frac{dy}{dt}$

สมการพลังงานจลน์ $K(\cdot)$ จะเป็นผลรวมของพลังงานจลน์ของรถ (K_{cart}) และพลังงานจลน์ของลูกตุ้ม (K_{pend}) ดังแสดงในสมการที่ (ข-2)

$$K(\phi, y, \dot{\phi}, \dot{y}) = K_{cart} + K_{pend}$$

$$K(\phi, y, \dot{\phi}, \dot{y}) = \frac{1}{2}M(v_{cart})^2 + \frac{1}{2}m(v_{pend})^2 \quad (\text{ข-2})$$

เมื่อ $v_{cart} = \dot{y}$ ขณะที่ $v_{pend} = [v_{pend_horizontal}, v_{pend_vertical}]$ ซึ่งจะต้องพิจารณาดังนี้คือ เมื่อกำหนดให้ลูกตุ้มแกว่งไกวอยู่ด้านล่างของรถ $v_{pend} = [\dot{y} + l \cos(\phi)\dot{\phi}, l \sin(\phi)\dot{\phi}]$ และ เมื่อกำหนดให้ลูกตุ้มแกว่งไกวอยู่ด้านบนของรถ $v_{pend} = [\dot{y} + l \cos(\phi)\dot{\phi}, -l \sin(\phi)\dot{\phi}]$

ดังนั้นสมการที่ (ข-2) สามารถเขียนได้ใหม่เป็นดังสมการที่ (ข-3) สมการดังกล่าวเป็นสมการพลังงานจลน์ของระบบ ที่สามารถใช้แทนได้ทั้งในกรณีที่ลูกตุ้มแกว่งไกวอยู่ด้านล่างและด้านบนของรถ

$$K(\phi, y, \dot{\phi}, \dot{y}) = \frac{1}{2}M(\dot{y})^2 + \frac{1}{2}m[(\dot{y})^2 + 2l\dot{\phi}\dot{y}\cos(\phi) + l^2(\dot{\phi})^2]$$

$$K(\phi, y, \dot{\phi}, \dot{y}) = \frac{1}{2} \begin{bmatrix} \dot{\phi} & \dot{y} \end{bmatrix} \begin{bmatrix} ml^2 & ml \cos(\phi) \\ ml \cos(\phi) & M + m \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{y} \end{bmatrix} \quad (\text{ข-3})$$

สำหรับสมการพลังงานศักย์ $V(\cdot)$ จะเป็นผลรวมของพลังงานศักย์ของรถ และพลังงานศักย์ของลูกตุ้ม แต่เนื่องจากพลังงานศักย์ของรถมีค่าเท่ากับศูนย์ (รถอยู่ในระดับอ้างอิง) ดังนั้นสมการพลังงานศักย์ $V(\cdot)$ จะพิจารณาจากพลังงานศักย์ของลูกตุ้มเพียงอย่างเดียว ในกรณีที่ลูกตุ้มแกว่งไกวอยู่ด้านล่างของรถ พลังงานศักย์จะแสดงดังสมการที่ (ข-4) และในกรณีที่ลูกตุ้มแกว่งไกวอยู่ด้านบนของรถ พลังงานศักย์จะแสดงดังสมการที่ (ข-5) ตามลำดับ

$$V(\phi) = -mgl \cos(\phi) \quad (\text{ข-4})$$

$$V(\phi) = mgl \cos(\phi) \quad (\text{ข-5})$$

กำหนดให้ $\alpha = ml^2$, $\beta = ml$, $\gamma = (M + m)$, และ $D = mgl$ เมื่อลูกตุ้มแกว่งไกวอยู่ด้านล่างของรถ และ $D = -mgl$ เมื่อลูกตุ้มแกว่งไกวอยู่ด้านบนของรถ ฟังก์ชันลากรางจ์ $L(\cdot)$ จะเขียนได้ดังสมการที่ (ข-6)

$$L(\phi, y, \dot{\phi}, \dot{y}) = \frac{1}{2} \left[\alpha(\dot{\phi})^2 + 2\beta \cos(\phi)\dot{\phi}\dot{y} + \gamma(\dot{y})^2 \right] + D \cos(\phi) \quad (\text{ข-6})$$

กำหนดให้ p_ϕ คือโมเมนตัมเชิงมุม และ p_y คือโมเมนตัมเชิงเส้น ซึ่งแสดงในสมการที่ (ข-7) และ (ข-8) ตามลำดับ

$$p_\phi = \frac{\partial L}{\partial \dot{\phi}} = \alpha \dot{\phi} + \beta \cos(\phi) \dot{y} \quad (\text{ข-7})$$

$$p_y = \frac{\partial L}{\partial \dot{y}} = \gamma \dot{y} + \beta \cos(\phi) \dot{\phi} \quad (\text{ข-8})$$

สมการการเคลื่อนที่ของลากรางจ์สำหรับระบบ CPP แสดงดังสมการที่ (ข-9) และ (ข-10)

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}} - \frac{\partial L}{\partial \phi} = 0 \quad (\text{ข-9})$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{y}} = f \quad (\text{ข-10})$$

จากสมการที่ (ข-9) และ (ข-10) สามารถเขียนได้ใหม่เป็นดังสมการที่ (ข-11) และ (ข-12) ตามลำดับ

$$\alpha \ddot{\phi} + \beta \cos(\phi) \ddot{y} + D \sin(\phi) = 0 \quad (\text{ข-11})$$

$$\gamma \ddot{y} + \beta \cos(\phi) \ddot{\phi} - \beta (\dot{\phi})^2 \sin(\phi) = f \quad (\text{ข-12})$$

เมื่อแทนค่า $\alpha, \beta, \gamma,$ และ D ลงในสมการที่ (ข-11) และ (ข-12) จะได้แบบจำลองทางคณิตศาสตร์ของระบบ CPP ในรูปของสมการอนุพันธ์ไม่เชิงเส้นดังนี้

เมื่อพิจารณาให้ลูกตุ้มแกว่งไกวอยู่ด้านล่างของรถ แบบจำลองทางคณิตศาสตร์ของระบบ CPP ไม่เชิงเส้น แสดงดังสมการที่ (ข-13) และ (ข-14) ตามลำดับ ซึ่งก็คือสมการที่ (7-27) และ (7-28) ในบทที่ 7 นั่นเอง

$$\ddot{\phi} = \frac{f \cos(\phi) + 0.5ml \sin(2\phi) (\dot{\phi})^2 + (M+m)g \sin(\phi)}{l [m \cos^2(\phi) - (M+m)]} \quad (\text{ข-13})$$

$$\ddot{y} = \frac{f + 0.5mg \sin(2\phi) + ml \sin(\phi)(\dot{\phi})^2}{[(M + m) - m \cos^2(\phi)]} \quad (\text{ข-14})$$

เมื่อพิจารณาให้ลูกตุ้มแกว่งไกวอยู่ด้านบนบนของรถ แบบจำลองทางคณิตศาสตร์ของระบบ CPP ไม่เชิงเส้น แสดงดังสมการที่ (ข-15) และ (ข-16) ตามลำดับ

$$\ddot{\phi} = \frac{f \cos(\phi) + 0.5ml \sin(2\phi)(\dot{\phi})^2 - (M + m)g \sin(\phi)}{l[m \cos^2(\phi) - (M + m)]} \quad (\text{ข-15})$$

$$\ddot{y} = \frac{f - 0.5mg \sin(2\phi) + ml \sin(\phi)(\dot{\phi})^2}{[(M + m) - m \cos^2(\phi)]} \quad (\text{ข-16})$$

แบบจำลองระบบ CPP เชิงเส้น

การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ CPP เชิงเส้น จะตั้งต้นพิจารณาจากแบบจำลองทางคณิตศาสตร์ของระบบ CPP ไม่เชิงเส้น ดังแสดงในสมการที่ (ข-11) และ (ข-12) อาศัยหลักการทำให้เป็นเชิงเส้น (linearization) โดยพิจารณาให้ลูกตุ้มแกว่งไกวด้วยมุม ϕ เล็ก ๆ ซึ่งจะสามารถประมาณพจน์ $\cos(\phi) \approx 1$ และพจน์ $\sin(\phi) \approx \phi$ ส่วนพจน์ $\dot{\phi}$ และ \ddot{y} ก็จะมีค่าน้อยมาก ดังนั้นเทอมยกกำลังสองของ $\dot{\phi}$ และ \ddot{y} สามารถประมาณให้เป็นศูนย์ได้ ดังนั้นสมการที่ (ข-11) และ (ข-12) สามารถเขียนได้เป็นดังสมการที่ (ข-17) และ (ข-18) ตามลำดับ

$$\alpha \ddot{\phi} + \beta \ddot{y} + D\phi = 0 \quad (\text{ข-17})$$

$$\gamma \ddot{y} + \beta \dot{\phi} = f \quad (\text{ข-18})$$

เมื่อแทนค่า $\alpha, \beta, \gamma,$ และ D ลงในสมการที่ (ข-17) และ (ข-18) จะได้แบบจำลองทางคณิตศาสตร์ของระบบ CPP ในรูปของสมการอนุพันธ์เชิงเส้นดังนี้

เมื่อพิจารณาให้ลูกตุ้มแกว่งไกวอยู่ด้านล่างของรถ แบบจำลองทางคณิตศาสตร์ของระบบ CPP เชิงเส้น แสดงดังสมการที่ (ข-19) และ (ข-20) ตามลำดับ ซึ่งก็คือสมการที่ (7-1) และ (7-2) ในบทที่ 7 นั่นเอง

$$\ddot{\phi} + \left[\frac{(M+m)g}{Ml} \right] \phi + \frac{f}{Ml} = 0 \quad (\text{ข-19})$$

$$\ddot{y} - \left[\frac{mg}{M} \right] \phi - \frac{f}{M} = 0 \quad (\text{ข-20})$$

เมื่อพิจารณาให้ลูกตุ้มแกว่งไกวอยู่ด้านบนบนของรถ แบบจำลองทางคณิตศาสตร์ของระบบ CPP เชิงเส้น แสดงดังสมการที่ (ข-21) และ (ข-22) ตามลำดับ

$$\ddot{\phi} - \left[\frac{(M+m)g}{Ml} \right] \phi + \frac{f}{Ml} = 0 \quad (\text{ข-21})$$

$$\ddot{y} + \left[\frac{mg}{M} \right] \phi - \frac{f}{M} = 0 \quad (\text{ข-22})$$

ภาคผนวก ค

โปรแกรมประมวลผลภาพระบบ CPP

โปรแกรมประมวลผลภาพระบบ CPP

โปรแกรมประมวลผลภาพระบบ CPP พัฒนาขึ้นโดยใช้โปรแกรม MATLAB™ แบ่งออกเป็นสองส่วนด้วยกันคือ โปรแกรมแยกแยะลำดับภาพจากไฟล์วิดีโอ และโปรแกรมสกัดข้อมูลพลวัต ดังรายละเอียดต่อไปนี้

โปรแกรมแยกแยะลำดับภาพจากไฟล์วิดีโอของระบบ CPP

```
% โปรแกรมแยกแยะลำดับภาพจากไฟล์วิดีโอที่พัฒนาด้วย MATLAB™
% พัฒนาโดยเฉชา พวงดาวเรือง (2002)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

clear all

n_frame = 2500; % กำหนด n_frame เท่ากับจำนวนภาพในไฟล์วิดีโอเท่ากับ 2,500 ภาพ
start = 1000; % กำหนดชื่อไฟล์ภาพเริ่มต้นเป็น 1000

for i = 1:n_frame
    mov = aviread('cpp01.avi',i); % อ่านไฟล์วิดีโอ cpp01.avi เก็บไว้ในตัวแปร mov
    % ซึ่งในงานวิจัยนี้มีไฟล์วิดีโอของระบบ CPP จำนวน 3 ไฟล์คือ
    % cpp01.avi, cpp02.avi, และ cpp03.avi ตามลำดับ

    start1 = start+i; % กำหนดให้ตัวแปรชื่อไฟล์ start1 = start + i
    str = num2str(start1); % แปลงค่าตัวแปร start1 ให้เป็นตัวเลขสตริง
    str = strcat(str,'.bmp'); % ผสมตัวแปรชื่อไฟล์เข้ากับ .bmp ("ชื่อไฟล์" + ".bmp" = "ชื่อไฟล์.bmp")
    imwrite(mov.cdata,str,'bmp'); % เขียนภาพชนิด bmp ในตัวแปร mov ลงในไฟล์ภาพ "ชื่อไฟล์.bmp"
end
```

โปรแกรมสกัดข้อมูลพลวัตของระบบ CPP

```
% โปรแกรมสกัดข้อมูลพลวัตของระบบ CPP จากลำดับภาพที่พัฒนาด้วย MATLAB™
% พัฒนาโดยเฉชา พวงดาวเรือง (2002)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

clear all

n_file = 2500; % กำหนด n_frame เท่ากับจำนวนภาพในไฟล์วิดีโอเท่ากับ 2,500 ภาพ
start = 1000; % กำหนดชื่อไฟล์ภาพเริ่มต้นเป็น 1000
r_ref = 158; % กำหนดตำแหน่งแถวอ้างอิง = 158 โดยคำนวณจากตำแหน่งรอยขณะหยุดนิ่ง
c_ref = 366; % กำหนดตำแหน่งคอลัมน์อ้างอิง = 366 โดยคำนวณจากตำแหน่งรอยขณะหยุดนิ่ง
for ii = 168:n_file
    start1 = start+ii; % กำหนดให้ตัวแปรชื่อไฟล์ start1 = start + ii
    str = num2str(start1); % แปลงค่าตัวแปร start1 ให้เป็นตัวเลขสตริง
```

```

str = strcat(str,'.bmp'); % ผูกตัวแปรชื่อไฟล์เข้ากับ .bmp (“ชื่อไฟล์” + “.bmp” = “ชื่อไฟล์.bmp”)
a = imread(str); % อ่านไฟล์ภาพ “ชื่อไฟล์.bmp” เก็บไว้ในตัวแปร a

% Start segmentation
[i,j,k] = size(a); % คำนี i, j, และ k คือขนาดของภาพ a
ij_y1 = zeros(0,2); % รีเซตตัวแปรที่ใช้เก็บข้อมูลตำแหน่งจุดภาพของรถ ij_y1
y1 = 1; % เซต y1 = 1
y2 = 1; % เซต y2 = 1

% Segment the cart position
for o = 130:180 % ช่วงแถวที่ 130 ถึง 180 คือช่วงแถวที่รถจะเคลื่อนที่ผ่าน
    for n = 1:j
        for m = 1:k
            x(m) = a(o,n,m); % เก็บค่าสี R, G, และ B ของแต่ละจุดภาพไว้ใน x(1), x(2), และ x(3)
        end
        % กำหนดจุดเปลี่ยนค่าของรถ
        if x(1)>=215&x(1)<=248 % ช่วงสี R ของรถคือ 215 ≤ R ≤ 248
            if x(2)>=193&x(2)<=220 % ช่วงสี G ของรถคือ 193 ≤ G ≤ 220
                if x(3)>=68&x(3)<=125 % ช่วงสี B ของรถคือ 68 ≤ B ≤ 125
                    ij_y1(y1,y2) = o; % เก็บค่าแถวของตำแหน่งจุดภาพของรถที่มีค่าสีอยู่ในช่วงของจุดเปลี่ยนค่าไว้ที่ ij_y1
                    ij_y1(y1,y2+1) = n;
                    % เก็บค่าคอลัมน์ของตำแหน่งจุดภาพของรถที่มีค่าสี
                    % อยู่ในช่วงของจุดเปลี่ยนค่าไว้ที่ ij_y1
                    y1 = y1+1; % เพิ่มค่า y1 ขึ้น 1
                    y2 = 1; % เซต y2 = 1
                end
            end
        end
        for p = 1:k
            x(p) = 0; % รีเซตค่าสี R, G, และ B ใน x(1), x(2), และ x(3)
        end
    end
end
end

max_1 = max(ij_y1,[],1); % หาค่าแถวและคอลัมน์ของรถที่มีค่ามากที่สุดเก็บไว้ใน max_1
min_1 = min(ij_y1,[],1); % หาค่าแถวและคอลัมน์ของรถที่มีค่าน้อยที่สุดเก็บไว้ใน min_1
r_1 = ((max_1(1,1)-min_1(1,1))/2+min_1(1,1));
% คำนวณตำแหน่งแถวของรถจากความสัมพันธ์
%  $\frac{\text{row}_{\max} - \text{row}_{\min}}{2} + \text{row}_{\min}$  เก็บไว้ใน r_1

c_1 = ((max_1(1,2)-min_1(1,2))/2+min_1(1,2));
% คำนวณตำแหน่งคอลัมน์ของรถจากความสัมพันธ์
%  $\frac{\text{column}_{\max} - \text{column}_{\min}}{2} + \text{column}_{\min}$  เก็บไว้ใน c_1

CG_r_cart = round(r_1); % CG_r_cart คือตำแหน่งแถวของรถที่ปิดศยแล้ว

```

```

CG_c_cart = round(c_1); % CG_c_cart คือตำแหน่งคอลัมน์ของรถที่ปิดเศษแล้ว
% Calculate Cart position in 'mm'
Cart_pos_pix = CG_c_cart - c_ref; % จำนวนตำแหน่งคอลัมน์ของรถที่ห่างจากตำแหน่งคอลัมน์อ้างอิง
Cart_pos_mm(ii) = Cart_pos_pix*2.1285;
% จำนวนตำแหน่งคอลัมน์ของรถในหน่วยมิลลิเมตร
% จากความสัมพันธ์ 459 จุดภาพ = 977 มิลลิเมตร
% ดังนั้น 1 จุดภาพ = 2.1285 มิลลิเมตร แล้วเก็บไว้ใน Cart_pos_mm

% Segment the pendulum angle
ij_y2 = zeros(0,2); % รีเซตตัวแปรที่ใช้เก็บข้อมูลตำแหน่งจุดภาพของลูกตุ้ม ij_y2
y1 = 1; % เซต y1 = 1
y2 = 1; % เซต y2 = 1
for o = 300:408 % ช่วงแถวที่ 300 ถึง 408 คือช่วงแถวที่ลูกตุ้มจะเคลื่อนที่ผ่าน
    for n = 1:j
        for m = 1:k
            x(m) = a(o,n,m); % เก็บค่าสี R, G, และ B ของแต่ละจุดภาพไว้ใน x(1), x(2), และ x(3)
        end
        % กำหนดจุดเปลี่ยนค่าของลูกตุ้ม
        if x(1)>=215&x(1)<=248 % ช่วงสี R ของลูกตุ้มคือ 215 ≤ R ≤ 248
            if x(2)>=193&x(2)<=220 % ช่วงสี G ของลูกตุ้มคือ 193 ≤ G ≤ 220
                if x(3)>=68&x(3)<=125 % ช่วงสี B ของลูกตุ้มคือ 68 ≤ B ≤ 125
                    ij_y2(y1,y2) = o;% เก็บค่าแถวของตำแหน่งจุดภาพของลูกตุ้มที่มีค่าสี
                    % อยู่ในช่วงของจุดเปลี่ยนค่าไว้ที่ ij_y2
                    ij_y2(y1,y2+1) = n;
                    % เก็บค่าคอลัมน์ของตำแหน่งจุดภาพของลูกตุ้ม
                    % ที่มีค่าสีอยู่ในช่วงของจุดเปลี่ยนค่าไว้ที่ ij_y2
                    y1 = y1+1; % เพิ่มค่า y1 ขึ้น 1
                    y2 = 1; % เซต y2 = 1
                end
            end
        end
        for p = 1:k
            x(p) = 0; % รีเซตค่าสี R, G, และ B ใน x(1), x(2), และ x(3)
        end
    end
end
end
end
max_2 = max(ij_y2,[],1); % หาค่าแถวและคอลัมน์ของลูกตุ้มที่มีค่ามากที่สุดเก็บไว้ใน max_2
min_2 = min(ij_y2,[],1); % หาค่าแถวและคอลัมน์ของลูกตุ้มที่มีค่าน้อยที่สุดเก็บไว้ใน min_2
r_2 = ((max_2(1,1)-min_2(1,1))/2+min_2(1,1));
% จำนวนตำแหน่งแถวของลูกตุ้มจากความสัมพันธ์
%  $\frac{\text{row}_{\max} - \text{row}_{\min}}{2} + \text{row}_{\min}$  เก็บไว้ใน r_2
c_2 = ((max_2(1,2)-min_2(1,2))/2+min_2(1,2));

```



```

% คำนวณตำแหน่งคอลัมน์ของลูกตุ้มจากความสัมพันธ์
%  $\frac{\text{column}_{\max} - \text{column}_{\min}}{2} + \text{column}_{\min}$  เก็บไว้ใน c_2
CG_r_Pen_ang = round(r_2); % CG_r_Pen_ang คือตำแหน่งแถวของลูกตุ้มที่ปิดเศษแล้ว
CG_c_Pen_ang = round(c_2); % CG_c_Pen_ang คือตำแหน่งคอลัมน์ของลูกตุ้มที่ปิดเศษแล้ว
% Calculate Pendulum angle in 'rad'
del_x = CG_c_cart - CG_c_Pen_ang; % จำนวนระยะห่างระหว่างตำแหน่งคอลัมน์ของรถและ
% ตำแหน่งคอลัมน์ของลูกตุ้ม เก็บไว้ใน del_x
del_y = CG_r_Pen_ang - CG_r_cart; % จำนวนระยะห่างระหว่างตำแหน่งแถวของลูกตุ้มกับตำแหน่งแถวของรถ
% เก็บไว้ใน del_y
theta_rad(ii) = atan(del_x/del_y); % คำนวณมุมของลูกตุ้มในหน่วยเรเดียนจากความสัมพันธ์
%  $\tan^{-1} \frac{\text{del}_x}{\text{del}_y}$  เก็บไว้ใน theta_rad
theta_deg(ii) = theta_rad(ii)*180/pi; % คำนวณมุมของลูกตุ้มในหน่วยองศาจากความสัมพันธ์
%  $\frac{\text{theta\_rad} \times 180}{\pi}$  เก็บไว้ใน theta_deg
end

```

ภาคผนวก ง

โปรแกรมระบุเอกลักษณ์ระบบ **CPP**

โปรแกรมระบุเอกลักษณ์ระบบ CPP

โปรแกรมระบุเอกลักษณ์ระบบ CPP แบ่งออกเป็นสองส่วนคือ โปรแกรมระบุเอกลักษณ์ระบบ CPP เชิงเส้น ซึ่งเป็นการระบุเอกลักษณ์ด้วยวิธีการดั้งเดิม ได้รับการพัฒนาขึ้นโดยใช้โปรแกรม MATLAB™ และโปรแกรมระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น โดยอาศัยการค้นหาแบบตามเชิงปรับตัว ซึ่งได้รับการพัฒนาขึ้นโดยใช้โปรแกรม C ดังรายละเอียดต่อไปนี้

โปรแกรมระบุเอกลักษณ์ระบบ CPP เชิงเส้น

```
% โปรแกรมนี้เป็นโปรแกรมระบุเอกลักษณ์ระบบ CPP เชิงเส้น
% เป็นการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม
% เพื่อพิจารณาอันดับที่เหมาะสมของแบบจำลอง BJ จากอันดับ 2 ถึง 20 ที่พัฒนาด้วย MATLAB™
% พัฒนาโดยเคชา พวงดาวเรือง (2003)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

load DATA_PENDULUM_AFTER_CORRECTION_TAKE_1
% ทำการโหลดข้อมูลพลวัตของระบบ CPP จากไฟล์
% DATA_PENDULUM_AFTER_CORRECTION_TAKE_1
% ซึ่งบรรจุข้อมูลจากการทดสอบตอนที่ 1 ดังต่อไปนี้
% U_Sensor คือข้อมูลอินพุต, AP_Sensor คือตำแหน่งเชิงมุมของลูกตุ้ม
% ที่ได้จากตัวตรวจรู้, AP_Image คือตำแหน่งเชิงมุมของลูกตุ้มที่ได้จาก
% การประมวลผลภาพ, และ AP_New คือตำแหน่งเชิงมุมของลูกตุ้ม
% จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
t = 0:0.15:2099*0.15; % กำหนดเวลา t ให้สอดคล้องกับจำนวนข้อมูล 2,100 ข้อมูล
% มีช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที

AP_Iden=[AP_Sensor(1:700);AP_Sensor(1:700);AP_Sensor(1:700)];
% ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 700 ข้อมูล
% ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลมีลักษณะเป็นรายคาบที่ยาวนาน
% แล้วเก็บไว้ใน AP_Iden เพื่อใช้ในการระบุเอกลักษณ์

U_Iden=[U_Sensor(1:700);U_Sensor(1:700);U_Sensor(1:700)];
% ข้อมูลอินพุตจำนวน 700 ข้อมูล ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูล
% มีลักษณะเป็นรายคาบที่ยาวนาน แล้วเก็บไว้ใน U_Iden เพื่อใช้ในการระบุเอกลักษณ์

AP_Comp=[AP_Sensor(1:700);AP_Sensor(1:700);AP_Sensor(1:700)];
% ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 700 ข้อมูล
% ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลมีลักษณะเป็นรายคาบที่ยาวนาน
% เก็บไว้ใน AP_Comp เพื่อใช้ในการเปรียบเทียบ

AP_Valid=AP_Sensor(701:900); % ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 200 ข้อมูล
```

```

% เก็บไว้ใน AP_Valid เพื่อใช้ในการตรวจสอบความถูกต้องของแบบจำลอง
U_Valid=U_Sensor(701:900); % ข้อมูลอินพุตจำนวน 200 ข้อมูล ถูกเก็บไว้ใน U_Valid เพื่อใช้ในการ
% ตรวจสอบความถูกต้องของแบบจำลอง
z=[AP_Iden U_Iden]; % เก็บข้อมูล AP_Iden และ U_Iden ไว้ใน z เพื่อรับการระบุเอกลักษณ์
th=bj(z,[2,1,2,2,1]); % ให้แบบจำลอง BJ มีอันดับ 2 ถึง 20
% กรณีแบบจำลองอันดับ 2 กำหนดให้ z,[2,1,2,2,1]
% กรณีแบบจำลองอันดับ 3 กำหนดให้ z,[3,2,2,3,1]
% กรณีแบบจำลองอันดับ 4 กำหนดให้ z,[4,4,1,4,1]
% กรณีแบบจำลองอันดับ 5 กำหนดให้ z,[5,1,1,5,1]
% กรณีแบบจำลองอันดับ 6 กำหนดให้ z,[6,1,1,6,1]
% กรณีแบบจำลองอันดับ 7 กำหนดให้ z,[7,4,1,7,1]
% กรณีแบบจำลองอันดับ 8 กำหนดให้ z,[8,2,1,8,1]
% กรณีแบบจำลองอันดับ 9 กำหนดให้ z,[9,5,1,9,1]
% กรณีแบบจำลองอันดับ 10 กำหนดให้ z,[10,9,1,10,1]
% กรณีแบบจำลองอันดับ 11 กำหนดให้ z,[11,3,1,11,1]
% กรณีแบบจำลองอันดับ 12 กำหนดให้ z,[12,1,1,12,1]
% กรณีแบบจำลองอันดับ 13 กำหนดให้ z,[13,1,1,13,1]
% กรณีแบบจำลองอันดับ 14 กำหนดให้ z,[14,1,1,14,1]
% กรณีแบบจำลองอันดับ 15 กำหนดให้ z,[15,2,1,15,1]
% กรณีแบบจำลองอันดับ 16 กำหนดให้ z,[16,4,1,16,1]
% กรณีแบบจำลองอันดับ 17 กำหนดให้ z,[17,2,1,17,1]
% กรณีแบบจำลองอันดับ 18 กำหนดให้ z,[18,3,1,18,1]
% กรณีแบบจำลองอันดับ 19 กำหนดให้ z,[19,2,1,19,1]
% กรณีแบบจำลองอันดับ 20 กำหนดให้ z,[20,3,1,20,1]
% ค่าพารามิเตอร์ของแบบจำลอง BJ บรรจุอยู่ในตัวแปร th
th=setf(th,0.15); % กำหนดช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
[numd,dend]=th2tf(th); % แปลงแบบจำลอง BJ ให้อยู่ในรูปฟังก์ชันถ่ายโอนโดเมน z
[numc,denc]=d2cm(numd,dend,0.15,'tustin'); % แปลงฟังก์ชันถ่ายโอนโดเมน z เป็นฟังก์ชันถ่ายโอนโดเมน s
% โดยใช้การแปลงแบบ Tustin
sys=tf(numc,denc); % แสดงแบบจำลองระบบ CPP ในรูปฟังก์ชันถ่ายโอนโดเมน s
y=lsim(sys,U_Iden,t); % ทำการจำลองผลแบบจำลองด้วยข้อมูลอินพุต U_Iden
% แล้วเก็บผลการจำลองไว้ใน y
% identification
figure(1)
plot(t,AP_Iden,'k'); % แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Iden ใน figure 1
hold on
plot(t,y,'b'); % แสดงการจำลองผลแบบจำลองระบบ CPP ด้วยข้อมูลใน y
% เพื่อเปรียบเทียบกับผลการทดสอบใน figure 1
abs_er(i)=sum(abs(y-AP_Comp)) % จำนวนค่าผลรวมของค่าความคลาดเคลื่อนสัมบูรณ์
ss_er(i)=sum((y-AP_Comp).^2) % จำนวนค่าผลรวมค่าความคลาดเคลื่อนกำลังสอง
ms_er(i)=ss_er(i)/length(y) % จำนวนค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง

```

```

rms_er(i)=(ms_er(i)).^0.5           % จำนวนค่ารากที่สองของค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง
% validation
figure(2)
t1=0:0.15:199*0.15;               % กำหนดเวลา t1 ให้สอดคล้องกับจำนวนข้อมูล 200 ข้อมูล
                                   % มีช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
y1=lsim(sys,U_Valid,t1);           % ทำการจำลองผลเพื่อตรวจสอบแบบจำลองที่ได้
                                   % ด้วยข้อมูลอินพุต U_Valid แล้วเก็บผลการจำลองไว้ใน y1
plot(t1,AP_Valid,'k');             % แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Valid ใน figure 2
hold on
plot(t1,y1,'b');                   % แสดงผลการตรวจสอบแบบจำลองระบบ CPP ด้วยข้อมูลใน y1
                                   % เพื่อเปรียบเทียบกับผลการทดสอบใน figure 2

% โปรแกรมนี้เป็นโปรแกรมระบุเอกลักษณ์ระบบ CPP เชิงเส้น ที่พัฒนาด้วย MATLAB™
% เป็นการระบุเอกลักษณ์ระบบด้วยวิธีการดั้งเดิม
% โดยใช้แบบจำลอง BJ อันดับ 10 อาศัยข้อมูลของระบบที่ได้จากตัวตรวจรู้ จากการประมวลผลภาพ
% และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
% พัฒนาโดยเคซา พวงดาวเรือง (2003)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

clear all
% ##### Sensory data #####
load DATA_PENDULUM_AFTER_CORRECTION_TAKE_1
                                   % ทำการโหลดข้อมูลพลวัตของระบบ CPP จากไฟล์
                                   % DATA_PENDULUM_AFTER_CORRECTION_TAKE_1
                                   % ซึ่งบรรจุข้อมูลจากการทดสอบตอนที่ 1 ดังต่อไปนี้
                                   % U_Sensor คือข้อมูลอินพุต, AP_Sensor คือตำแหน่งเชิงมุมของลูกตุ้ม
                                   % ที่ได้จากตัวตรวจรู้, AP_Image คือตำแหน่งเชิงมุมของลูกตุ้มที่ได้จาก
                                   % การประมวลผลภาพ, และ AP_New คือตำแหน่งเชิงมุมของลูกตุ้ม
                                   % จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
t = 0:0.15:2099*0.15;             % กำหนดเวลา t ให้สอดคล้องกับจำนวนข้อมูล 2100 ข้อมูล
                                   % มีช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที

AP_IdeN=[AP_Sensor(1:700);AP_Sensor(1:700);AP_Sensor(1:700)];
                                   % ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 700 ข้อมูล
                                   % ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลมีลักษณะเป็นรายคาบที่ยาวนาน
                                   % แล้วเก็บไว้ใน AP_IdeN เพื่อใช้ในการระบุเอกลักษณ์

U_IdeN=[U_Sensor(1:700);U_Sensor(1:700);U_Sensor(1:700)];
                                   % ข้อมูลอินพุตจำนวน 700 ข้อมูล ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูล
                                   % มีลักษณะเป็นรายคาบที่ยาวนาน แล้วเก็บไว้ใน U_IdeN เพื่อใช้ในการระบุเอกลักษณ์

AP_Comp=[AP_Sensor(1:700);AP_Sensor(1:700);AP_Sensor(1:700)];
                                   % ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 700 ข้อมูล
                                   % ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลมีลักษณะเป็นรายคาบที่ยาวนาน
                                   % เก็บไว้ใน AP_Comp เพื่อใช้ในการเปรียบเทียบ

AP_Valid=AP_Sensor(701:900);      % ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 200 ข้อมูล

```

```

% เก็บไว้ใน AP_Valid เพื่อใช้ในการตรวจสอบความถูกต้องของแบบจำลอง
U_Valid=U_Sensor(701:900);
% ข้อมูลอินพุตจำนวน 200 ข้อมูล ถูกเก็บไว้ใน U_Valid เพื่อใช้ในการ
% ตรวจสอบความถูกต้องของแบบจำลอง

z=[AP_Iden U_Iden];
% เก็บข้อมูล AP_Iden และ U_Iden ไว้ใน z เพื่อรับการระบุเอกลักษณ์
th=bj(z,[10,9,1,10,1]);
% กำหนดแบบจำลอง BJ อันดับ 10
% ค่าพารามิเตอร์ของแบบจำลอง BJ ในรูปของแบบจำลองกล่องดำ
% เชิงเส้นชนิดเวลาเต็มหน่วย บรรจุอยู่ในตัวแปร th

th=sett(th,0.15);
% กำหนดช่วงเวลาซีกตัวอย่างเท่ากับ 0.15 วินาที
[numd,dend]=th2tf(th);
% แปลงแบบจำลอง BJ ให้อยู่ในรูปฟังก์ชันถ่ายโอนโดเมน z
[numc,denc]=d2cm(numd,dend,0.15,'tustin');
% แปลงฟังก์ชันถ่ายโอนโดเมน z เป็นฟังก์ชันถ่ายโอนโดเมน s
% โดยใช้การแปลงแบบ Tustin

sys=tf(numc,denc);
% แสดงแบบจำลองระบบ CPP ในรูปฟังก์ชันถ่ายโอนโดเมน s
poles=pole(sys)
% แสดงค่าโพลของแบบจำลอง
zeros=zero(sys)
% แสดงค่าซีโรของแบบจำลอง
zpk(sys)
% แสดงแบบจำลองในรูปของตัวประกอบซีโร โพล และอัตราขยาย
zplane(zeros,poles)
% พล็อตตำแหน่งโพล และซีโร
y=lsim(sys,U_Iden,t);
% ทำการจำลองผลแบบจำลองที่ได้ด้วยข้อมูลอินพุต U_Iden
% แล้วเก็บผลการจำลองไว้ใน y

% identification
figure(1)
plot(t,AP_Iden,'k');
hold on
plot(t,y,'b');
% แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Iden ใน figure 1
% แสดงการจำลองผลแบบจำลองระบบ CPP ด้วยข้อมูลใน y
% เพื่อเปรียบเทียบกับผลการทดสอบใน figure 1
% คำนวณค่าผลรวมของค่าความคลาดเคลื่อนสัมบูรณ์
abs_er(i)=sum(abs(y-AP_Comp))
% คำนวณค่าผลรวมค่าความคลาดเคลื่อนกำลังสอง
ss_er(i)=sum((y-AP_Comp).^2)
ms_er(i)=ss_er(i)/length(y)
% คำนวณค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง
rms_er(i)=(ms_er(i)).^0.5
% ของผลรวมค่าความคลาดเคลื่อนกำลังสอง

% validation
figure(2)
t1=0:0.15:199*0.15;
% กำหนดเวลา t1 ให้สอดคล้องกับจำนวนข้อมูล 200 ข้อมูล
% มีช่วงเวลาซีกตัวอย่างเท่ากับ 0.15 วินาที
y1=lsim(sys,U_Valid,t1);
% ทำการจำลองผลเพื่อตรวจสอบแบบจำลองที่ได้
% ด้วยข้อมูลอินพุต U_Valid แล้วเก็บผลการจำลองไว้ใน y1
plot(t1,AP_Valid,'k');
hold on
plot(t1,y1,'b');
% แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Valid ใน figure 2
% แสดงผลการตรวจสอบแบบจำลองระบบ CPP ด้วยข้อมูลใน y1
% เพื่อเปรียบเทียบกับผลการทดสอบใน figure 2

figure(3)
bode(numc,denc)
hold on
pause
% แสดงการจำลองระบบ CPP ในโดเมนความถี่

```

```

% ##### Image data #####
clear all
load DATA_PENDULUM_AFTER_CORRECTION_TAKE_1

% ทำการโหลดข้อมูลพลวัตของระบบ CPP จากไฟล์
% DATA_PENDULUM_AFTER_CORRECTION_TAKE_1
% ซึ่งบรรจุข้อมูลจากการทดสอบตอนที่ 1 ดังต่อไปนี้
% U_Sensor คือข้อมูลอินพุต, AP_Sensor คือตำแหน่งเชิงมุมของลูกตุ้ม
% ที่ได้จากตัวตรวจรู้, AP_Image คือตำแหน่งเชิงมุมของลูกตุ้มที่ได้จาก
% การประมวลผลภาพ, และ AP_New คือตำแหน่งเชิงมุมของลูกตุ้ม
% จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
t = 0:0.15:2099*0.15; % กำหนดเวลา t ให้สอดคล้องกับจำนวนข้อมูล 2100 ข้อมูล
% มีช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที

AP_Iden=[AP_Image(1:700);AP_Image(1:700);AP_Image(1:700)];
% ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากการประมวลผลภาพจำนวน 700
% ข้อมูล ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลมีลักษณะเป็นรายการที่
% ยาวนานแล้วเก็บไว้ใน AP_Iden เพื่อใช้ในการระบุเอกลักษณ์

U_Iden=[U_Sensor(1:700);U_Sensor(1:700);U_Sensor(1:700)];
% ข้อมูลอินพุตจำนวน 700 ข้อมูล ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูล
% มีลักษณะเป็นรายการที่ยาวนาน แล้วเก็บไว้ใน U_Iden เพื่อใช้ในการระบุเอกลักษณ์

AP_Comp=[AP_Sensor(1:700);AP_Sensor(1:700);AP_Sensor(1:700)];
% ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 700 ข้อมูล
% ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลมีลักษณะเป็นรายการที่ยาวนาน
% เก็บไว้ใน AP_Comp เพื่อใช้ในการเปรียบเทียบ

AP_Valid=AP_Image(701:900); % ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากการประมวลผลภาพจำนวน 200
% ข้อมูล เก็บไว้ใน AP_Valid เพื่อใช้ในการตรวจสอบความถูกต้องของแบบจำลอง

U_Valid=U_Sensor(701:900); % ข้อมูลอินพุตจำนวน 200 ข้อมูล ถูกเก็บไว้ใน U_Valid เพื่อใช้ในการ
% ตรวจสอบความถูกต้องของแบบจำลอง

z=[AP_Iden U_Iden]; % เก็บข้อมูล AP_Iden และ U_Iden ไว้ใน z เพื่อรับการระบุเอกลักษณ์
th=bj(z,[10,8,1,10,1]); % กำหนดแบบจำลอง BJ อันดับ 10
% ค่าพารามิเตอร์ของแบบจำลอง BJ ในรูปของแบบจำลองกล่องดำ
% เชิงเส้นชนิดเวลาเต็มหน่วย บรรจุอยู่ในตัวแปร th
th=sett(th,0.15); % กำหนดช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
[numd,dend]=th2tf(th); % แปลงแบบจำลอง BJ ให้อยู่ในรูปฟังก์ชันถ่ายโอน โดเมน z
[numc,denc]=d2cm(numd,dend,0.15,'tustin');
% แปลงฟังก์ชันถ่ายโอน โดเมน z เป็นฟังก์ชันถ่ายโอน โดเมน s
% โดยใช้การแปลงแบบ Tustin

sys=tf(numc,denc); % แสดงแบบจำลองระบบ CPP ในรูปฟังก์ชันถ่ายโอน โดเมน s
poles=pole(sys) % แสดงค่าโพลของแบบจำลอง
zeros=zero(sys) % แสดงค่าซีโรของแบบจำลอง
zpk(sys) % แสดงแบบจำลองในรูปของตัวประกอบซีโร โพล และอัตราขยาย
zplane(zeros,poles) % พล็อตตำแหน่งโพล และซีโร
y=lsim(sys,U_Iden,t); % ทำการจำลองผลแบบจำลองที่ได้ด้วยข้อมูลอินพุต U_Iden

```

```

% แล้วเก็บผลการจำลองไว้ใน y

% identification
figure(1)
plot(t,AP_Iden,'k'); % แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Iden ใน figure 1
hold on
plot(t,y,'b'); % แสดงผลการจำลองผลแบบจำลองระบบ CPP ด้วยข้อมูลใน y
% เพื่อเปรียบเทียบกับผลการทดสอบใน figure 1

abs_er(i)=sum(abs(y-AP_Comp)) % จำนวนค่าผลรวมของค่าความคลาดเคลื่อนสัมบูรณ์
ss_er(i)=sum((y-AP_Comp).^2) % จำนวนค่าผลรวมค่าความคลาดเคลื่อนกำลังสอง
ms_er(i)=ss_er(i)/length(y) % จำนวนค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง
rms_er(i)=(ms_er(i)).^0.5 % จำนวนค่ารากที่สองของค่าเฉลี่ย
% ของผลรวมค่าความคลาดเคลื่อนกำลังสอง

% validation
figure(2)
t1=0:0.15:199*0.15; % กำหนดเวลา t1 ให้สอดคล้องกับจำนวนข้อมูล 200 ข้อมูล
% มีช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
y1=lsim(sys,U_Valid,t1); % ทำการจำลองผลเพื่อตรวจสอบแบบจำลองที่ได้
% ด้วยข้อมูลอินพุต U_Valid แล้วเก็บผลการจำลองไว้ใน y1
plot(t1,AP_Valid,'k'); % แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Valid ใน figure 2
hold on
plot(t1,y1,'b'); % แสดงผลการตรวจสอบแบบจำลองระบบ CPP ด้วยข้อมูลใน y1
% เพื่อเปรียบเทียบกับผลการทดสอบใน figure 2

figure(3)
bode(numc,denc) % แสดงการจำลองระบบ CPP ใน โดเมนความถี่
hold on
pause
% ##### Image data after correction #####
clear all
load DATA_PENDULUM_AFTER_CORRECTION_TAKE_1
% ทำการโหลดข้อมูลพลวัตของระบบ CPP จากไฟล์
% DATA_PENDULUM_AFTER_CORRECTION_TAKE_1
% ซึ่งบรรจุข้อมูลจากการทดสอบตอนที่ 1 ดังต่อไปนี้
% U_Sensor คือข้อมูลอินพุต, AP_Sensor คือตำแหน่งเชิงมุมของลูกตุ้ม
% ที่ได้จากตัวตรวจรู้, AP_Image คือตำแหน่งเชิงมุมของลูกตุ้มที่ได้จาก
% การประมวลผลภาพ, และ AP_New คือตำแหน่งเชิงมุมของลูกตุ้ม
% จากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
t = 0:0.15:2099*0.15; % กำหนดเวลา t ให้สอดคล้องกับจำนวนข้อมูล 2100 ข้อมูล
% มีช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที

AP_Iden=[AP_New(1:700);AP_New(1:700);AP_New(1:700)];
% ตำแหน่งเชิงมุมของลูกตุ้มจากการประมวลผลภาพที่ได้รับการแก้ไข
% ความคลาดเคลื่อน จำนวน 700 ข้อมูล ถูกนำมาเรียงต่อกัน 3 ชุด
% เพื่อให้ข้อมูลมีลักษณะเป็นรายคาบที่ยาวนาน
% แล้วเก็บไว้ใน AP_Iden เพื่อใช้ในการระบุเอกลักษณ์

U_Iden=[U_Sensor(1:700);U_Sensor(1:700);U_Sensor(1:700)];

```



```

% ข้อมูลอินพุตจำนวน 700 ข้อมูล ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูล
% มีลักษณะเป็นรายการที่ยาวนาน แล้วเก็บไว้ใน U_Iden เพื่อใช้ในการระบุเอกลักษณ์
AP_Comp=[AP_Sensor(1:700);AP_Sensor(1:700);AP_Sensor(1:700)];
% ตำแหน่งเชิงมุมของลูกตุ้มที่ได้จากตัวตรวจรู้จำนวน 700 ข้อมูล
% ถูกนำมาเรียงต่อกัน 3 ชุด เพื่อให้ข้อมูลมีลักษณะเป็นรายการที่ยาวนาน
% เก็บไว้ใน AP_Comp เพื่อใช้ในการเปรียบเทียบ
AP_Valid=AP_New(701:900);
% ตำแหน่งเชิงมุมของลูกตุ้มจากการประมวลผลภาพที่ได้รับการแก้ไข
% ความคลาดเคลื่อน จำนวน 200 ข้อมูล เก็บไว้ใน AP_Valid
% เพื่อใช้ในการตรวจสอบความถูกต้องของแบบจำลอง
U_Valid=U_Sensor(701:900);
% ข้อมูลอินพุตจำนวน 200 ข้อมูล ถูกเก็บไว้ใน U_Valid เพื่อใช้ในการ
% ตรวจสอบความถูกต้องของแบบจำลอง
z=[AP_Iden U_Iden];
% เก็บข้อมูล AP_Iden และ U_Iden ไว้ใน z เพื่อรับการระบุเอกลักษณ์
th=bj(z,[10,5,1,10,1]);
% กำหนดแบบจำลอง BJ อันดับ 10
% ค่าพารามิเตอร์ของแบบจำลอง BJ ในรูปของแบบจำลองกล่องดำ
% เชิงเส้นชนิดเวลาเต็มหน่วย บรรจุอยู่ในตัวแปร th
th=sett(th,0.15);
% กำหนดช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
[numd,denc]=th2tf(th);
% แปลงแบบจำลอง BJ ให้อยู่ในรูปฟังก์ชันถ่ายโอน โดเมน z
[numc,denc]=d2cm(numd,denc,0.15,'tustin');
% แปลงฟังก์ชันถ่ายโอน โดเมน z เป็นฟังก์ชันถ่ายโอน โดเมน s
% โดยใช้การแปลงแบบ Tustin
sys=tf(numc,denc);
% แสดงแบบจำลองระบบ CPP ในรูปฟังก์ชันถ่ายโอน โดเมน s
poles=pole(sys)
% แสดงค่าโพลของแบบจำลอง
zeros=zero(sys)
% แสดงค่าซีโรของแบบจำลอง
zpk(sys)
% แสดงแบบจำลองในรูปของตัวประกอบซีโร โพล และอัตราขยาย
zplane(zeros,poles)
% พล็อตตำแหน่งโพล และซีโร
y=lsim(sys,U_Iden,t);
% ทำการจำลองผลแบบจำลองที่ได้ด้วยข้อมูลอินพุต U_Iden
% แล้วเก็บผลการจำลองไว้ใน y

% identification
figure(1)
plot(t,AP_Iden,'k');
hold on
plot(t,y,'b');
% แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Iden ใน figure 1
% แสดงการจำลองผลแบบจำลองระบบ CPP ด้วยข้อมูลใน y
% เพื่อเปรียบเทียบกับผลการทดสอบใน figure 1
abs_er(i)=sum(abs(y-AP_Comp))
% จำนวนค่าผลรวมของค่าความคลาดเคลื่อนสัมบูรณ์
ss_er(i)=sum((y-AP_Comp).^2)
% จำนวนค่าผลรวมค่าความคลาดเคลื่อนกำลังสอง
ms_er(i)=ss_er(i)/length(y)
% จำนวนค่าเฉลี่ยของผลรวมค่าความคลาดเคลื่อนกำลังสอง
rms_er(i)=(ms_er(i)).^0.5
% จำนวนค่ารากที่สองของค่าเฉลี่ย
% ของผลรวมค่าความคลาดเคลื่อนกำลังสอง

% validation
figure(2)
t1=0:0.15:199*0.15;
% กำหนดเวลา t1 ให้สอดคล้องกับจำนวนข้อมูล 200 ข้อมูล
% มีช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
y1=lsim(sys,U_Valid,t1);
% ทำการจำลองผลเพื่อตรวจสอบแบบจำลองที่ได้

```

```

% ด้วยข้อมูลอินพุต U_Valid แล้วเก็บผลการจำลองไว้ใน y1
plot(t1,AP_Valid,'k'); % แสดงผลการทดสอบระบบ CPP ด้วยข้อมูลใน AP_Valid ใน figure 2
hold on
plot(t1,y1,'b'); % แสดงผลการตรวจสอบแบบจำลองระบบ CPP ด้วยข้อมูลใน y1
% เพื่อเปรียบเทียบกับผลการทดสอบใน figure 2

figure(3)
bode(numc,denc) % แสดงการจำลองระบบ CPP ในโดเมนความถี่

```

โปรแกรมระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น

```

// โปรแกรมนี้เป็น โปรแกรมระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น
// เป็นการระบุเอกลักษณ์ระบบด้วยการค้นหาแบบความเชิงปรับตัว
// เพื่อพิจารณาอันดับที่เหมาะสมของแรง  $f$  ที่เป็นพหุนามของ  $u$  จากอันดับ 2 ถึง 7
// โปรแกรมได้รับการพัฒนาด้วยภาษา C เพื่อการระบุเอกลักษณ์ระบบโดยใช้ข้อมูลที่ได้จากตัวตรวจรู้
// พัฒนาโดยเดชา พวงดาวเรือง (2004)
// สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#define para 9 // กำหนดจำนวนพารามิเตอร์ที่ต้องการค้นหาเท่ากับ 9
#define row 5 // กำหนดจำนวนการสุ่มค่าตอบเริ่มต้นเท่ากับ 5 ชุด
#define n_ne 30 // กำหนดจำนวนการสุ่มค่าใกล้เคียงเท่ากับ 30
float obj1(float x2); // กำหนดฟังก์ชันวัตถุประสงค์ obj1
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj2
float obj3(float x4); // กำหนดฟังก์ชันวัตถุประสงค์ obj3
float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj4
float model(float s0_ini_0,float s0_ini_1,float s0_ini_2,float s0_ini_3,float s0_ini_4,float s0_ini_5,float s0_ini_6,float s0_ini_7,float s0_ini_8);
// กำหนดฟังก์ชันวัตถุประสงค์ model
void main()
{
/*****
// ประกาศตัวแปรใช้งาน
*****/
clock_t start,end; // ประกาศตัวแปรเพื่อคำนวณเวลาการค้นหา
int st_ini; // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int i,i_rank,i_tb_list,ii,j,ct,ct_0,o,ct_0_ini,i_ini,j_ini,ct_ini,ct_tb,ct_tb_0,count;
// ตัวแปรใช้งานเพื่อการนับค่าและวนรอบ

```

```

int st; // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int n; // ตัวนับจำนวนการซ้ำของค่าคำตอบ
int n_back; // ตัวนับจำนวนการใช้กลไกย้อนรอยการค้นหา
int ct_max; // จำนวนรอบการค้นหาสูงสุด
float p[para][2]; // ตัวแปร p คือตัวแปรกำหนดขอบเขตของพารามิเตอร์แต่ละตัว
// มีขนาดเท่ากับจำนวนพารามิเตอร์ × 2 (ขอบบน, ขอบล่าง)
float f_ini, f_min_ini; // f_ini คือค่าฟังก์ชันวัตถุประสงค์เริ่มต้นในแต่ละรอบการค้นหา
// f_min_ini คือค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
float s0_ini[row][para]; // ค่าพารามิเตอร์เริ่มต้นจำนวน 5 × 2 ชุด (เพื่อเป็นตัวอย่าง)
float s; // ตัวแปรชั่วคราวทำหน้าที่รับส่งค่าพารามิเตอร์
float f_min_best_ini[1]; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
float s0_best_ini[1][para]; // ค่าพารามิเตอร์ของ f_min_best_ini
float div, rd, rd_0; // ตัวแปรเกี่ยวกับการสุ่มค่า
// div = 32767 เป็นตัวหารค่าที่สุ่มได้เพื่อทำให้มีค่าสูงสุดเท่ากับ 1
// rd คือค่าที่สุ่มได้จากคำสั่ง rand() มีค่าบวกสูงสุดเท่ากับ 32767
// rd_0 = rd/div คือค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1
float f_tb[tb_list]; // ค่าฟังก์ชันวัตถุประสงค์ที่เก็บใน TL
float f_tb_rank[tb_list]; // ค่าฟังก์ชันวัตถุประสงค์ที่เรียงลำดับแล้ว
float f_tb1; // ค่าฟังก์ชันวัตถุประสงค์เพื่อใช้ในการเปรียบเทียบ
float s_tb[tb_list][para]; // ค่าพารามิเตอร์ของ f_tb
float s_tb_rank[tb_list][para]; // ค่าพารามิเตอร์ของ f_tb_rank
float a, b; // ขอบเขตของค่าสุ่ม a = -1 (ขอบล่าง) b = 1 (ขอบบน)
float c; // ค่าที่สุ่มได้จากคำสั่ง rand() มีค่าบวกสูงสุดเท่ากับ 32767
float rd0; // rd_0 = rd/div คือค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1
float r; // ค่าสุ่ม r = a+[r0×(b-a)]
float rad; // ตัวแปรรัศมีการค้นหา
float f_min0[1]; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุดมีค่าเท่ากับ f_min_best_ini
float s0[1][para]; // ค่าพารามิเตอร์ของ f_min0 มีค่าเท่ากับ s0_best_ini
float s1[n_ne][para]; // ค่าพารามิเตอร์ของค่าใกล้เคียงในแต่ละรอบการค้นหา
float del[1][para]; // ช่วงของค่าพารามิเตอร์แต่ละตัว
float f_min1_old; // ค่าฟังก์ชันวัตถุประสงค์เพื่อการเปรียบเทียบหาค่าที่ดีกว่า
float f_min1[n_ne]; // ค่าตอบทั้งหมดในแต่ละรอบการค้นหา
float f_min1_best[1]; // ค่าตอบที่ดีที่สุดในแต่ละรอบการค้นหา
float s1_best[1][para]; // ค่าพารามิเตอร์ของค่าตอบที่ดีที่สุดในแต่ละรอบการค้นหา
float f_min0_old, f_min0_new; // f_min0_old คือค่าตอบที่ดีที่สุดในอดีต
// f_min0_new คือค่าตอบที่ดีที่สุดในปัจจุบัน
float local_f[1]; // ค่าตอบวงแคบเฉพาะถิ่น
float local_ne[1][para]; // ค่าพารามิเตอร์ของ local_f
float global_f[1]; // ค่าตอบวงกว้าง
float global_ne[1][para]; // ค่าพารามิเตอร์ของ global_f

```

```

/*****
// คุ่มคำตอบเริ่มต้นภายในปริภูมิการค้นหา
/*****

start=clock(); // เริ่มต้นคำนวณเวลาการค้นหา
ct=0; // รีเซต ct = 0
ct_0_ini=0; // รีเซต ct_0_ini = 0
count=0; // รีเซต count = 0
ct_tb=0; // รีเซต ct_tb = 0
ct_tb_0=0; // รีเซต ct_tb_0 = 0
i_tb_list=0; // รีเซต i_tb_list = 0
f_min_ini=32767; // กำหนด f_min_ini = 32767 ไว้เพื่อการเปรียบเทียบ (เพื่อเป็นตัวอย่าง)
div=32767; // กำหนด div = 32767
p[0][0]=0.5; //M // กำหนดขอบเขตล่างของพารามิเตอร์ M = 0.5 (พิจารณาเป็น  $x_1$ )
p[0][1]=1.5; // กำหนดขอบเขตบนของพารามิเตอร์ M = 1.5 (พิจารณาเป็น  $x_n$ )
p[1][0]=4; //A // กำหนดขอบเขตล่างของพารามิเตอร์ A = 4.0 (พิจารณาเป็น  $x_1$ )
p[1][1]=5; // กำหนดขอบเขตบนของพารามิเตอร์ A = 5.0 (พิจารณาเป็น  $x_n$ )
p[2][0]=6; //B // กำหนดขอบเขตล่างของพารามิเตอร์ B = 6.0 (พิจารณาเป็น  $x_1$ )
p[2][1]=7; // กำหนดขอบเขตบนของพารามิเตอร์ B = 7.0 (พิจารณาเป็น  $x_n$ )
p[3][0]=-2; //C // กำหนดขอบเขตล่างของพารามิเตอร์ C = -2.0 (พิจารณาเป็น  $x_1$ )
p[3][1]=-1; // กำหนดขอบเขตบนของพารามิเตอร์ C = -1.0 (พิจารณาเป็น  $x_n$ )
p[4][0]=-6; //D // กำหนดขอบเขตล่างของพารามิเตอร์ D = -6.0 (พิจารณาเป็น  $x_1$ )
p[4][1]=-5; // กำหนดขอบเขตบนของพารามิเตอร์ D = -5.0 (พิจารณาเป็น  $x_n$ )
p[5][0]=-5; //E // กำหนดขอบเขตล่างของพารามิเตอร์ E = -5.0 (พิจารณาเป็น  $x_1$ )
p[5][1]=5; // กำหนดขอบเขตบนของพารามิเตอร์ E = 5.0 (พิจารณาเป็น  $x_n$ )
p[6][0]=-5; //F // กำหนดขอบเขตล่างของพารามิเตอร์ F = -5.0 (พิจารณาเป็น  $x_1$ )
p[6][1]=5; // กำหนดขอบเขตบนของพารามิเตอร์ F = 5.0 (พิจารณาเป็น  $x_n$ )
p[7][0]=-5; //G // กำหนดขอบเขตล่างของพารามิเตอร์ G = -5.0 (พิจารณาเป็น  $x_1$ )
p[7][1]=5; // กำหนดขอบเขตบนของพารามิเตอร์ G = 5.0 (พิจารณาเป็น  $x_n$ )
p[8][0]=-1; //H // กำหนดขอบเขตล่างของพารามิเตอร์ H = -1.0 (พิจารณาเป็น  $x_1$ )
p[8][1]=1; // กำหนดขอบเขตบนของพารามิเตอร์ H = 1.0 (พิจารณาเป็น  $x_n$ )
// ในการหาค่าอันดับที่เหมาะสมของแรง  $f$  ที่เป็นพหุนามของ  $u$ 
// จากอันดับ 2 ถึง 7 ค่าพารามิเตอร์ A, B,...,H จะถูกกำหนดให้สอดคล้อง
// กับอันดับที่พิจารณา ยกตัวอย่างเช่น เมื่อพิจารณาให้แรง  $f$ 
// เป็นพหุนามอันดับ 2 ของ  $u$  ค่าพารามิเตอร์ A, B,...,E จะถูกกำหนดให้
// มีค่าขอบเขตล่างและบนเท่ากับ 0 เป็นต้น

printf("Enter seed of random no.\nfor finding initial feasible solution = ");
scanf("%d",&st_ini); // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด
srand(st_ini); // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
for(i_ini=0;i_ini<row;i_ini++)
{
    j_ini=0; // ตัวชี้ค่าพารามิเตอร์ M

```

```

rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[0][1]-p[0][0])*rd_0)+p[0][0];
// สุ่มค่าพารามิเตอร์ M จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ M ที่สุ่มได้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=1; // ตัวชี้ค่าพารามิเตอร์ A
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[1][1]-p[1][0])*rd_0)+p[1][0]; // สุ่มค่าพารามิเตอร์ A จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ A ที่สุ่มได้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=2; // ตัวชี้ค่าพารามิเตอร์ B
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[2][1]-p[2][0])*rd_0)+p[2][0]; // สุ่มค่าพารามิเตอร์ B จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ B ที่สุ่มได้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=3; // ตัวชี้ค่าพารามิเตอร์ C
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[3][1]-p[3][0])*rd_0)+p[3][0]; // สุ่มค่าพารามิเตอร์ C จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ C ที่สุ่มได้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=4; // ตัวชี้ค่าพารามิเตอร์ D
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[4][1]-p[4][0])*rd_0)+p[4][0]; // สุ่มค่าพารามิเตอร์ D จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 

```

```

s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ D ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=5; // ตัวชี้ค่าพารามิเตอร์ E
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[5][1]-p[5][0])*rd_0)+p[5][0]; // สุ่มค่าพารามิเตอร์ E จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ E ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=6; // ตัวชี้ค่าพารามิเตอร์ F
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[6][1]-p[6][0])*rd_0)+p[6][0]; // สุ่มค่าพารามิเตอร์ F จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ F ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=7; // ตัวชี้ค่าพารามิเตอร์ G
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[7][1]-p[7][0])*rd_0)+p[7][0]; // สุ่มค่าพารามิเตอร์ G จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ G ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=8; // ตัวชี้ค่าพารามิเตอร์ H
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[8][1]-p[8][0])*rd_0)+p[8][0]; // สุ่มค่าพารามิเตอร์ H จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ H ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
}
/*****/
// ประเมินค่าพารามิเตอร์ที่สุ่มได้โดยฟังก์ชันวัตถุประสงค์ obj
/*****/

```

```

for(i_ini=0;i_ini<row;i_ini++)
{
    f_ini=model(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]);
    if(f_ini<=f_min_ini) // เปรียบเทียบ f_ini และ f_min_ini เพื่อหาค่าน้อยที่สุด
    {
        f_min_ini=f_ini; // ถ้า f_ini ≤ f_min_ini แล้วให้ f_min_ini = f_ini
        ct_ini=ct_0_ini; // ปรับตัวนับ ct_ini = ct_0_ini
    }
    ct_0_ini=ct_0_ini+1; // เพิ่มค่า ct_0_ini ขึ้น 1
}
f_min_best_ini[0]=f_min_ini; // ให้ f_min_best_ini = f_min_ini
f_tb[i_tb_list]=f_min_best_ini[0]; // เก็บค่า f_min_best_ini ไว้ใน f_tb
printf("f_min_best_ini[0] = %.4e\n",f_min_best_ini[0]);
// พิมพ์ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่น้อยที่สุด

for(j=0;j<para;j++)
{
    s0_best_ini[0][j]=s0_ini[ct_ini][j];
    s_tb[i_tb_list][j]=s0_ini[ct_ini][j];
} // เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นมีค่าน้อยที่สุด
// ไว้ใน s0_best_ini และ s_tb
i_tb_list=i_tb_list+1; // เพิ่มค่า i_tb_list ขึ้น 1
/*****
// เริ่มต้นการค้นหาแบบตามเชิงปรับตัว
*****/

n_back=0; // รีเซต n_back = 0
del[0][0]=1.0; // กำหนดช่วงพารามิเตอร์ M (p[0][1]-p[0][0])
del[0][1]=1.0; // กำหนดช่วงพารามิเตอร์ A (p[1][1]-p[1][0])
del[0][2]=1.0; // กำหนดช่วงพารามิเตอร์ B (p[2][1]-p[2][0])
del[0][3]=1.0; // กำหนดช่วงพารามิเตอร์ C (p[3][1]-p[3][0])
del[0][4]=1.0; // กำหนดช่วงพารามิเตอร์ D (p[4][1]-p[4][0])
del[0][5]=10.0; // กำหนดช่วงพารามิเตอร์ E (p[5][1]-p[5][0])
del[0][6]=10.0; // กำหนดช่วงพารามิเตอร์ F (p[6][1]-p[6][0])
del[0][7]=10.0; // กำหนดช่วงพารามิเตอร์ G (p[7][1]-p[7][0])
del[0][8]=2.0; // กำหนดช่วงพารามิเตอร์ H (p[8][1]-p[8][0])
s0[0][0]=s0_best_ini[0][0];
s0[0][1]=s0_best_ini[0][1];
s0[0][2]=s0_best_ini[0][2];
s0[0][3]=s0_best_ini[0][3];
s0[0][4]=s0_best_ini[0][4];
s0[0][5]=s0_best_ini[0][5];

```

```

s0[0][6]=s0_best_ini[0][6];
s0[0][7]=s0_best_ini[0][7];
s0[0][8]=s0_best_ini[0][8];           // เก็บ s0_best_ini ไว้ใน s0
f_min0[0]=f_min_best_ini[0];         // เก็บ f_min_best_ini ไว้ใน f_min0
f_min0_old=f_min0[0];                // เก็บ f_min0 ไว้ใน f_min0_old เพื่อเปรียบเทียบ
ct_0=0;                               // รีเซต ct_0 = 0
rad=0.1;                              // กำหนดรัศมีการค้นหาเริ่มต้นเท่ากับ 10%
a=-1;                                 // ขอบเขตล่างของค่าสุ่ม a = -1
b=1;                                  // ขอบเขตบนของค่าสุ่ม b = 1
n=0;                                  // รีเซต n = 0
local_f[0]=f_min0[0];                 // เก็บ f_min0 ไว้ใน local_f
local_ne[0][0]=s0[0][0];
local_ne[0][1]=s0[0][1];
local_ne[0][2]=s0[0][2];
local_ne[0][3]=s0[0][3];
local_ne[0][4]=s0[0][4];
local_ne[0][5]=s0[0][5];
local_ne[0][6]=s0[0][6];
local_ne[0][7]=s0[0][7];
local_ne[0][8]=s0[0][8];             // เก็บ s0 ไว้ใน local_ne
global_f[0]=local_f[0];               // เก็บ local_f ไว้ใน global_f
global_ne[0][0]=local_ne[0][0];
global_ne[0][1]=local_ne[0][1];
global_ne[0][2]=local_ne[0][2];
global_ne[0][3]=local_ne[0][3];
global_ne[0][4]=local_ne[0][4];
global_ne[0][5]=local_ne[0][5];
global_ne[0][6]=local_ne[0][6];
global_ne[0][7]=local_ne[0][7];
global_ne[0][8]=local_ne[0][8];     // เก็บ local_ne ไว้ใน global_ne
printf("Enter seed of random no. for ATS = ");
scanf("%d",&st);                     // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด
srand(st);                            // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
printf("Enter number of ct_max = ");
scanf("%d",&ct_max);                 // รับค่าจำนวนรอบการค้นหาสูงสุดผ่านคีย์บอร์ด (1,000 รอบ)
x2: for(o=0;o<ct_max;o++)
{
/*****
// กลไกปรับรัศมีการค้นหา
*****/
if(global_f[0]>2.0)

```



```

{
    rad=0.2*rad;          // ถ้า global_f[0] > 2.0 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
else if((global_f[0]>1.34)&&(global_f[0] <=2.0))
{
    rad=0.2*rad; //      // ถ้า 1.34 < global_f ≤ 2.0 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
else if((global_f[0]>1.32)&&(global_f[0]<=1.34))
{
    rad=0.2*rad;          // ถ้า 1.32 < global_f ≤ 1.34 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
else if(global_f[0]<=1.32)
{
    rad=0.2*rad;          // ถ้า global_f ≤ 1.32 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
}

/*****
// สุ่มค่าใกล้เคียงจำนวน n_ne ชุด แล้วประเมินด้วยฟังก์ชันวัตถุประสงค์
*****/

for(i=0;i<n_ne;i++)
{
    for(j=0;j<para;j++)
    {
        c=rand();          // ค่า c จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
        r0=c/div;          // ค่า r0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
        r=a+r0*(b-a);      // ค่า r = a+[r0×(b-a)]
        s1[i][j]=s0[0][j]+r*rad*del[0][j]; // คำนวณค่าใกล้เคียง
        if((s1[i][j]<p[j][0])||(s1[i][j]>p[j][1]))
        {
            while((s1[i][j]<p[j][0])||(s1[i][j]>p[j][1]))
            {
                // ค่าใกล้เคียงที่ได้จะต้องอยู่ในขอบเขตบนและขอบเขตล่าง
                // ของค่าพารามิเตอร์ ถ้าค่าใกล้เคียงที่สุ่มได้มีค่าเกินขอบเขตบนและขอบเขตล่าง
                // ที่กำหนด ให้ทำการสุ่มค่าใกล้เคียงใหม่

                c=rand();
                r0=c/div;
                r=a+r0*(b-a);
                s1[i][j]=s0[0][j]+r*rad*del[0][j];
            }
        }
    }
}
}

```

```

/*****/
// ประเมินค่าพารามิเตอร์ของค่าใกล้เคียงที่สุ่มได้โดยฟังก์ชันวัตถุประสงค์ obj
/*****/

for(i=0;i<n_ne;i++)
{
    f_min1[i]=model(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]);
}
// เก็บค่าฟังก์ชันวัตถุประสงค์จากค่าใกล้เคียงแต่ละค่าไว้ใน f_min1
f_min1_old=f_min1[0]; // เก็บ f_min1 ตัวแรก (f_min1[0]) ไว้ใน f_min1_old เพื่อเปรียบเทียบ
for(i=0;i<n_ne;i++)
{
    if(f_min1[i]<f_min1_old) // ถ้า f_min1 < f_min1_old
    {
        f_min1_old=f_min1[i]; // ให้ f_min1_old = f_min1
        ct=ct_0; // ปรับค่า ct = ct_0
    }
    ct_0=ct_0+1; // เพิ่มค่า ct_0 ขึ้น 1
}
f_min1_best[0]=f_min1_old; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในบรรดาค่าใกล้เคียงไว้ใน f_min1_best
for(j=0;j<para;j++)
{
    s1_best[0][j]=s1[ct][j]; // เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดไว้ใน s1_best
}
f_tb[i_tb_list]=f_min1_old; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในบรรดาค่าใกล้เคียงไว้ใน f_tb
for(j=0;j<para;j++)
{
    s_tb[i_tb_list][j]=s1[ct][j]; // เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดไว้ใน s_tb
}
i_tb_list=i_tb_list+1; // เพิ่มค่า i_tb_list ขึ้น 1
ct_0=0; // รีเซ็ต ct_0 = 0
ct=0; // รีเซ็ต ct = 0
/*****/
// เปรียบเทียบค่าที่ดีที่สุดปัจจุบันกับค่าที่ดีที่สุดในอดีต เพื่อพิจารณาการซ้ำของค่าคำตอบ
/*****/

f_min0_new=f_min1_best[0]; // เก็บ f_min1_best ไว้ใน f_min0_new
if(f_min0_new>=f_min0_old) // ตรวจสอบการซ้ำของค่าคำตอบโดยการเปรียบเทียบระหว่าง
// f_min0_new และ f_min0_old
{
    n=n+1; // ถ้ามีการซ้ำของค่าคำตอบ ให้ปรับเพิ่มค่า n ขึ้น 1
}
else // ไม่เช่นนั้นแล้ว
{

```

```

n=0; //รีเซ็ตค่า n = 0
f_min0_old=f_min0_new; //ปรับค่า f_min0_old = f_min0_new
}

/*****
// กลไกย้อนรอยการค้นหา
*****/

if(n>=5) //เมื่อจำนวนการซ้ำของค่าคำตอบเท่ากับ 5
{ //เรียกใช้กลไกย้อนรอยการค้นหา
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; //ปรับเพิ่ม n_back ขึ้น 1
goto x1; //กระโดดไปทำงานที่ลาเบล x1
}

/*****
// ปรับค่าคำตอบและพารามิเตอร์
*****/

if(f_min1_best[0]<f_min0[0]) // ถ้า f_min1_best < f_min0
{
f_min0[0]=f_min1_best[0]; //เก็บ f_min1_best ไว้ใน f_min0
for(j=0;j<para;j++)
{
s0[0][j]=s1_best[0][j]; //เก็บค่าพารามิเตอร์ s1_best ไว้ใน s0
}
}

/*****
// ตรวจสอบจำนวนรอบการค้นหา
*****/

if(count>=ct_max) // ถ้า count ≥ ct_max
{
goto x3; //กระโดดไปทำงานที่ลาเบล x3
}
count=count+1; //เพิ่ม count ขึ้น 1
}

/*****
// เปรียบเทียบคำตอบวงแคบเฉพาะถิ่นและคำตอบวงกว้าง
*****/

x1: if(local_f[0]<global_f[0]) // ถ้า local_f < global_f
{

```

```

global_f[0]=local_f[0];          // เก็บ local_f ไว้ใน global_f
for(j=0;j<para;j++)
{
    global_ne[0][j]=local_ne[0][j];
}                                // เก็บค่าพารามิเตอร์ local_ne ไว้ใน global_ne
printf("%d\t%.1e\t%.4e\n",count,rad,global_f[0]);
}                                // พิมพ์ค่า count, rad, และ global_f
if(global_f[0]<=1.32)            // ตรวจสอบเกณฑ์ยุติการค้นหา เมื่อ global_f ≤ 1.32
{
    goto x3;                    // กระโดดไปทำงานที่ลาเบล x3
}

/*****
// จัดเรียงลำดับ (ranking) คำตอบที่บรรจุใน TL
*****/

i_rank=0;                       // รีเซ็ตค่า i_rank
for(ii=0;ii<i_tb_list;ii++)
{
    f_tb1=f_tb[0];              // เก็บค่า f_tb ตัวแรก (f_tb[0]) ไว้ใน f_tb1 เพื่อเปรียบเทียบ
    for(i=0;i<i_tb_list;i++)
    {
        if(f_tb[i]<f_tb1)        // ถ้า f_tb < f_tb1
        {
            f_tb1=f_tb[i];      // เก็บ f_tb ไว้ใน f_tb1
            ct_tb=ct_tb_0;      // ให้ ct_tb = ct_tb_0
        }
        ct_tb_0=ct_tb_0+1;      // เพิ่มค่า ct_tb_0 ขึ้น 1
    }
    f_tb[ct_tb]=1e7;            // แทนค่า f_tb ที่พบว่ามีความน้อยกว่า f_tb1 ด้วยค่ามาก ๆ
                                // ((1×107) ทั้งนี้เพื่อให้การหาค่าน้อยที่สุดในรอบต่อไป
                                // จะไม่นำเอาค่า f_tb ที่พบแล้วมาพิจารณาซ้ำอีก
    f_tb_rank[i_rank]=f_tb1;    // เก็บ f_tb1 ไว้ใน f_tb_rank
                                // เมื่อโปรแกรมในส่วนนี้สิ้นสุดลง ค่าที่ถูกเก็บใน f_tb_rank
                                // จะเรียงจากน้อยสุดไปมากที่สุด
    for(j=0;j<para;j++)
    {
        s_tb_rank[i_rank][j]=s_tb[ct_tb][j];
    }                            // เก็บค่าพารามิเตอร์ของ f_tb_rank ไว้ใน s_tb_rank
    ct_tb=0;                     // รีเซ็ต ct_tb = 0
    ct_tb_0=0;                   // รีเซ็ต ct_tb_0 = 0
    i_rank=i_rank+1;            // เพิ่มค่า i_rank ขึ้น 1
}

```

```

/*****
// แทนค่าตอบเริ่มต้นใหม่จากคำตอบที่เรียงลำดับแล้ว
/*****

i_tb_list=0; // รีเซต i_tb_list = 0
f_min0[0]=f_tb_rank[4]; // แทนค่า f_min0 ด้วยค่าลำดับที่ 5 (-5) ใน f_tb_rank (f_tb_rank[4])
for(j=0;j<para;j++)
{
    s0[0][j]=s_tb_rank[4][j]; // แทนค่าพารามิเตอร์ s0 ด้วยค่าพารามิเตอร์ชุดที่ 5 ใน s_tb_rank
}
f_min0_old=f_min0[0]; // แทน f_min0_old ด้วย f_min0
n=0; // รีเซต n = 0
count=count+1; // เพิ่มค่า count ขึ้น 1
goto x2; // กระโดดไปทำงานที่ลาเบล x2
/*****

// แสดงผลการค้นหา
/*****

x3: end=clock(); // ยุติการค้นหา และยุติการคำนวณเวลาการค้นหา
printf("stoped count = %d\n",count); // พิมพ์ค่า count สุดท้ายที่ยุติ
printf("global_f = %.4e\n",global_f[0]); // พิมพ์ค่าตอบวงกว้าง global_f
printf("Back tracking %d\n",n_back); // พิมพ์จำนวนการเรียกใช้กลไกย้อนรอยการค้นหา
printf("radius = %.2e\n",rad); // พิมพ์รัศมีการค้นหาสุดท้าย
printf("Execution time = %f second\n",(end-start)/CLK_TCK);
// พิมพ์เวลาทั้งหมดที่ใช้ในการค้นหา
printf("M = %.4e\nA = %.4e\nB = %.4e\nC = %.4e\n",global_ne[0][0],global_ne[0][1],global_ne[0][2],global_ne[0][3]);
printf("D = %.4e\nE = %.4e\nF = %.4e\nG = %.4e\nH =
%.4e\n",global_ne[0][4],global_ne[0][5],global_ne[0][6],global_ne[0][7],global_ne[0][8]);
// พิมพ์ค่าพารามิเตอร์ของ global_f (M, A,..., H)
printf("End of program"); // พิมพ์โปรแกรมสิ้นสุด
getch();
}
/*****

// ฟังก์ชันวัตถุประสงค์ model เป็นแบบจำลองของระบบ CPP ไม่เชิงเส้น
/*****

float model(float s0_ini_0,float s0_ini_1,float s0_ini_2,float s0_ini_3,float s0_ini_4,float s0_ini_5,float s0_ini_6,float
s0_ini_7,float s0_ini_8)
{
    float sq_er,s_sq_er,output; // ค่าความคลาดเคลื่อนกำลังสอง, ผลรวมค่าความคลาดเคลื่อนกำลังสอง,
// และเอาต์พุตของโปรแกรมย่อย
    float k1,k2,k3,k4; // ประกาศตัวแปรเพื่อใช้สำหรับการคำนวณผลเฉลย
    float m1,m2,m3,m4; // ของสมการอนุพันธ์ไม่เชิงเส้น ด้วยระเบียบวิธีรุงเง-คุดตา (Runge-Kutta method)
    float n1,n2,n3,n4;
}

```

```

float p1,p2,p3,p4;
float del_x1,del_x2,del_x3,del_x4;
float x,h; // ค่าเวลาเริ่มต้นการคำนวณ และช่วงเวลาชักตัวอย่าง
float x1,x2,x3,x4; //  $\phi$ ,  $\dot{\phi}$ ,  $y$ , และ  $\dot{y}$ 
float M,m,l,g,A,B,C,D,E,F,G,H; // ค่าพารามิเตอร์ของแบบจำลอง
float solu[1000]; // ค่าคำตอบของการจำลองผล
int i,iii; // ตัวแปรที่ใช้ในการนับ และวนรอบ
float ua_pen[1000],ub_pen[1000],x_pen[1000]; // อินพุต และเอาต์พุตสำหรับการจำลองผลการระบุเอกลักษณ์ระบบ
FILE *fp1,*fp2,*fp3;
fp1=fopen("ua_pen.dat","rb"); // ไฟล์ ua_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
fread(ua_pen,sizeof(float),700,fp1); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ua_pen
fclose(fp1); // ปิดไฟล์ ua_pen.dat
fp2=fopen("ub_pen.dat","rb"); // ไฟล์ ub_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
// ที่ได้รับการคำนวณค่าในช่วง เพื่อใช้คำนวณผลเฉลยตามระเบียบวิธีรุ่งง-คุตดา
fread(ub_pen,sizeof(float),700,fp2); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ub_pen
fclose(fp2); // ปิดไฟล์ ub_pen.dat
fp3=fopen("x_pen.dat","rb"); // ไฟล์ x_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลเอาต์พุต (ตำแหน่งเชิงมุมของลูกตุ้ม)
// จากตัวตรวจรู้ จากการทดสอบระบบ CPP ตอนที่ 3
fread(x_pen,sizeof(float),700,fp3); // อ่านข้อมูลเอาต์พุตมาเก็บไว้ในตัวแปร x_pen
fclose(fp3); // ปิดไฟล์ x_pen.dat
x=0; // กำหนดค่าเวลาเริ่มต้นการคำนวณเท่ากับ 0
h=0.15; // กำหนดช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
x1=0; // กำหนดค่าเริ่มต้นของ  $\phi$  เท่ากับ 0
x2=0; // กำหนดค่าเริ่มต้นของ  $\dot{\phi}$  เท่ากับ 0
x3=0; // กำหนดค่าเริ่มต้นของ  $y$  เท่ากับ 0
x4=0; // กำหนดค่าเริ่มต้นของ  $\dot{y}$  เท่ากับ 0
m=0.251; // ลูกตุ้มมีมวลเท่ากับ 0.251 กิโลกรัม
l=0.40; // ก้านของลูกตุ้มมีความยาวเท่ากับ 0.4 เมตร
g=9.81; // ความเร่งเนื่องจากแรงโน้มถ่วงเท่ากับ 9.81 เมตร/วินาที2
s_sq_er=0; // กำหนดผลรวมค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
sq_er=0; // กำหนดค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
M=s0_ini_0; // ให้ M=s0_ini_0;
A=s0_ini_1; // ให้ A=s0_ini_1;
B=s0_ini_2; // ให้ B=s0_ini_2;
C=s0_ini_3; // ให้ C=s0_ini_3;
D=s0_ini_4; // ให้ D=s0_ini_4;
E=s0_ini_5; // ให้ E=s0_ini_5;
F=s0_ini_6; // ให้ F=s0_ini_6;

```

```

G=s0_ini_7; // ให้ G=s0_ini_7;
H=s0_ini_8; // ให้ H=s0_ini_8;
for(i=100;i<=250;i++)
{
// จำนวนผลเฉลยของสมการอนุพันธ์ไม่เชิงเส้นด้วยระเบียบวิธีรุงง-คูดตา
// สำหรับรายละเอียดการคำนวณ ผู้อ่านสามารถศึกษาได้จากตำรา
// การคำนวณเชิงตัวเลขทั่วไป อาทิเช่น ปราโมทย์ เชะอำไพ (2544) หรือ
// Mathews และ Fink (2004) เป็นต้น

k1=h*obj1(x2);
m1=h*obj2(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n1=h*obj3(x4);
p1=h*obj4(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k2=h*obj1(x2+m1/2);
m2=h*obj2(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n2=h*obj3(x4+p1/2);
p2=h*obj4(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k3=h*obj1(x2+m2/2);
m3=h*obj2(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n3=h*obj3(x4+p2/2);
p3=h*obj4(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k4=h*obj1(x2+m3);
m4=h*obj2(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
n4=h*obj3(x4+p3);
p4=h*obj4(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
del_x1=(k1+2*k2+2*k3+k4)/6;
del_x2=(m1+2*m2+2*m3+m4)/6;
del_x3=(n1+2*n2+2*n3+n4)/6;
del_x4=(p1+2*p2+2*p3+p4)/6;
solu[i]=x1; // เก็บผลเฉลยที่คำนวณได้ไว้ใน solu
x1=x1+del_x1;
x2=x2+del_x2;
x3=x3+del_x3;
x4=x4+del_x4;
x=x+h; // ปรับค่าเวลาคำนวณสำหรับรอบต่อไป
}
for(iii=100;iii<=250;iii++)
{
sq_er=(solu[iii]-x_pen[iii])*(solu[iii]-x_pen[iii]); // จำนวนค่าความคลาดเคลื่อนกำลังสอง
s_sq_er=s_sq_er+sq_er; // และผลรวมค่าความคลาดเคลื่อนกำลังสอง
}
output=s_sq_er;
return(output);

```

```

}
/*****/
// ฟังก์ชันวัตถุประสงค์ obj1
/*****/
float obj1(float x2)
{
    float output1;           // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj1
    output1=x2;
    return(output1);
}
/*****/
// ฟังก์ชันวัตถุประสงค์ obj2
/*****/
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
    float output2;           // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj2
    output2=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)
    +G*u+H)*cos(x1)+0.5*m*l*x2*x2*sin(2*x1)+(M+m)*g*sin(x1))/(l*(m*cos(x1)*cos(x1)-(M+m)));
    return(output2);
}
/*****/
// ฟังก์ชันวัตถุประสงค์ obj3
/*****/
float obj3(float x4)
{
    float output3;           // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj3
    output3=x4;
    return(output3);
}
/*****/
// ฟังก์ชันวัตถุประสงค์ obj4
/*****/
float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
    float output4;           // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj4
    output4=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)+G*u+H)
    +l*m*x2*x2*sin(x1)+0.5*g*m*sin(2*x1))/((M+m)-m*cos(x1)*cos(x1));
    return(output4);
}
/*****/

```



```

// โปรแกรมนี้เป็นโปรแกรมจำลองผล และตรวจสอบความถูกต้องของแบบจำลองระบบ CPP ไม่เชิงเส้น
// เมื่อพิจารณาให้แรง  $f$  ที่เป็นพหุนามของ  $u$  อันดับ 2 ถึง 7
// โปรแกรมได้รับการพัฒนาด้วยภาษา C เพื่อการจำลองผลโดยใช้ข้อมูลที่ได้จากตัวตรวจรู้
// พัฒนาโดยเดชา พวงดาวเรือง (2004)
// สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

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

float obj1(float x2); // กำหนดฟังก์ชันวัตถุประสงค์ obj1
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj2
float obj3(float x4); // กำหนดฟังก์ชันวัตถุประสงค์ obj3
float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj4

void main()
{
float sq_er,s_sq_er,output; // ค่าความคลาดเคลื่อนกำลังสอง, ผลรวมค่าความคลาดเคลื่อนกำลังสอง,
// และเอาต์พุตของโปรแกรมน้อย

float k1,k2,k3,k4; // ประกาศตัวแปรเพื่อใช้สำหรับการคำนวณผลเฉลย
float m1,m2,m3,m4; // ของสมการอนุพันธ์ไม่เชิงเส้น ด้วยระเบียบวิธีรุ่งง-กูดดา
float n1,n2,n3,n4;
float p1,p2,p3,p4;
float del_x1,del_x2,del_x3,del_x4;
float x,h; // ค่าเวลาเริ่มต้นการคำนวณ และช่วงเวลาชักตัวอย่าง
float x1,x2,x3,x4; //  $\phi$ ,  $\dot{\phi}$ ,  $y$ , และ  $\dot{y}$ 
float M,m,l,g,A,B,C,D,E,F,G,H; // ค่าพารามิเตอร์ของแบบจำลอง
float solu[1000]; // ค่าคำตอบของการจำลองผล
int i,iii; // ตัวแปรที่ใช้ในการนับ และวนรอบ
float ua_pen[1000],ub_pen[1000],x_pen[1000]; // อินพุต และเอาต์พุตสำหรับการจำลองผลการระบุเอกลักษณ์ระบบ
FILE *fp1,*fp2,*fp3;
FILE *fp4;
fp1=fopen("ua_pen.dat","rb"); // ไฟล์ ua_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
fread(ua_pen,sizeof(float),900,fp1); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ua_pen
fclose(fp1); // ปิดไฟล์ ua_pen.dat
fp2=fopen("ub_pen.dat","rb"); // ไฟล์ ub_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
// ที่ได้รับการคำนวณค่าในช่วง เพื่อใช้คำนวณผลเฉลยตามระเบียบวิธีรุ่งง-กูดดา
fread(ub_pen,sizeof(float),900,fp2); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ub_pen
fclose(fp2); // ปิดไฟล์ ub_pen.dat
fp3=fopen("x_pen.dat","rb"); // ไฟล์ x_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า

```

```

// ไฟล์ดังกล่าวบรรจุข้อมูลเอาต์พุต (ตำแหน่งเชิงมุมของลูกตุ้ม)
// จากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
fread(x_pen,sizeof(float),900,fp3); // อ่านข้อมูลเอาต์พุตมาเก็บไว้ในตัวแปร x_pen
fclose(fp3); // ปิดไฟล์ x_pen.dat
x=0; // กำหนดค่าเวลาเริ่มต้นการคำนวณเท่ากับ 0
h=0.15; // กำหนดช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
x1=0; // กำหนดค่าเริ่มต้นของ  $\phi$  เท่ากับ 0
x2=0; // กำหนดค่าเริ่มต้นของ  $\dot{\phi}$  เท่ากับ 0
x3=0; // กำหนดค่าเริ่มต้นของ  $y$  เท่ากับ 0
x4=0; // กำหนดค่าเริ่มต้นของ  $\dot{y}$  เท่ากับ 0
m=0.251; // ลูกตุ้มมีมวลเท่ากับ 0.251 กิโลกรัม
l=0.40; // ก้านของลูกตุ้มมีความยาวเท่ากับ 0.4 เมตร
g=9.81; // ความเร่งเนื่องจากแรงโน้มถ่วงเท่ากับ 9.81 เมตร/วินาที2
s_sq_er=0; // กำหนดผลรวมค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
sq_er=0; // กำหนดค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
M=XXXX;
A= XXXX;
B= XXXX;
C= XXXX;
D= XXXX;
E= XXXX;
F= XXXX;
G= XXXX;
H= XXXX; // ค่า XXXX ของพารามิเตอร์ M, A, B, ..., H
// เป็นค่าที่ได้จากการระบุเอกลักษณ์ด้วยการค้นหาแบบตามเชิงปรับตัว
// จากโปรแกรมระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น
// เพื่อพิจารณาอันดับของแรง  $f$  ที่เป็นพหุนามของ  $u$ 
// จากอันดับ 2 ถึง 7 ดังที่ได้แสดงไว้แล้วข้างต้น

for(i=100;i<=300;i++)
{ // คำนวณผลเฉลยของสมการอนุพันธ์ไม่เชิงเส้นด้วยระเบียบวิธีรุง-คุตดา
k1=h*obj1(x2);
m1=h*obj2(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n1=h*obj3(x4);
p1=h*obj4(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k2=h*obj1(x2+m1/2);
m2=h*obj2(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n2=h*obj3(x4+p1/2);
p2=h*obj4(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k3=h*obj1(x2+m2/2);
m3=h*obj2(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n3=h*obj3(x4+p2/2);

```

```

p3=h*obj4(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k4=h*obj1(x2+m3);
m4=h*obj2(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
n4=h*obj3(x4+p3);
p4=h*obj4(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
del_x1=(k1+2*k2+2*k3+k4)/6;
del_x2=(m1+2*m2+2*m3+m4)/6;
del_x3=(n1+2*n2+2*n3+n4)/6;
del_x4=(p1+2*p2+2*p3+p4)/6;
solu[i]=x1; // เก็บผลเฉลยที่คำนวณได้ไว้ใน solu
fp4=fopen("sim_sensor_order2.dat","wb+");
// ทำการเปิดไฟล์ sim_sensor_order2.dat
// เพื่อเก็บผลการจำลอง และตรวจสอบความถูกต้องของแบบจำลองระบบ
// CPP ไม่เชิงเส้น เมื่อแรง f ที่เป็นพหุนามอันดับ 2 ของ u
// สำหรับไฟล์ที่ใช้เก็บผลการจำลอง และตรวจสอบความถูกต้อง
// เมื่อแรง f ที่เป็นพหุนามอันดับ 3 ถึง 7 ของ u
// คือไฟล์ sim_sensor_order3.dat, sim_sensor_order4.dat,...,
// sim_sensor_order7.dat ตามลำดับ
// ไฟล์ดังกล่าวข้างต้น จะได้รับการอ่านและแสดงผล
// ด้วยโปรแกรม MATLAB ต่อไป ซึ่งจะไม่ขอแสดงรายละเอียดในที่นี้
fwrite(solu,sizeof(float),i,fp4); // ทำการเขียนผลการจำลอง และตรวจสอบความถูกต้องลงไฟล์
fclose(fp4); // ปิดไฟล์
x1=x1+del_x1;
x2=x2+del_x2;
x3=x3+del_x3;
x4=x4+del_x4;
x=x+h; // ปรับค่าเวลาคำนวณสำหรับรอบต่อไป
}
for(iii=100;iii<=300;iii++)
{
sq_er=(solu[iii]-x_pen[iii])*(solu[iii]-x_pen[iii]);
s_sq_er=s_sq_er+sq_er; // คำนวณค่าความคลาดเคลื่อนกำลังสอง
// และผลรวมค่าความคลาดเคลื่อนกำลังสอง
}
getch();
}
/*****
// ฟังก์ชันวัตถุประสงก์ obj1
*****/
float obj1(float x2)
{
float output1; // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงก์ obj1

```

```

    output1=x2;
    return(output1);
}

/*****/
// ฟังก์ชันวัตถุประสงค์ obj2
/*****/
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
    float output2;           // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj2
    output2=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)
    +G*u+H)*cos(x1)+0.5*m*1*x2*x2*sin(2*x1)+(M+m)*g*sin(x1))/(l*(m*cos(x1)*cos(x1)-(M+m)));
    return(output2);
}

/*****/
// ฟังก์ชันวัตถุประสงค์ obj3
/*****/
float obj3(float x4)
{
    float output3;           // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj3
    output3=x4;
    return(output3);
}

/*****/
// ฟังก์ชันวัตถุประสงค์ obj4
/*****/
float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
    float output4;           // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj4
    output4=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)+G*u+H)
    +1*m*x2*x2*sin(x1)+0.5*g*m*sin(2*x1))/((M+m)-m*cos(x1)*cos(x1));
    return(output4);
}

/*****/

// โปรแกรมนี้เป็น โปรแกรมระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น
// เป็นการระบุเอกลักษณ์ระบบด้วยการค้นหาแบบตามเชิงปรับตัว
// โดยกำหนดให้แรง  $f$  ที่เป็นพหุนามอันดับ 7 ของ  $u$ 
// โปรแกรมได้รับการพัฒนาด้วยภาษา C เพื่อการระบุเอกลักษณ์ระบบ โดยใช้ข้อมูลที่ได้จากตัวตรวจรู้
// จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
// พัฒนาโดยเดชา พวงดาวเรือง (2004)
// สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

```

```

#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#define para 9 // กำหนดจำนวนพารามิเตอร์ที่ต้องการค้นหาเท่ากับ 9
#define row 5 // กำหนดจำนวนการสุ่มค่าตอบเริ่มต้นเท่ากับ 5 ชุด
#define n_ne 30 // กำหนดจำนวนการสุ่มค่าใกล้เคียงเท่ากับ 30
float obj1(float x2); // กำหนดฟังก์ชันวัตถุประสงค์ obj1
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj2
float obj3(float x4); // กำหนดฟังก์ชันวัตถุประสงค์ obj3
float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj4
float model(float s0_ini_0,float s0_ini_1,float s0_ini_2,float s0_ini_3,float s0_ini_4,float s0_ini_5,float s0_ini_6,float
s0_ini_7,float s0_ini_8); // กำหนดฟังก์ชันวัตถุประสงค์ model
void main()
{
/*****
// ประกาศตัวแปรใช้งาน
*****/
clock_t start,end; // ประกาศตัวแปรเพื่อคำนวณเวลาการค้นหา
int st_ini; // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int i,j_rank,i_tb_list,ii,j,ct,ct_0,o,ct_0_ini,i_ini,j_ini,ct_ini,ct_tb,ct_tb_0,count;
// ตัวแปรใช้งานเพื่อการนับค่าและวนรอบ
int st; // ตัวแปรกำหนดค่าเริ่มต้นการสุ่ม
int n; // ตัวนับจำนวนการซ้ำของค่าคำตอบ
int n_back; // ตัวนับจำนวนการใช้กลไกย้อนรอยการค้นหา
int ct_max; // จำนวนรอบการค้นหาสูงสุด
float p[para][2]; // ตัวแปร p คือตัวแปรกำหนดขอบเขตของพารามิเตอร์แต่ละตัว
// มีขนาดเท่ากับจำนวนพารามิเตอร์ × 2 (ขอบบน,ขอบล่าง)
float f_ini,f_min_ini; // f_ini คือค่าฟังก์ชันวัตถุประสงค์เริ่มต้นในแต่ละรอบการค้นหา
// f_min_ini คือค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
float s0_ini[row][para]; // ค่าพารามิเตอร์เริ่มต้นจำนวน 5 × 2 ชุด (เพื่อเป็นตัวอย่าง)
float s; // ตัวแปรชั่วคราวทำหน้าที่รับส่งค่าพารามิเตอร์
float f_min_best_ini[1]; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุด
float s0_best_ini[1][para]; // ค่าพารามิเตอร์ของ f_min_best_ini
float div,rd,rd_0; // ตัวแปรเกี่ยวกับการสุ่มค่า
// div = 32767 เป็นตัวหารค่าที่สุ่มได้เพื่อทำให้มีค่าสูงสุดเท่ากับ 1
// rd คือค่าที่สุ่มได้จากคำสั่ง rand() มีค่าบวกสูงสุดเท่ากับ 32767
// rd_0 = rd/div คือค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1

```

```

float f_tb[tb_list]; // ค่าฟังก์ชันวัตถุประสงค์ที่เก็บใน TL
float f_tb_rank[tb_list]; // ค่าฟังก์ชันวัตถุประสงค์ที่เรียงลำดับแล้ว
float f_tb1; // ค่าฟังก์ชันวัตถุประสงค์เพื่อใช้ในการเปรียบเทียบ
float s_tb[tb_list][para]; // ค่าพารามิเตอร์ของ f_tb
float s_tb_rank[tb_list][para]; // ค่าพารามิเตอร์ของ f_tb_rank
float a,b; // ขอบเขตของค่าสุ่ม a = -1 (ขอบล่าง) b = 1 (ขอบบน)
float c; // ค่าที่สุ่มได้จากคำสั่ง rand() มีค่าบวกสูงสุดเท่ากับ 32767
float rd0; // rd_0 = rd/div คือค่าสุ่มที่ปรับให้อยู่ในช่วง 0 ถึง 1
float r; // ค่าสุ่ม r = a+[r0*(b-a)]
float rad; // ตัวแปรสุ่มมีการค้นหา
float f_min0[1]; // ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่ดีที่สุดมีค่าเท่ากับ f_min_best_ini
float s0[1][para]; // ค่าพารามิเตอร์ของ f_min0 มีค่าเท่ากับ s0_best_ini
float s1[n_ne][para]; // ค่าพารามิเตอร์ของค่าใกล้เคียงในแต่ละรอบการค้นหา
float del[1][para]; // ช่วงของค่าพารามิเตอร์แต่ละตัว
float f_min1_old; // ค่าฟังก์ชันวัตถุประสงค์เพื่อการเปรียบเทียบหาค่าที่ดีกว่า
float f_min1[n_ne]; // ค่าตอบทั้งหมดในแต่ละรอบการค้นหา
float f_min1_best[1]; // ค่าตอบที่ดีที่สุดในแต่ละรอบการค้นหา
float s1_best[1][para]; // ค่าพารามิเตอร์ของค่าตอบที่ดีที่สุดในแต่ละรอบการค้นหา
float f_min0_old,f_min0_new; // f_min0_old คือค่าตอบที่ดีที่สุดในอดีต
// f_min0_new คือค่าตอบที่ดีที่สุดในปัจจุบัน

float local_f[1]; // ค่าตอบวงแคบเฉพาะถิ่น
float local_ne[1][para]; // ค่าพารามิเตอร์ของ local_f
float global_f[1]; // ค่าตอบวงกว้าง
float global_ne[1][para]; // ค่าพารามิเตอร์ของ global_f

/*****
// สุ่มค่าตอบเริ่มต้นภายในปริภูมิการค้นหา
*****/

start=clock(); // เริ่มต้นคำนวณเวลาการค้นหา
ct=0; // รีเซต ct = 0
ct_0_ini=0; // รีเซต ct_0_ini = 0
count=0; // รีเซต count = 0
ct_tb=0; // รีเซต ct_tb = 0
ct_tb_0=0; // รีเซต ct_tb_0 = 0
i_tb_list=0; // รีเซต i_tb_list = 0
f_min_ini=32767; // กำหนด f_min_ini = 32767 ไว้เพื่อการเปรียบเทียบ (เพื่อเป็นตัวอย่าง)
div=32767; // กำหนด div = 32767
p[0][0]=0.5; //M // กำหนดขอบเขตล่างของพารามิเตอร์ M = 0.5 (พิจารณาเป็น x1)
p[0][1]=1.5; // // กำหนดขอบเขตบนของพารามิเตอร์ M = 1.5 (พิจารณาเป็น xn)
p[1][0]=4; //A // กำหนดขอบเขตล่างของพารามิเตอร์ A = 4.0 (พิจารณาเป็น x1)
p[1][1]=5; // // กำหนดขอบเขตบนของพารามิเตอร์ A = 5.0 (พิจารณาเป็น xn)
p[2][0]=6; //B // กำหนดขอบเขตล่างของพารามิเตอร์ B = 6.0 (พิจารณาเป็น x1)

```

```

p[2][1]=7; // กำหนดขอบเขตบนของพารามิเตอร์ B = 7.0 (พิจารณาเป็น  $x_h$ )
p[3][0]=-2; //C // กำหนดขอบเขตล่างของพารามิเตอร์ C = -2.0 (พิจารณาเป็น  $x_i$ )
p[3][1]=-1; // กำหนดขอบเขตบนของพารามิเตอร์ C = -1.0 (พิจารณาเป็น  $x_h$ )
p[4][0]=-6; //D // กำหนดขอบเขตล่างของพารามิเตอร์ D = -6.0 (พิจารณาเป็น  $x_i$ )
p[4][1]=-5; // กำหนดขอบเขตบนของพารามิเตอร์ D = -5.0 (พิจารณาเป็น  $x_h$ )
p[5][0]=-5; //E // กำหนดขอบเขตล่างของพารามิเตอร์ E = -5.0 (พิจารณาเป็น  $x_i$ )
p[5][1]=5; // กำหนดขอบเขตบนของพารามิเตอร์ E = 5.0 (พิจารณาเป็น  $x_h$ )
p[6][0]=-5; //F // กำหนดขอบเขตล่างของพารามิเตอร์ F = -5.0 (พิจารณาเป็น  $x_i$ )
p[6][1]=5; // กำหนดขอบเขตบนของพารามิเตอร์ F = 5.0 (พิจารณาเป็น  $x_h$ )
p[7][0]=-5; //G // กำหนดขอบเขตล่างของพารามิเตอร์ G = -5.0 (พิจารณาเป็น  $x_i$ )
p[7][1]=5; // กำหนดขอบเขตบนของพารามิเตอร์ G = 5.0 (พิจารณาเป็น  $x_h$ )
p[8][0]=-1; //H // กำหนดขอบเขตล่างของพารามิเตอร์ H = -1.0 (พิจารณาเป็น  $x_i$ )
p[8][1]=1; // กำหนดขอบเขตบนของพารามิเตอร์ H = 1.0 (พิจารณาเป็น  $x_h$ )

printf("Enter seed of random no.\nfor finding initial feasible solution = ");
scanf("%d",&st_ini); // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด
srand(st_ini); // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand

for(i_ini=0;i_ini<row;i_ini++)
{
    j_ini=0; // ตัวชี้ค่าพารามิเตอร์ M
    rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
    rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
    s=((p[0][1]-p[0][0])*rd_0)+p[0][0]; // สุ่มค่าพารามิเตอร์ M จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
    // เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
    s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ M ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
    j_ini=1; // ตัวชี้ค่าพารามิเตอร์ A
    rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
    rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
    s=((p[1][1]-p[1][0])*rd_0)+p[1][0]; // สุ่มค่าพารามิเตอร์ A จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
    // เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
    s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ A ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
    j_ini=2; // ตัวชี้ค่าพารามิเตอร์ B
    rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
    rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
    s=((p[2][1]-p[2][0])*rd_0)+p[2][0]; // สุ่มค่าพารามิเตอร์ B จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
    // เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 

```

```

s0_ini[i_ini][j_ini]=s;           // เก็บค่าพารามิเตอร์ B ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=3;                          // ตัวชี้ค่าพารามิเตอร์ C
rd=rand();                         // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div;                       // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[3][1]-p[3][0])*rd_0)+p[3][0]; // สุ่มค่าพารามิเตอร์ C จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s;           // เก็บค่าพารามิเตอร์ C ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=4;                          // ตัวชี้ค่าพารามิเตอร์ D
rd=rand();                         // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div;                       // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[4][1]-p[4][0])*rd_0)+p[4][0]; // สุ่มค่าพารามิเตอร์ D จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s;           // เก็บค่าพารามิเตอร์ D ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=5;                          // ตัวชี้ค่าพารามิเตอร์ E
rd=rand();                         // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div;                       // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[5][1]-p[5][0])*rd_0)+p[5][0]; // สุ่มค่าพารามิเตอร์ E จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s;           // เก็บค่าพารามิเตอร์ E ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=6;                          // ตัวชี้ค่าพารามิเตอร์ F
rd=rand();                         // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div;                       // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[6][1]-p[6][0])*rd_0)+p[6][0]; // สุ่มค่าพารามิเตอร์ F จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพันธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s;           // เก็บค่าพารามิเตอร์ F ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=7;                          // ตัวชี้ค่าพารามิเตอร์ G

```



```

rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[7][1]-p[7][0])*rd_0)+p[7][0]; // สุ่มค่าพารามิเตอร์ G จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพัทธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ G ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
for(i_ini=0;i_ini<row;i_ini++)
{
j_ini=8; // ตัวชี้ค่าพารามิเตอร์ H
rd=rand(); // ค่า rd จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
rd_0=rd/div; // ค่า rd_0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
s=((p[8][1]-p[8][0])*rd_0)+p[8][0]; // สุ่มค่าพารามิเตอร์ H จำนวน 5 ชุดเพื่อหาค่าตอบเริ่มต้นที่ดีที่สุด
// เก็บไว้ใน s โดยใช้ความสัมพัทธ์  $(x_{h,i} - x_{l,i})rand + x_{l,i}$ , for  $i = 1, \dots, 5$ 
s0_ini[i_ini][j_ini]=s; // เก็บค่าพารามิเตอร์ H ที่สุ่มได้ไว้ใน s0_ini ตำแหน่ง i_ini, j_ini
}
/*****
// ประเมินค่าพารามิเตอร์ที่สุ่มได้โดยฟังก์ชันวัตถุประสงค์ obj
*****/
for(i_ini=0;i_ini<row;i_ini++)
{
f_ini=model(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]);
if(f_ini<=f_min_ini) // เปรียบเทียบ f_ini และ f_min_ini เพื่อหาค่าน้อยที่สุด
{
f_min_ini=f_ini; // ถ้า  $f_{ini} \leq f_{min\_ini}$  แล้วให้  $f_{min\_ini} = f_{ini}$ 
ct_ini=ct_0_ini; // ปรับตัวนับ ct_ini = ct_0_ini
}
ct_0_ini=ct_0_ini+1; // เพิ่มค่า ct_0_ini ขึ้น 1
}
f_min_best_ini[0]=f_min_ini; // ให้ f_min_best_ini = f_min_ini
f_tb[i_tb_list]=f_min_best_ini[0]; // เก็บค่า f_min_best_ini ไว้ใน f_tb
printf("f_min_best_ini[0] = %.4e\n",f_min_best_ini[0]);
// พิมพ์ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นที่น้อยที่สุด
for(j=0;j<para;j++)
{
s0_best_ini[0][j]=s0_ini[ct_ini][j]; // เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์เริ่มต้นมีค่าน้อยที่สุด
s_tb[i_tb_list][j]=s0_ini[ct_ini][j]; // ไว้ใน s0_best_ini และ s_tb
}
i_tb_list=i_tb_list+1; // เพิ่มค่า i_tb_list ขึ้น 1

```

```

/*****
// เริ่มต้นการค้นหาแบบตามูเชิงปรับตัว
*****/

n_back=0; // รีเซต n_back = 0
del[0][0]=1.0; // กำหนดช่วงพารามิเตอร์ M (p[0][1]-p[0][0])
del[0][1]=1.0; // กำหนดช่วงพารามิเตอร์ A (p[1][1]-p[1][0])
del[0][2]=1.0; // กำหนดช่วงพารามิเตอร์ B (p[2][1]-p[2][0])
del[0][3]=1.0; // กำหนดช่วงพารามิเตอร์ C (p[3][1]-p[3][0])
del[0][4]=1.0; // กำหนดช่วงพารามิเตอร์ D (p[4][1]-p[4][0])
del[0][5]=10.0; // กำหนดช่วงพารามิเตอร์ E (p[5][1]-p[5][0])
del[0][6]=10.0; // กำหนดช่วงพารามิเตอร์ F (p[6][1]-p[6][0])
del[0][7]=10.0; // กำหนดช่วงพารามิเตอร์ G (p[7][1]-p[7][0])
del[0][8]=2.0; // กำหนดช่วงพารามิเตอร์ H (p[8][1]-p[8][0])

s0[0][0]=s0_best_ini[0][0];
s0[0][1]=s0_best_ini[0][1];
s0[0][2]=s0_best_ini[0][2];
s0[0][3]=s0_best_ini[0][3];
s0[0][4]=s0_best_ini[0][4];
s0[0][5]=s0_best_ini[0][5];
s0[0][6]=s0_best_ini[0][6];
s0[0][7]=s0_best_ini[0][7];
s0[0][8]=s0_best_ini[0][8]; // เก็บ s0_best_ini ไว้ใน s0
f_min0[0]=f_min_best_ini[0]; // เก็บ f_min_best_ini ไว้ใน f_min0
f_min0_old=f_min0[0]; // เก็บ f_min0 ไว้ใน f_min0_old เพื่อเปรียบเทียบ
ct_0=0; // รีเซต ct_0 = 0
rad=0.1; // กำหนดรัศมีการค้นหาเริ่มต้นเท่ากับ 10%
a=-1; // ขอบเขตล่างของค่าสุ่ม a = -1
b=1; // ขอบเขตบนของค่าสุ่ม b = 1
n=0; // รีเซต n = 0
local_f[0]=f_min0[0]; // เก็บ f_min0 ไว้ใน local_f
local_ne[0][0]=s0[0][0];
local_ne[0][1]=s0[0][1];
local_ne[0][2]=s0[0][2];
local_ne[0][3]=s0[0][3];
local_ne[0][4]=s0[0][4];
local_ne[0][5]=s0[0][5];
local_ne[0][6]=s0[0][6];
local_ne[0][7]=s0[0][7];
local_ne[0][8]=s0[0][8]; // เก็บ s0 ไว้ใน local_ne
global_f[0]=local_f[0]; // เก็บ local_f ไว้ใน global_f
global_ne[0][0]=local_ne[0][0];

```

```

global_ne[0][1]=local_ne[0][1];
global_ne[0][2]=local_ne[0][2];
global_ne[0][3]=local_ne[0][3];
global_ne[0][4]=local_ne[0][4];
global_ne[0][5]=local_ne[0][5];
global_ne[0][6]=local_ne[0][6];
global_ne[0][7]=local_ne[0][7];
global_ne[0][8]=local_ne[0][8];      // เก็บ local_ne ไว้ใน global_ne
printf("Enter seed of random no. for ATS = ");
scanf("%d",&st);                    // รับค่าเริ่มต้นการสุ่มผ่านคีย์บอร์ด
srand(st);                           // เก็บค่าเริ่มต้นการสุ่มไว้ในคำสั่ง srand
printf("Enter number of ct_max = ");
scanf("%d",&ct_max);                // รับค่าจำนวนรอบการค้นหาสูงสุดผ่านคีย์บอร์ด (1,000 รอบ)
x2: for(o=0;o<ct_max;o++)
{
/*****
// กลไกปรับรัศมีการค้นหา
*****/

if(global_f[0]>2.0)
{
rad=0.2*rad;                        // ถ้า global_f[0] > 2.0 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
else if((global_f[0]>1.34)&&(global_f[0] <=2.0))
{
rad=0.2*rad;                        // ถ้า 1.34 < global_f ≤ 2.0 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
else if((global_f[0]>1.32)&&(global_f[0]<=1.34))
{
rad=0.2*rad;                        // ถ้า 1.32 < global_f ≤ 1.34 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
else if(global_f[0]<=1.32)
{
rad=0.2*rad;                        // ถ้า global_f ≤ 1.32 แล้ว radius เท่ากับ 20% ของค่ารัศมีก่อนหน้า
}
/*****
// สุ่มค่าใกล้เคียงจำนวน n_ne ชุด แล้วประเมินด้วยฟังก์ชันวัตถุประสงค์
*****/

for(i=0;i<n_ne;i++)
{
for(j=0;j<para;j++)
{

```

```

c=rand(); // ค่า c จากการสุ่มจะอยู่ในช่วง 0 ถึง 32767
r0=c/div; // ค่า r0 จะเป็นค่าสุ่มที่อยู่ในช่วง 0 ถึง 1
r=a+r0*(b-a); // ค่า r = a+[r0*(b-a)]
s1[i][j]=s0[0][j]+r*rad*del[0][j]; // คำนวณค่าใกล้เคียง
if((s1[i][j]<p[j][0])||(s1[i][j]>p[j][1]))
{
    while((s1[i][j]<p[j][0])||(s1[i][j]>p[j][1]))
    {
        // ค่าใกล้เคียงที่ได้จะต้องอยู่ในขอบเขตบนและขอบเขตล่าง
        // ของค่าพารามิเตอร์ ถ้าค่าใกล้เคียงที่สุ่มได้มีค่าเกินขอบเขตบนและขอบเขตล่าง
        // ที่กำหนด ให้ทำการสุ่มค่าใกล้เคียงใหม่

        c=rand();
        r0=c/div;
        r=a+r0*(b-a);
        s1[i][j]=s0[0][j]+r*rad*del[0][j];
    }
}
}

/*****
// ประเมินค่าพารามิเตอร์ของค่าใกล้เคียงที่สุ่มได้โดยฟังก์ชันวัตถุประสงค์ obj
*****/

for(i=0;i<n_ne;i++)
{
    f_min1[i]=model(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]);
} // เก็บค่าฟังก์ชันวัตถุประสงค์จากค่าใกล้เคียงแต่ละค่าไว้ใน f_min1
f_min1_old=f_min1[0]; // เก็บ f_min1 ตัวแรก (f_min1[0]) ไว้ใน f_min1_old เพื่อเปรียบเทียบ
for(i=0;i<n_ne;i++)
{
    if(f_min1[i]<f_min1_old) // ถ้า f_min1 < f_min1_old
    {
        f_min1_old=f_min1[i]; // ให้ f_min1_old = f_min1
        ct=ct_0; // ปรับค่า ct = ct_0
    }
    ct_0=ct_0+1; // เพิ่มค่า ct_0 ขึ้น 1
}
f_min1_best[0]=f_min1_old; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในบรรดาค่าใกล้เคียงไว้ใน f_min1_best
for(j=0;j<para;j++)
{
    s1_best[0][j]=s1[ct][j]; // เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์ดีที่สุดไว้ใน s1_best
}

```

```

f_tb[i_tb_list]=f_min1_old; // เก็บค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดในรอบค่าค่าใกล้เคียงไว้ใน f_tb
for(j=0;j<para;j++)
{
    s_tb[i_tb_list][j]=s1[ct][j]; // เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดไว้ใน s_tb
}
i_tb_list=i_tb_list+1; // เพิ่มค่า i_tb_list ขึ้น 1
ct_0=0; // รีเซต ct_0 = 0
ct=0; // รีเซต ct = 0
/*****
// เปรียบเทียบค่าที่ดีที่สุดปัจจุบันกับค่าที่ดีที่สุดในอดีต เพื่อพิจารณาการซ้ำของค่าคำตอบ
/*****/

f_min0_new=f_min1_best[0]; // เก็บ f_min1_best ไว้ใน f_min0_new
if(f_min0_new>=f_min0_old) // ตรวจสอบการซ้ำของค่าคำตอบโดยการเปรียบเทียบระหว่าง
// f_min0_new และ f_min0_old

{
    n=n+1; // ถ้ามีการซ้ำของค่าคำตอบ ให้ปรับเพิ่มค่า n ขึ้น 1
}
else // ไม่เช่นนั้นแล้ว
{
    n=0; // รีเซตค่า n = 0
    f_min0_old=f_min0_new; // ปรับค่า f_min0_old = f_min0_new
}
/*****
// กลไกย้อนรอยการค้นหา
/*****/

if(n>=5) // เมื่อจำนวนการซ้ำของค่าคำตอบเท่ากับ 5
{
    // เรียกใช้กลไกย้อนรอยการค้นหา
    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; // ปรับเพิ่ม n_back ขึ้น 1
    goto x1; // กระโดดไปทำงานที่ลาเบล x1
}
/*****
// ปรับค่าคำตอบและพารามิเตอร์
/*****/

if(f_min1_best[0]<f_min0[0]) // ถ้า f_min1_best < f_min0
{
    f_min0[0]=f_min1_best[0]; // เก็บ f_min1_best ไว้ใน f_min0
}

```

```

    for(j=0;j<para;j++)
    {
        s0[0][j]=s1_best[0][j];          // เก็บค่าพารามิเตอร์ s1_best ไว้ใน s0
    }
}

/*****

// ตรวจสอบจำนวนรอบการค้นหา

/*****

if(count>=ct_max)          // ถ้า count ≥ ct_max
{
    goto x3;                // กระโดดไปทำงานที่ลาเบล x3
}
count=count+1 ;           // เพิ่ม count ขึ้น 1
}

/*****

// เปรียบเทียบคำตอบวงแคบเฉพาะถิ่นและคำตอบวงกว้าง

/*****

x1: if(local_f[0]<global_f[0])      // ถ้า local_f < global_f
{
    global_f[0]=local_f[0];        // เก็บ local_f ไว้ใน global_f
    for(j=0;j<para;j++)
    {
        global_ne[0][j]=local_ne[0][j];
    }
    // เก็บค่าพารามิเตอร์ local_ne ไว้ใน global_ne
    printf("%d\t%.1e\t%.4e\n",count,rad,global_f[0]);
}
// พิมพ์ค่า count, rad, และ global_f
if(global_f[0]<=1.32)              // ตรวจสอบเกณฑ์ยุติการค้นหา เมื่อ global_f ≤ 1.32
{
    goto x3;                       // กระโดดไปทำงานที่ลาเบล x3
}

/*****

// จัดเรียงลำดับ (ranking) คำตอบที่บรรจุใน TL

/*****

i_rank=0;                          // รีเซ็ตค่า i_rank
for(ii=0;ii<i_tb_list;ii++)
{
    f_tb1=f_tb[0];                  // เก็บค่า f_tb ตัวแรก (f_tb[0]) ไว้ใน f_tb1 เพื่อเปรียบเทียบ
    for(i=0;i<i_tb_list;i++)
    {
        if(f_tb[i]<f_tb1)           // ถ้า f_tb < f_tb1
        {

```

```

        f_tb1=f_tb[i]; // เก็บ f_tb ไว้ใน f_tb1
        ct_tb=ct_tb_0; // ให้ ct_tb = ct_tb_0
    }
    ct_tb_0=ct_tb_0+1; // เพิ่มค่า ct_tb_0 ขึ้น 1
}
f_tb[ct_tb]=1e7; // แทนค่า f_tb ที่พบว่ามีความน้อยกว่า f_tb1 ด้วยค่ามาก ๆ
// (1×107) ทั้งนี้เพื่อให้การหาค่าน้อยที่สุดในรอบต่อไป
// จะไม่นำเอาค่า f_tb ที่พบแล้วมาพิจารณาซ้ำอีก
f_tb_rank[i_rank]=f_tb1; // เก็บ f_tb1 ไว้ใน f_tb_rank
// เมื่อโปรแกรมในส่วนนี้สิ้นสุดลง ค่าที่ถูกเก็บใน f_tb_rank
// จะเรียงจากน้อยสุดไปมากที่สุด

for(j=0;j<para;j++)
{
    s_tb_rank[i_rank][j]=s_tb[ct_tb][j];
} // เก็บค่าพารามิเตอร์ของ f_tb_rank ไว้ใน s_tb_rank
ct_tb=0; // รีเซ็ต ct_tb = 0
ct_tb_0=0; // รีเซ็ต ct_tb_0 = 0
i_rank=i_rank+1; // เพิ่มค่า i_rank ขึ้น 1
}

/*****
// แทนคำตอบเริ่มต้นใหม่จากคำตอบที่เรียงลำดับแล้ว
*****/

i_tb_list=0; // รีเซ็ต i_tb_list = 0
f_min0[0]=f_tb_rank[4]; // แทนค่า f_min0 ด้วยค่าลำดับที่ 5 (-5) ใน f_tb_rank (f_tb_rank[4])
for(j=0;j<para;j++)
{
    s0[0][j]=s_tb_rank[4][j]; // แทนค่าพารามิเตอร์ s0 ด้วยค่าพารามิเตอร์ชุดที่ 5
} // ใน s_tb_rank
f_min0_old=f_min0[0]; // แทน f_min0_old ด้วย f_min0
n=0; // รีเซ็ต n = 0
count=count+1; // เพิ่มค่า count ขึ้น 1
goto x2; // กระโดดไปทำงานที่ลาเบล x2

/*****
// แสดงผลการค้นหา
*****/

x3: end=clock(); // ยุติการค้นหา และยุติการคำนวณเวลาการค้นหา
printf("stoped count = %d\n",count); // พิมพ์ค่า count สุดท้ายที่ยุติ
printf("global_f = %.4e\n",global_f[0]); // พิมพ์คำตอบวงกว้าง global_f
printf("Back tracking %d\n",n_back); // พิมพ์จำนวนการเรียกใช้กลไกย้อนรอยการค้นหา
printf("radius = %.2e\n",rad); // พิมพ์รัศมีการค้นหาสุดท้าย
printf("Execution time = %f second\n",(end-start)/CLK_TCK);
// พิมพ์เวลาทั้งหมดที่ใช้ในการค้นหา

```

```

printf("M = %.4e\nA = %.4e\nB = %.4e\nC = %.4e\n",global_ne[0][0],global_ne[0][1],global_ne[0][2],global_ne[0][3]);
printf("D = %.4e\nE = %.4e\nF = %.4e\nG = %.4e\nH =
%.4e\n",global_ne[0][4],global_ne[0][5],global_ne[0][6],global_ne[0][7],global_ne[0][8]);
// พิมพ์ค่าพารามิเตอร์ของ global_f (M, A,..., H)
printf("End of program"); // พิมพ์โปรแกรมสิ้นสุด
getch();
}
/*****
// ฟังก์ชันวัตถุประสงค์ model เป็นแบบจำลองของระบบ CPP ไม่เชิงเส้น
*****/
float model(float s0_ini_0,float s0_ini_1,float s0_ini_2,float s0_ini_3,float s0_ini_4,float s0_ini_5,float s0_ini_6,float
s0_ini_7,float s0_ini_8)
{
float sq_er,s_sq_er,output; // ค่าความคลาดเคลื่อนกำลังสอง, ผลรวมค่าความคลาดเคลื่อนกำลังสอง,
// และเอาต์พุตของโปรแกรมย่อย
float k1,k2,k3,k4; // ประกาศตัวแปรเพื่อใช้สำหรับการคำนวณผลเฉลย
float m1,m2,m3,m4; // ของสมการอนุพันธ์ไม่เชิงเส้น ด้วยระเบียบวิธีรุงเง-คุตดา
float n1,n2,n3,n4;
float p1,p2,p3,p4;
float del_x1,del_x2,del_x3,del_x4;
float x,h; // ค่าเวลาเริ่มต้นการคำนวณ และช่วงเวลาชักตัวอย่าง
float x1,x2,x3,x4; //  $\phi$ ,  $\dot{\phi}$ ,  $y$ , และ  $\dot{y}$ 
float M,m,l,g,A,B,C,D,E,F,G,H; // ค่าพารามิเตอร์ของแบบจำลอง
float solu[1000]; // ค่าคำตอบของการจำลองผล
int i,iii; // ตัวแปรที่ใช้ในการนับ และวนรอบ
float ua_pen[1000],ub_pen[1000],x_pen[1000]; // อินพุต และเอาต์พุตสำหรับการจำลองผลการระบุเอกลักษณ์ระบบ
FILE *fp1,*fp2,*fp3;
fp1=fopen("ua_pen.dat","rb"); // ไฟล์ ua_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
fread(ua_pen,sizeof(float),700,fp1); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ua_pen
fclose(fp1); // ปิดไฟล์ ua_pen.dat
fp2=fopen("ub_pen.dat","rb"); // ไฟล์ ub_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
// ที่ได้รับการคำนวณค่าในช่วง เพื่อใช้คำนวณผลเฉลยตามระเบียบวิธีรุงเง-คุตดา
fread(ub_pen,sizeof(float),700,fp2); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ub_pen
fclose(fp2); // ปิดไฟล์ ub_pen.dat
fp3=fopen("x_pen.dat","rb"); // ไฟล์ x_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลเอาต์พุต (ตำแหน่งเชิงมุมของลูกตุ้ม)
// จากตัวตรวจรู้ หรือจากการประมวลผลภาพ
// หรือจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
// จากการทดสอบระบบ CPP ตอนที่ 3

```



```

fread(x_pen,sizeof(float),700,fp3); // อ่านข้อมูลเอาต์พุตมาเก็บไว้ในตัวแปร x_pen
fclose(fp3); // ปิดไฟล์ x_pen.dat
x=0; // กำหนดค่าเวลาเริ่มต้นการคำนวณเท่ากับ 0
h=0.15; // กำหนดช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
x1=0; // กำหนดค่าเริ่มต้นของ  $\phi$  เท่ากับ 0
x2=0; // กำหนดค่าเริ่มต้นของ  $\dot{\phi}$  เท่ากับ 0
x3=0; // กำหนดค่าเริ่มต้นของ  $y$  เท่ากับ 0
x4=0; // กำหนดค่าเริ่มต้นของ  $\dot{y}$  เท่ากับ 0
m=0.251; // ลูกตุ้มมีมวลเท่ากับ 0.251 กิโลกรัม
l=0.40; // ก้านของลูกตุ้มมีความยาวเท่ากับ 0.4 เมตร
g=9.81; // ความเร่งเนื่องจากแรงโน้มถ่วงเท่ากับ 9.81 เมตร/วินาที2
s_sq_er=0; // กำหนดผลรวมค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
sq_er=0; // กำหนดค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
M=s0_ini_0; // ให้ M=s0_ini_0;
A=s0_ini_1; // ให้ A=s0_ini_1;
B=s0_ini_2; // ให้ B=s0_ini_2;
C=s0_ini_3; // ให้ C=s0_ini_3;
D=s0_ini_4; // ให้ D=s0_ini_4;
E=s0_ini_5; // ให้ E=s0_ini_5;
F=s0_ini_6; // ให้ F=s0_ini_6;
G=s0_ini_7; // ให้ G=s0_ini_7;
H=s0_ini_8; // ให้ H=s0_ini_8;
for(i=100;i<=250;i++)
{
// คำนวณผลเฉลยของสมการอนุพันธ์ไม่เชิงเส้นด้วยระเบียบวิธีรุงง-คุตดา
k1=h*obj1(x2);
m1=h*obj2(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n1=h*obj3(x4);
p1=h*obj4(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k2=h*obj1(x2+m1/2);
m2=h*obj2(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n2=h*obj3(x4+p1/2);
p2=h*obj4(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k3=h*obj1(x2+m2/2);
m3=h*obj2(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
n3=h*obj3(x4+p2/2);
p3=h*obj4(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
k4=h*obj1(x2+m3);
m4=h*obj2(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
n4=h*obj3(x4+p3);
p4=h*obj4(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
del_x1=(k1+2*k2+2*k3+k4)/6;

```

```

del_x2=(m1+2*m2+2*m3+m4)/6;
del_x3=(n1+2*n2+2*n3+n4)/6;
del_x4=(p1+2*p2+2*p3+p4)/6;
solu[i]=x1; // เก็บผลเฉลยที่คำนวณได้ไว้ใน solu
x1=x1+del_x1;
x2=x2+del_x2;
x3=x3+del_x3;
x4=x4+del_x4;
x=x+h; // ปรับค่าเวลาคำนวณสำหรับรอบต่อไป
}
for(iii=100;iii<=250;iii++)
{
sq_er=(solu[iii]-x_pen[iii])*(solu[iii]-x_pen[iii]);
s_sq_er=s_sq_er+sq_er; // คำนวณค่าความคลาดเคลื่อนกำลังสอง
// และผลรวมค่าความคลาดเคลื่อนกำลังสอง
}
output=s_sq_er;
return(output);
}
/*****
// ฟังก์ชันวัตถุประสงค์ obj1
*****/
float obj1(float x2)
{
float output1; // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj1
output1=x2;
return(output1);
}
/*****
// ฟังก์ชันวัตถุประสงค์ obj2
*****/
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
float output2; // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj2
output2=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)
+G*u+H)*cos(x1)+0.5*m*1*x2*x2*sin(2*x1)+(M+m)*g*sin(x1))/(l*(m*cos(x1)*cos(x1)-(M+m)));
return(output2);
}
/*****
// ฟังก์ชันวัตถุประสงค์ obj3
*****/
float obj3(float x4)

```

```

{
    float output3;                // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj3
    output3=x4;
    return(output3);
}

/*****/

// ฟังก์ชันวัตถุประสงค์ obj4

/*****/

float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
    float output4;                // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj4
    output4=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)+G*u+H)
    +1*m*x2*x2*sin(x1)+0.5*g*m*sin(2*x1))/(M+m)-m*cos(x1)*cos(x1));
    return(output4);
}

/*****/

// โปรแกรมนี้เป็นโปรแกรมจำลองผล และตรวจสอบความถูกต้องของแบบจำลองระบบ CPP ไม่เชิงเส้น
// เมื่อพิจารณาให้แรง  $f$  ที่เป็นพหุนามอันดับ 7 ของ  $u$ 
// โปรแกรมได้รับการพัฒนาด้วยภาษา C เพื่อการจำลองผลโดยใช้ข้อมูลที่ได้จากตัวตรวจรู้
// จากการประมวลผลภาพ และจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
// พัฒนาโดยเดชา พวงดาวเรือง (2004)
// สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

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

float obj1(float x2);                // กำหนดฟังก์ชันวัตถุประสงค์ obj1
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj2
float obj3(float x4);                // กำหนดฟังก์ชันวัตถุประสงค์ obj3
float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g);
// กำหนดฟังก์ชันวัตถุประสงค์ obj4

void main()
{
    float sq_er,s_sq_er,output;        // ค่าความคลาดเคลื่อนกำลังสอง, ผลรวมค่าความคลาดเคลื่อนกำลังสอง,
// และเอาต์พุตของโปรแกรมย่อย
    float k1,k2,k3,k4;                // ประกาศตัวแปรเพื่อใช้สำหรับการคำนวณผลเฉลย
    float m1,m2,m3,m4;                // ของสมการอนุพันธ์ไม่เชิงเส้น ด้วยระเบียบวิธีรุงเง-คุตดา
    float n1,n2,n3,n4;
    float p1,p2,p3,p4;
    float del_x1,del_x2,del_x3,del_x4;

```

```

float x,h; // ค่าเวลาเริ่มต้นการคำนวณ และช่วงเวลาชักตัวอย่าง
float x1,x2,x3,x4; //  $\phi$ ,  $\dot{\phi}$ ,  $y$ , และ  $\dot{y}$ 
float M,m,l,g,A,B,C,D,E,F,G,H; // ค่าพารามิเตอร์ของแบบจำลอง
float solu[1000]; // ค่าคำตอบของการจำลองผล
int i,iii; // ตัวแปรที่ใช้ในการนับ และวนรอบ
float ua_pen[1000],ub_pen[1000],x_pen[1000]; // อินพุต และเอาต์พุตสำหรับการจำลองผลการระบุเอกลักษณ์ระบบ

FILE *fp1,*fp2,*fp3;
FILE *fp4;
fp1=fopen("ua_pen.dat","rb"); // ไฟล์ ua_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
fread(ua_pen,sizeof(float),900,fp1); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ua_pen
fclose(fp1); // ปิดไฟล์ ua_pen.dat
fp2=fopen("ub_pen.dat","rb"); // ไฟล์ ub_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลอินพุตจากตัวตรวจรู้จากการทดสอบระบบ CPP ตอนที่ 3
// ที่ได้รับการคำนวณค่าในช่วง เพื่อใช้คำนวณผลเฉลยตามระเบียบวิธีรุ่งง-คุดตา
fread(ub_pen,sizeof(float),900,fp2); // อ่านข้อมูลอินพุตมาเก็บไว้ในตัวแปร ub_pen
fclose(fp2); // ปิดไฟล์ ub_pen.dat
fp3=fopen("x_pen.dat","rb"); // ไฟล์ x_pen.dat ได้รับการเปิดเพื่ออ่านข้อมูลเข้า
// ไฟล์ดังกล่าวบรรจุข้อมูลเอาต์พุต (ตำแหน่งเชิงมุมของลูกตุ้ม)
// จากตัวตรวจรู้ หรือจากการประมวลผลภาพ
// หรือจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
// จากการทดสอบระบบ CPP ตอนที่ 3
fread(x_pen,sizeof(float),900,fp3); // อ่านข้อมูลเอาต์พุตมาเก็บไว้ในตัวแปร x_pen
fclose(fp3); // ปิดไฟล์ x_pen.dat
x=0; // กำหนดค่าเวลาเริ่มต้นการคำนวณเท่ากับ 0
h=0.15; // กำหนดช่วงเวลาชักตัวอย่างเท่ากับ 0.15 วินาที
x1=0; // กำหนดค่าเริ่มต้นของ  $\phi$  เท่ากับ 0
x2=0; // กำหนดค่าเริ่มต้นของ  $\dot{\phi}$  เท่ากับ 0
x3=0; // กำหนดค่าเริ่มต้นของ  $y$  เท่ากับ 0
x4=0; // กำหนดค่าเริ่มต้นของ  $\dot{y}$  เท่ากับ 0
m=0.251; // ลูกตุ้มมีมวลเท่ากับ 0.251 กิโลกรัม
l=0.40; // ก้านของลูกตุ้มมีความยาวเท่ากับ 0.4 เมตร
g=9.81; // ความเร่งเนื่องจากแรงโน้มถ่วงเท่ากับ 9.81 เมตร/วินาที2
s_sq_er=0; // กำหนดผลรวมค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
sq_er=0; // กำหนดค่าความคลาดเคลื่อนกำลังสองเท่ากับ 0
M=XXXX;
A= XXXX;
B= XXXX;
C= XXXX;
D= XXXX;
E= XXXX;

```

```

F= XXXX;
G= XXXX;
H= XXXX;                                // ค่า XXXX ของพารามิเตอร์ M, A, B,..., H
                                           // เป็นค่าที่ได้จากการระบุเอกลักษณ์ด้วยการค้นหาแบบตามเชิงปรับตัว
                                           // จากโปรแกรมระบุเอกลักษณ์ระบบ CPP ไม่เชิงเส้น
                                           // เมื่อพิจารณาให้แรง  $f$  ที่เป็นพหุนามอันดับ 7 ของ  $u$ 
                                           // โดยใช้ข้อมูลจากตัวตรวจรู้ หรือจากการประมวลผลภาพ
                                           // หรือจากการประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน
                                           // ดังที่ได้แสดงไว้แล้วข้างต้น

for(i=100;i<=300;i++)
{
    // คำนวณผลเฉลยของสมการอนุพันธ์ไม่เชิงเส้นด้วยระเบียบวิธีรุง-กูดตา
    k1=h*obj1(x2);
    m1=h*obj2(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
    n1=h*obj3(x4);
    p1=h*obj4(x1,x2,ua_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
    k2=h*obj1(x2+m1/2);
    m2=h*obj2(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
    n2=h*obj3(x4+p1/2);
    p2=h*obj4(x1+k1/2,x2+m1/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
    k3=h*obj1(x2+m2/2);
    m3=h*obj2(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
    n3=h*obj3(x4+p2/2);
    p3=h*obj4(x1+k2/2,x2+m2/2,ub_pen[i],A,B,C,D,E,F,G,H,M,m,l,g);
    k4=h*obj1(x2+m3);
    m4=h*obj2(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
    n4=h*obj3(x4+p3);
    p4=h*obj4(x1+k3,x2+m3,ua_pen[i+1],A,B,C,D,E,F,G,H,M,m,l,g);
    del_x1=(k1+2*k2+2*k3+k4)/6;
    del_x2=(m1+2*m2+2*m3+m4)/6;
    del_x3=(n1+2*n2+2*n3+n4)/6;
    del_x4=(p1+2*p2+2*p3+p4)/6;
    solu[i]=x1;                            // เก็บผลเฉลยที่คำนวณได้ไว้ใน solu
    fp4=fopen("sim_sensor_order7.dat","wb+");
                                           // ทำการเปิดไฟล์ sim_sensor_order7.dat
                                           // เพื่อเก็บผลการจำลอง และตรวจสอบความถูกต้องของแบบจำลองระบบ
                                           // CPP ไม่เชิงเส้น เมื่อแรง  $f$  ที่เป็นพหุนามอันดับ 7 ของ  $u$ 
                                           // โดยใช้ข้อมูลจากตัวตรวจรู้
                                           // สำหรับไฟล์ที่ใช้เก็บผลการจำลอง และตรวจสอบความถูกต้อง
                                           // ของแบบจำลองระบบ CPP ไม่เชิงเส้น เมื่อแรง  $f$  ที่เป็นพหุนามอันดับ
                                           // 7 ของ  $u$  โดยใช้ข้อมูลจากการประมวลผลภาพ และข้อมูลจากการ
                                           // ประมวลผลภาพที่ได้รับการแก้ไขความคลาดเคลื่อน

```

```

// คือไฟล์ sim_image_order7.dat และไฟล์ sim_new_image_order7.dat
// ตามลำดับ ไฟล์ดังกล่าวข้างต้น จะได้รับการอ่านและแสดงผล
// ด้วยโปรแกรม MATLAB ต่อไป ซึ่งจะไม่ขอแสดงรายละเอียดในที่นี้
fwrite(solu,sizeof(float),i,fp4); // ทำการเขียนผลการจำลอง และตรวจสอบความถูกต้องลงไฟล์
fclose(fp4); // ปิดไฟล์
x1=x1+del_x1;
x2=x2+del_x2;
x3=x3+del_x3;
x4=x4+del_x4;
x=x+h; // ปรับค่าเวลาคำนวณสำหรับรอบต่อไป
}
for(iii=100;iii<=300;iii++)
{
sq_er=(solu[iii]-x_pen[iii])*(solu[iii]-x_pen[iii]);
s_sq_er=s_sq_er+sq_er; // คำนวณค่าความคลาดเคลื่อนกำลังสอง
// และผลรวมค่าความคลาดเคลื่อนกำลังสอง
}
getch();
}
/*****
// ฟังก์ชันวัตถุประสงค์ obj1
*****/
float obj1(float x2)
{
float output1; // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj1
output1=x2;
return(output1);
}
/*****
// ฟังก์ชันวัตถุประสงค์ obj2
*****/
float obj2(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
float output2; // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj2
output2=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)
+G*u+H)*cos(x1)+0.5*m*1*x2*x2*sin(2*x1)+(M+m)*g*sin(x1))/(l*(m*cos(x1)*cos(x1)-(M+m)));
return(output2);
}
/*****
// ฟังก์ชันวัตถุประสงค์ obj3
*****/
float obj3(float x4)

```

```

{
    float output3;                // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj3
    output3=x4;
    return(output3);
}
/*****/
// ฟังก์ชันวัตถุประสงค์ obj4
/*****/
float obj4(float x1,float x2,float u,float A,float B,float C,float D,float E,float F,float G,float H,float M,float m,float l,float g)
{
    float output4;                // ประกาศตัวแปรเอาต์พุตของฟังก์ชันวัตถุประสงค์ obj4
    output4=((A*pow(u,7)+B*pow(u,6)+C*pow(u,5)+D*pow(u,4)+E*pow(u,3)+F*pow(u,2)+G*u+H)
    +l*m*x2*x2*sin(x1)+0.5*g*m*sin(2*x1))/((M+m)-m*cos(x1)*cos(x1));
    return(output4);
}
/*****/

```

ภาคผนวก จ

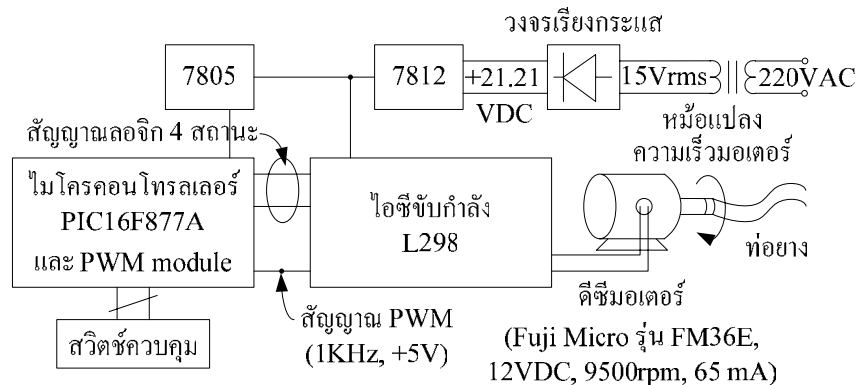
การควบคุมระบบ VTS

การควบคุมระบบ VTS

การควบคุมระบบ VTS ในงานวิจัยนี้ เป็นการควบคุมแบบวงรอบเปิด สัญญาณอินพุตที่ใช้ในการควบคุมระบบ เป็นสัญญาณทวิภาคสี่เหลี่ยมที่ผลิตจากไมโครคอนโทรลเลอร์ สัญญาณดังกล่าวจะได้รับการขับเร้าเพื่อเพิ่มกำลังด้วยวงจรขับกำลัง (power driver) และถูกส่งไปขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง แรงจากเพลลาของมอเตอร์จะส่งผ่านไปตามท่อยาง ทำให้ท่อยางเกิดการเคลื่อนที่แบบ 3 มิติขึ้น โครงสร้างและชุดควบคุมระบบ VTS ในงานวิจัยนี้แสดงดังรูปที่ จ.1 และ จ.2 ตามลำดับ ขณะที่แผนภาพวงจรควบคุมระบบ VTS แสดงในรูปที่ จ.3

รายละเอียดของอุปกรณ์หลักที่ใช้ควบคุมระบบ VTS มีดังต่อไปนี้

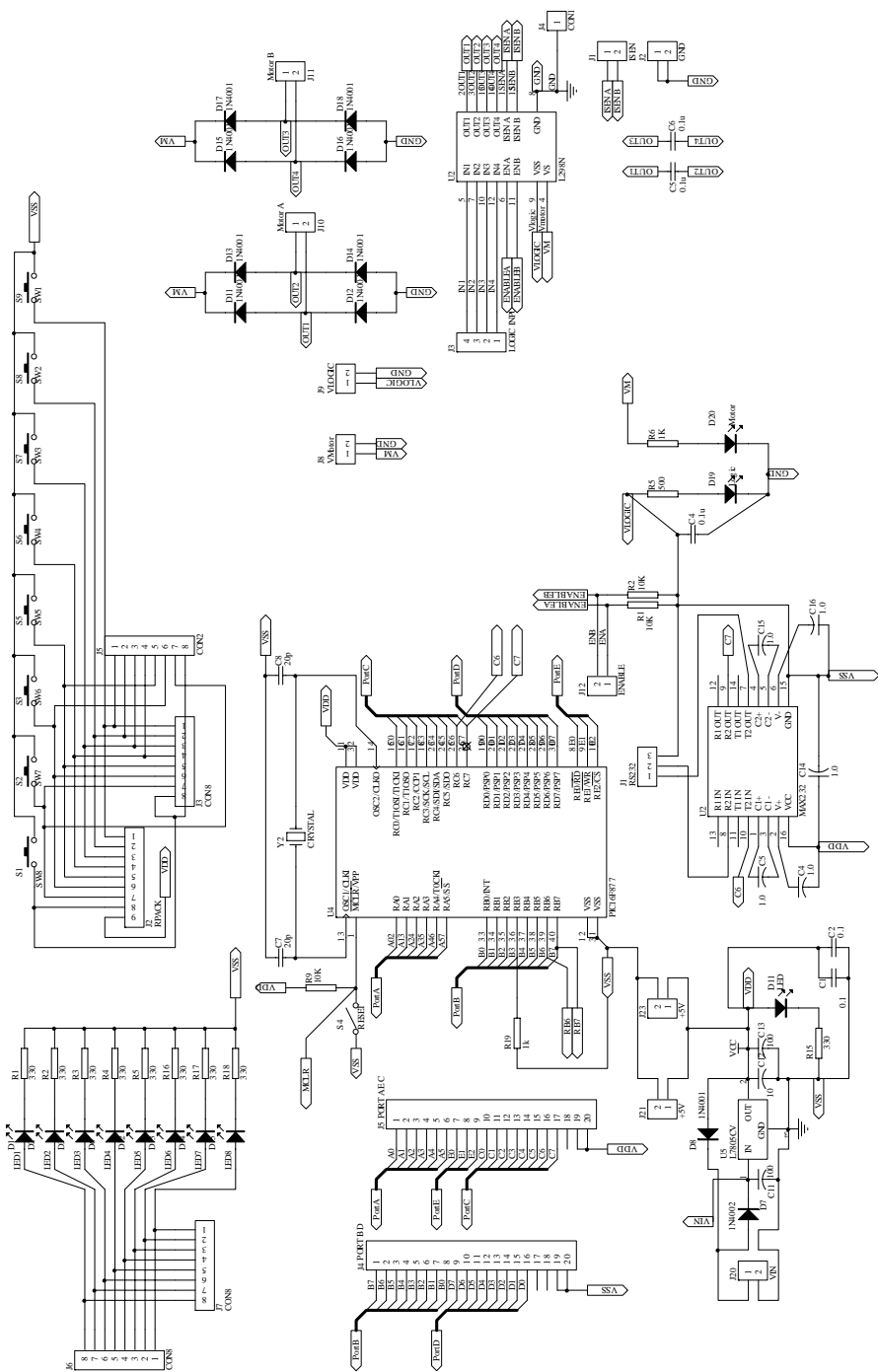
1. ไมโครคอนโทรลเลอร์ PIC16F877A มีขนาดความแยกชัด 8 บิต ทำงานบนอัตราสัญญาณนาฬิกา 20 เมกกะเฮิร์ตซ์ ไมโครคอนโทรลเลอร์ตัวนี้ได้รับการพัฒนาขึ้นเพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงโดยเฉพาะ และมีโมดูล PWM อยู่ภายใน



รูปที่ จ.1 โครงสร้างการควบคุมระบบ VTS



รูปที่ จ.2 ชุดควบคุมระบบ VTS



2. ไอซีขับกำลัง L298 ทำหน้าที่เป็นวงจรขับกำลังสัญญาณ PWM
3. มอเตอร์ไฟฟ้ากระแสตรง ของบริษัท Fuji Micro รุ่น FM36E ขนาด 9,500 รอบต่อนาที 12 โวลต์ 65 มิลลิแอมแปร์

สำหรับซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรมมีดังนี้

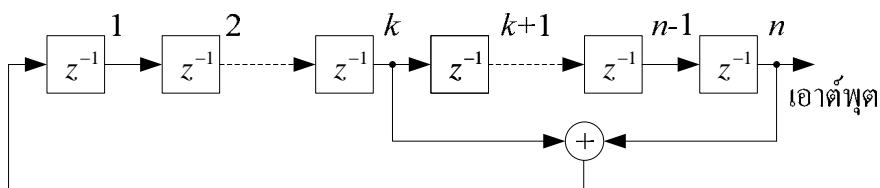
1. ดีบั๊กเกอร์ Micro Code Studio
2. ดาวนโหลดเดอร์ Epic Win (2.44 Beta)
3. แอสเซมเบลเลอร์ หรือคอมไพเลอร์ MP-Lab (Micro Chip) หรือ PIC-Basic Pro

หลักการควบคุมระบบ VTS เริ่มต้นจากการประดิษฐ์โปรแกรมภาษาเบสิกของไมโครคอนโทรลเลอร์ PIC16F877A ด้วยดีบั๊กเกอร์ เพื่อผลิตค่าสัญญาณทวิภาคสุมเทียม จากนั้นทำการแปลงโปรแกรมหากกล่าวไปเป็นภาษาเครื่องด้วยแอสเซมเบลเลอร์ และทำการดาวนโหลดลงสู่ไมโครคอนโทรลเลอร์ด้วยดาวนโหลดเดอร์ จากโปรแกรมที่พัฒนาขึ้น ไมโครคอนโทรลเลอร์ PIC16F877A จะมีสัญญาณเอาต์พุต 2 สัญญาณคือ สัญญาณลอจิก 4 สถานะ และสัญญาณ PWM (ดูรูปที่ จ.1 ประกอบ) สัญญาณลอจิก 4 สถานะจะเป็นสัญญาณที่กำหนดให้มอเตอร์หมุนซ้าย หมุนขวา หยุด และเบรก ส่วนสัญญาณ PWM มีขนาด 5 โวลต์ 1 กิโลเฮิร์ตซ์ ค่า duty cycle ของสัญญาณ PWM จะแปรเปลี่ยนไปตามค่าทวิภาคสุมเทียมที่ไมโครคอนโทรลเลอร์ผลิตขึ้น สัญญาณดังกล่าวจะถูกส่งไปยังไอซี L298 เพื่อเพิ่มกำลัง มอเตอร์ไฟฟ้ากระแสตรงจะถูกขับเคลื่อนด้วยสัญญาณกำลังจากไอซี L298 แรงจากเพลลาของมอเตอร์จะส่งผ่านไปทางล้อ และทำให้ล้อเกิดการเคลื่อนที่แบบ 3 มิติขึ้น

ดังได้กล่าวมาแล้วข้างต้น สัญญาณอินพุตที่ใช้ในการควบคุมระบบ VTS เป็นสัญญาณทวิภาคสุมเทียม สัญญาณดังกล่าวมีลักษณะเป็นลำดับพัลส์สองระดับที่ผ่านการมอดูเลตเชิงความกว้าง และมีลักษณะเป็นรายคาบที่ยาวนานมาก การสร้างสัญญาณทวิภาคสุมเทียมมักใช้หลักการของรีจิสเตอร์เลื่อนที่มีการป้อนกลับ ควบคู่กับพีชคณิตมอดูโล (Modulo algebra) ดังปรากฏในตำราการระบุเอกลักษณ์ระบบทั่วไป อาทิเช่น Davies (1970), Eykhoff (1974), Landau (1990), และ Johansson (1993) เป็นต้น วงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับแสดงดังรูปที่ จ.4

จากรูปที่ จ.4 ค่า n คือ จำนวนรีจิสเตอร์เลื่อนทั้งหมด k คือ จำนวนรีจิสเตอร์เลื่อนก่อนการป้อนกลับครั้งแรก และ \oplus คือ สัญลักษณ์การปฏิบัติการของพีชคณิตมอดูโล ซึ่งมีความสัมพันธ์ดังสมการที่ (จ-1)

$$x \oplus y = \begin{cases} 0, & \text{if } x = y \\ 1, & \text{if } x \neq y \end{cases} \quad (\text{จ-1})$$



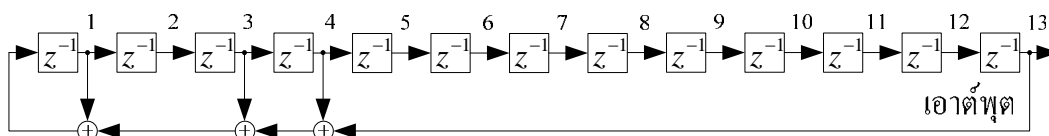
รูปที่ จ.4 วงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับ

ปฏิบัติการของพีชคณิตมอดูโลจะเป็นการเปรียบเทียบค่าสัญญาณ x และสัญญาณ y กล่าวคือ ถ้าสัญญาณ x และ y มีค่าเท่ากัน ผลของพีชคณิตมอดูโลจะให้ค่าเป็น 0 และถ้าสัญญาณ x และ y มีค่าไม่เท่ากัน ผลของพีชคณิตมอดูโลจะให้ค่าเป็น 1

คาบเวลาของสัญญาณทวิภาคสุ่มเทียมที่ได้จากวงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับ ควบคุมกับไปกับพีชคณิตมอดูโล แสดงดังสมการที่ (จ-2)

$$N = (2^n - 1)h \tag{จ-2}$$

เมื่อ N คือ คาบเวลาของสัญญาณทวิภาคสุ่มเทียม และ h คือ ช่วงเวลาประวิงของรีจิสเตอร์เลื่อน จะเห็นได้ว่า คาบเวลาของสัญญาณทวิภาคสุ่มเทียมจะแปรผันตรงกับจำนวนรีจิสเตอร์เลื่อนที่ใช้



รูปที่ จ.5 วงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับจำนวน 13 ตัว

ในงานวิจัยนี้เลือกใช้โครงสร้างวงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับ เพื่อสร้างสัญญาณทวิภาคสุ่มเทียม ดังแสดงในรูปที่ จ.5 (Johansson, 1993) วงจรดังกล่าวมีรีจิสเตอร์เลื่อนจำนวน 13 ตัว ($n = 13$) การป้อนกลับเกิดขึ้นที่รีจิสเตอร์ตัวที่ 1, 3, 4, และ 13 ($k = 1$) จากวงจรรีจิสเตอร์เลื่อนที่มีการป้อนกลับดังรูปที่ จ.5 สามารถเขียนอธิบายในรูปของพหุนาม z ได้ดังสมการที่ (จ-3)

$$P(z^{-1}) = 1 \oplus z^{-1} \oplus z^{-3} \oplus z^{-4} \oplus z^{-13} \tag{จ-3}$$

เมื่อเทอม z^{-1} คือ ช่วงเวลาประวิงของรีจิสเตอร์เลื่อน

จากรูปที่ จ.5 พบว่าสัญญาณทวิภาคสุ่มเทียมจะมีคาบเวลาเป็นดังสมการที่ (จ-4)

$$N = (2^{13} - 1)h = 8191h \quad (\text{จ-4})$$

รายละเอียดโปรแกรมภาษาเบสิกของไมโครคอนโทรลเลอร์ PIC16F877A ที่พัฒนาขึ้น เพื่อผลิตค่าสัญญาณทวิภาคสุ่มเทียม และเพื่อควบคุมระบบ VTS แสดงดังต่อไปนี้

- โปรแกรมภาษาเบสิกของไมโครคอนโทรลเลอร์ PIC16F877A
- ที่พัฒนาขึ้นเพื่อผลิตค่าสัญญาณทวิภาคสุ่มเทียม และเพื่อควบคุมระบบ VTS
- พัฒนาโดยเฉชา พวงดาวเรือง (2004)
- สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

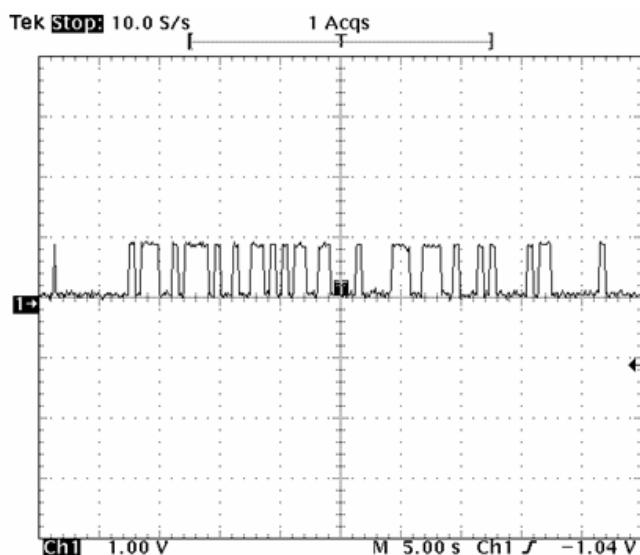
```

define osc 20          ' กำหนดค่าความถี่ของสัญญาณนาฬิกาหลัก 20 เมกกะเฮิร์ตซ์
define ccp1_reg portc ' กำหนดคอมดูล PWM ให้อยู่ที่พอร์ต C และอยู่ที่
define ccp1_bit 2     ' ตำแหน่งบิตที่ 2
value_rand var word   ' กำหนดตัวแปร value_rand ขนาด 16 บิต รับค่าทวิภาคสุ่มเทียม
t1 var bit            ' กำหนดตัวแปรชั่วคราว t1 ขนาด 1 บิต
t2 var bit            ' กำหนดตัวแปรชั่วคราว t2 ขนาด 1 บิต
speed var byte       ' กำหนดตัวแปร speed ขนาด 8 บิต เพื่อเก็บค่าความเร็วมอเตอร์
trisd=0              ' กำหนดให้ทุกบิตของพอร์ต D เป็นพอร์ตเอาต์พุต
portd=0              ' กำหนดให้พอร์ต D แต่ละบิตมีค่าเท่ากับ 0 (มอเตอร์หยุดหมุน)
trisc=128            ' กำหนดให้บิตที่ 7 ของพอร์ต C เป็นพอร์ตอินพุต (1)
                    ' และกำหนดให้บิตที่เหลือของพอร์ต C เป็นพอร์ตเอาต์พุต (0)
trisb=255            ' กำหนดให้ทุกบิตของพอร์ต B เป็นพอร์ตอินพุต
value_rand=65535    ' ให้ตัวแปร value_rand มีค่าเท่ากับ 65535 (มีค่า 1 ทั้ง 16 บิต)
main:
    if portb.0 = 0 then ' ถ้า SW1 ที่ต่ออยู่กับพอร์ต B บิต 0 ได้รับการกด
        pause 1000    ' ตรวจสอบการกด SW1 โดยการหน่วงเวลา 1,000 มิลลิวินาที
        goto work     ' กระโดดไปทำงานที่ลาเบล work
    endif
    goto main         ' กระโดดไปทำงานที่ลาเบล main
speed=50              ' กำหนดความเร็วมอเตอร์เริ่มต้นเป็น 420 รอบต่อนาที
work:
    t1=value_rand.12 xor value_rand.3 ' ผลิตค่าสัญญาณทวิภาคสุ่มเทียมตามสมการที่ (จ-3)
    ' โดยที่ปฏิบัติการของพีชคณิตมอดูโล จะใช้การ
    ' จำนวนแบบลอจิกเกต XOR
    ' value_rand บิตที่ 13 XOR กับ value_rand บิตที่ 4
    ' และเก็บผลลัพธ์ไว้ใน t1
    t2=t1 xor value_rand.2 ' t1 XOR กับ value_rand บิตที่ 3 และเก็บผลลัพธ์ไว้ใน t2

```

value_rand.0=t2 xor value_rand.0	‘ t2 XOR กับ value_rand บิตที่ 1 และเก็บผลลัพธ์ไว้ใน value_rand บิตที่ 1
value_rand=value_rand<<1	‘ เลื่อนทุกบิตใน value_rand ไปทางซ้าย 1 ครั้ง
if value_rand.0 = 1 then	‘ ถ้า value_rand บิตที่ 1 มีค่าเป็น 1 แล้ว
gsub forward	‘ กระโดดไปทำงานที่ลาเบล forward
else	‘ ไม่เช่นนั้นแล้ว
gsub backward	‘ กระโดดไปทำงานที่ลาเบล backward
endif	
if portb.1 = 0 then	‘ ถ้า SW2 ที่ต่ออยู่กับพอร์ต B บิต 1 ได้รับการกด
gsub incPWM	‘ กระโดดไปทำงานที่ลาเบล incPWM
endif	
if portb.2 = 0 then	‘ ถ้า SW3 ที่ต่ออยู่กับพอร์ต B บิต 2 ได้รับการกด
gsub decPWM	‘ กระโดดไปทำงานที่ลาเบล decPWM
endif	
pause 200	‘ หน่วงเวลา 200 มิลลิวินาที
if portb.0 = 0 then	‘ ถ้า SW1 ที่ต่ออยู่กับพอร์ต B บิต 0 ได้รับการกด
portd=0	‘ กำหนดให้พอร์ต D แต่ละบิตมีค่าเท่ากับ 0 (มอเตอร์หยุดหมุน)
pause 1000	‘ ตรวจสอบการกด SW1 โดยการหน่วงเวลา 1,000 มิลลิวินาที
goto main	‘ กระโดดไปทำงานที่ลาเบล main
endif	
goto work	‘ กระโดดไปทำงานที่ลาเบล work
incPWM:	
speed = speed + 2	‘ เพิ่มความเร็วมอเตอร์
if speed>250 then	‘ ถ้าความเร็วมอเตอร์มีค่ามากกว่าความเร็วมอเตอร์สูงสุด
speed=250	‘ ให้ความเร็วมอเตอร์มีค่าเท่ากับความเร็วมอเตอร์สูงสุด
endif	
HPWM 1,speed,1000	‘ กำหนดให้มอดูล PWM มีความถี่เท่ากับ 1,000 เฮิรตซ์
Return	‘ ที่ความเร็วปัจจุบัน
decPWM:	
speed = speed – 2	‘ ลดความเร็วมอเตอร์
if speed<5 then	‘ ถ้าความเร็วมอเตอร์มีค่าน้อยกว่าความเร็วขณะหยุดหมุน
speed=5	‘ ให้มอเตอร์หยุดหมุน
endif	
HPWM 1,speed,1000	‘ กำหนดให้มอดูล PWM มีความถี่เท่ากับ 1,000 เฮิรตซ์
Return	‘ ที่ความเร็วปัจจุบัน
forward:	
portd=%01010101	‘ กำหนดให้มอเตอร์หมุนตามเข็มนาฬิกา
return	
backward:	
portd=%10101010	‘ กำหนดให้มอเตอร์หมุนทวนเข็มนาฬิกา
return	

เนื่องจากโปรแกรมที่พัฒนาขึ้นมีลักษณะการทำงานแบบวนรอบ แต่ละรอบของโปรแกรม จะใช้เวลาในการเอ็ชชีควิต์ (ซึ่งก็คือค่า h) ประมาณ 800 มิลลิวินาที ดังนั้นคาบเวลาของสัญญาณ ทวิภาคสุม่เทียม (N) ที่ผลิตได้จะมีค่าประมาณ 1.82 ชั่วโมง สัญญาณทวิภาคสุม่เทียมที่ผลิตขึ้น ได้รับการบันทึกโดยออสซิลโลสโคปแบบดิจิตอล ดังแสดงในรูปที่ จ.6



รูปที่ จ.6 สัญญาณทวิภาคสุม่เทียมสำหรับควบคุมระบบ VTS

ภาคผนวก จ

การเปรียบเทียบกล้องวิดีโอดิจิทัล

การปรับเทียบกล้องวิดีโอดิจิทัล

การปรับเทียบกล้อง (camera calibration) เป็นกระบวนการวินิจฉัยลักษณะสมบัติภายในของกล้อง และความสัมพันธ์ระหว่างตำแหน่งและท่าทางของกล้องที่กระทำกับวัตถุ (Heikkila and Silven, 1997; Bakstein, 1999) การปรับเทียบกล้องจะให้ความสัมพันธ์ดังกล่าวออกมาในรูปของแบบจำลองกล้อง (camera model) เมื่อนำกล้องไปใช้งาน ภาพทุก ๆ ภาพที่ได้จากกล้องบันทึกภาพจะมีความเพี้ยน (distortion) และจำเป็นต้องได้รับการแก้ไขความเพี้ยนที่เกิดขึ้นในภาพ ก่อนที่จะนำภาพไปทำการประมวลผล โดยอาศัยข้อมูลความรู้ที่ปรากฏในแบบจำลองกล้องซึ่งได้มาจากการปรับเทียบกล้องนั่นเอง ดังนั้นในภาคผนวก จ. นี้ จะกล่าวถึงการปรับเทียบกล้องวิดีโอดิจิทัลที่ใช้บันทึกภาพการเคลื่อนที่ของระบบกวัดแกว่งท่ออย่าง หรือระบบ VTS โดยเริ่มต้นจากรูปแบบความเพี้ยน แบบจำลองกล้อง เทคนิคการปรับเทียบกล้อง และผลการปรับเทียบกล้องวิดีโอดิจิทัลที่ใช้ในงานวิจัยนี้

ความเพี้ยน

ความเพี้ยนของภาพที่ได้จากกล้องบันทึกภาพ อาจเกิดจากหลายสาเหตุ เช่น ความเพี้ยนของเลนส์ การผลิตและการติดตั้งตัวตรวจรู้ของกล้องที่ไม่เหมาะสม อุณหภูมิในการใช้งานกล้อง และการสั่นสะเทือนขณะใช้งาน (Nalwa, 1993) อย่างไรก็ตาม สาเหตุหลักที่ก่อให้เกิดความเพี้ยนขึ้นในภาพคือความเพี้ยนของเลนส์

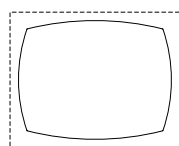
ความเพี้ยนของเลนส์ (lens distortion) เป็นความเพี้ยนเชิงเรขาคณิต (geometrical distortion) แบบไม่เชิงเส้น แบ่งออกได้เป็น 4 ลักษณะคือ ความเพี้ยนเชิงรัศมี (radial distortion) ความเพี้ยนเชิงผิวสัมผัส (tangential distortion) ความเพี้ยนเยื้องศูนย์กลาง (decentering distortion) และความเพี้ยนปริซึมบาง (thin prism distortion)

ความเพี้ยนเชิงรัศมี จะทำให้จุดวัตถุที่ปรากฏบนระนาบภาพ (image plane) เลื่อนไปจากจุดศูนย์กลางภาพอย่างสมมาตรเชิงวงกลม เมื่อจุดวัตถุเลื่อนเข้าหาจุดศูนย์กลางภาพ จะทำให้เกิดความเพี้ยนรูปถัง (barrel distortion) และเมื่อจุดวัตถุเลื่อนออกจากจุดศูนย์กลางภาพ จะทำให้เกิดความเพี้ยนรูปหมอนปีกเข็ม (pincushion distortion) ดังแสดงในรูปที่ จ.1 (ก) และ (ข) ตามลำดับ

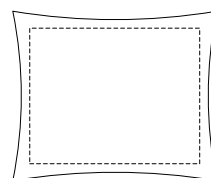
ความเพี้ยนเชิงผิวสัมผัส จะทำให้จุดวัตถุที่ปรากฏบนระนาบภาพ เลื่อนไปจากจุดวัตถุจริง การเลื่อนของจุดวัตถุดังกล่าวเป็นการเลื่อนเชิงผิวสัมผัส

ความเพี้ยนเชิงศูนย์กลาง เป็นผลของการเกิดความเพี้ยนเชิงรัศมี และความเพี้ยนเชิงผิวสัมผัส มีสาเหตุมาจากการที่จุดศูนย์กลางของเลนส์ซึ่งมีพื้นผิวโค้ง ไม่มีความเป็นเชิงเส้นร่วมแนว (collinear) กันอย่างแท้จริง ความเพี้ยนเชิงศูนย์กลางแสดงดังรูปที่ ๑.๒ โดยที่ Δr คือความเพี้ยนเชิงรัศมี และ Δt คือความเพี้ยนเชิงผิวสัมผัส จะสังเกตเห็นได้ว่าจุดวัตถุจะเลื่อนตำแหน่งไปเป็นจุดภาพที่ปรากฏบนระนาบภาพ ทำให้จุดศูนย์กลางภาพปรากฏการเยื้องศูนย์กลางเกิดขึ้น นอกจากนี้ ความเพี้ยนเชิงศูนย์กลางอาจมีสาเหตุมาจากจุดศูนย์กลางของชิ้นเลนส์ต่าง ๆ ที่ประกอบในตัวเลนส์ ไม่มีความเป็นเชิงเส้นร่วมแนวกันอย่างแท้จริง

ความเพี้ยนปริซึมบาง เป็นความเพี้ยนที่เกิดจากการขั้นตอนการออกแบบ การผลิต และการประกอบเลนส์ที่ไม่สมบูรณ์ อย่างไรก็ตาม ความเพี้ยนดังกล่าวส่งผลต่อความเพี้ยนโดยรวมน้อยมาก



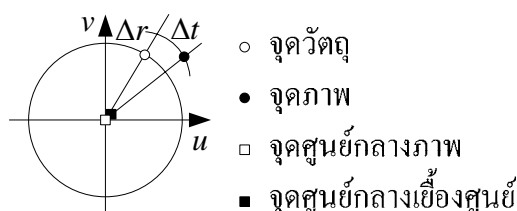
(ก)



(ข)

รูปที่ ๑.๑ ความเพี้ยนเชิงรัศมี

(ก) ความเพี้ยนรูปถัง (ข) ความเพี้ยนรูปหมอนปีกเข็ม

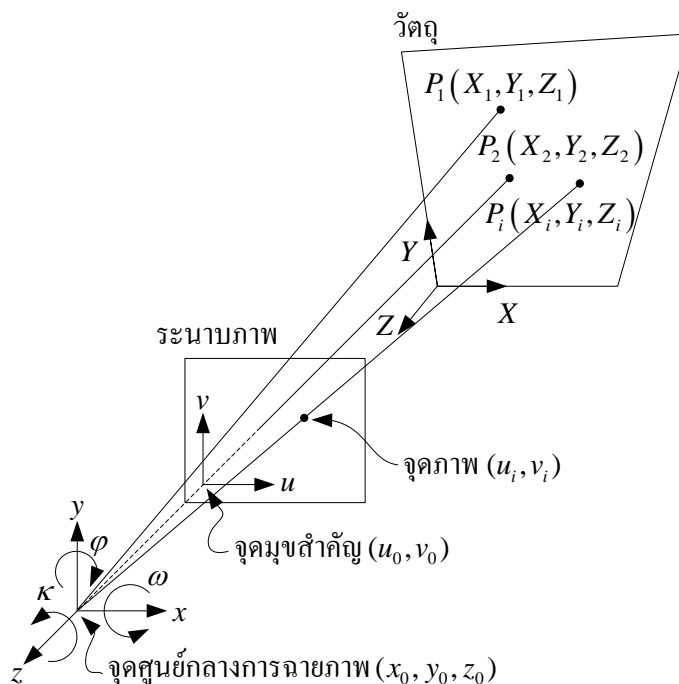


รูปที่ ๑.๒ ความเพี้ยนเชิงศูนย์กลาง

แบบจำลองกล้อง

แบบจำลองกล้อง เป็นแบบจำลองที่อธิบายลักษณะสมบัติทางกายภาพภายในกล้อง ซึ่งแสดงด้วยค่าพารามิเตอร์แบบอินทรินซิก (intrinsic parameter) และความสัมพันธ์ระหว่างตำแหน่งและท่าทางของกล้องที่กระทำกับวัตถุ ซึ่งแสดงด้วยค่าพารามิเตอร์แบบเอ็กทรินซิก (extrinsic parameter) โดยทั่วไป การพัฒนาแบบจำลองกล้องจะอาศัยแบบจำลองรูเข็ม (pinhole model) ที่แสดงการหมุน (rotation) และการเลื่อนขนาน (translation) ของกล้องตามท่าทางการรับภาพวัตถุ แบบจำลองรูเข็มตั้งอยู่บนพื้นฐานของความเป็นเชิงเส้นร่วมแนว กล่าวคือ จุดวัตถุในปริภูมิ

จะได้รับการฉายภาพโดยเส้นตรง ไปยังจุดศูนย์กลางการฉายภาพ (project center) ผ่านระนาบภาพ ดังแสดงในรูปที่ ๓.3 จุดกำเนิด (origin) ของระบบพิกัดกล้องจะอยู่ที่จุดศูนย์กลางการฉายภาพ ที่ตำแหน่ง (x_0, y_0, z_0) แกน z ของระบบพิกัดกล้องจะตั้งฉากกับระนาบภาพ และการหมุนของกล้องถูกกำกับโดยมุมของออยเลอร์ (Euler angle) ω , φ , และ κ ที่กระทำในทิศทางตามเข็มนาฬิกากับแกน x , y , และ z ของกล้องตามลำดับ



รูปที่ ๓.3 เรขาคณิตการฉายภาพ

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{๓-1}$$

$$\left. \begin{aligned} m_{11} &= \cos \varphi \cos \kappa \\ m_{12} &= \sin \omega \sin \varphi \cos \kappa - \cos \omega \sin \kappa \\ m_{13} &= \cos \omega \sin \varphi \cos \kappa + \sin \omega \sin \kappa \\ m_{21} &= \cos \varphi \sin \kappa \\ m_{22} &= \sin \omega \sin \varphi \sin \kappa + \cos \omega \cos \kappa \\ m_{23} &= \cos \omega \sin \varphi \sin \kappa - \sin \omega \cos \kappa \\ m_{31} &= -\sin \varphi \\ m_{32} &= \sin \omega \cos \varphi \\ m_{33} &= \cos \omega \cos \varphi \end{aligned} \right\} \tag{๓-2}$$

การฉายภาพจุดวัตถุ P_i ที่ตำแหน่ง (X_i, Y_i, Z_i) ในปริภูมิลงบนระนาบภาพ จำเป็นต้องทำการแปลงจุดวัตถุ P_i ให้ไปอยู่ในระบบพิกัดกล้อง (x_i, y_i, z_i) ก่อน การแปลงจะประกอบด้วย การหมุนและการเลื่อนขนาน ซึ่งเป็นค่าพารามิเตอร์แบบเอ็ทรินซิกของแบบจำลองกล้อง ดังแสดงในสมการที่ (ฉ-1) องค์ประกอบ m_{11}, \dots, m_{33} ในสมการที่ (ฉ-1) แสดงดังสมการที่ (ฉ-2)

จากรูปที่ ฉ.3 จุดमुखสำคัญ (principal point) คือจุดภาพ (u_0, v_0) บนระนาบภาพที่ได้รับการฉายภาพจากจุดวัตถุเป็นเส้นตรงที่ตั้งฉากกับระนาบภาพ จุดमुखสำคัญนี้คือจุดศูนย์กลางภาพ (image center) นั่นเอง

ค่าพารามิเตอร์แบบอินทรินซิก ประกอบด้วยความยาวโฟกัส (f) ตัวประกอบสเกล (s_u) และจุดศูนย์กลางภาพ (u_0, v_0) โดยใช้แบบจำลองรูเข็ม จุดวัตถุ P_i ที่ปรากฏอยู่ในระบบพิกัดกล้อง (x_i, y_i, z_i) จะได้รับการฉายภาพเพื่อแสดงค่าในระบบพิกัดภาพ ดังแสดงในสมการที่ (ฉ-3)

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{f}{z} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (\text{ฉ-3})$$

ในทางปฏิบัติโดยทั่วไปของการประมวลผลภาพ จะกำหนดให้จุดกำเนิดของระบบพิกัดภาพ อยู่ที่ตำแหน่งมุมบนด้านซ้ายมือของภาพ ดังนั้นค่าในระบบพิกัดภาพในสมการที่ (ฉ-3) จะได้รับการแปลงให้มีหน่วยเป็นจุดภาพ (pixel) และมีจุดกำเนิดอยู่ที่ตำแหน่งมุมบนด้านซ้ายมือของภาพ ดังสมการที่ (ฉ-4) เมื่อ D_u และ D_v คือค่าสัมประสิทธิ์การแปลงระบบพิกัดภาพ

$$\begin{bmatrix} u'_i \\ v'_i \end{bmatrix} = \begin{bmatrix} D_u s_u \tilde{u}_i \\ D_v \tilde{v}_i \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad (\text{ฉ-4})$$

เนื่องจากแบบจำลองรูเข็มไม่ได้อธิบายถึงความเพี้ยนที่เกิดขึ้นในจุดภาพ ดังนั้นการแก้ไขความเพี้ยนเชิงรัศมี และความเพี้ยนเชิงผิวสัมผัส จึงได้รับการพิจารณาเพิ่มเติมดังนี้

ความเพี้ยนเชิงรัศมีสามารถแสดงได้ดังสมการที่ (ฉ-5) ค่า k_1, k_2, \dots คือค่าสัมประสิทธิ์ความเพี้ยนเชิงรัศมี โดยที่ค่า $r_i = \sqrt{(\tilde{u}_i)^2 + (\tilde{v}_i)^2}$

$$\begin{bmatrix} \delta u_i^{(r)} \\ \delta v_i^{(r)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_i (k_1 r_i^2 + k_2 r_i^4 + \dots) \\ \tilde{v}_i (k_1 r_i^2 + k_2 r_i^4 + \dots) \end{bmatrix} \quad (\text{ฉ-5})$$

$$\begin{bmatrix} \delta u_i^{(t)} \\ \delta v_i^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1 \tilde{u}_i \tilde{v}_i + p_2 (r_i^2 + 2\tilde{u}_i^2) \\ p_1 (r_i^2 + 2\tilde{v}_i^2) + 2p_2 \tilde{u}_i \tilde{v}_i \end{bmatrix} \quad (\text{ฉ-6})$$

ความเพี้ยนเชิงพิกัดสามมิติสามารถแสดงได้ดังสมการที่ (ฉ-6) ค่า p_1 และ p_2 คือค่าสัมประสิทธิ์ความเพี้ยนเชิงพิกัดสามมิติ สำหรับความเพี้ยนรูปแบบอื่นจะมีผลน้อยมากและละเลยได้

ดังนั้นแบบจำลองกล้องที่ได้รับการพัฒนาขึ้นจากแบบจำลองรูเข็ม ผนวกกับการแก้ไขความเพี้ยนเชิงรัศมี และความเพี้ยนเชิงพิกัดสามมิติ แสดงดังสมการที่ (ฉ-7)

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_u s_u (\tilde{u}_i + \delta u_i^{(r)} + \delta u_i^{(t)}) \\ D_v (\tilde{v}_i + \delta v_i^{(r)} + \delta v_i^{(t)}) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad (\text{ฉ-7})$$

อย่างไรก็ตาม แบบจำลองรูเข็มไม่ได้อธิบายถึงการเลื่อนของจุดกำเนิด ความต่างของสเกล และการไม่ตั้งฉากกันระหว่างแกนภาพ จึงได้รับการพัฒนาเป็นแบบจำลองการแปลงเชิงเส้นโดยตรง (direct linear transform (DLT) model) (Abdel-Aziz and Karara, 1971) พิจารณาจุดวัตถุ P_i ที่ตำแหน่ง (X_i, Y_i, Z_i) เมื่อได้รับการแปลงเชิงเส้นลงสู่ระบบพิกัดภาพ (u_i, v_i) โดยใช้เมทริกซ์ \mathbf{A} แสดงดังสมการที่ (ฉ-8) ค่าพารามิเตอร์ a_{11}, \dots, a_{34} ของแบบจำลอง DLT สามารถคำนวณได้โดยกำจัด w_i และกำหนดให้ \mathbf{L} และ \mathbf{a} เป็นดังสมการที่ (ฉ-9) และ (ฉ-10) ตามลำดับ

$$\begin{bmatrix} u_i w_i \\ v_i w_i \\ w_i \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (\text{ฉ-8})$$

$$\mathbf{L} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1 u_1 & -Y_1 u_1 & -Z_1 u_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1 v_1 & -Y_1 v_1 & -Z_1 v_1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -X_i u_i & -Y_i u_i & -Z_i u_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -X_i v_i & -Y_i v_i & -Z_i v_i & -v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -X_N u_N & -Y_N u_N & -Z_N u_N & -u_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -X_N v_N & -Y_N v_N & -Z_N v_N & -v_N \end{bmatrix} \quad (\text{ฉ-9})$$

$$\mathbf{a} = [a_{11} \ a_{12} \ a_{13} \ a_{14} \ a_{21} \ a_{22} \ a_{23} \ a_{24} \ a_{31} \ a_{32} \ a_{33} \ a_{34}]^T \quad (\text{ฉ-10})$$

$$\mathbf{L}\mathbf{a} = \mathbf{0} \quad (\text{ฉ-11})$$

จากสมการที่ (ฉ-9) และ (ฉ-10) สามารถเขียนเป็นสมการเมตริกซ์เชิงเส้นสำหรับการพิจารณาจุดภาพจำนวน N จุดภาพ ดังสมการที่ (ฉ-11) การคำนวณค่าพารามิเตอร์ a_{11}, \dots, a_{34} โดยทั่วไป จะอาศัยระเบียบวิธีกำลังสองน้อยที่สุด โดยการแทนจุดภาพ (u_i, v_i) ด้วยจุดภาพที่ได้จากการสังเกด (U_i, V_i)

ค่าพารามิเตอร์ a_{11}, \dots, a_{34} ที่ได้จากการคำนวณที่ผ่านมา ไม่ได้สื่อความหมายทางกายภาพของกล้องแต่อย่างใด การสกัดค่าพารามิเตอร์ที่สื่อความหมายทางกายภาพของกล้องจากแบบจำลอง DLT แสดงดังสมการที่ (ฉ-12)

$$\mathbf{A} = \lambda \mathbf{V}^{-1} \mathbf{B}^{-1} \mathbf{F} \mathbf{M} \mathbf{T} \quad (\text{ฉ-12})$$

เมื่อ λ คือ ตัวประกอบสเกลรวม เมตริกซ์ \mathbf{F} , \mathbf{M} , และ \mathbf{T} แสดงความยาวโฟกัส การหมุน และการเลื่อนขนาน ดังสมการที่ (ฉ-13) ถึง (ฉ-15) ตามลำดับ เมตริกซ์ \mathbf{V} คือการเลื่อนของจุดกำเนิดภาพ และเมตริกซ์ \mathbf{B} เป็นการชดเชยความเพี้ยนเชิงเส้นเนื่องจากความต่างของสเกล และความไม่ตั้งฉากกันระหว่างแกนภาพ ดังสมการที่ (ฉ-16) และ (ฉ-17) ตามลำดับ

$$\mathbf{F} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{ฉ-13})$$

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (\text{ฉ-14})$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \end{bmatrix} \quad (\text{ฉ-15})$$

องค์ประกอบ m_{11}, \dots, m_{33} ในสมการที่ (ฉ-14) แสดงดังสมการที่ (ฉ-2)

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{ฉ-16})$$

$$\mathbf{B} = \begin{bmatrix} 1+b_1 & b_2 & 0 \\ b_2 & 1-b_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{ฉ-17})$$

ค่า (u_0, v_0) ในเมตริกซ์ \mathbf{V} คือ จุดมุงสำคัญ และค่า (b_1, b_2) ในเมตริกซ์ \mathbf{B} คือ ค่าสัมประสิทธิ์ความเพี้ยนเชิงเส้น

เมื่อพิจารณาความสัมพันธ์ระหว่างสมการที่ (ฉ-8) และ (ฉ-12) จะได้เป็นดังสมการที่ (ฉ-18) ซึ่งสามารถคำนวณค่าพารามิเตอร์ที่สื่อความหมายทางกายภาพของกล้องจากแบบจำลอง DLT ได้ โดยอาศัยการคำนวณเชิงตัวเลข

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} = \lambda \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1+b_1 & b_2 & 0 \\ b_2 & 1-b_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \end{bmatrix} \quad (\text{ฉ-18})$$

เทคนิคการปรับเทียบกล้อง

เทคนิคการปรับเทียบกล้องสามารถแบ่งได้เป็นสองประเภทคือ เทคนิคการปรับเทียบกล้องแบบเชิงเส้น และเทคนิคการปรับเทียบกล้องแบบไม่เชิงเส้น สำหรับเทคนิคการปรับเทียบกล้องแบบเชิงเส้นมีข้อดีคือ มีอัลกอริทึมไม่ซับซ้อน ทำให้เวลาที่ใช้ในการคำนวณค่อนข้างเร็ว แต่ข้อเสียคือผลการคำนวณมีความถูกต้องต่ำ ขณะที่เทคนิคการปรับเทียบกล้องแบบไม่เชิงเส้น ซึ่งรวมถึงการปรับเทียบกล้องโดยใช้เทคนิคทางปัญญาประดิษฐ์ด้วยนั้น มีข้อดีคือ ให้ผลการคำนวณที่มีความถูกต้องสูง แต่เนื่องจากมีอัลกอริทึมค่อนข้างซับซ้อน และอาจประสบปัญหาการกำหนดค่าเริ่มต้นที่ไม่เหมาะสม จนอาจทำให้เวลาที่ใช้ในการคำนวณค่อนข้างนาน

ในช่วงปี ค.ศ. 1987 ถึง 1988 Tsai และ Lenz ได้นำเสนอเทคนิคการปรับเทียบกล้องโดยใช้วิธีการเชิงเส้นเพื่อหาค่าคำตอบเริ่มต้น และใช้วิธีการไม่เชิงเส้นเพื่อปรับปรุงคำตอบให้เหมาะสมที่สุด (Tsai, 1987; Lenz and Tsai, 1988) เทคนิคการปรับเทียบกล้องของ Tsai และ Lenz ได้รับความนิยมอย่างแพร่หลาย เพราะได้รับการพัฒนาเป็น โปรแกรมสำเร็จที่สามารถหาใช้ได้ทางอินเทอร์เน็ต อย่างไรก็ตาม เทคนิคการปรับเทียบกล้องของ Tsai และ Lenz ไม่ครอบคลุมการแก้ไขความเพี้ยนที่เกิดขึ้นทั้งหมด กล่าวคือ จะให้ความสนใจแก้ไขเฉพาะความเพี้ยนเชิงรัศมีการเลือนของจุดกำเนิดภาพ และความต่างของสเกลในแกนภาพเท่านั้น

สำหรับเทคนิคการปรับเทียบกล้องที่ใช้ในงานวิจัยนี้ คือเทคนิคที่ได้รับการนำเสนอโดย Heikkila และ Silven (1997) ซึ่งใช้เทคนิคการหาค่าที่เหมาะสมที่สุด (optimization technique)

ในการคำนวณค่าพารามิเตอร์ของกล้อง ค่าพารามิเตอร์เริ่มต้นที่ใช้ในการหาค่าเหมาะที่สุดได้มาจากแบบจำลอง DLT เทคนิคของ Heikkila และ Silven สามารถคำนวณค่าพารามิเตอร์ของกล้องได้อย่างรวดเร็ว และสามารถแก้ไขความเพี้ยนเชิงรัศมี ความเพี้ยนเชิงพิวสัมพันธ์ ความเพี้ยนเชิงศูนย์กลาง การเลื่อนของจุดกำเนิดภาพ และความต่างของสเกลในแกนภาพ ได้อย่างมีประสิทธิภาพ นอกจากนี้เทคนิคการปรับเทียบกล้องของ Heikkila และ Silven ยังได้รับการพัฒนาให้เป็นกล่องเครื่องมือสำหรับปฏิบัติงานบน MATLAB อีกด้วย

การคำนวณค่าพารามิเตอร์ของกล้องตามเทคนิคของ Heikkila และ Silven จะใช้การหาค่าน้อยที่สุดของฟังก์ชันวัตถุประสงค์ ดังสมการที่ (ฉ-19) เมื่อ N คือ จำนวนจุดภาพ (U_i, V_i) คือ จุดภาพที่ได้จากการสังเกต และ (u_i, v_i) คือ จุดภาพที่ได้จากแบบจำลองในสมการที่ (ฉ-7)

$$F = \sum_{i=1}^N (U_i - u_i)^2 + \sum_{i=1}^N (V_i - v_i)^2 \quad (\text{ฉ-19})$$

เทคนิคการหาค่าเหมาะที่สุด ที่ Heikkila และ Silven ใช้ในการคำนวณค่าพารามิเตอร์ของกล้องคือ วิธีการหาค่าเหมาะที่สุดแบบ Gradient-descent (ดังรายละเอียดที่ปรากฏใน Heikkila และ Silven (1997)) ทั้งนี้เพราะสามารถเข้าสู่หาผลเฉลยได้อย่างรวดเร็วในกรณีที่ค่าคำตอบเริ่มต้นมีความเหมาะสม และเพื่อป้องกันการลื่นไถลโดยคำตอบวงแคบเฉพาะถิ่น Heikkila และ Silven ได้ใช้ค่าพารามิเตอร์ที่ได้จากแบบจำลอง DLT เป็นค่าคำตอบเริ่มต้นของกระบวนการค้นหา ดังนั้นคำตอบวงกว้างของฟังก์ชันวัตถุประสงค์ในสมการที่ (ฉ-19) จะถูกค้นพบภายในเวลาอันรวดเร็ว

การแก้ไขความเพี้ยน หรือการชดเชยความเพี้ยนของภาพ ตามวิธีของ Heikkila และ Silven จะเป็นการฉายภาพกลับ (back-projection) โดยอาศัยการคำนวณจุดภาพ $(\tilde{u}_i, \tilde{v}_i)$ จากจุดภาพ (u_i, v_i) ดังสมการที่ (ฉ-20) และ (ฉ-21)

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{1}{G} \begin{bmatrix} \hat{u}_i + \hat{u}_i(a_1 r_i^2 + a_2 r_i^4) + 2a_3 \hat{u}_i \hat{v}_i + a_4 (r_i^2 + 2\hat{u}_i^2) \\ \hat{v}_i + \hat{v}_i(a_1 r_i^2 + a_2 r_i^4) + a_3 (r_i^2 + 2\hat{v}_i^2) + 2a_4 \hat{u}_i \hat{v}_i \end{bmatrix} \quad (\text{ฉ-20})$$

$$G = (a_5 r_i^2 + a_6 \hat{u}_i + a_7 \hat{v}_i + a_8) r_i^2 + 1 \quad (\text{ฉ-21})$$

เมื่อ $\hat{u}_i = (u_i - u_0) / (D_u s_u)$, $\hat{v}_i = (v_i - v_0) / D_v$, และ $r_i = \sqrt{\hat{u}_i^2 + \hat{v}_i^2}$

จุดภาพ $(\tilde{u}_i, \tilde{v}_i)$ ที่ได้จากการคำนวณในสมการที่ (ฉ-20) และ (ฉ-21) จะถูกนำไปสร้างภาพที่ได้รับการแก้ไข หรือชดเชยความเพี้ยน โดยอาศัยการประมาณค่าในช่วง

สำหรับรายละเอียดของการเปรียบเทียบกล้อง และการแก้ไขความเพี้ยนของภาพ โดยเทคนิคของ Heikkila และ Silven ผู้อ่านสามารถศึกษาเพิ่มเติมได้จาก (Heikkila and Silven, 1997)

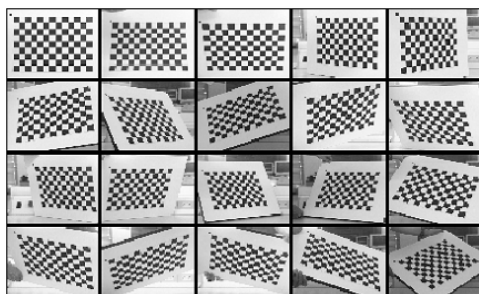
ผลการเปรียบเทียบกล้องวิดีโอดิจิทัล

การบันทึกภาพการเคลื่อนที่ของระบบกวัดแกว่งท่อยาง หรือระบบ VTS ในงานวิจัยนี้ จะใช้กล้องวิดีโอดิจิทัลจำนวน 2 ตัว โดยที่กล้องตัวที่ 1 ได้รับการติดตั้งในแนวแกน z เพื่อให้สามารถบันทึกการเคลื่อนไหวของระบบในระนาบ x, y และกล้องตัวที่ 2 ได้รับการติดตั้งในแนวแกน y เพื่อให้สามารถบันทึกการเคลื่อนไหวของระบบในระนาบ z, x กล้องวิดีโอดิจิทัลทั้ง 2 ตัวจะต้องมีคุณลักษณะสมบัติที่เหมือนกันทุกประการ ดังนั้นจึงพิจารณาเลือกกล้องวิดีโอดิจิทัลทั้ง 2 ตัวเป็นกล้องวิดีโอดิจิทัลยี่ห้อ Sony Digital Handycam DCR-VX1000E ผนวกเลนส์ 5.9 ถึง 59 มิลลิเมตร ซึ่งมีอัตราการภาพเท่ากับ 25 ภาพต่อวินาที

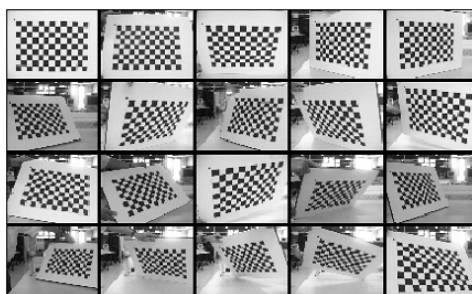
การเปรียบเทียบกล้องวิดีโอดิจิทัลทั้ง 2 ตัวอาศัยเทคนิคของ Heikkila และ Silven ซึ่งมีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1. นำกล้องวิดีโอดิจิทัลทั้ง 2 ตัวไปทำการบันทึกภาพแผ่นกระดานตรวจสอบแบบเรียบ (planer checkerboard) ซึ่งแผ่นกระดานตรวจสอบแบบเรียบดังกล่าว จะได้รับการจัดทำให้เป็นตารางสี่เหลี่ยมจัตุรัสสีขาวสลับดำ จำนวน 9×13 ช่อง แต่ละช่องมีขนาด 30×30 ตาราง มิลลิเมตร การบันทึกภาพโดยกล้องวิดีโอดิจิทัลแต่ละตัว จะเป็นการบันทึกในลักษณะภาพหนึ่งของแผ่นกระดานตรวจสอบแบบเรียบตำแหน่งใด ๆ ที่ให้ภาพชัดเจนจำนวน 20 ภาพ ดังแสดงในรูปที่ 4 (ก) และ (ข) ตามลำดับ

ขั้นตอนที่ 2. ทำการสกัดจุดภาพ (U_i, V_i) จากทุกภาพที่บันทึกได้โดยกล้องแต่ละตัว



(ก)



(ข)

รูปที่ 4. ภาพแผ่นกระดานตรวจสอบแบบเรียบที่ได้จากกล้องวิดีโอดิจิทัลทั้ง 2 ตัว

(ก) ภาพจากกล้องวิดีโอดิจิทัลตัวที่ 1 (ข) ภาพจากกล้องวิดีโอดิจิทัลตัวที่ 2

ขั้นตอนที่ 3. ทำการคำนวณค่าพารามิเตอร์เริ่มต้นจากแบบจำลอง DLT ซึ่งประกอบด้วยค่าพารามิเตอร์แบบอินทรีนซิก ค่าพารามิเตอร์แบบเอ็กทรีนซิก ค่าสัมประสิทธิ์ความเพี้ยนเชิงรัศมี และค่าสัมประสิทธิ์ความเพี้ยนเชิงผิวสัมผัส

ขั้นตอนที่ 4. ทำการคำนวณค่าพารามิเตอร์ที่เหมาะสมที่สุด โดยใช้การค้นหาแบบ Gradient-descent ซึ่งอาศัยค่าพารามิเตอร์เริ่มต้นที่ได้จากขั้นตอนที่ 3 เพื่อหาค่าน้อยที่สุดของฟังก์ชันวัตถุประสงค์ ในสมการที่ (ฉ-19)

ขั้นตอนที่ 5. ทำการแก้ไขหรือชดเชยความเพี้ยนของภาพ ตามอัลกอริทึมในสมการที่ (ฉ-20) และ (ฉ-21)

จากผลการปรับเทียบกล้องโดยเทคนิคของ Heikkila และ Silven จะให้ค่าพารามิเตอร์ที่เหมาะสมของแบบจำลองกล้องออกมาสองชุดคือ ค่าพารามิเตอร์แบบอินทรีนซิก และค่าพารามิเตอร์แบบเอ็กทรีนซิก

ค่าพารามิเตอร์แบบอินทรีนซิกที่ได้จากการปรับเทียบกล้องจะอยู่ในตัวแปรต่าง ๆ ดังนี้

1. ความยาวโฟกัส จะอยู่ในตัวแปร **fc** ซึ่งเป็นเวกเตอร์ขนาด 2×1
2. จุดมูข่าสำคัญ จะอยู่ในตัวแปร **cc** ซึ่งเป็นเวกเตอร์ขนาด 2×1
3. ค่าสัมประสิทธิ์ความเฉ (skew coefficient) ระหว่างแกนของจุดภาพ x และ y จะอยู่ในตัวแปร **alpha_c** (**alpha_c** = 0 นั่นคือแกนของจุดภาพ x และ y ตั้งฉากกันอย่างแท้จริง)
4. ค่าสัมประสิทธิ์ความเพี้ยนของภาพ (ค่าสัมประสิทธิ์ความเพี้ยนเชิงรัศมี และค่าสัมประสิทธิ์ความเพี้ยนเชิงผิวสัมผัส) จะอยู่ในตัวแปร **kc** ซึ่งเป็นเวกเตอร์ขนาด 5×1

เมื่อกำหนดให้ **P** เป็นจุดภาพในระบบพิกัดกล้อง $\mathbf{XX}_c = [\mathbf{X}_c; \mathbf{Y}_c; \mathbf{Z}_c]$ และทำการฉายภาพจุด **P** ดังกล่าวลงบนระนาบภาพโดยใช้ค่าพารามิเตอร์ **fc**, **cc**, **alpha_c**, และ **kc** กำหนดให้ \mathbf{x}_n เป็นการฉายภาพตามแบบจำลองรูเข็มที่ผ่านการนอร์มัลไลซ์ ดังสมการที่ (ฉ-22)

$$\mathbf{x}_n = \begin{bmatrix} \mathbf{X}_c / \mathbf{Z}_c \\ \mathbf{Y}_c / \mathbf{Z}_c \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \quad (\text{ฉ-22})$$

เมื่อ $\mathbf{r}^2 = \mathbf{x}^2 + \mathbf{y}^2$ ระบบพิกัดภาพที่รวมความเพี้ยนของเลนส์, \mathbf{x}_d , แสดงดังสมการที่ (ฉ-23)

$$\mathbf{x}_d = \begin{bmatrix} \mathbf{x}_d(1) \\ \mathbf{x}_d(2) \end{bmatrix} = (\mathbf{1} + \mathbf{kc}(1)\mathbf{r}^2 + \mathbf{kc}(2)\mathbf{r}^4 + \mathbf{kc}(5)\mathbf{r}^6) \mathbf{x}_n + \mathbf{dx} \quad (\text{ฉ-23})$$

เมื่อ **dx** คือเวกเตอร์ความเพี้ยนเชิงผิวสัมผัส ดังสมการที่ (ฉ-24)

$$dx = \begin{bmatrix} 2kc(3)xy + kc(4)(r^2 + 2x^2) \\ kc(3)(r^2 + 2y^2) + 2kc(4)xy \end{bmatrix} \quad (จ-24)$$

ดังนั้นระบบพิกัดภาพ $x_{\text{pixel}} = [x_p; y_p]$ ของการฉายภาพจุด P ลงบนระนาบภาพที่รวมความเพี้ยนของเลนส์แสดงดังสมการที่ (จ-25)

$$\left. \begin{aligned} x_p &= fc(1)(x_d(1) + \alpha_c \cdot x_d(2)) + cc(1) \\ y_p &= fc(2)x_d(2) + cc(2) \end{aligned} \right\} \quad (จ-25)$$

และความสัมพันธ์ระหว่างระบบพิกัดภาพ x_{pixel} และ x_d ในสมการที่ (จ-25) สามารถเขียนในรูปเมตริกซ์เชิงเส้นได้ดังสมการที่ (จ-26) เมื่อ KK คือเมตริกซ์ที่บรรจุค่าพารามิเตอร์ fc , cc , และ α_c ดังสมการที่ (จ-27)

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = KK \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix} \quad (จ-26)$$

$$KK = \begin{bmatrix} fc(1) & \alpha_c \cdot fc(1) & cc(1) \\ 0 & fc(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix} \quad (จ-27)$$

เมื่อเปรียบเทียบค่าพารามิเตอร์แบบอินทรีนซิกในตัวแปร fc , cc , α_c , และ kc กับสมการที่ (จ-3) ถึง (จ-7) จะพบว่า $fc(1) = fD_u s_u$, $fc(2) = fD_v$, $cc(1) = u_0$, $cc(2) = v_0$, $\alpha_c = 0$, $kc(1) = f^3 k_1$, $kc(2) = f^5 k_2$, $kc(3) = f^2 p_1$, $kc(4) = f^2 p_2$, และ $kc(5) = 0$ ค่าพารามิเตอร์ f , D_u , D_v , และ s_u โดยทั่วไปไม่สามารถคำนวณแยกส่วนกันได้ ยกเว้นทราบค่าพารามิเตอร์อย่างน้อย 2 ตัว สำหรับค่า $kc(5) = 0$ เมื่อพิจารณาจากสมการที่ (จ-5) และ (จ-23) พบว่าความเพี้ยนเชิงรัศมีถูกกำหนดให้มีอันดับ 4 ซึ่งถือว่าเพียงพอสำหรับกรณีส่วนใหญ่

อัตราส่วน $fc(2)/fc(1)$ เป็นที่รู้จักกันในนาม อัตราส่วนภาพ (aspect ratio) จะมีค่าเท่ากับ 1 ในกรณีที่จุดภาพมีความเป็นสี่เหลี่ยมจัตุรัสอย่างแท้จริง

ค่าพารามิเตอร์แบบเอ็กทรีนซิกที่ได้จากการปรับเทียบกล้องจะอยู่ในตัวแปรต่าง ๆ ดังนี้

1. เมตริกซ์การหมุน ขนาด 3×3 จำนวนเท่ากับจำนวนภาพที่ใช้ในการปรับเทียบกล้อง (20 ภาพ) จะอยู่ในตัวแปร **Rc_1, Rc_2, ..., Rc_20**

2. เวกเตอร์การเลื่อนขนาน ขนาด 3×1 จำนวนเท่ากับจำนวนภาพที่ใช้ในการปรับเทียบกล้อง จะอยู่ในตัวแปร **Tc_1, Tc_2, ..., Tc_20**

กำหนดให้ **P** เป็นจุดภาพในระบบพิกัดวัตถุ $\mathbf{XX}=[\mathbf{X};\mathbf{Y};\mathbf{Z}]$ และ $\mathbf{XX}_c=[\mathbf{X}_c;\mathbf{Y}_c;\mathbf{Z}_c]$ เป็นระบบพิกัดกล้องของจุดภาพ **P** ความสัมพันธ์ระหว่าง \mathbf{XX} และ \mathbf{XX}_c ที่ได้จากภาพที่ 1 แสดงดังสมการที่ (ฉ-28) ซึ่งสอดคล้องกับสมการที่ (ฉ-1)

$$\mathbf{XX}_c = \mathbf{Rc}_1 \cdot \mathbf{XX} + \mathbf{Tc}_1 \quad (\text{ฉ-28})$$

ผลการปรับเทียบกล้องตัวที่ 1 โดยเทคนิคของ Heikkila และ Silven ให้ค่าความคลาดเคลื่อนของจุดภาพในแนวแกน x และ y เท่ากับ 0.11741 และ 0.14405 จุดภาพตามลำดับค่าพารามิเตอร์แบบอินทรีนซิก แสดงดังสมการที่ (ฉ-29) และ (ฉ-30)

สำหรับค่าพารามิเตอร์แบบเอ็กทรีนซิกที่ได้จากปรับเทียบกล้องตัวที่ 1 จะอยู่ในตัวแปร **Rc_1, Rc_2, ..., Rc_20** และ **Tc_1, Tc_2, ..., Tc_20** ซึ่งจะไม่ขอแสดงรายละเอียดในที่นี้

$$\left. \begin{aligned} \mathbf{fc} &= [1187.76764 \quad 1175.74776] \\ \mathbf{cc} &= [199.98271 \quad 127.63545] \\ \mathbf{alpha}_c &= 0 \\ \mathbf{kc} &= [0.11211 \quad 2.83911 \quad -0.00559 \quad 0.00152 \quad 0] \end{aligned} \right\} \quad (\text{ฉ-29})$$

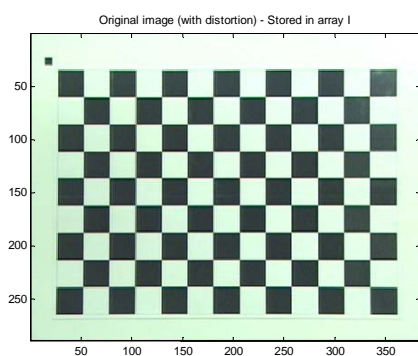
$$\mathbf{KK} = \begin{bmatrix} 1187.77 & 0 & 199.98 \\ 0 & 1175.75 & 127.64 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{ฉ-30})$$

ผลการปรับเทียบกล้องตัวที่ 2 โดยเทคนิคของ Heikkila และ Silven ให้ค่าความคลาดเคลื่อนของจุดภาพในแนวแกน x และ y เท่ากับ 0.14860 และ 0.22651 จุดภาพตามลำดับค่าพารามิเตอร์แบบอินทรีนซิก แสดงดังสมการที่ (ฉ-31) และ (ฉ-32) สำหรับค่าพารามิเตอร์แบบเอ็กทรีนซิกจะไม่ขอแสดงรายละเอียดในที่นี้

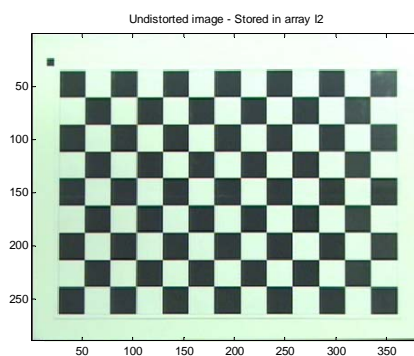
ผลการแก้ไขความเพี้ยนของภาพที่ได้จากกล้องตัวที่ 1 และ 2 แสดงดังรูปที่ ฉ.5 และ ฉ.6 ตามลำดับ ซึ่งพบว่ากล้องตัวที่ 1 ปรากฏความเพี้ยนรูปหมอนปีกเข็ม และกล้องตัวที่ 2 ปรากฏความเพี้ยนรูปถัง การแก้ไขความเพี้ยนของภาพให้ผลที่น่าพอใจ

$$\left. \begin{aligned} \mathbf{fc} &= [478.68507 \quad 474.48586] \\ \mathbf{cc} &= [199.29024 \quad 141.49148] \\ \mathbf{alpha_c} &= 0 \\ \mathbf{kc} &= [-0.23212 \quad 0.30327 \quad 0.00022 \quad 0.00016 \quad 0] \end{aligned} \right\} \quad (ก-31)$$

$$\mathbf{KK} = \begin{bmatrix} 478.69 & 0 & 199.29 \\ 0 & 474.49 & 141.49 \\ 0 & 0 & 1 \end{bmatrix} \quad (ก-32)$$



(ก)

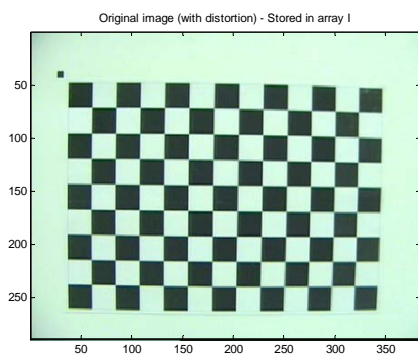


(ข)

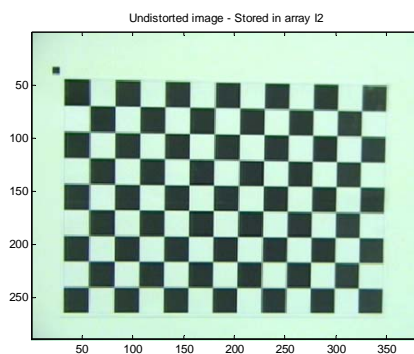
รูปที่ ๕.5 ผลการแก้ไขความเพี้ยนของภาพที่ได้จากกล้องตัวที่ 1

(ก) ภาพปรากฏความเพี้ยนรูปหมอนปีกเข้ม

(ข) ภาพภายหลังการแก้ไขความเพี้ยน



(ก)



(ข)

รูปที่ ๕.6 ผลการแก้ไขความเพี้ยนของภาพที่ได้จากกล้องตัวที่ 2

(ก) ภาพปรากฏความเพี้ยนรูปปลั่ง

(ข) ภาพภายหลังการแก้ไขความเพี้ยน

ภาคผนวก ข

แบบจำลองทางคณิตศาสตร์ของระบบ VTS

แบบจำลองทางคณิตศาสตร์ของระบบ VTS

ระบบกวัดแกว่งท่อยาง หรือระบบ VTS ดังแสดงในรูปที่ 8.1 เป็นระบบที่ประกอบด้วยท่อยางที่ได้รับการต่อปลายด้านหนึ่งเข้ากับเพลาของมอเตอร์ไฟฟ้ากระแสตรง และต่อปลายอีกด้านหนึ่งเข้ากับจุดยึดซึ่งสามารถหมุนได้รอบตัว การต่อท่อยางในลักษณะดังกล่าวทำให้ปลายทั้งสองด้านของท่อยางไม่สามารถเคลื่อนที่ไปตามแนวอนได้ การเคลื่อนที่ของท่อยางจะมีลักษณะการเคลื่อนที่แบบ 3 มิติที่ขึ้นกับแรงของมอเตอร์ที่ส่งผ่านมาตามเพลา

การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS อาศัยหลักการสร้างแบบจำลองเส้นลวดสั่น (vibrating string model) และเยื่อ (membrane) ในรูปของสมการคลื่น (wave equation) ซึ่งถือเป็นสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก (hyperbolic equation) โดยกำหนดข้อตกลงเบื้องต้นดังต่อไปนี้

1. พิจารณาให้ท่อยางไม่มีการยืดหยุ่นภายใน มีคุณสมบัติเช่นเดียวกับเส้นลวด
2. พิจารณาให้ท่อยางมีเนื้อเดียวกัน (homogeneous) ตลอดความยาว
3. ไม่พิจารณาแรงลมที่กระทำกับท่อยาง

การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS แบ่งออกเป็นสองส่วน คือแบบจำลองระบบ VTS แบบ 2 มิติ และแบบจำลองระบบ VTS แบบ 3 มิติ ดังรายละเอียดต่อไปนี้

แบบจำลองระบบ VTS แบบ 2 มิติ

พิจารณาระบบ VTS แบบ 2 มิติในรูปที่ ๗.1 ตัวแปรที่สนใจคือตำแหน่งของท่อยาง y ที่แปรผันตามระยะ x และเวลา t ซึ่งแสดงด้วย $y(x,t)$

จากรูปที่ ๗.1 อาศัยกฎการเคลื่อนที่ของนิวตัน โดยยังไม่พิจารณาความเร่งเนื่องจากแรงโน้มถ่วง จะได้ผลรวมของแรง F ในแนวแกน x และ y ดังสมการที่ (๗-1) และ (๗-2) ตามลำดับ

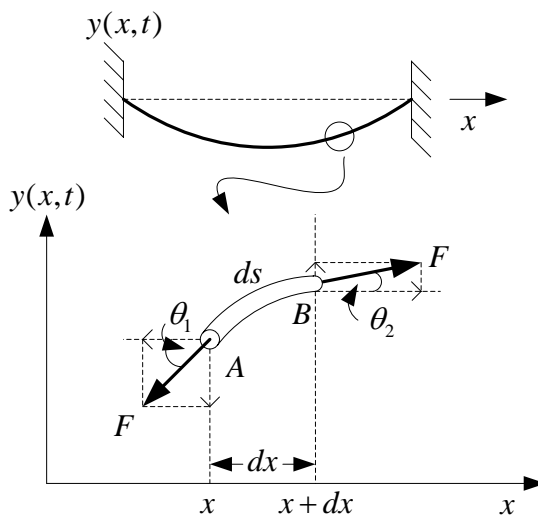
$$\sum F_x = F \cos \theta_2 - F \cos \theta_1 \quad (๗-1)$$

$$\sum F_y = F \sin \theta_2 - F \sin \theta_1 \quad (๗-2)$$

เนื่องจากมุม θ_1 และ θ_2 มีค่าน้อยมาก ส่งผลทำให้ $\cos \theta_1 \approx \cos \theta_2$ และทำให้สมการที่ (๗-1) มีค่าเท่ากับศูนย์ นั่นคือไม่มีแรงกระทำกับท่อยางในแนวแกน x

เมื่อมุม θ_1 และ θ_2 มีค่าน้อยมาก ๆ จะทำให้ $\sin \theta_1 \approx \tan \theta_1$ และ $\sin \theta_2 \approx \tan \theta_2$ และสมการที่ (ซ-2) สามารถเขียนใหม่ได้ดังสมการที่ (ซ-3)

$$\sum F_y = F \tan \theta_2 - F \tan \theta_1 \quad (\text{ซ-3})$$



รูปที่ ซ.1 ระบบ VTS แบบ 2 มิติ

เมื่อ m คือ มวลของท่อยาง และ a_y คือ ความเร่งในแนวแกน y อาศัยกฎการเคลื่อนที่ของนิวตัน

$$\sum F_y = ma_y = m \frac{\partial^2 y}{\partial t^2} \quad (\text{ซ-4})$$

จากรูปที่ ซ.1 ถ้า $ds \approx dx$ จะทำให้ $m = \mu dx$ เมื่อ μ คือมวลของท่อยางต่อความยาวหนึ่งหน่วย ดังนั้น

$$\mu dx \frac{\partial^2 y}{\partial t^2} = F (\tan \theta_2 - \tan \theta_1) \quad (\text{ซ-5})$$

และเมื่อ $\tan \theta = \frac{\partial y}{\partial x}$ (ความชัน) สามารถเขียนสมการที่ (ซ-5) ได้ดังสมการที่ (ซ-6)

$$\mu dx \frac{\partial^2 y}{\partial t^2} = F \left[\left(\frac{\partial y}{\partial x} \right)_B - \left(\frac{\partial y}{\partial x} \right)_A \right] \quad (\text{ข-6})$$

พิจารณานุกรมเทย์เลอร์ (Taylor series) ดังสมการที่ (ข-7)

$$f(x+dx) = f(x) + f'(x)dx + \frac{f''(x)(dx)^2}{2!} + \dots \quad (\text{ข-7})$$

ทำการประยุกต์อนุกรมเทย์เลอร์กับความชันที่จุด B ของท่อนางในรูปที่ ข.1 จะได้เป็น

$$\left(\frac{\partial y}{\partial x} \right)_B = \left(\frac{\partial y}{\partial x} \right)_A + \left(\frac{\partial^2 y}{\partial x^2} \right)_A dx + \frac{1}{2} \left(\frac{\partial^3 y}{\partial x^3} \right)_A (dx)^2 + \dots \quad (\text{ข-8})$$

เนื่องจากเทอม dx ยกกำลัง 2 เป็นต้นไปจะถือว่าน้อยมาก ๆ และละเลยได้ ทำให้สมการที่ (ข-8) สามารถเขียนได้ดังสมการที่ (ข-9)

$$\left(\frac{\partial y}{\partial x} \right)_B \approx \left(\frac{\partial y}{\partial x} \right)_A + \left(\frac{\partial^2 y}{\partial x^2} \right)_A dx \quad (\text{ข-9})$$

จากนั้นทำการแทนสมการที่ (ข-9) ลงในสมการที่ (ข-6) จะได้

$$\mu dx \frac{\partial^2 y}{\partial t^2} = F \left[\left(\frac{\partial y}{\partial x} \right)_A + \left(\frac{\partial^2 y}{\partial x^2} \right)_A dx - \left(\frac{\partial y}{\partial x} \right)_A \right] \quad (\text{ข-10})$$

แบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 2 มิติ เมื่อไม่พิจารณาความเร่งเนื่องจากแรงโน้มถ่วง แสดงดังสมการที่ (ข-11)

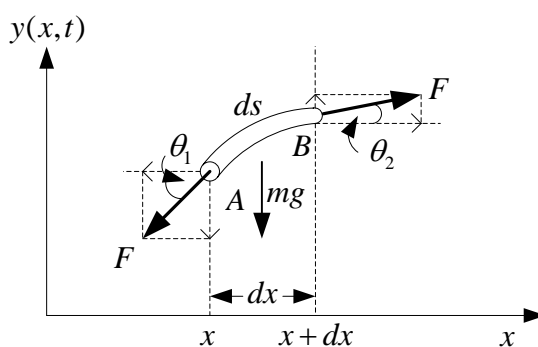
$$\frac{\partial^2 y}{\partial t^2} = \frac{F}{\mu} \frac{\partial^2 y}{\partial x^2} \quad (\text{ข-11})$$

ในกรณีที่พิจารณาความเร่งเนื่องจากแรงโน้มถ่วง ดังแสดงในรูปที่ ข.2 สมการที่ (ข-2) และ (ข-3) สามารถเขียนใหม่ได้ดังสมการที่ (ข-12) และ (ข-13) ตามลำดับ

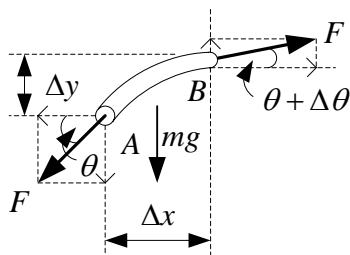
$$\sum F_y = F \sin \theta_2 - F \sin \theta_1 - mg \quad (\text{ข-12})$$

$$\sum F_y = F \tan \theta_2 - F \tan \theta_1 - mg \quad (\text{ข-13})$$

เมื่อ g คือ ความเร่งเนื่องจากแรงโน้มถ่วง และ $m = \mu dx$ จากสมการที่ (ข-5) และ (ข-10) สามารถเขียนใหม่ได้ดังสมการที่ (ข-14) และ (ข-15) ตามลำดับ



รูปที่ ข.2 ระบบ VTS แบบ 2 มิติเมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง



รูปที่ ข.3 ระบบ VTS แบบ 2 มิติเมื่อกำหนดให้ $\sin(\theta) \approx \theta$

$$\mu dx \frac{\partial^2 y}{\partial t^2} = F(\tan \theta_2 - \tan \theta_1) - \mu g dx \quad (\text{ข-14})$$

$$\mu dx \left(\frac{\partial^2 y}{\partial t^2} + g \right) = F \left[\left(\frac{\partial y}{\partial x} \right)_B + \left(\frac{\partial^2 y}{\partial x^2} \right)_A dx - \left(\frac{\partial y}{\partial x} \right)_A \right] \quad (\text{ข-15})$$

ดังนั้นแบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 2 มิติ เมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง แสดงดังสมการที่ (ข-16)

$$\frac{\partial^2 y}{\partial t^2} = \frac{F}{\mu} \frac{\partial^2 y}{\partial x^2} - g \quad (\text{ข-16})$$

กำหนดให้ $\sin(\theta) \approx \theta$ จากสมการที่ (ข-12) และ (ข-14) สามารถเขียนเป็นสมการเดียวกันได้ดังสมการที่ (ข-17) และรูปที่ ข.2 สามารถวาดขึ้นใหม่ได้ดังรูปที่ ข.3

$$\sum F_y = F(\theta_2 - \theta_1) - mg = \mu dx \frac{\partial^2 y}{\partial t^2} \quad (\text{ข-17})$$

กำหนดให้ $\theta_1 = \theta$, $\theta_2 = \theta + \Delta\theta$, และ $dx = \Delta x$ จากสมการที่ (ข-17) จะได้

$$\sum F_y = F(\theta + \Delta\theta - \theta) - \mu g \Delta x = \mu \Delta x \frac{\partial^2 y}{\partial t^2}$$

$$F \Delta\theta - \mu g \Delta x = \mu \Delta x \frac{\partial^2 y}{\partial t^2} \quad (\text{ข-18})$$

$$\tan \theta = \frac{\partial y}{\partial x} \quad (\text{ข-19})$$

$$\sec^2 \theta \Delta\theta = \frac{\partial^2 y}{\partial x^2} \Delta x \quad (\text{ข-20})$$

เนื่องจาก $\sec \theta \approx 1$ ดังนั้น

$$\Delta\theta \approx \frac{\partial^2 y}{\partial x^2} \Delta x \quad (\text{ข-21})$$

สมการที่ (ข-18) สามารถเขียนใหม่ได้ดังสมการที่ (ข-22)

$$F \frac{\partial^2 y}{\partial x^2} \Delta x = \mu \Delta x \left(\frac{\partial^2 y}{\partial t^2} + g \right) \quad (\text{ข-22})$$

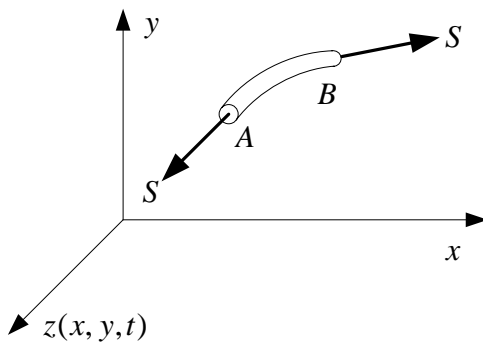
แบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 2 มิติ เมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง และกำหนดให้ $\sin(\theta) \approx \theta$ แสดงดังสมการที่ (ข-23) ซึ่งตรงกับสมการที่ (ข-16)

$$\frac{\partial^2 y}{\partial t^2} = \frac{F}{\mu} \frac{\partial^2 y}{\partial x^2} - g \tag{ซ-23}$$

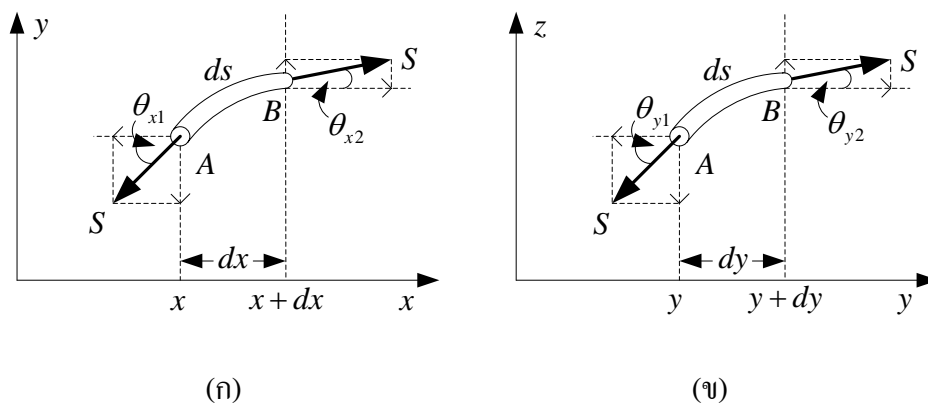
แบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 2 มิติ จะได้รับการพัฒนาให้เป็นแบบจำลองแบบ 3 มิติ ดังรายละเอียดในหัวข้อต่อไป

แบบจำลองระบบ VTS แบบ 3 มิติ

ระบบ VTS แบบ 3 มิติแสดงในรูปที่ ซ.4 โดยตัวแปรที่สนใจคือตำแหน่งของท่อยาง z จะแปรผันตามระยะ x ระยะ y และเวลา t ซึ่งแสดงด้วย $z(x, y, t)$ ทำการพิจารณาระบบในระนาบ x, y และ y, z ดังรูปที่ ซ.5 (ก) และ (ข) ตามลำดับ



รูปที่ ซ.4 ระบบ VTS แบบ 3 มิติ



รูปที่ ซ.5 ระบบ VTS แบบ 3 มิติเมื่อพิจารณาแบบ 2 มิติ

(ก) พิจารณาแรงในระนาบ x, y (ข) พิจารณาแรงในระนาบ y, z

การพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 3 มิติจะอาศัยพื้นฐานของการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 2 มิติ และหลักการสร้างแบบจำลองทางคณิตศาสตร์ของเยื่อ (French, 1992) จากรูปที่ ข.5 (ก) อาศัยกฎการเคลื่อนที่ของนิวตัน โดยยังไม่พิจารณาความเร่งเนื่องจากแรงโน้มถ่วง จะได้ผลรวมของแรง S ที่กระทำกับท่ออย่างตามแนวแกน x และ y ดังสมการที่ (ข-24) และ (ข-25) ตามลำดับ

$$\sum S_x = S \cos \theta_{x2} - S \cos \theta_{x1} \quad (\text{ข-24})$$

$$\sum S_y = S \sin \theta_{x2} - S \sin \theta_{x1} \quad (\text{ข-25})$$

พิจารณาเช่นเดียวกับการพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 2 มิติ เนื่องจากมุม θ_{x1} และ θ_{x2} มีค่าน้อยมาก ส่งผลทำให้ $\cos \theta_{x1} \approx \cos \theta_{x2}$ และทำให้สมการที่ (ข-24) มีค่าเท่ากับศูนย์ นั่นคือไม่มีแรงกระทำกับท่ออย่างในแนวแกน x

จากสมการที่ (ข-25) เมื่อมุม θ_{x1} และ θ_{x2} มีค่าน้อยมาก ๆ จะทำให้ $\sin \theta_{x1} \approx \tan \theta_{x1}$ และ $\sin \theta_{x2} \approx \tan \theta_{x2}$ สมการที่ (ข-25) สามารถเขียนใหม่ได้ดังสมการที่ (ข-26) หรือ (ข-27)

$$\sum S_y = S \tan \theta_{x2} - S \tan \theta_{x1} \quad (\text{ข-26})$$

$$\sum S_y = S \left[\left(\frac{\partial z}{\partial x} \right)_B - \left(\frac{\partial z}{\partial x} \right)_A \right] \quad (\text{ข-27})$$

จากอนุกรมเทย์เลอร์ในสมการที่ (ข-7) ทำการประยุกต์กับความชันที่จุด B ของท่ออย่างในรูปที่ ข.5 (ก) จะได้

$$\left(\frac{\partial z}{\partial x} \right)_B = \left(\frac{\partial z}{\partial x} \right)_A + \left(\frac{\partial^2 z}{\partial x^2} \right)_A dx dy + \frac{1}{2} \left(\frac{\partial^3 z}{\partial x^3} \right)_A (dx)^2 (dy)^2 + \dots \quad (\text{ข-28})$$

เนื่องจากเทอม dx ยกกำลัง 2 เป็นต้นไปจะถือว่าน้อยมาก ๆ และละเลยได้ ทำให้สมการที่ (ข-28) สามารถเขียนได้ดังสมการที่ (ข-29)

$$\left(\frac{\partial z}{\partial x} \right)_B \approx \left(\frac{\partial z}{\partial x} \right)_A + \left(\frac{\partial^2 z}{\partial x^2} \right)_A dx dy \quad (\text{ข-29})$$

จากนั้นแทนสมการที่ (ข-29) ลงในสมการที่ (ข-27) จะได้

$$\begin{aligned}\sum S_y &= S \left[\left(\frac{\partial z}{\partial x} \right)_A + \left(\frac{\partial^2 z}{\partial x^2} \right)_A dx dy - \left(\frac{\partial z}{\partial x} \right)_A \right] \\ \sum S_y &= S \left(\frac{\partial^2 z}{\partial x^2} \right) dx dy\end{aligned}\quad (ข-30)$$

และจากรูปที่ ข.5 (ข) อาศัยกฎการเคลื่อนที่ของนิวตัน โดยยังไม่พิจารณาความเร่งเนื่องจากแรงโน้มถ่วง จะได้ผลรวมของแรง S ที่กระทำกับท่ออย่างตามแนวแกน y และ z ดังสมการที่ (ข-31) และ (ข-32) ตามลำดับ

$$\sum S_y = S \cos \theta_{y2} - S \cos \theta_{y1} \quad (ข-31)$$

$$\sum S_z = S \sin \theta_{y2} - S \sin \theta_{y1} \quad (ข-32)$$

เนื่องจากมุม θ_{y1} และ θ_{y2} มีค่าน้อยมาก ส่งผลทำให้ $\cos \theta_{y1} \approx \cos \theta_{y2}$ และทำให้สมการที่ (ข-31) มีค่าเท่ากับศูนย์ นั่นคือไม่มีแรงกระทำกับท่ออย่างในแนวแกน y

จากสมการที่ (ข-32) เมื่อมุม θ_{y1} และ θ_{y2} มีค่าน้อยมาก ๆ จะทำให้ $\sin \theta_{y1} \approx \tan \theta_{y1}$ และ $\sin \theta_{y2} \approx \tan \theta_{y2}$ สมการที่ (ข-32) สามารถเขียนใหม่ได้ดังสมการที่ (ข-33) หรือ (ข-34)

$$\sum S_z = S \tan \theta_{y2} - S \tan \theta_{y1} \quad (ข-33)$$

$$\sum S_z = S \left[\left(\frac{\partial z}{\partial y} \right)_B - \left(\frac{\partial z}{\partial y} \right)_A \right] \quad (ข-34)$$

ทำการประยุกต์อนุกรมเทย์เลอร์ในสมการที่ (ข-7) กับความชันที่จุด B ของท่ออย่างในรูปที่ ข.5 (ข) และพิจารณาให้ทอม dx ยกกำลัง 2 เป็นต้นไปมีค่าน้อยมาก ๆ และละเลยได้

$$\left(\frac{\partial z}{\partial y} \right)_B \approx \left(\frac{\partial z}{\partial y} \right)_A + \left(\frac{\partial^2 z}{\partial y^2} \right)_A dx dy \quad (ข-35)$$

จากนั้นแทนสมการที่ (ข-35) ลงในสมการที่ (ข-34) จะได้

$$\begin{aligned}\sum S_z &= S \left[\left(\frac{\partial z}{\partial y} \right)_A + \left(\frac{\partial^2 z}{\partial y^2} \right)_A dx dy - \left(\frac{\partial z}{\partial y} \right)_A \right] \\ \sum S_z &= S \left(\frac{\partial^2 z}{\partial y^2} \right) dx dy\end{aligned}\quad (\text{ข-36})$$

กำหนดให้ σ คือ มวลของท่ออย่างต่อความยาวหนึ่งหน่วยในแนวแกน x และ y (ในกรณีของเยื่อ ค่า σ คือมวลต่อพื้นที่ผิวหนึ่งหน่วย) ดังนั้นจากรูปที่ ข.5 (ก) และ (ข) ถ้า $ds \approx dx \approx dy$ จะทำให้ $m = \sigma dx dy$ และเมื่ออาศัยกฎการเคลื่อนที่ของนิวตันจะได้

$$\sum S = \sigma dx dy \frac{\partial^2 z}{\partial t^2} \quad (\text{ข-37})$$

ผลรวมของแรงทั้งหมดที่กระทำในท่ออย่าง แสดงดังสมการที่ (ข-38)

$$\sum S = \sum S_y + \sum S_z \quad (\text{ข-38})$$

ทำการแทนสมการที่ (ข-30), (ข-36), และ (ข-37) ลงในสมการที่ (ข-38) จะได้

$$\sigma dx dy \frac{\partial^2 z}{\partial t^2} = S \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) dx dy \quad (\text{ข-39})$$

แบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 3 มิติ เมื่อไม่พิจารณาความเร่งเนื่องจากแรงโน้มถ่วง แสดงดังสมการที่ (ข-40)

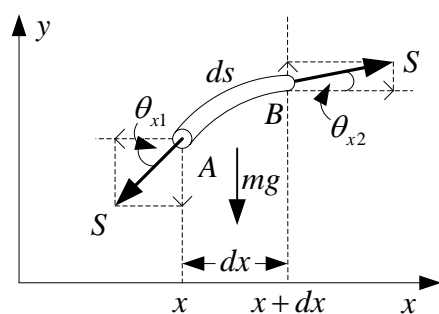
$$\frac{\partial^2 z}{\partial t^2} = \frac{S}{\sigma} \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) \quad (\text{ข-40})$$

ในกรณีที่พิจารณาความเร่งเนื่องจากแรงโน้มถ่วง รูปที่ ข.5 (ก) และ (ข) สามารถวาดขึ้นใหม่ได้ดังรูปที่ ข.6 (ก) และ (ข) ตามลำดับ

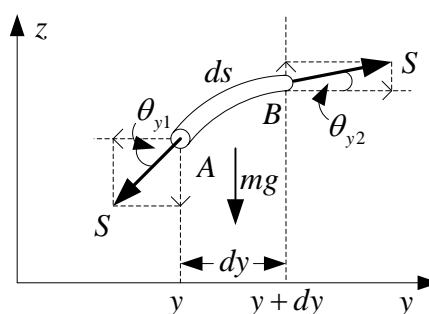
เมื่อพิจารณาจากรูปที่ ข.6 (ก) สมการที่ (ข-27) และ (ข-30) สามารถเขียนใหม่ได้ดังสมการที่ (ข-41) และ (ข-42) ตามลำดับ

$$\sum S_y = S \left[\left(\frac{\partial z}{\partial x} \right)_B - \left(\frac{\partial z}{\partial x} \right)_A \right] - mg \quad (\text{ข-41})$$

$$\sum S_y = S \left(\frac{\partial^2 z}{\partial x^2} \right) dx dy - mg \quad (\text{ข-42})$$



(ก)



(ข)

รูปที่ ข.6 ระบบ VTS แบบ 3 มิติเมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง

(ก) พิจารณาแรงในระนาบ x, y (ข) พิจารณาแรงในระนาบ y, z

และเมื่อพิจารณาจากรูปที่ ข.6 (ข) สมการที่ (ข-34) และ (ข-36) สามารถเขียนใหม่ได้ดังสมการที่ (ข-43) และ (ข-44) ตามลำดับ

$$\sum S_z = S \left[\left(\frac{\partial z}{\partial y} \right)_B - \left(\frac{\partial z}{\partial y} \right)_A \right] - mg \quad (\text{ข-43})$$

$$\sum S_z = S \left(\frac{\partial^2 z}{\partial y^2} \right) dx dy - mg \quad (\text{ข-44})$$

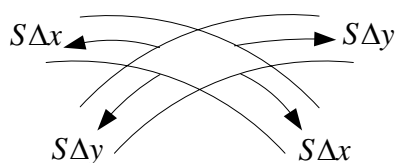
เมื่อกำหนดให้ σ คือ มวลของท่อยางต่อความยาวหนึ่งหน่วยในแนวแกน x และ y จากรูปที่ ข.6 (ก) และ (ข) ถ้า $ds \approx dx \approx dy$ จะทำให้ $m = \sigma dx dy$ เมื่อแทนสมการที่ (ข-37),

(ข-42), และ (ข-44) ลงในสมการที่ (ข-38) จะได้ดังสมการที่ (ข-45) แบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 3 มิติ เมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง แสดงดังสมการที่ (ข-46)

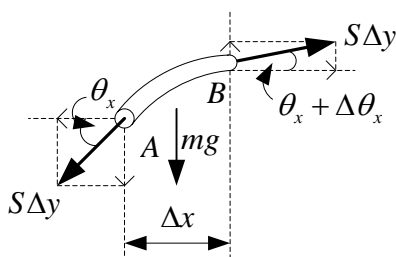
$$\sigma dx dy \frac{\partial^2 z}{\partial t^2} = S \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) dx dy - \sigma g dx dy \quad (\text{ข-45})$$

$$\frac{\partial^2 z}{\partial t^2} = \frac{S}{\sigma} \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - g \quad (\text{ข-46})$$

เมื่อกำหนดให้ $\sin(\theta) \approx \theta$ เพื่อพัฒนาแบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 3 มิติ พิจารณาแรงที่กระทำบนพื้นที่เล็ก ๆ ($\Delta x \Delta y$) และภาพตัดของแรง $S \Delta y$ ในระนาบ z, x ดังแสดงในรูปที่ ข.7 และ ข.8 ตามลำดับ



รูปที่ ข.7 ไลอะแกรมของแรงที่กระทำบนพื้นที่ $\Delta x \Delta y$



รูปที่ ข.8 ภาพตัดของแรง $S \Delta y$ ในระนาบ z, x

ตัวแปรพลวัตที่สนใจคือ $z(x, y, t)$ พิจารณาการเคลื่อนที่ของท่ออย่างในปริภูมิด้วยพื้นที่เล็ก ๆ ($\Delta x \Delta y$) ดังรูปที่ ข.7 ค่า S คือแรงกระทำต่อความยาวหนึ่งหน่วย ซึ่งมีทิศทางตั้งฉากกับระนาบที่สนใจ กล่าวคือ ถ้าความยาวหนึ่งหน่วยที่พิจารณาคือ Δx และ Δy ตามแนวแกน x และ y ตามลำดับ แรงที่กระทำในระนาบ z, x คือ $S \Delta y$ และแรงที่กระทำในระนาบ y, z คือ $S \Delta x$ ตามลำดับ

พิจารณาให้ $m = \sigma \Delta x \Delta y$ คือมวลของท่ออย่างที่เกิดขึ้นที่ในปริภูมิด้วยพื้นที่เล็ก ๆ และเมื่อพิจารณาภาพตัดของแรง $S \Delta y$ ในระนาบ z, x ดังรูปที่ ข.8 พบว่ามีลักษณะเช่นเดียวกับรูปที่ ข.3 ดังนั้นสามารถพิจารณาเฉพาะแรงที่กระทำในแนวตั้ง ดังสมการที่ (ข-47)

$$\sum S \Delta y = S \Delta y [\sin(\theta_x + \Delta \theta_x) - \sin \theta_x] - mg \quad (\text{ข-47})$$

ทำการแทน $\sin \theta \approx \theta$ ลงในสมการที่ (ข-47) จะได้

$$\sum S \Delta y = S \Delta y (\theta_x + \Delta \theta_x - \theta_x) - mg = S \Delta y \Delta \theta_x - mg \quad (\text{ข-48})$$

ในทำนองเดียวกัน เมื่อพิจารณาภาพตัดของแรง $S \Delta x$ ในระนาบ y, z จะได้แรงที่กระทำในแนวตั้ง ดังสมการที่ (ข-49)

$$\sum S \Delta x = S \Delta x [\sin(\theta_y + \Delta \theta_y) - \sin \theta_y] - mg \quad (\text{ข-49})$$

ทำการแทน $\sin \theta \approx \theta$ ลงในสมการที่ (ข-49) จะได้

$$\sum S \Delta x = S \Delta x (\theta_y + \Delta \theta_y - \theta_y) - mg = S \Delta x \Delta \theta_y - mg \quad (\text{ข-50})$$

ประยุกต์สมการที่ (ข-19) และ (ข-20) จะได้สมการที่ (ข-51) และ (ข-52) ตามลำดับ

$$\tan \theta_x = \frac{\partial z}{\partial x} \quad (\text{ข-51})$$

$$\sec^2 \theta_x \Delta \theta_x = \frac{\partial^2 z}{\partial x^2} \Delta x \quad (\text{ข-52})$$

เนื่องจาก $\sec \theta \approx 1$ ดังนั้น

$$\Delta \theta_x \approx \frac{\partial^2 z}{\partial x^2} \Delta x \quad (\text{ข-53})$$

ดังนั้นสมการที่ (ข-48) สามารถเขียนใหม่ได้ดังสมการที่ (ข-54)

$$S\Delta y\Delta\theta_x - mg = S \frac{\partial^2 z}{\partial x^2} \Delta x\Delta y \quad (\text{ข-54})$$

ในทำนองเดียวกัน เมื่อประยุกต์สมการที่ (ข-19) และ (ข-20) จะได้สมการที่ (ข-55) และ (ข-56) ตามลำดับ

$$\tan \theta_y = \frac{\partial z}{\partial y} \quad (\text{ข-55})$$

$$\sec^2 \theta_y \Delta\theta_y = \frac{\partial^2 z}{\partial y^2} \Delta y \quad (\text{ข-56})$$

เนื่องจาก $\sec \theta \approx 1$ ดังนั้น

$$\Delta\theta_y \approx \frac{\partial^2 z}{\partial y^2} \Delta y \quad (\text{ข-57})$$

ดังนั้นสมการที่ (ข-50) สามารถเขียนใหม่ได้ดังสมการที่ (ข-58)

$$S\Delta x\Delta\theta_y - mg = S \frac{\partial^2 z}{\partial y^2} \Delta x\Delta y \quad (\text{ข-58})$$

เมื่ออาศัยกฎการเคลื่อนที่ของนิวตัน และทำการประยุกต์สมการที่ (ข-37) จะได้

$$\sum S = m \frac{\partial^2 z}{\partial t^2} = \sigma\Delta x\Delta y \frac{\partial^2 z}{\partial t^2} \quad (\text{ข-59})$$

ผลรวมของแรงทั้งหมดที่กระทำในท่อยาง เมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง แสดงดังสมการที่ (ข-60)

$$\sigma \Delta x \Delta y \frac{\partial^2 z}{\partial t^2} = S \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) \Delta x \Delta y - \sigma g \Delta x \Delta y \quad (\text{ข-60})$$

ดังนั้นแบบจำลองทางคณิตศาสตร์ของระบบ VTS แบบ 3 มิติ เมื่อพิจารณาความเร่งเนื่องจากแรงโน้มถ่วง และกำหนดให้ $\sin(\theta) \approx \theta$ แสดงดังสมการที่ (ข-61) ซึ่งตรงกับสมการที่ (ข-46)

$$\frac{\partial^2 z}{\partial t^2} = \frac{S}{\sigma} \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - g \quad (\text{ข-61})$$

ค่า g ในสมการที่ (ข-61) คือความเร่งเนื่องจากแรงโน้มถ่วงที่กระทำในแนวแกน y กำหนดให้เป็น g_y ดังนั้นสมการที่ (ข-61) สามารถเขียนได้ใหม่เป็นดังสมการที่ (ข-62)

$$\frac{\partial^2 z}{\partial t^2} = \frac{S}{\sigma} \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - g_y \quad (\text{ข-62})$$

สมการที่ (ข-62) คือแบบจำลองทั่วไป (generalized model) ที่ให้การอธิบายการเคลื่อนที่ของระบบ VTS แบบ 3 มิติ และสามารถให้การอธิบายการเคลื่อนที่ของระบบ VTS แบบ 2 มิติ ในระนาบ 2 มิติใด ๆ ได้โดยการลดรูปแบบจำลอง ดังรายละเอียดต่อไปนี้

เมื่อต้องการอธิบายการเคลื่อนที่ของระบบ VTS แบบ 2 มิติในระนาบ x, y แบบจำลองทั่วไปดังสมการที่ (ข-62) จะได้รับการลดรูปโดยอาศัยความสัมพันธ์ตามสมการที่ (ข-63)

$$\left. \begin{array}{l} S \rightarrow F \\ \sigma \rightarrow \mu \\ g_y \rightarrow g \\ z(x, y, t) \rightarrow y(x, t) \end{array} \right\} \quad (\text{ข-63})$$

เมื่อพิจารณาสมการที่ (ข-62) และ (ข-63) จะพบว่า $\frac{\partial^2 z}{\partial t^2} = \frac{\partial^2 y}{\partial t^2}$, $\frac{\partial^2 z}{\partial x^2} = \frac{\partial^2 y}{\partial x^2}$, และ $\frac{\partial^2 z}{\partial y^2} = 0$ ทั้งนี้เนื่องจาก $z(x, y, t)$ ได้รับการลดรูปเป็น $y(x, t)$ ตามความสัมพันธ์ในสมการที่ (ข-63) อนุพันธ์ย่อยของ $y(x, t)$ เทียบกับ y มีค่าเป็นศูนย์เนื่องจาก $y(x, t)$ แปรผันตามระยะ x

และเวลา t แต่ไม่แปรผันตามระยะ y ดังนั้นแบบจำลองทั่วไปที่ได้รับการลดรูปเพื่อให้การอธิบายการเคลื่อนที่ของระบบ VTS แบบ 2 มิติในระนาบ x, y แสดงดังสมการที่ (ซ-64) ซึ่งก็คือสมการที่ (9-1) ในบทที่ 9 นั่นเอง

$$\frac{\partial^2 y}{\partial t^2} = \frac{F}{\mu} \left(\frac{\partial^2 y}{\partial x^2} \right) - g \quad (\text{ซ-64})$$

สำหรับการอธิบายการเคลื่อนที่ของระบบ VTS แบบ 2 มิติในระนาบ z, x แบบจำลองทั่วไปดังสมการที่ (ซ-62) จะได้รับการลดรูปโดยอาศัยความสัมพันธ์ตามสมการที่ (ซ-65)

$$\left. \begin{array}{l} S \rightarrow F \\ \sigma \rightarrow \mu \\ g_y \rightarrow 0 \\ z(x, y, t) \rightarrow z(x, t) \end{array} \right\} \quad (\text{ซ-65})$$

เมื่อพิจารณาสมการที่ (ซ-62) และ (ซ-65) จะพบว่า g_y ได้รับการพิจารณาให้มีค่าเป็นศูนย์ ทั้งนี้เนื่องจาก g_y คือความเร่งเนื่องจากแรงโน้มถ่วงที่กระทำในแนวแกน y ซึ่งมีทิศทางที่ตั้งฉากกับระนาบที่กำลังพิจารณา ค่า $\frac{\partial^2 z}{\partial t^2} = \frac{\partial^2 z}{\partial t^2}$, $\frac{\partial^2 z}{\partial x^2} = \frac{\partial^2 z}{\partial x^2}$, และ $\frac{\partial^2 z}{\partial y^2} = 0$ เนื่องจาก $z(x, y, t)$ ได้รับการลดรูปเป็น $z(x, t)$ ตามความสัมพันธ์ในสมการที่ (ซ-65) อนุพันธ์ย่อยของ $z(x, t)$ เทียบกับ y มีค่าเป็นศูนย์เนื่องจาก $z(x, t)$ แปรผันตามระยะ x และเวลา t แต่ไม่แปรผันตามระยะ y ดังนั้นแบบจำลองทั่วไปที่ได้รับการลดรูปเพื่อให้การอธิบายการเคลื่อนที่ของระบบ VTS แบบ 2 มิติในระนาบ z, x แสดงดังสมการที่ (ซ-66) ซึ่งก็คือสมการที่ (9-4) ในบทที่ 9 นั่นเอง

$$\frac{\partial^2 z}{\partial t^2} = \frac{F}{\mu} \left(\frac{\partial^2 z}{\partial x^2} \right) \quad (\text{ซ-66})$$

ภาคผนวก ซ

โปรแกรมประมวลผลภาพระบบ VTS

โปรแกรมประมวลผลภาพระบบ VTS

โปรแกรมประมวลผลภาพระบบ VTS พัฒนาขึ้นโดยใช้โปรแกรม MATLAB™ แบ่งออกเป็นสามส่วนด้วยกันคือ โปรแกรมแยกแยะลำดับภาพจากไฟล์วิดีโอ โปรแกรมสกัดข้อมูลพลวัตของระบบ VTS จากกล้องตัวที่ 1 และโปรแกรมสกัดข้อมูลพลวัตของระบบ VTS จากกล้องตัวที่ 2 ดังรายละเอียดต่อไปนี้

โปรแกรมแยกแยะลำดับภาพจากไฟล์วิดีโอของระบบ VTS

```
% โปรแกรมแยกแยะลำดับภาพจากไฟล์วิดีโอที่พัฒนาด้วย MATLAB™
% พัฒนาโดยเคชา พวงดาวเรือง (2005)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

n_frame = 500; % กำหนด n_frame เท่ากับจำนวนภาพในไฟล์วิดีโอ 500 ภาพ
start = 100000; % กำหนดชื่อไฟล์ภาพเริ่มต้นด้วย 100000

for i = 1:n_frame
    mov = aviread('vts_cam1.avi',i); % อ่านไฟล์วิดีโอ vts_cam1.avi เก็บไว้ในตัวแปร mov
    % ในงานวิจัยนี้มีไฟล์วิดีโอของระบบ VTS จำนวน 2 ไฟล์คือ
    % ไฟล์ vts_cam1.avi ซึ่งได้จากกล้องตัวที่ 1 และ
    % ไฟล์ vts_cam2.avi ซึ่งได้จากกล้องตัวที่ 2

    start1 = start+i; % กำหนดให้ตัวแปรชื่อไฟล์ start1 = start + i
    str = num2str(start1); % แปลงค่าตัวแปร start1 ให้เป็นตัวเลขสตริง
    str = strcat(str,'.bmp'); % ผูกตัวแปรชื่อไฟล์เข้ากับ .bmp ("ชื่อไฟล์" + ".bmp" = "ชื่อไฟล์.bmp")
    imwrite(mov.cdata,str,'bmp'); % เขียนภาพชนิด bmp ในตัวแปร mov ลงในไฟล์ภาพ "ชื่อไฟล์.bmp"
end
```

โปรแกรมสกัดข้อมูลพลวัตของระบบ VTS จากกล้องตัวที่ 1

```
% โปรแกรมสกัดข้อมูลพลวัตของระบบ VTS จากลำดับภาพที่ได้จากกล้องตัวที่ 1 ซึ่งพัฒนาด้วย MATLAB™
% พัฒนาโดยเคชา พวงดาวเรือง (2005)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

clear all
n_file = 500; % กำหนด n_file เท่ากับจำนวนภาพในไฟล์วิดีโอ 500 ภาพ
start = 100000; % กำหนดชื่อไฟล์ภาพเริ่มต้นด้วย 100000
for iii = 1:n_file % เริ่มต้นสกัดข้อมูลพลวัตจากไฟล์ลำดับที่ 1 (100001)
    start1 = start+iii; % กำหนดให้ตัวแปรชื่อไฟล์ start1 = start + iii
    str = num2str(start1); % แปลงค่าตัวแปร start1 ให้เป็นตัวเลขสตริง
```

```

str = strcat(str,'_undist_color_deacha.bmp');
                                %ผนวกตัวแปรชื่อไฟล์เข้ากับ .bmp (“ชื่อไฟล์” + “_undist_color_deacha.bmp”)
a = imread(str);                %อ่านไฟล์ภาพ “ชื่อไฟล์.bmp” เก็บไว้ในตัวแปร a
% Start segmentation
for i=120:166                    % ช่วงแถวที่ 120 ถึง 166 คือช่วงแถวที่ทอยางเคลื่อนที่
    for j=10:375                % ช่วงแถวที่ 10 ถึง 375 คือช่วงคอลัมน์ที่ทอยางเคลื่อนที่
        if a(i,j,1)>=115&a(i,j,1)<=250 & a(i,j,2)>=120&a(i,j,2)<=255 & a(i,j,3)>=50&a(i,j,3)<=240
                                % กำหนดจุดเปลี่ยนค่าของทอยาง
                                % ถ้าช่วงสี R ของทอยางอยู่ในช่วง  $115 \leq R \leq 250$ 
                                % และถ้าช่วงสี G ของทอยางอยู่ในช่วง  $120 \leq G \leq 255$ 
                                % และถ้าช่วงสี B ของทอยางอยู่ในช่วง  $50 \leq B \leq 240$ 
                                % ให้ b2 ที่ตำแหน่ง i,j มีค่า 255
            b2(i,j)=255;
        else                    % ไม่เช่นนั้นแล้ว
            b2(i,j)=0;        % ให้ b2 ที่ตำแหน่ง i,j มีค่า 0
        end
    end
end
for i=1:47                      % กำหนดช่วงแถวที่ทอยางเคลื่อนที่
    for j=1:366                % กำหนดช่วงคอลัมน์ที่ทอยางเคลื่อนที่
        b3(i,j)=b2(i+119,j+9); % ให้ b3 ที่ตำแหน่ง i,j มีค่าเท่ากับ b2 ที่ตำแหน่ง i+119,j+9
    end
end
[xxx,yyy] = size(b3);          % กำหนดดัชนีให้กับ b3
ij_cam1 = zeros(0,2);         % รีเซตตัวแปรที่ใช้เก็บข้อมูลตำแหน่งจุดภาพของทอยาง ij_cam1
y1 = 1;                       % เซต y1 = 1
y2 = 1;                       % เซต y2 = 1
for j=1:yyy
    for i=1:xxx
        if b3(i,j)==255;      % ถ้า b3 ที่ตำแหน่ง i,j มีค่า 255
            ij_cam1(y1,y2) = j; % เก็บค่า j ไว้ใน ij_cam1 ที่ตำแหน่ง y1,y2
            ij_cam1(y1,y2+1) = i; % เก็บค่า i ไว้ใน ij_cam1 ที่ตำแหน่ง y1,y2+1
            y1 = y1+1;        % เพิ่มค่า y1 ขึ้น 1
            y2 = 1;          % เซต y2 = 1
        end
    end
end
mina = min(ij_cam1,[],1);     % หาค่าแถวและคอลัมน์ของทอยางที่มีค่าน้อยที่สุด และเก็บไว้ใน mina
maxa = max(ij_cam1,[],1);     % หาค่าแถวและคอลัมน์ของทอยางที่มีค่ามากที่สุด และเก็บไว้ใน maxa
if maxa(1,2)~=mina(1,2);     % ถ้า maxa ≠ mina
    p = ((maxa(1,2)-mina(1,2))/2+mina(1,2));
                                % คำนวณตำแหน่งคอลัมน์ของทอยางจากความสัมพัทธ์
                                %  $\frac{\text{column}_{\max} - \text{column}_{\min}}{2} + \text{column}_{\min}$  แล้วเก็บไว้ใน p

```



```

        f(j) = round(p);           % ตำแหน่งคอลัมน์ของตัวอย่างที่ปิดเศษแล้ว
    end
    if maxa(1,2)==mina(1,2);     % ถ้า maxa = mina
        p = maxa(1,2);         % เก็บตำแหน่งคอลัมน์ของตัวอย่างไว้ใน p
        f(j) = round(p);       % ตำแหน่งคอลัมน์ของตัวอย่างที่ปิดเศษแล้ว
    end
    end
    ij_cam1 = zeros(0,2);       % วิเซตตัวแปรที่ใช้เก็บข้อมูลตำแหน่งจุดภาพของตัวอย่าง ij_cam1
    y1 = 1;                     % เซต y1 = 1
    y2 = 1;                     % เซต y2 = 1
end
for i=1:61                     % จัดเก็บข้อมูลไว้เพียง 61 จุดจาก 366 จุด
    k=i*6;                     %
    f1(i)=f(k);                % ข้อมูลตำแหน่งของตัวอย่าง 61 ข้อมูลถูกเก็บไว้ใน f1
end
for i=1:61
    y(i) = -1.6257*(f1(i)-f1(1)); % กำหนดตำแหน่งตัวอย่างในหน่วยมิลลิเมตร และเก็บไว้ใน y
                                % โดยกำหนดให้ข้อมูลที่ f1(1) เป็นตำแหน่งอ้างอิง
                                % (1.6257 มิลลิเมตร ได้มาจาก 595 มิลลิเมตร/366 จุดภาพ)
end
XX = 0:9.7541:9.7541*61;      % กำหนดช่วงตำแหน่ง x ของตัวอย่าง และเก็บไว้ใน XX
                                % (0.97541 ได้มาจาก 595 มิลลิเมตร/61 ข้อมูล)
YY = [0,y(1:59),0,0];        % เก็บตำแหน่ง y ของตัวอย่างไว้ใน YY
XXX_CAM1(iii,:)=XX;         % เก็บตำแหน่ง x ของตัวอย่างจากทุก ๆ ภาพไว้ใน XXX
YYY_CAM1(iii,:)=YY;         % เก็บตำแหน่ง y ของตัวอย่างจากทุก ๆ ภาพไว้ใน YYY
end

```

โปรแกรมสกัดข้อมูลพลวัตของระบบ VTS จากกล้องตัวที่ 2

```

% โปรแกรมสกัดข้อมูลพลวัตของระบบ VTS จากลำดับภาพที่ได้จากกล้องตัวที่ 2 ซึ่งพัฒนาด้วย MATLAB™
% พัฒนาโดยเฉชา พวงดาวเรือง (2005)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

clear all
n_file = 500;                 % กำหนด n_file เท่ากับจำนวนภาพในไฟล์วิดีโอ 500 ภาพ
start = 100000;              % กำหนดชื่อไฟล์ภาพเริ่มต้นด้วย 100000
for iii = 1:n_file           % เริ่มต้นสกัดข้อมูลพลวัตจากไฟล์ลำดับที่ 1 (100001)
    start1 = start+iii;      % กำหนดให้ตัวแปรชื่อไฟล์ start1 = start + iii
    str = num2str(start1);   % แปลงค่าตัวแปร start1 ให้เป็นตัวเลขสตริง
    str = strcat(str,'_undist_color_deacha.bmp');
                                % ผนวกตัวแปรชื่อไฟล์เข้ากับ .bmp (“ชื่อไฟล์” + “_undist_color_deacha.bmp”)
    a = imread(str);        % อ่านไฟล์ภาพ “ชื่อไฟล์.bmp” เก็บไว้ในตัวแปร a
% Start segmentation

```

```

for i=120:166 % ช่วงแถวที่ 117 ถึง 177 คือช่วงแถวที่ท่อยางเคลื่อนที่
    for j=10:375 % ช่วงแถวที่ 10 ถึง 375 คือช่วงคอลัมน์ที่ท่อยางเคลื่อนที่
        if a(i,j,1)>=75&a(i,j,1)<=250 & a(i,j,2)>=60&a(i,j,2)<=255 & a(i,j,3)>=20&a(i,j,3)<=170
            % กำหนดจุดเปลี่ยนค่าของท่อยาง
            % ถ้าช่วงสี R ของท่อยางอยู่ในช่วง  $75 \leq R \leq 250$ 
            % และถ้าช่วงสี G ของท่อยางอยู่ในช่วง  $60 \leq G \leq 255$ 
            % และถ้าช่วงสี B ของท่อยางอยู่ในช่วง  $20 \leq B \leq 170$ 
            b2(i,j)=255; % ให้ b2 ที่ตำแหน่ง i,j มีค่า 255
        else % ไม่เช่นนั้นแล้ว
            b2(i,j)=0; % ให้ b2 ที่ตำแหน่ง i,j มีค่า 0
        end
    end
end

for i=1:60 % กำหนดช่วงแถวที่ท่อยางเคลื่อนที่
    for j=1:366 % กำหนดช่วงคอลัมน์ที่ท่อยางเคลื่อนที่
        b3(i,j)=b2(i+119,j+9); % ให้ b3 ที่ตำแหน่ง i,j มีค่าเท่ากับ b2 ที่ตำแหน่ง i+119,j+9
    end
end

[xxx,yyy] = size(b3); % กำหนดดัชนีให้กับ b3
ij_cam2 = zeros(0,2); % เริ่ชดตัวแปรที่ใช้เก็บข้อมูลตำแหน่งจุดภาพของท่อยาง ij_cam2
y1 = 1; % เซต y1 = 1
y2 = 1; % เซต y2 = 1
for j=1:yyy
    for i=1:xxx
        if b3(i,j)==255; % ถ้า b3 ที่ตำแหน่ง i,j มีค่า 255
            ij_cam2(y1,y2) = j; % เก็บค่า j ไว้ใน ij_cam2 ที่ตำแหน่ง y1,y2
            ij_cam2(y1,y2+1) = i; % เก็บค่า i ไว้ใน ij_cam2 ที่ตำแหน่ง y1,y2+1
            y1 = y1+1; % เพิ่มค่า y1 ขึ้น 1
            y2 = 1; % เซต y2 = 1
        end
    end

    mina = min(ij_cam2,[],1); % หาค่าแถวและคอลัมน์ของท่อยางที่มีค่าน้อยที่สุด และเก็บไว้ใน mina
    maxa = max(ij_cam2,[],1); % หาค่าแถวและคอลัมน์ของท่อยางที่มีค่ามากที่สุด และเก็บไว้ใน maxa
    if maxa(1,2)~=mina(1,2); % ถ้า maxa ≠ mina
        p = ((maxa(1,2)-mina(1,2))/2+mina(1,2));
        % คำนวณตำแหน่งคอลัมน์ของท่อยางจากความสัมพันธ์
        %  $\frac{\text{column}_{\max} - \text{column}_{\min}}{2} + \text{column}_{\min}$  แล้วเก็บไว้ใน p
    end

    f(j) = round(p); % ตำแหน่งคอลัมน์ของท่อยางที่ปิดเศษแล้ว
end

if maxa(1,2)==mina(1,2); % ถ้า maxa = mina
    p = maxa(1,2); % เก็บตำแหน่งคอลัมน์ของท่อยางไว้ใน p
end

```

```

        f(j) = round(p);           % ตำแหน่งคอลัมน์ของตัวอย่างที่ปิดเศษแล้ว
    end
    ij_cam2 = zeros(0,2);        % รีเซตตัวแปรที่ใช้เก็บข้อมูลตำแหน่งจุดภาพของตัวอย่าง ij_cam2
    y1 = 1;                       % เซต y1 = 1
    y2 = 1;                       % เซต y2 = 1
end
for i=1:61                        % จัดเก็บข้อมูลไว้เพียง 61 จุดจาก 366 จุด
    k=i*6;
    f1(i)=f(k);                   % ข้อมูลตำแหน่งของตัวอย่าง 61 ข้อมูลถูกเก็บไว้ใน f1
end
for i=1:61
    z(i) = -1.6257*(f1(i)-f1(1)); % จำนวนตำแหน่งตัวอย่างในหน่วยมิลลิเมตร และเก็บไว้ใน z
                                % โดยกำหนดให้ข้อมูลที่ f1(1) เป็นตำแหน่งอ้างอิง
                                % (1.6257 มิลลิเมตร ได้มาจาก 595 มิลลิเมตร/366 จุดภาพ)
end
XX = 0:9.7541:9.7541 *61;        % กำหนดช่วงตำแหน่ง x ของตัวอย่าง และเก็บไว้ใน XX
                                % (0.97541 ได้มาจาก 595 มิลลิเมตร/61 ข้อมูล)
ZZ = [0,z(1:59),0,0];           % เก็บตำแหน่ง z ของตัวอย่างไว้ใน ZZ
XXX_CAM2(iii,:)=XX;            % เก็บตำแหน่ง x ของตัวอย่างจากทุก ๆ ภาพไว้ใน XXX
ZZZ_CAM2(iii,:)=ZZ;            % เก็บตำแหน่ง z ของตัวอย่างจากทุก ๆ ภาพไว้ใน ZZZ
end
end

```

ภาคผนวก ฅ

ระเบียบวิธีผลต่างสี่เหลี่ยม

ระเบียบวิธีผลต่างสี่เหลี่ยม

ระเบียบวิธีผลต่างสี่เหลี่ยม (finite difference method) สำหรับคำนวณผลเฉลยของแบบจำลองทางคณิตศาสตร์ของระบบ VTS ในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก มีรายละเอียดดังต่อไปนี้

จากบทที่ 9 แบบจำลองทางคณิตศาสตร์ของระบบ VTS ที่แสดงการเคลื่อนที่ของท่อภายในระนาบ x, y แสดงดังสมการที่ (ฉ-1)

$$\frac{\partial^2 y(x,t)}{\partial t^2} = \left(\frac{K_D K_M u}{\mu} \right) \left(\frac{\partial^2 y(x,t)}{\partial x^2} \right) - g \quad (\text{ฉ-1})$$

กำหนดให้ $y_{tt}(x,t) = \frac{\partial^2 y(x,t)}{\partial t^2}$, $y_{xx}(x,t) = \frac{\partial^2 y(x,t)}{\partial x^2}$, และ $c^2 = \frac{K_D K_M u}{\mu}$ เมื่อ μ คือมวลของท่อต่อความยาวหนึ่งหน่วย (กิโลกรัม/เมตร) g คือ ความเร่งเนื่องจากแรงโน้มถ่วง (เมตร/วินาที²) u คือ สัญญาณอินพุตของมอเตอร์ไฟฟ้ากระแสตรง (โวลต์) K_D คือ อัตราขยายที่ให้การประมาณแบบจำลองของส่วนเพิ่มกำลัง และ K_M คือ อัตราขยายที่ให้การประมาณแบบจำลองของมอเตอร์ไฟฟ้ากระแสตรง สมการที่ (ฉ-1) สามารถเขียนใหม่ได้ดังสมการที่ (ฉ-2) ที่ซึ่งความยาวของท่อ x อยู่ในช่วง 0 ถึง a และเวลา t อยู่ในช่วง 0 ถึง b

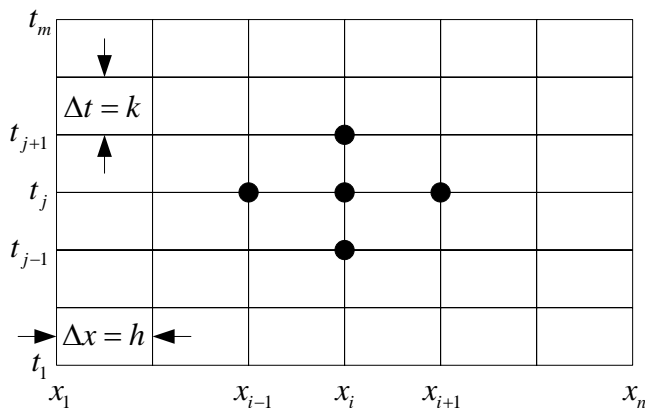
$$y_{tt}(x,t) = c^2 y_{xx}(x,t) - g, \quad 0 < x < a, \quad 0 < t < b \quad (\text{ฉ-2})$$

แบบจำลองในสมการที่ (ฉ-2) เป็นสมการคลื่น ในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก ที่พิจารณาผลของความเร่งเนื่องจากแรงโน้มถ่วง แบบจำลองดังกล่าวให้การอธิบายพลวัตของท่อภายในระนาบ x, y โดยที่ปลายด้านหนึ่งของท่ออยู่ที่ตำแหน่ง $x=0$ และปลายอีกด้านหนึ่งของท่ออยู่ที่ตำแหน่ง $x=a$ เงื่อนไขขอบเขตได้รับการกำหนดดังสมการที่ (ฉ-3)

$$\text{ที่ซึ่ง } y_t(x,t) = \frac{\partial y(x,t)}{\partial t}$$

$$\left. \begin{aligned} y(0,t) = 0, \quad y(a,t) = 0, \quad 0 \leq t \leq b \\ y(x,0) = f(x), \quad 0 \leq x \leq a \\ y_t(x,0) = g(x), \quad 0 < x < a \end{aligned} \right\} \quad (\text{ฉ-3})$$

กำหนดให้ $R = \{(x,t) : 0 \leq x \leq a, 0 \leq t \leq b\}$ เป็นกริด (grid) ที่ประกอบด้วยสี่เหลี่ยมจำนวน $n-1 \times m-1$ โดยที่ $\Delta x = h$ และ $\Delta t = k$ ดังแสดงในรูปที่ ฅ.1 เพื่อใช้คำนวณผลเฉลยของสมการคลื่น ผลเฉลยในแถวแรกที่เวลา $t = t_1 = 0$ คือ $y(x_i, t_1) = f(x_i)$ จะถูกใช้เพื่อคำนวณผลเฉลย $y(x_i, t_j)$ ในแถวถัดไป



รูปที่ ฅ.1 กริดสำหรับคำนวณผลเฉลยของสมการคลื่น

โดยอาศัยผลต่างแบบตรงกลาง (central difference) เพื่อประมาณเทอม $y_{tt}(x, t)$ และ $y_{xx}(x, t)$ ดังแสดงในสมการที่ (ฅ-4) และ (ฅ-5) ตามลำดับ

$$y_{tt}(x, t) = \frac{y(x, t+k) - 2y(x, t) + y(x, t-k)}{k^2} + O(k^2) \tag{ฅ-4}$$

$$y_{xx}(x, t) = \frac{y(x+h, t) - 2y(x, t) + y(x-h, t)}{h^2} + O(h^2) \tag{ฅ-5}$$

จากรูปที่ ฅ.1 กริดจะมีลักษณะที่สม่ำเสมอทั้งในส่วนของแถวและคอลัมน์ ทำให้ $x_{i+1} = x_i + h$, $x_{i-1} = x_i - h$, $t_{j+1} = t_j + k$, และ $t_{j-1} = t_j - k$ สมการที่ (ฅ-4) และ (ฅ-5) จะได้รับการประมาณโดยการละเลยเทอม $O(k^2)$ และ $O(h^2)$ และทำการแทน $y(x_i, t_j)$ ด้วย $y_{i,j}$ เมื่อนำสมการที่ (ฅ-4) และ (ฅ-5) แทนลงในสมการที่ (ฅ-2) จะได้

$$\left(\frac{y_{i,j+1} - 2y_{i,j} + y_{i,j-1}}{k^2} \right) = c^2 \left(\frac{y_{i+1,j} - 2y_{i,j} + y_{i-1,j}}{h^2} \right) - g \tag{ฅ-6}$$

กำหนดให้ $r = ck/h$ สมการที่ (ฉ-6) สามารถเขียนใหม่ได้ดังสมการที่ (ฉ-7)

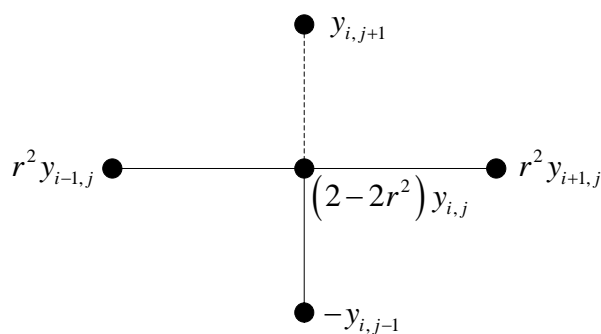
$$(y_{i,j+1} - 2y_{i,j} + y_{i,j-1}) = r^2(y_{i+1,j} - 2y_{i,j} + y_{i-1,j}) - gk^2 \quad (\text{ฉ-7})$$

สมการที่ (ฉ-7) จะถูกใช้เพื่อคำนวณผลเฉลยในแถว $j+1$ โดยอาศัยผลเฉลยที่ทราบแล้วในอดีตจากแถว j และ $j-1$ ดังแสดงในสมการที่ (ฉ-8) และรูปที่ ฉ.2 ตามลำดับ

$$y_{i,j+1} = (2 - 2r^2)y_{i,j} + r^2(y_{i+1,j} + y_{i-1,j}) - y_{i,j-1} - gk^2, \quad i = 2, 3, \dots, n-1 \quad (\text{ฉ-8})$$

$$y_{i,2} = (1 - r^2)f_i + kg_i + \frac{r^2}{2}(f_{i+1} - f_{i-1}), \quad i = 2, 3, \dots, n-1 \quad (\text{ฉ-9})$$

สมการที่ (ฉ-8) เป็นผลเฉลยในแถว $j = 3, 4, \dots, m$ ซึ่งจะมีความถูกต้องแม่นยำก็ต่อเมื่อ $r = ck/h \leq 1$ (Mathews and Fink, 2004) สำหรับผลเฉลยในแถว $j=1$ คือ $y_{i,1} = f(x)$ และผลเฉลยในแถว $j=2$ คือ $y_{i,2}$ ดังแสดงในสมการที่ (ฉ-9) ผู้อ่านสามารถศึกษารายละเอียดของระเบียบวิธีผลต่างสี่เหลี่ยม เพื่อคำนวณผลเฉลยของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก จากตำราการคำนวณเชิงตัวเลขทั่วไป เช่น ปราโมทย์ เคะอำไพ (2544) และ Mathews และ Fink (2004) เป็นต้น



รูปที่ ฉ.2 การคำนวณผลเฉลย $y_{i,j+1}$

สำหรับแบบจำลองทางคณิตศาสตร์ของระบบ VTS ที่แสดงการเคลื่อนที่ของท่อภายในระนาบ z, x แสดงดังสมการที่ (ฉ-10)

$$\frac{\partial^2 z(x,t)}{\partial t^2} = \left(\frac{K_D K_M u}{\mu} \right) \left(\frac{\partial^2 z(x,t)}{\partial x^2} \right) \quad (\text{ฉ-10})$$

กำหนดให้ $z_{tt}(x,t) = \frac{\partial^2 z(x,t)}{\partial t^2}$, $z_{xx}(x,t) = \frac{\partial^2 z(x,t)}{\partial x^2}$, และ $c^2 = \frac{K_D K_M u}{\mu}$ สมการที่ (ฉ-10) สามารถเขียนใหม่ได้ตั้งสมการที่ (ฉ-11) ที่ซึ่งความยาวของท่ออย่าง x อยู่ในช่วง 0 ถึง a และเวลา t อยู่ในช่วง 0 ถึง b แบบจำลองในสมการที่ (ฉ-11) เป็นสมการคลื่น ในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก ที่ไม่พิจารณาผลของความเร่งเนื่องจากแรงโน้มถ่วง แบบจำลองดังกล่าวให้การอธิบายพลวัตของท่อภายในระนาบ z, x โดยที่เงื่อนไขขอบเขตได้รับการกำหนดตั้งสมการที่ (ฉ-12) ที่ซึ่ง $z_t(x,0) = \frac{\partial z(x,t)}{\partial t}$

$$z_{tt}(x,t) = c^2 z_{xx}(x,t), \quad 0 < x < a, \quad 0 < t < b \quad (\text{ฉ-11})$$

$$\left. \begin{aligned} z(0,t) = 0, \quad z(a,t) = 0, \quad 0 \leq t \leq b \\ z(x,0) = f(x), \quad 0 \leq x \leq a \\ z_t(x,0) = g(x), \quad 0 < x < a \end{aligned} \right\} \quad (\text{ฉ-12})$$

โดยประยุกต์สมการที่ (ฉ-8) และ (ฉ-9) ผลเฉลยของแบบจำลองในสมการที่ (ฉ-11) แสดงตั้งสมการที่ (ฉ-13) และ (ฉ-14) ตามลำดับ

$$z_{i,j+1} = (2 - 2r^2)z_{i,j} + r^2(z_{i+1,j} + z_{i-1,j}) - z_{i,j-1}, \quad i = 2, 3, \dots, n-1 \quad (\text{ฉ-13})$$

$$z_{i,2} = (1 - r^2)f_i + kg_i + \frac{r^2}{2}(f_{i+1} - f_{i-1}), \quad i = 2, 3, \dots, n-1 \quad (\text{ฉ-14})$$

ในการระบุเอกลักษณ์ระบบ VTS โดยอาศัยการค้นหาแบบตาบู่เชิงปรับตัว ค่าพารามิเตอร์ μ , K_D , และ K_M จะได้รับการค้นหา การกำหนดขอบเขตปริภูมิการค้นหาของค่าพารามิเตอร์แต่ละตัว จะต้องสอดคล้องกับเงื่อนไข $r = ck/h \leq 1$ โดยที่ $c = \sqrt{(K_D K_M u)/\mu}$ ทั้งนี้เพื่อให้ผลเฉลยที่ได้จากการคำนวณตามระเบียบวิธีผลต่างสืบเนื่องมีความถูกต้องแม่นยำมากที่สุด

ภาคผนวก ๑

โปรแกรมระบุเอกลักษณ์ระบบ VTS

โปรแกรมระบุเอกลักษณ์ระบบ VTS

โปรแกรมระบุเอกลักษณ์ระบบ VTS แบ่งออกเป็นสองส่วนคือ โปรแกรมระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y และโปรแกรมระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x โดยอาศัยการค้นหาแบบตามเชิงปรับตัวเพื่อค้นหาค่าพารามิเตอร์ที่เหมาะสม การคำนวณผลเฉลยของแบบจำลองในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิกอาศัยระเบียบวิธีผลต่างสี่เหลี่ยม โปรแกรมระบุเอกลักษณ์ระบบ VTS ทั้งสองส่วนได้รับการพัฒนาขึ้นโดยโปรแกรม MATLAB ซึ่งมีรายละเอียดดังต่อไปนี้

โปรแกรมระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y

% โปรแกรมนี้เป็นโปรแกรมระบุเอกลักษณ์ระบบ VTS ในระนาบ x, y
% โดยอาศัยการค้นหาแบบตามเชิงปรับตัว
% พัฒนาโดยเฉชา พวงดาวเรือง (2005)
% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรม VTS_ATS.m

```
function [best_neighbor,best_error,time,radius,count,local]=VTS_ATS(N)
% โปรแกรมหลัก มี N เป็นอินพุตของโปรแกรม
% N คือจำนวนพารามิเตอร์ที่ต้องการค้นหา (N = 3 คือ  $K_D$ ,  $K_M$ , และ  $\mu$ )
% เอาต์พุตของโปรแกรมคือ
% ค่าพารามิเตอร์ที่ดีที่สุดหรือค่าคำตอบวงกว้าง (best_neighbor)
% ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด (best_error)
% ค่าเวลาที่ใช้ในการค้นหา (time), ค่ารัศมีการค้นหา (radius)
% ค่าจำนวนรอบการค้นหาทั้งหมด (count)
% ค่าคำตอบวงแคบเฉพาะถิ่น (local)
xlimit=[0.25 0.25 0.1;0 0 0]; % กำหนดปริภูมิการค้นหาของพารามิเตอร์
% แถวที่ 1 ของ xlimit คือขอบเขตบนของพารามิเตอร์  $K_D$ ,  $K_M$ , และ  $\mu$ 
% แถวที่ 2 ของ xlimit คือขอบเขตล่างของพารามิเตอร์  $K_D$ ,  $K_M$ , และ  $\mu$ 
for r=1:5
    S(r,:)=(xlimit(1,:)-xlimit(2,:)).*rand(1,N)+xlimit(2,:);
end % สุ่มคำตอบเริ่มต้นภายในปริภูมิการค้นหาของพารามิเตอร์แต่ละตัว
% จำนวน 5 ชุด
% โดยอาศัยความสัมพัทธ์  $(x_{h,i} - x_{l,i}) \text{rand} + x_{l,i}$ , for  $i = 1, \dots, 5$ 
for k=1:size(S,1)
```

```

ysim=Obj_VTS(S(k,1),S(k,2),S(k,3));
costvalue(k,1)=ysim;
end % ประเมินค่าตอบเริ่มต้นด้วยฟังก์ชันวัตถุประสงค์ใน โปรแกรม Obj_VTS
% อินพุตของโปรแกรม Obj_VTS คือค่าตอบเริ่มต้นที่สุ่มได้จำนวน 5 ชุด
% เอาต์พุตคือค่าฟังก์ชันวัตถุประสงค์ที่ได้จากคำตอบแต่ละชุด
% เก็บค่าฟังก์ชันวัตถุประสงค์ทั้ง 5 ค่าไว้ใน costvalue
[best_error,index]=min(costvalue); % หาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดใน costvalue
% เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดไว้ใน best_error
S0=S(index,:); % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุดไว้ใน S0
max_count=1000; % กำหนดจำนวนรอบสูงสุดในการค้นหาเท่ากับ 1,000 รอบ
% เพื่อใช้เป็นเกณฑ์ยุติการค้นหา
best_neighbor=S0; % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
% ไว้ใน best_neighbor
neighbor_list=zeros(5,N); % รีเซ็ตค่า neighbor_list
radius=0.1; % กำหนดรัศมีการค้นหาเริ่มต้นเท่ากับ 10%
Number_neighb=30; % กำหนดจำนวนการสุ่มค่าใกล้เคียงเท่ากับ 30 ตัว
overall_best_error=best_error; % ปรับค่า overall_best_error = best_error
overall_neighbor=best_neighbor; % ปรับค่า overall_neighbor = best_neighbor
n=0; % กำหนดค่าเริ่มต้นของจำนวนการซ้ำของค่าคำตอบ
t=0; % กำหนดค่าตัวเลื่อนสำหรับ local
tt=0; % กำหนดค่าตัวเลื่อนสำหรับ tabu_list
ttt=0; % กำหนดค่าตัวเลื่อนสำหรับ best_error_list
count=0; % กำหนดค่าเริ่มต้นของจำนวนรอบการค้นหา
n_back_tracking=0; % กำหนดค่าเริ่มต้นของจำนวนการเรียกใช้กลไกย้อนรอยการค้นหา
tic; % เริ่มต้นคำนวณเวลาที่ใช้ในการค้นหา
% Start ATS
t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
ttt=ttt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ best_error_list
disp([count best_error overall_best_error])
% แสดงค่า count, best_error, และ overall_best_error
local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
local(t,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ local
local(t,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ local
tabu_list(tt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ tabu_list
tabu_list(tt,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ tabu_list
tabu_list(tt,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ tabu_list
best_error_list(ttt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ best_error_list
best_error_list(ttt,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ best_error_list
best_error_list(ttt,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ best_error_list
for count=1:max_count % วนรอบการค้นหา

```

```

S1=random_neigh(Number_neighb,radius,xlimit,S0);
% เรียกใช้โปรแกรม random_neigh เพื่อสุ่มค่าใกล้เคียงรอบคำตอบ S0
% ภายในปริภูมิการค้นหาย่อยปัจจุบัน
% อินพุตของโปรแกรมนี้คือ Number_neighb, radius, xlimit, และ S0
% เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ Number_neighb (30 ตัว)
[best_error1,best_neighbor1,best_error,best_neighbor]=Objective_VTS(S1,best_error,S0);
% เรียกใช้โปรแกรม Objective_VTS เพื่อประเมินค่าใกล้เคียงที่สุ่มได้
% อินพุตของโปรแกรมนี้คือ S1, best_error, และ S0
% เอาต์พุตคือ best_error1, best_neighbor1,
% best_error, และ best_neighbor
neighbor_list(k,1:size(S0,2))=[best_neighbor1];
% เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
% ภายในปริภูมิการค้นหาย่อยปัจจุบันไว้ในคอลัมน์ที่ 1 ถึง 3 ของ neighbor_list
neighbor_list(k,size(S0,2)+1)=best_error1;
% เก็บค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
% ภายในปริภูมิการค้นหาย่อยปัจจุบันไว้ในคอลัมน์ที่ 4 ของ neighbor_list
if (best_error1-best_error)>1e-18
    n=n+1;
else
    n=0;
end
% ตรวจสอบการซ้ำของค่าคำตอบโดยการเปรียบเทียบระหว่าง
% best_error1 และ best_error ถ้ามีการซ้ำของค่าคำตอบ ให้ปรับเพิ่มค่า n
tt=tt+1;
% ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
tabu_list(tt,1)=count;
% เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ tabu_list
tabu_list(tt,2:4)=best_neighbor1;
% เก็บค่า best_neighbor1 ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ tabu_list
tabu_list(tt,5)=best_error1;
% เก็บค่า best_error1 ไว้ในคอลัมน์ที่ 5 ของ tabu_list
ttt= ttt+1;
% ปรับเพิ่มค่าตัวเลื่อนสำหรับ best_error_list
best_error_list(ttt,1)=count;
% เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ best_error_list
best_error_list(ttt,2:4)=best_neighbor;
% เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ best_error_list
best_error_list(ttt,5)=best_error;
% เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ best_error_list
disp([count best_error overall_best_error])
% แสดงค่า count, best_error, และ overall_best_error

% Start AR mechanism
if best_error<=25.0
% เงื่อนไขที่ 1) ถ้า best_error ≤ 25.0 แล้ว radius = 20% ของ radius ก่อนหน้า
    radius=0.2*radius;
end
if best_error<=24.5
% เงื่อนไขที่ 2) ถ้า best_error ≤ 24.5 แล้ว radius = 20% ของ radius ก่อนหน้า
    radius=0.2*radius;
end
if best_error<=24.0
% เงื่อนไขที่ 3) ถ้า best_error ≤ 24.0 แล้ว radius = 20% ของ radius ก่อนหน้า

```

```

radius=0.2*radius;
end
if (best_error<23.26) % ตรวจสอบเกณฑ์ยุติการค้นหา เมื่อ best_error < 23.26
    t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
    tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
    disp([count best_error overall_best_error])
    % แสดงค่า count, best_error, และ overall_best_error
    local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
    local(t,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ local
    local(t,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ local
    break; % ยุติการค้นหา
end
% Start BT mechanism
% เมื่อจำนวนการซ้ำของค่าคำตอบเท่ากับ 5
if n>=5
    n_back_tracking=n_back_tracking+1;
    % เพิ่มค่า n_back_tracking
    TEMP=tabu_list(count-3:count+1,:);
    % ทำการจัดลำดับ (ranking) ค่าฟังก์ชันวัตถุประสงค์
    % และค่าพารามิเตอร์ 5 ชุดสุดท้ายก่อนมีการเรียกใช้กลไกย้อนรอยการค้นหา
    % TEMP จะเก็บค่าฟังก์ชันวัตถุประสงค์ และค่าพารามิเตอร์ 5 ชุดสุดท้าย
    % ก่อนทำการจัดลำดับ
    [MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
    RANK(5,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 5 ใน RANK
    TEMP(INDEX,4)=0; % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 4 ค่า
    [MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
    RANK(4,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 4 ใน RANK
    TEMP(INDEX,4)=0; % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 3 ค่า
    [MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
    RANK(3,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 3 ใน RANK
    TEMP(INDEX,4)=0; % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์

```

```

% ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 2 ค่า
[MAX,INDEX] = max(TEMP(:,4));
% หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(2,:) = TEMP(INDEX,:);
% จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
% และค่าพารามิเตอร์ไว้เป็นลำดับที่ 2 ใน RANK
TEMP(INDEX,4)= 0; % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
% ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 1 ค่า
[MAX,INDEX] = max(TEMP(:,4));
% หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(1,:) = TEMP(INDEX,:);
% จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
% และค่าพารามิเตอร์ไว้เป็นลำดับที่ 1 ใน RANK
TEMP(INDEX,4)= 0; % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
% ดังนั้นใน TEMP ขณะนี้จะไม่มีค่าฟังก์ชันวัตถุประสงค์ให้ทำการจัดลำดับอีกต่อไป
neighbor=RANK(5,2:4); % แทนค่า neighbor ด้วยค่าพารามิเตอร์ลำดับที่ 5 ใน RANK
% ซึ่งเป็นค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์แตกต่างจาก
% ค่าคำตอบที่ซ้ำมากที่สุดเมื่อเทียบกับอีก 4 ลำดับที่เหลือใน RANK
S0=neighbor; % ปรับค่า S0 = neighbor
if best_error<overall_best_error
    overall_best_error=best_error;
    overall_best_neighbor=best_neighbor;
    % ปรับค่า overall_best_error
    % ถ้า best_error < overall_best_error
    % ให้แทนค่า overall_best_error ด้วย best_error
    % และแทนค่าพารามิเตอร์ overall_best_neighbor ด้วย best_neighbor
    t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
    local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
    local(t,2:4)=best_neighbor;
    % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ local
    local(t,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ local
end
best_error=RANK(5,5); % แทนค่า best_error ด้วยค่าฟังก์ชันวัตถุประสงค์ลำดับที่ 5 ใน RANK
% ซึ่งเป็นค่าฟังก์ชันวัตถุประสงค์ที่แตกต่างจาก
% ค่าคำตอบที่ซ้ำมากที่สุดเมื่อเทียบกับอีก 4 ลำดับที่เหลือใน RANK
n=0; % ปรับค่า n = 0
else % ไม่เช่นนั้นแล้ว
    S0=best_neighbor; % แทนค่า S0 ด้วย best_neighbor
    best_error=best_error;% แทนค่า best_error ด้วย best_error
end
end
end

```

```

if overall_best_error<best_error
    best_error=overall_best_error;
    best_neighbor=overall_best_neighbor;

    % ปรับค่า best_error
    % ถ้า overall_best_error < best_error
    % ให้แทนค่า best_error ด้วย overall_best_error
    % และแทนค่าพารามิเตอร์ best_neighbor ด้วย overall_best_neighbor

end

time=toc; % ยุติการคำนวณเวลาการค้นหา
return

```

โปรแกรม Obj_VTS.m

```

function ysim=Obj_VTS(km,kd,mu) % โปรแกรม Obj_VTS คือโปรแกรมคำนวณค่าฟังก์ชันวัตถุประสงค์
% ซึ่งเป็นการคำนวณผลเฉลยของแบบจำลองระบบ VTS ในระนาบ  $x, y$ 
% ในรูปของสมการอนุพันธ์อันดับสองแบบไฮเพอร์โบลิก
% ที่พิจารณาผลของความเร่งเนื่องจากแรงโน้มถ่วง
% ด้วยระเบียบวิธีผลต่างสี่เหลี่ยม (ดูรายละเอียดในภาคผนวก ฉ.)
% อินพุตของโปรแกรมคือคำตอบหรือค่าพารามิเตอร์เริ่มต้นที่สุ่มได้
% km คือ  $K_M$ , kd คือ  $K_D$  และ mu คือ  $\mu$  ของแบบจำลองระบบ VTS
% เอาต์พุต ysim คือค่าฟังก์ชันวัตถุประสงค์ที่ได้จากคำตอบ km, kd, และ mu

load VTS_DATA_AFTER_CALIBRATION_XYZ % โหลดข้อมูลอินพุตและเอาต์พุตของระบบ VTS ในระนาบ  $x, y$ 

h=0.0097541; % ค่าระยะของข้อมูลตามแนวแกน  $x$  ซึ่ง  $h = \Delta x$ 
k=0.04; % ค่าระยะของข้อมูลตามแนวแกน  $t$  ซึ่ง  $k = \Delta t$ 
g=9.81; % ความเร่งเนื่องจากแรงโน้มถ่วง
% จำนวนคำตอบตามระเบียบวิธีผลต่างสี่เหลี่ยม

c=(km*kd*u/mu).^0.5; % จำนวนค่า  $c = \sqrt{(K_D K_M u) / \mu}$ 
r=c*k/h; % จำนวนค่า  $r = ck / h$ 
if (r>1.0) % ถ้า  $r > 1.0$ 
    ysim=5000; % กำหนดให้ส่ง ysim ที่มีค่ามาก ๆ (5,000) กลับสู่โปรแกรมหลัก
    return
end
for t=1:40
    Y(1,t)=0; % จำนวนเงื่อนไขว้ขอบเขต
    Y(59,t)=0;
end
for x=2:59
    Y(x,1)=YYY_CAM1(24,x+1); % จำนวนคำตอบแถวที่ 1
end
for x=2:58
    % จำนวนคำตอบแถวที่ 2

```

```

Y(x,2)=((1-r*r)*Y(x,1))+(r*r*(Y(x-1,1)+Y(x+1,1))/2)-g*k*k;
end
for t=3:40 % จำนวนคำตอบแถวที่ 3 ถึง 40
    for x=2:58
        Y(x,t)=((2-2*r*r)*Y(x,t-1))+r*r*(Y(x+1,t-1)+Y(x-1,t-1))-Y(x,t-2)-g*k*k;
    end
end
Y1=Y';
for i=24:63
    for j=2:60
        Y_New(i-23,j-1)=y(i,j); % จัดสรรข้อมูลจากการทดสอบเพื่อเปรียบเทียบกับผลจากแบบจำลอง
    end
end
ss_er=sum(sum((Y1-Y_New).^2)); % จำนวน SSE
ms_er=ss_er/(length(i)*length(j)); % จำนวน MSE
ysim=ms_er; % ส่งเอาต์พุต ysim เท่ากับ MSE กลับสู่โปรแกรมหลัก
return

```

โปรแกรม Objective_VTS.m

```

function [best_error1,best_neighbor1,best_error,best_neighbor]=Objective_VTS(S1,best_error,S0)
% โปรแกรม Objective_VTS เป็นโปรแกรมประเมินค่าใกล้เคียงที่สุ่มได้รอบ S0
% อินพุตของโปรแกรมนี้คือ S1, best_error, และ S0
% เอาต์พุตคือ best_error1, best_neighbor1,
% best_error, และ best_neighbor
error=[]; % ตัวแปร error สำหรับรองรับค่าฟังก์ชันวัตถุประสงค์
% ของค่าใกล้เคียงที่ได้จากการสุ่ม
for k=1:size(S1,1)
    ysim=Obj_VTS(S1(k,1),S1(k,2),S1(k,3));
    % ประเมินค่าใกล้เคียงใน S1 ด้วยฟังก์ชันวัตถุประสงค์ในโปรแกรม Obj_VTS
    error(k,1)=ysim; % เก็บค่าฟังก์ชันวัตถุประสงค์ของค่าใกล้เคียงทั้งหมด
end
[best_error1,index]=min(error); % หาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในบรรดาค่าใกล้เคียง
% และเก็บไว้ใน best_error1
best_neighbor1=S1(index,:); % เก็บค่าพารามิเตอร์ของค่าใกล้เคียงที่ทำให้
% ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุดไว้ใน best_neighbor1
if best_error1<best_error % ปรับค่า best_error
    best_error=best_error1; % ถ้า best_error1 < best_error
    % ให้แทนค่า best_error ด้วย best_error1
    best_neighbor=S1(index,:); % และแทนค่าพารามิเตอร์ best_neighbor ด้วย S1(index,:)
else % ไม่เช่นนั้นแล้ว

```



```

        best_neighbor=S0;          % แทนค่า best_neighbor ด้วย S0
    end
return

```

โปรแกรม rand1.m

```

function x=rand1(a,b)           % โปรแกรม rand1 เป็นโปรแกรมสุ่มค่าพารามิเตอร์ในช่วงที่กำหนด
                                % อินพุตของโปรแกรมคือขอบเขตบนของพารามิเตอร์ (b) และ
                                % ขอบเขตล่างของพารามิเตอร์ (a)
                                % โดยอาศัยความสัมพันธ์  $(b-a)\text{rand} + a$ 
                                % เอาต์พุตของโปรแกรมคือผลของการสุ่มค่าพารามิเตอร์ในช่วง a ถึง b

    x=a+rand*(b-a);
return

```

โปรแกรม random_neigh.m

```

function S1=random_neigh(Number_neighb,radius,xlimit,S0)
                                % โปรแกรม random_neigh เป็นโปรแกรมสุ่มค่าใกล้เคียงรอบคำตอบ S0
                                % อินพุตของโปรแกรมนี้คือ Number_neighb, radius, xlimit, และ S0
                                % เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ Number_neighb

    for u=1:Number_neighb
        for k=1:size(xlimit,2)
            S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
                                % ทำการสุ่มค่าใกล้เคียงโดยอาศัยความสัมพันธ์
                                %  $S_{1,i} = S_0 + (\text{radius} \cdot \text{rand}_1(-1,1) \cdot (x_{h,i} - x_{l,i}))$ ,
                                % for  $i = 1, \dots, \text{Number\_neighb}$ 
                                % เรียกใช้โปรแกรม rand1 โดยให้ทำการสุ่มค่าในช่วง -1 ถึง 1
                                % นั่นหมายความว่า การสุ่มมีพื้นที่อยู่ในช่วง  $S_0 \pm (\text{radius} \cdot (x_{h,i} - x_{l,i}))$ 

            while ( S1(u,k)>xlimit(1,k) | S1(u,k)<xlimit(2,k) )
                % ค่าใกล้เคียงที่สุ่มได้จะต้องอยู่ในขอบเขตบนและขอบเขตล่าง
                % ของค่าพารามิเตอร์ ถ้าค่าใกล้เคียงที่สุ่มได้
                % มีค่าเกินขอบเขตบนและขอบเขตล่างที่กำหนด
                % ให้ทำการสุ่มค่าใกล้เคียงใหม่

                S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
            end
        end
    end
return

```

โปรแกรมระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x

% โปรแกรมนี้เป็นโปรแกรมระบุเอกลักษณ์ระบบ VTS ในระนาบ z, x

% โดยอาศัยการค้นหาแบบตามเชิงปรับตัว

% พัฒนาโดยเดชา พวงดาวเรือง (2005)

% สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรม VTS_ATS.m

```
function [best_neighbor,best_error,time,radius,count,local]=VTS_ATS(N)
% โปรแกรมหลัก มี N เป็นอินพุตของโปรแกรม
% N คือจำนวนพารามิเตอร์ที่ต้องการค้นหา (N = 3 คือ  $K_D$ ,  $K_M$ , และ  $\mu$ )
% เอาต์พุตของโปรแกรมคือ
% ค่าพารามิเตอร์ที่ดีที่สุดหรือค่าคำตอบวงกว้าง (best_neighbor)
% ค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด (best_error)
% ค่าเวลาที่ใช้ในการค้นหา (time)
% ค่ารัศมีการค้นหา (radius)
% ค่าจำนวนรอบการค้นหาทั้งหมด (count)
% ค่าคำตอบวงแคบเฉพาะถิ่น (local)
xlimit=[0.25 0.25 0.1;0 0 0]; % กำหนดปริภูมิการค้นหาของพารามิเตอร์
% แถวที่ 1 ของ xlimit คือขอบเขตบนของพารามิเตอร์  $K_D$ ,  $K_M$ , และ  $\mu$ 
% แถวที่ 2 ของ xlimit คือขอบเขตล่างของพารามิเตอร์  $K_D$ ,  $K_M$ , และ  $\mu$ 
for r=1:5
    S(r,:)=(xlimit(1,:)-xlimit(2,:)).*rand(1,N)+xlimit(2,:);
end % สุ่มคำตอบเริ่มต้นภายในปริภูมิการค้นหาของพารามิเตอร์แต่ละตัว
% จำนวน 5 ชุด
% โดยอาศัยความสัมพันธ์  $(x_{h,i} - x_{l,i}) \text{rand} + x_{l,i}$ , for  $i = 1, \dots, 5$ 
for k=1:size(S,1)
    zsim=Obj_VTS(S(k,1),S(k,2),S(k,3));
    costvalue(k,1)=zsim;
end % ประเมินคำตอบเริ่มต้นด้วยฟังก์ชันวัตถุประสงค์ในโปรแกรม Obj_VTS
% อินพุตของโปรแกรม Obj_VTS คือคำตอบเริ่มต้นที่สุ่มได้จำนวน 5 ชุด
% เอาต์พุตคือค่าฟังก์ชันวัตถุประสงค์ที่ได้จากคำตอบแต่ละชุด
% เก็บค่าฟังก์ชันวัตถุประสงค์ทั้ง 5 ค่าไว้ใน costvalue
[best_error,index]=min(costvalue); % หาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดใน costvalue
% เก็บค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดไว้ใน best_error
S0=S(index,:); % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุดใน S0
max_count=1000; % กำหนดจำนวนรอบสูงสุดในการค้นหาเท่ากับ 1,000 รอบ
% เพื่อใช้เป็นเกณฑ์ยุติการค้นหา
best_neighbor=S0; % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
% ไว้ใน best_neighbor
neighbor_list=zeros(5,N); % รีเซ็ตค่า neighbor_list
```

```

radius=0.1; % กำหนดรัศมีการค้นหาเริ่มต้นเท่ากับ 10%
Number_neighb=30; % กำหนดจำนวนการสุ่มค่าใกล้เคียงเท่ากับ 30 ตัว
overall_best_error=best_error; % ปรับค่า overall_best_error = best_error
overall_neighbor=best_neighbor; % ปรับค่า overall_neighbor = best_neighbor
n=0; % กำหนดค่าเริ่มต้นของจำนวนการซ้ำของค่าคำตอบ
t=0; % กำหนดค่าตัวเลื่อนสำหรับ local
tt=0; % กำหนดค่าตัวเลื่อนสำหรับ tabu_list
ttt=0; % กำหนดค่าตัวเลื่อนสำหรับ best_error_list
count=0; % กำหนดค่าเริ่มต้นของจำนวนรอบการค้นหา
n_back_tracking=0; % กำหนดค่าเริ่มต้นของจำนวนการเรียกใช้กลไกย้อนรอยการค้นหา
tic; % เริ่มต้นคำนวณเวลาที่ใช้ในการค้นหา

% Start ATS
t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
ttt=ttt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ best_error_list
disp([count best_error overall_best_error])
% แสดงค่า count, best_error, และ overall_best_error
local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
local(t,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ local
local(t,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ local
tabu_list(tt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ tabu_list
tabu_list(tt,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ tabu_list
tabu_list(tt,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ tabu_list
best_error_list(ttt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ best_error_list
best_error_list(ttt,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ best_error_list
best_error_list(ttt,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ best_error_list
for count=1:max_count % วนรอบการค้นหา
    S1=random_neigh(Number_neighb,radius,xlimit,S0);
    % เรียกใช้โปรแกรม random_neigh เพื่อสุ่มค่าใกล้เคียงรอบคำตอบ S0
    % ภายในปฏิบัติการค้นหาอยู่ปัจจุบัน
    % อินพุตของโปรแกรมนี้คือ Number_neighb, radius, xlimit, และ S0
    % เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ Number_neighb (30 ตัว)
    [best_error1,best_neighbor1,best_error,best_neighbor]=Objective_VTS(S1,best_error,S0);
    % เรียกใช้โปรแกรม Objective_VTS เพื่อประเมินค่าใกล้เคียงที่สุ่มได้
    % อินพุตของโปรแกรมนี้คือ S1, best_error, และ S0
    % เอาต์พุตคือ best_error1, best_neighbor1,
    % best_error, และ best_neighbor
    neighbor_list(k,1:size(S0,2))=[best_neighbor1];
    % เก็บค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุด
    % ภายในปฏิบัติการค้นหาอยู่ปัจจุบันไว้ในคอลัมน์ที่ 1 ถึง 3 ของ neighbor_list
    neighbor_list(k,size(S0,2)+1)=best_error1;

```

```

% เก็บค่าฟังก์ชันวัตถุประสงค์ที่มีค่าน้อยที่สุด
% ภายในปริภูมิการค้นหาของปัจจุบันไว้ในคอลัมน์ที่ 4 ของ neighbor_list

if (best_error1-best_error)>1e-18
    n=n+1;
else
    n=0;
end
% ตรวจสอบการซ้ำของค่าคำตอบโดยการเปรียบเทียบระหว่าง
% best_error1 และ best_error ถ้ามีการซ้ำของค่าคำตอบ ให้ปรับเพิ่มค่า n
tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
tabu_list(tt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ tabu_list
tabu_list(tt,2:4)=best_neighbor1; % เก็บค่า best_neighbor1 ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ tabu_list
tabu_list(tt,5)=best_error1; % เก็บค่า best_error1 ไว้ในคอลัมน์ที่ 5 ของ tabu_list
ttt= ttt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ best_error_list
best_error_list(ttt,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ best_error_list
best_error_list(ttt,2:4)=best_neighbor;
% เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ best_error_list
best_error_list(ttt,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ best_error_list
disp([count best_error overall_best_error])
% แสดงค่า count, best_error, และ overall_best_error

% Start AR mechanism
if best_error<=34.0 % เงื่อนไขที่ 1) ถ้า best_error ≤ 34.0 แล้ว radius = 20% ของ radius ก่อนหน้า
    radius=0.2*radius;
end
if best_error<=33.0 % เงื่อนไขที่ 2) ถ้า best_error ≤ 33.0 แล้ว radius = 20% ของ radius ก่อนหน้า
    radius=0.2*radius;
end
if best_error<=32.5 % เงื่อนไขที่ 3) ถ้า best_error ≤ 32.5 แล้ว radius = 20% ของ radius ก่อนหน้า
    radius=0.2*radius;
end
if (best_error<32.38) % ตรวจสอบเกณฑ์ยุติการค้นหา เมื่อ best_error < 32.38
    t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
    tt=tt+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ tabu_list
    disp([count best_error overall_best_error])
    % แสดงค่า count, best_error, และ overall_best_error
    local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
    local(t,2:4)=best_neighbor; % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ local
    local(t,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ local
    break; % ยุติการค้นหา
end

% Start BT mechanism
% เมื่อจำนวนการซ้ำของค่าคำตอบเท่ากับ 5
if n>=5

```

```

n_back_tracking=n_back_tracking+1;
    % เพิ่มค่า n_back_tracking
TEMP=tabu_list(count-3:count+1,:);
    % ทำการจัดลำดับ (ranking) ค่าฟังก์ชันวัตถุประสงค์
    % และค่าพารามิเตอร์ 5 ชุดสุดท้ายก่อนมีการเรียกใช้กลไกย้อนรอยการค้นหา
    % TEMP จะเก็บค่าฟังก์ชันวัตถุประสงค์ และค่าพารามิเตอร์ 5 ชุดสุดท้าย
    % ก่อนทำการจัดลำดับ
[MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(5,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 5 ใน RANK
TEMP(INDEX,4)= 0;
    % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 4 ค่า
[MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(4,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 4 ใน RANK
TEMP(INDEX,4)= 0;
    % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 3 ค่า
[MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(3,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 3 ใน RANK
TEMP(INDEX,4)= 0;
    % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 2 ค่า
[MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(2,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 2 ใน RANK
TEMP(INDEX,4)= 0;
    % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์
    % ดังนั้นใน TEMP ขณะนี้จะเหลือค่าฟังก์ชันวัตถุประสงค์เพียง 1 ค่า
[MAX,INDEX] = max(TEMP(:,4));
    % หากค่าฟังก์ชันวัตถุประสงค์สูงสุด และลำดับที่ของค่าฟังก์ชันวัตถุประสงค์สูงสุด
RANK(1,:) = TEMP(INDEX,:);
    % จัดลำดับค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุด
    % และค่าพารามิเตอร์ไว้เป็นลำดับที่ 1 ใน RANK
TEMP(INDEX,4)= 0;
    % แทนค่าฟังก์ชันวัตถุประสงค์ที่มีค่าสูงสุดใน TEMP ด้วยศูนย์

```

```

% ดังนั้นใน TEMP ขณะนี้จะไม่มีค่าฟังก์ชันวัตถุประสงค์ให้ทำการจัดลำดับอีกต่อไป
neighbor=RANK(5,2:4); % แทนค่า neighbor ด้วยค่าพารามิเตอร์ลำดับที่ 5 ใน RANK
% ซึ่งเป็นค่าพารามิเตอร์ที่ทำให้ค่าฟังก์ชันวัตถุประสงค์แตกต่างจาก
% ค่าคำตอบที่ซ้ำมากที่สุดเมื่อเทียบกับอีก 4 ลำดับที่เหลือใน RANK
S0=neighbor; % ปรับค่า S0 = neighbor
if best_error<overall_best_error
    overall_best_error=best_error;
    overall_best_neighbor=best_neighbor;
    % ปรับค่า overall_best_error
    % ถ้า best_error < overall_best_error
    % ให้แทนค่า overall_best_error ด้วย best_error
    % และแทนค่าพารามิเตอร์ overall_best_neighbor ด้วย best_neighbor
    t=t+1; % ปรับเพิ่มค่าตัวเลื่อนสำหรับ local
    local(t,1)=count; % เก็บค่า count ไว้ในคอลัมน์ที่ 1 ของ local
    local(t,2:4)=best_neighbor;
    % เก็บค่า best_neighbor ไว้ในคอลัมน์ที่ 2 ถึง 4 ของ local
    local(t,5)=best_error; % เก็บค่า best_error ไว้ในคอลัมน์ที่ 5 ของ local
end
best_error=RANK(5,5); % แทนค่า best_error ด้วยค่าฟังก์ชันวัตถุประสงค์ลำดับที่ 5 ใน RANK
% ซึ่งเป็นค่าฟังก์ชันวัตถุประสงค์ที่แตกต่างจาก
% ค่าคำตอบที่ซ้ำมากที่สุดเมื่อเทียบกับอีก 4 ลำดับที่เหลือใน RANK
n=0; % ปรับค่า n = 0
else % ไม่เช่นนั้นแล้ว
    S0=best_neighbor; % แทนค่า S0 ด้วย best_neighbor
    best_error=best_error;% แทนค่า best_error ด้วย best_error
end
end
if overall_best_error<best_error
    best_error=overall_best_error;
    best_neighbor=overall_best_neighbor;
    % ปรับค่า best_error
    % ถ้า overall_best_error < best_error
    % ให้แทนค่า best_error ด้วย overall_best_error
    % และแทนค่าพารามิเตอร์ best_neighbor ด้วย overall_best_neighbor
end
time=toc; % ยุติการคำนวณเวลาการค้นหา
return

```

โปรแกรม Obj_VTS.m

```

function zsim=Obj_VTS(km,kd,mu) % โปรแกรม Obj_VTS คือโปรแกรมคำนวณค่าฟังก์ชันวัตถุประสงค์
% ซึ่งเป็นการคำนวณผลเฉลยของแบบจำลองระบบ VTS ในระนาบ z, x

```

```

% ในรูปของสมการอนุพันธ์ย่อยแบบไฮเพอร์โบลิก
% ที่ไม่พิจารณาผลของความเร่งเนื่องจากแรงโน้มถ่วง
% ด้วยระเบียบวิธีผลต่างสี่เหลี่ยม (ดูรายละเอียดในภาคผนวก ฉ.)
% อินพุตของ โปรแกรมคือค่าตอบหรือค่าพารามิเตอร์เริ่มต้นที่สุ่มได้
% km คือ  $K_M$  , kd คือ  $K_D$  และ mu คือ  $\mu$  ของแบบจำลองระบบ VTS
% เอาต์พุต zsim คือค่าฟังก์ชันวัตถุประสงค์ที่ได้จากคำตอบ km, kd, และ mu

load VTS_DATA_AFTER_CALIBRATION_XYZ

% โหลดข้อมูลอินพุตและเอาต์พุตของระบบ VTS ในระนาบ z,x
h=0.0097541; % ค่าระยะของข้อมูลตามแนวแกน x ซึ่ง  $h = \Delta x$ 
k=0.04; % ค่าระยะของข้อมูลตามแนวแกน t ซึ่ง  $k = \Delta t$ 
g=9.81; % ความเร่งเนื่องจากแรงโน้มถ่วง

% จำนวนคำตอบตามระเบียบวิธีผลต่างสี่เหลี่ยม
c=(km*kd*u/mu).^0.5; % จำนวนค่า  $c = \sqrt{(K_D K_M u) / \mu}$ 
r=c*k/h; % จำนวนค่า  $r = ck / h$ 
if (r>1.0) % ถ้า  $r > 1.0$ 
    zsim=5000; % กำหนดให้ส่ง zsim ที่มีค่ามาก ๆ (5,000) กลับสู่โปรแกรมหลัก
    return
end
for t=1:40
    Z(1,t)=0; % จำนวนเงินโอนไปขอขอบเขต
    Z(59,t)=0;
end
for x=2:59
    Z(x,1)=ZZZ_CAM2(24,x+1); % จำนวนคำตอบแถวที่ 1
end
for x=2:58 % จำนวนคำตอบแถวที่ 2
    Z(x,2)=((1-r*r)*Z(x,1))+(r*r*(Z(x-1,1)+Z(x+1,1)))/2;
end
for t=3:40 % จำนวนคำตอบแถวที่ 3 ถึง 40
    for x=2:58
        Z(x,t)=((2-2*r*r)*Z(x,t-1))+r*r*(Z(x+1,t-1)+Z(x-1,t-1))-Z(x,t-2);
    end
end
Z1=Z';
for i=24:63
    for j=2:60
        Z_New(i-23,j-1)=z(i,j); % จัดสรรข้อมูลจากการทดสอบเพื่อเปรียบเทียบกับผลจากแบบจำลอง
    end
end
ss_er=sum(sum((Z1-Z_New).^2)); % จำนวน SSE
ms_er=ss_er/(length(i)*length(j)); % จำนวน MSE

```

```

zsim=ms_er; % ส่งเอาต์พุต zsim เท่ากับ MSE กลับสู่โปรแกรมหลัก
return

```

โปรแกรม Objective_VTS.m

```

function [best_error1,best_neighbor1,best_error,best_neighbor]=Objective_VTS(S1,best_error,S0)
% โปรแกรม Objective_VTS เป็น โปรแกรมประเมินค่าใกล้เคียงที่สุ่มได้รอบ S0
% อินพุตของโปรแกรมนี้คือ S1, best_error, และ S0
% เอาต์พุตคือ best_error1, best_neighbor1,
% best_error, และ best_neighbor
error=[]; % ตัวแปร error สำหรับรองรับค่าฟังก์ชันวัตถุประสงค์
% ของค่าใกล้เคียงที่ได้จากการสุ่ม
for k=1:size(S1,1)
    zsim=Obj_VTS(S1(k,1),S1(k,2),S1(k,3)); % ประเมินค่าใกล้เคียงใน S1 ด้วยฟังก์ชันวัตถุประสงค์ใน โปรแกรม Obj_VTS
    error(k,1)=zsim; % เก็บค่าฟังก์ชันวัตถุประสงค์ของค่าใกล้เคียงทั้งหมด
end
[best_error1,index]=min(error); % หาค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในบรรดาค่าใกล้เคียง
% และเก็บไว้ใน best_error1
best_neighbor1=S1(index,:); % เก็บค่าพารามิเตอร์ของค่าใกล้เคียงที่ทำให้
% ค่าฟังก์ชันวัตถุประสงค์มีค่าน้อยที่สุดไว้ใน best_neighbor1
if best_error1<best_error % ปรับค่า best_error
    best_error=best_error1; % ถ้า best_error1 < best_error
% ให้แทนค่า best_error ด้วย best_error1
    best_neighbor=S1(index,:); % และแทนค่าพารามิเตอร์ best_neighbor ด้วย S1(index,:)
else % ไม่เช่นนั้นแล้ว
    best_neighbor=S0; % แทนค่า best_neighbor ด้วย S0
end
return

```

โปรแกรม rand1.m

```

function x=rand1(a,b) % โปรแกรม rand1 เป็นโปรแกรมสุ่มค่าพารามิเตอร์ในช่วงที่กำหนด
% อินพุตของโปรแกรมคือขอบเขตบนของพารามิเตอร์ (b) และ
% ขอบเขตล่างของพารามิเตอร์ (a)
% โดยอาศัยความสัมพันธ์  $(b-a)\text{rand} + a$ 
% เอาต์พุตของโปรแกรมคือผลของการสุ่มค่าพารามิเตอร์ในช่วง a ถึง b
x=a+rand*(b-a);
return

```


โปรแกรม random_neigh.m

```
function S1=random_neigh(Number_neighb,radius,xlimit,S0)
% โปรแกรม random_neigh เป็นโปรแกรมสุ่มค่าใกล้เคียงรอบคำตอบ S0
% อินพุตของโปรแกรมนี้คือ Number_neighb, radius, xlimit, และ S0
% เอาต์พุตคือค่าใกล้เคียง (S1) จำนวนเท่ากับ Number_neighb

for u=1:Number_neighb
    for k=1:size(xlimit,2)
        S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
        % ทำการสุ่มค่าใกล้เคียงโดยอาศัยความสัมพันธ์
        %  $S_{1,i} = S_0 + (\text{radius} \cdot \text{rand1}_i(-1,1) \cdot (x_{n,i} - x_{1,i}))$ ,
        % for i = 1, ..., Number_neighb
        % เรียกใช้โปรแกรม rand1 โดยให้ทำการสุ่มค่าในช่วง -1 ถึง 1
        % นั่นหมายความว่าค่าสุ่มมีพื้นที่อยู่ในช่วง  $S_0 \pm (\text{radius} \cdot (x_{n,i} - x_{1,i}))$ 

        while ( S1(u,k)>xlimit(1,k) | S1(u,k)<xlimit(2,k) )
            % ค่าใกล้เคียงที่สุ่มได้จะต้องอยู่ในขอบเขตบนและขอบเขตล่าง
            % ของค่าพารามิเตอร์ ถ้าค่าใกล้เคียงที่สุ่มได้
            % มีค่าเกินขอบเขตบนและขอบเขตล่างที่กำหนด
            % ให้ทำการสุ่มค่าใกล้เคียงใหม่

            S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
        end
    end
end
return
```

ภาคผนวก ๓

รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์ รายการจดลิขสิทธิ์
และ **CDROM** ภาพวีดีโอ

รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์ รายการจดลิขสิทธิ์
และ CDROM ภาพวีดีโอ

รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์เผยแพร่

- รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์ในวารสารวิชาการนานาชาติ

[1] D. Puangdownreong, S. Sujitjorn, and T. Kulworawanichpong, “Convergence Analysis of Adaptive Tabu Search”, **ScienceAsia Journal of the Science Society of Thailand**, vol. 30, no. 2, pp. 183-190, 2004.

[2] T. Kulworawanichpong, D. Puangdownreong, and S. Sujitjorn, “Finite Convergence of Adaptive Tabu Search”, **ASEAN Journal on Science and Technology for Development**, vol. 21, no. 2&3, pp. 103-115, 2004.

[3] D. Puangdownreong, T. Kulworawanichpong, and S. Sujitjorn, “Finite Convergence and Performance Evaluation of Adaptive Tabu Search”, **LNCS (Lecture Notes in Computer Science)**, Springer-Verlag Heidelberg, vol. 3215, pp. 710-717, 2004.

- รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์ในการประชุมวิชาการนานาชาติ

[1] D. Puangdownreong, K-N. Areerak, A. Srikaew, S. Sujitjorn, and P. Totarong, “System Identification via Adaptive Tabu Search”, In **Proc. IEEE Int. Conf. on Industrial Technology (ICIT’02)**, Bangkok, Thailand, vol. 2, pp. 915-920, 2002.

[2] S. Sujitjorn, A. Srikaew, D. Puangdownreong, K. Attakitmongcol, and P. Totarong, "Model Identification using Image Processing Technique", In **Proc. CCCT'03 Int. Conf. on Computer, Communication and Control Technologies**, Florida, USA, vol. 3, pp. 530-535, 2003.

[3] D. Puangdownreong, K-N. Areerak, K-L. Areerak, T. Kulworawanichpong, and S. Sujitjorn, "Application of Adaptive Tabu Search to System Identification", In **Proc. the 24th IASTED Int. Conf. on Modelling, Identification, and Control (MIC2005)**, Innsbruck, Austria, pp. 178-183, 2005.

รายการจดลิขสิทธิ์

จดแจ้งลิขสิทธิ์ประเภทวรรณกรรม (โปรแกรมคอมพิวเตอร์) ต่อกรมทรัพย์สินทางปัญญา จำนวน 1 รายการคือ

[1] สรวาภูมิ สุจิตจร และ เดชา พวงดาวเรือง. "โปรแกรมการค้นหาแบบตามูเชิงปรับตัว (ภาษา C และ MATLAB)", 19 พฤศจิกายน 2547, เลขที่คำขอ 90322.

CDROM ภาพวิดีโอ

ไฟล์ภาพวิดีโอที่บรรจุอยู่ใน CDROM เป็นไฟล์ภาพการเคลื่อนที่ของระบบ CPP และระบบ VTS ดังรายละเอียดต่อไปนี้

ระบบ CPP มีไฟล์ภาพวิดีโอที่บรรจุอยู่ใน CDROM จำนวน 3 ไฟล์คือ

1. ไฟล์ cpp01.avi เป็นไฟล์ภาพการเคลื่อนที่ของระบบ CPP จากการทดสอบตอนที่ 1
2. ไฟล์ cpp02.avi เป็นไฟล์ภาพการเคลื่อนที่ของระบบ CPP จากการทดสอบตอนที่ 2
3. ไฟล์ cpp03.avi เป็นไฟล์ภาพการเคลื่อนที่ของระบบ CPP จากการทดสอบตอนที่ 3

ระบบ VTS มีไฟล์ภาพวิดีโอที่บรรจุอยู่ใน CDROM จำนวน 2 ไฟล์คือ

1. ไฟล์ vts_cam1.avi เป็นไฟล์ภาพการเคลื่อนที่ของระบบ VTS ที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 1
2. ไฟล์ vts_cam2.avi เป็นไฟล์ภาพการเคลื่อนที่ของระบบ VTS ที่ได้จากกล้องวิดีโอ ดิจิตอลตัวที่ 2

ประวัติผู้เขียน

นายเดชา พวงดาวเรือง เกิดวันพฤหัสบดี ที่ 3 ธันวาคม พ.ศ. 2513 ที่ตำบลตาลาน อำเภอฝักไห้ จังหวัดพระนครศรีอยุธยา สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) จากมหาวิทยาลัยเอเชียอาคเนย์ เมื่อปี พ.ศ. 2536 หลังจากสำเร็จการศึกษาได้บรรจุเข้าเป็นอาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเอเชียอาคเนย์ ต่อมาได้สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมระบบควบคุม) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อปี พ.ศ. 2540 ในระหว่างศึกษาปริญญาวิศวกรรมศาสตรมหาบัณฑิต ได้เป็นอาจารย์พิเศษที่ภาควิชาวิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และที่สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี พ.ศ. 2543 ได้รับการแต่งตั้งให้ดำรงตำแหน่งผู้ช่วยศาสตราจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเอเชียอาคเนย์ ต่อมาในปี พ.ศ. 2544 ได้เข้าศึกษาต่อในระดับปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี มีบทความวิจัยและบทความวิชาการตีพิมพ์ทั้งในประเทศและต่างประเทศรวมทั้งสิ้นประมาณ 25 บทความ ขณะศึกษาปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต มีบทความวิจัยตีพิมพ์ในวารสารวิชาการนานาชาติจำนวน 3 บทความ การประชุมวิชาการนานาชาติ จำนวน 3 บทความ และจดแจ้งลิขสิทธิ์ 1 รายการ มีความเชี่ยวชาญทางด้านระบบควบคุมอัตโนมัติ การระบุเอกลักษณ์ระบบพลวัต การประมวลผลภาพ และการประยุกต์เทคนิคทางปัญญาประดิษฐ์ เพื่อแก้ปัญหาทางวิศวกรรม ปัจจุบันเป็นอาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเอเชียอาคเนย์