

การควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทนแบบป้อนกลับสัญญาณขาออกสำหรับ
ระบบสี่ถัง



นายกรภัทร ไยฤทธิ

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเคมี ภาควิชาวิศวกรรมเคมี

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2556

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR) are the thesis authors' files submitted through the University Graduate School.

ROBUST CONSTRAINED OUTPUT FEEDBACK MODEL PREDICTIVE CONTROL FOR
QUADRUPLE TANK SYSTEM



Mr. Korrapat Yairit

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Chemical Engineering

Department of Chemical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2013

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทน
โดย	แบบป้อนกลับสัญญาณขาออกสำหรับระบบสี่ถึง
สาขาวิชา	นายกรภัทร ไยฤทธิ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	วิศวกรรมเคมี
	ผู้ช่วยศาสตราจารย์ ดร.สุรเทพ เขียวหอม

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.กษิติศ หนูทอง)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.สุรเทพ เขียวหอม)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อมรชัย อภรณ์วิชานพ)

.....กรรมการภายนอกมหาวิทยาลัย
(ดร.พรชัย บำรุงศรี)

กรภัทร ไยฤทธิ : การควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทนแบบ
 ป้อนกลับสัญญาณขาออกสำหรับระบบสี่ถัง. (ROBUST CONSTRAINED OUTPUT
 FEEDBACK MODEL PREDICTIVE CONTROL FOR QUADRUPLE TANK SYSTEM)
 อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.สุรเทพ เขียวหอม, 129 หน้า.

ในงานวิจัยนี้พิจารณาปัญหาการควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทน
 แบบป้อนกลับสัญญาณขาออกสำหรับระบบแบบไม่ต่อเนื่องทางเวลาภายใต้ขอบเขตความไม่
 แน่นนอนแบบโพลีโทพ ระเบียบวิธีการควบคุมที่ได้เสนอสามารถคำนวณปัญหาออปติไมซ์แบบ
 ออฟไลน์เพื่อหาค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออก โดยกฎการควบคุม
 แบบป้อนกลับค่าสัญญาณขาออกจะเป็นฟังก์ชันกับตัวแปรสถานะจากการประมาณซึ่งคำนวณจาก
 ค่าสัญญาณขาออก ณ เวลาปัจจุบันด้วยตัวประมาณค่าตัวแปรสถานะ นอกจากนี้เซตยืนยันทรงรี
 จะถูกสร้างขึ้นโดยแต่ละเซตยืนยันทรงรีจะมีค่าอัตราขยายของตัวควบคุมแบบป้อนกลับกำกับแต่
 ละเซต ในขั้นตอนการควบคุมระบบ การสุ่มรับค่าแต่ละครั้งจะคำนวณหาค่าตัวแปรสถานะจาก
 การประมาณด้วยค่าสัญญาณขาออกโดยใช้ตัวประมาณค่าตัวแปรสถานะ จากนั้นจะนำค่าตัวแปร
 สถานะจากการประมาณมาคำนวณค่าอัตราขยายแบบป้อนกลับ โดยเลือกค่าอัตราขยายแบบ
 ป้อนกลับของเซตยืนยันทรงรีที่เล็กที่สุดซึ่งค่าตัวแปรสถานะจากการประมาณเป็นสมาชิกของเซต
 นั้น จากนั้นจะคำนวณค่าตัวแปรปรับเพื่อควบคุมระบบ นอกจากนี้ในงานวิจัยยังได้ประยุกต์ใช้การ
 ประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกระหว่างเซตยืนยันทรงรีซึ่งตัว
 แปรสถานะจากการประมาณเป็นสมาชิกกับเซตยืนยันทรงรีถัดไปที่มีขนาดเล็กกว่าโดยคำนวณค่า
 อัตราขยายมากที่สุดภายใต้เงื่อนไขทางการควบคุม ระเบียบวิธีการควบคุมที่งานวิจัยได้เสนอนำ
 ไปทดสอบกับระบบสี่ถังซึ่งเป็นระบบที่คำนึงถึงผลความไม่แน่นอนภายใต้ขอบเขตความไม่
 แน่นนอนของโพลีโทพ โดยระบบดังกล่าวมีลักษณะแบบจำลองแบบไม่เป็นเชิงเส้นซึ่งมีจำนวนตัว
 แปรหลายตัวรวมทั้งมีขอบเขตจำกัดของตัวแปรสัญญาณขาเข้าและตัวแปรสัญญาณขาออก ในการ
 ทดสอบระบบสี่ถังสามารถดำเนินการได้ภายใต้สถานะเฟสต่ำสุดและไม่มีเฟสต่ำสุดโดยอาศัยการ
 ปรับสัดส่วนของวาล์วตัวที่หนึ่งและตัวที่สอง ระดับน้ำในถังที่หนึ่งและถังที่สองถูกควบคุมโดยการ
 ปรับค่าความถี่ของกระแสไฟฟ้าที่ป้อนสู่มอเตอร์แต่ละตัว ผลการทดสอบแสดงให้เห็นว่าระเบียบวิธีการ
 ควบคุมที่เสนอสามารถรักษาเสถียรภาพของระบบได้ทั้งสองสถานะและระเบียบวิธีการประมาณค่า
 สามารถปรับปรุงสมรรถนะในการควบคุมได้

ภาควิชา วิศวกรรมเคมี

ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมเคมี

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2556

5570110721 : MAJOR CHEMICAL ENGINEERING

KEYWORDS: OFF-LINE ROBUST MPC/OUTPUT FEEDBACK CONTROL/LINEAR INTERPOLATION

KORRAPAT YAIRIT: ROBUST CONSTRAINED OUTPUT FEEDBACK MODEL PREDICTIVE CONTROL FOR QUADRUPLE TANK SYSTEM. ADVISOR: ASST. PROF. SOORATHEP KHEAWHOM, Ph.D., 129 pp.

This research considers the problem of output feedback robust control of polytopic uncertain discrete-time systems. The proposed algorithms solve off-line the optimal control optimization problem to determine the state feedback gain. The output feedback control law is parameterized as a function of estimated state determined by current output measured. An appropriate estimator is defined, and a sequence of feedback gains is computed. In addition, a sequence of ellipsoidal invariant set associated with each feedback gain pre-computed is then constructed. At each control iteration, the smallest ellipsoidal invariant set containing the currently measured output is determined, a corresponding feedback gain is implemented to the process. Further, the interpolation algorithm is proposed to improve control performance. The feedback gain is determined by maximizing the feedback gain with subjected to a set of constraints associated with the current invariant set and the adjacent smaller invariant set. The controller design is illustrated with a case study of quadruple tank system formulated as a polytopic uncertain system. This system is a nonlinear multi-variable system with input and output constraints. The system can operate in two operating schemes that are minimum phase and non-minimum phase by ratio adjustment of valve1 and valve2. Liquid level of tank1 and tank2 are controlled by manipulating frequency of the electrical power supplied to each pump. The results showed that the proposed algorithm can stabilize the system operated in both operating schemes. Interpolation algorithm applied can improve a control performance.

Department: Chemical Engineering Student's Signature

Field of Study: Chemical Engineering Advisor's Signature

Academic Year: 2013

กิตติกรรมประกาศ

งานวิจัยนี้จะสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับคำแนะนำ ดูแล ตลอดจน ข้อคิดเห็นในการแก้ไขปัญหาต่างๆ จากอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.สุรเทพ เขียวหอม ผู้วิจัยขอขอบพระคุณมา ณ ที่นี้

ขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร.กษิติศ หนูทอง ประธานในการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.อมรชัย อารมณ์วิฆานพ และ อาจารย์ ดร.พรชัย บำรุงศรี กรรมการในการสอบวิทยานิพนธ์ ที่กรุณาสละเวลาเพื่อให้ข้อเสนอแนะอันเป็นประโยชน์แก่งานวิจัยนี้

ขอขอบคุณบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ในการสนับสนุนเงินอุดหนุนค่าเล่าเรียนตลอดจนจบการศึกษา

ขอขอบคุณภาควิชาวิศวกรรมเคมี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ประสิทธิ์ประสาทความรู้ และประสบการณ์ทางวิชาชีพตลอดระยะเวลาที่ได้ศึกษาเล่าเรียนมา

ขอขอบคุณ คุณครู อาจารย์ทุกท่านที่ให้ความรู้ ความเข้าใจในสรรพวิชา และอบรมสั่งสอนในเรื่องต่างๆ

ขอขอบคุณเพื่อนพ้องพี่น้องในห้องปฏิบัติการ รวมถึงบุคคลอื่นอีกมากมาย ที่คอยให้กำลังใจ ให้ความช่วยเหลือและให้คำแนะนำที่ดีเสมอมา

ท้ายที่สุดนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา มารดา ผู้มีพระคุณที่ทำให้ทำงานได้ประสบความสำเร็จและมีทุกวันนี้ได้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญภาพ.....	ญ
คำอธิบายสัญลักษณ์และคำย่อ	ฎ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตงานวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	4
บทที่ 3 ทฤษฎีทางการควบคุม.....	7
3.1 ทฤษฎีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออนไลน์.....	7
3.2 ทฤษฎีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์	10
3.3 ทฤษฎีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ด้วยการประยุกต์ใช้ตัว ประมาณตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขา ออกระหว่างเซตยีนยงทรงรี.....	11
3.3.1 ตัวประมาณค่าตัวแปรสถานะ.....	11
3.3.2 การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่าง เซต ยีนยงทรงรี	11
3.3.3 อัลกอริทึมในการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ต่อระบบในกรณี ประยุกต์ใช้ตัวประมาณตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุม แบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี.....	13
บทที่ 4 ระบบสี่ถึงและตัวควบคุม	15
4.1 ชุดทดลองระบบสี่ถึง.....	15
4.2 อุปกรณ์ในระบบสี่ถึง	17

4.3	แบบจำลองทางคณิตศาสตร์ของระบบสี่ถึงอุปกรณ์ในระบบสี่ถึง	22
4.4	วิธีการสร้างแบบจำลองเชิงเส้นซึ่งมีการเปลี่ยนแปลงตามเวลาของระบบสี่ถึง	25
4.5	ระบบสี่ถึงภายใต้สภาวะเฟสต่ำสุดและสภาวะไม่มีเฟสต่ำสุด	27
4.6	ขอบเขตความไม่แน่นอนของแบบจำลองระบบสี่ถึง	28
4.7	ตัวควบคุมระบบสี่ถึง.....	28
บทที่ 5 วิธีการดำเนินงานวิจัย		32
5.1	การหาสภาวะคงตัวของระบบภายใต้พฤติกรรมแบบเฟสต่ำสุด	32
5.2	การหาสภาวะคงตัวของระบบภายใต้พฤติกรรมแบบไม่มีเฟสต่ำสุด	32
5.3	ขั้นตอนการควบคุมระบบ	33
บทที่ 6 ผลและวิเคราะห์ผลการทดลอง		34
6.1	ผลการทดสอบสมรรถนะการทำงานของปั๊ม.....	34
6.2	พฤติกรรมของระบบสี่ถึงภายใต้สภาวะเฟสต่ำสุดและไม่มีเฟสต่ำสุด	37
6.3	ผลการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกและค่าอัตราขยายของตัวประมาณตัวแปรสภาวะเก็บไว้ในฐานข้อมูล	43
6.3.1	ผลการคำนวณของระบบในสภาวะเฟสต่ำสุด.....	43
6.3.2	ผลการคำนวณของระบบในสภาวะไม่มีเฟสต่ำสุด.....	44
6.4	ผลการควบคุมระบบสี่ถึงด้วยระเบียบวิธีการควบคุมตามที่งานวิจัยเสนอ	46
6.4.1	ผลการควบคุมระบบสี่ถึงภายใต้สภาวะเฟสต่ำสุด.....	46
6.4.2	ผลการควบคุมระบบสี่ถึงภายใต้สภาวะไม่มีเฟสต่ำสุด	51
บทที่ 7 สรุปผลการวิจัย.....		57
7.1	ลักษณะการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ เมื่อมีการประยุกต์ใช้ตัวประมาณค่าตัวแปรสภาวะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี.....	57
7.2	ระเบียบวิธีการควบคุมที่งานวิจัยเสนอต่อพฤติกรรมการตอบสนองแบบตรงข้าม.....	57
7.3	สมรรถนะในการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี.....	58
รายการอ้างอิง		59
ประวัติผู้เขียนวิทยานิพนธ์		129

สารบัญตาราง

หน้า

ตารางที่ 4. 1 แสดงอัตราการไหลของน้ำที่ป้อนจากปั๊มตัวที่ 1 และ 2 เข้าสู่ระบบ.....	23
ตารางที่ 4. 2 แสดงค่าคงที่ในแบบจำลองทางคณิตศาสตร์ของระบบสี่ถัง.....	24
ตารางที่ 6. 1 ข้อมูลค่าพารามิเตอร์ที่สภาวะคงตัว.....	36
ตารางที่ 6. 2 ข้อมูลขอบเขตค่าพารามิเตอร์ในแบบจำลองของตัวควบคุมลูปลัก	37
ตารางที่ 6. 3 ข้อมูลขอบเขตค่าพารามิเตอร์ในแบบจำลองของตัวควบคุมลูปรอง.....	37
ตารางที่ 6. 4 ค่าอัตราการไหลของปั๊มในระบบสี่ถังที่สภาวะคงตัวภายใต้สภาวะเฟสต่ำสุดและ.....	38
ตารางที่ 6. 5 ค่าตัวแปรสภาวะจากการประมาณเพื่อคำนวณค่าอัตราขยายของตัวควบคุมแบบ	43
ตารางที่ 6. 6 ค่าตัวแปรสภาวะจากการประมาณเพื่อคำนวณค่าอัตราขยายของตัวควบคุมแบบ	45
ตารางที่ 6. 7 ข้อมูลค่าพารามิเตอร์ที่สภาวะคงตัว	56

สารบัญภาพ

หน้า

รูปที่ 3. 1 เขตขอบเขตความไม่แน่นอนแบบโพลีโทป.....	8
รูปที่ 4. 1 ภาพรวมของระบบสี่ถังและตัวแปรที่เกี่ยวข้อง.....	15
รูปที่ 4. 2 อาร์คูโน.....	16
รูปที่ 4. 3 แสดงภาพรวมการรับ-ส่ง สัญญาณ ระหว่าง คอมพิวเตอร์-อาร์คูโน-ระบบสี่ถัง.....	16
รูปที่ 4. 4 อุปกรณ์ในการทดลอง.....	18
รูปที่ 4. 5 แสดงความสัมพันธ์ระหว่างระบบควบคุมกับระบบสี่ถัง.....	29
รูปที่ 4. 6 ผลการตอบสนองของระบบแบบสแต็ป.....	30
รูปที่ 4. 7 เส้นโค้งการตอบสนองสำหรับการคำนวณค่าคงตัวของเวลา.....	31
รูปที่ 6. 1 ผลการทดสอบสมรรถนะปั๊มตัวที่สองในสภาวะเฟสต่ำสุด.....	34
รูปที่ 6. 2 ผลการทดสอบสมรรถนะปั๊มตัวที่สองในสภาวะเฟสต่ำสุดในช่วงเวลา.....	35
รูปที่ 6. 3 พฤติกรรมเฟสต่ำสุดของถังใบที่1ในระบบสี่ถัง.....	38
รูปที่ 6. 4 พฤติกรรมเฟสต่ำสุดของถังใบที่2ในระบบสี่ถัง.....	38
รูปที่ 6. 5 ลักษณะการควบคุมอัตราการไหลของตัวควบคุมลูปรองที่ 1.....	40
รูปที่ 6. 6 ลักษณะการควบคุมอัตราการไหลของตัวควบคุมลูปรองที่ 2.....	41
รูปที่ 6. 7 พฤติกรรมไม่มีเฟสต่ำสุดของถังใบที่ 1 ในระบบสี่ถัง.....	41
รูปที่ 6. 8 พฤติกรรมไม่มีเฟสต่ำสุดของถังใบที่ 2 ในระบบสี่ถัง.....	42
รูปที่ 6. 9 เขตยืนยงทรงรีจากการคำนวณของระบบในสภาวะเฟสต่ำสุด.....	44
รูปที่ 6. 10 เขตยืนยงทรงรีจากการคำนวณของระบบในสภาวะไม่มีเฟสต่ำสุด.....	45
รูปที่ 6. 11 ผลการควบคุมระดับน้ำในถังใบที่ 1 ในรูปตัวแปรเบี่ยงเบน.....	48
รูปที่ 6. 12 ผลการควบคุมระดับน้ำในถังใบที่ 2 ในรูปตัวแปรเบี่ยงเบน.....	49
รูปที่ 6. 13 ค่าตัวแปรปรับที่ป้อนให้แก่ปั๊มตัวที่ 1 ในรูปตัวแปรเบี่ยงเบน.....	50
รูปที่ 6. 14 ค่าตัวแปรปรับที่ป้อนให้แก่ปั๊มตัวที่ 2 ในรูปตัวแปรเบี่ยงเบน.....	50
รูปที่ 6. 15 เส้นทางการเปลี่ยนแปลงค่าตัวแปรสถานะจากการประมาณ.....	51
รูปที่ 6. 16 ผลการควบคุมระดับน้ำในถังใบที่ 1.....	53
รูปที่ 6. 17 ผลการควบคุมระดับน้ำในถังใบที่ 2.....	54
รูปที่ 6. 18 ค่าตัวแปรปรับที่ป้อนให้แก่ปั๊มตัวที่ 1 ในรูปตัวแปรเบี่ยงเบน.....	54
รูปที่ 6. 19 ค่าตัวแปรปรับที่ป้อนให้แก่ปั๊มตัวที่ 1 ในรูปตัวแปรเบี่ยงเบน.....	55
รูปที่ 6. 20 เส้นทางการเปลี่ยนแปลงค่าตัวแปรสถานะจากการประมาณ.....	55

คำอธิบายสัญลักษณ์และคำย่อ

A	พื้นที่หน้าตัดถ้ง
A_i	ค่าคงที่ในรูปเมตริกซ์บนโพลีโทพจุดที่ i
A_o	ค่าคงที่จากแบบจำลองที่สภาวะคงตัวในรูปเมตริกซ์
$A(k)$	ค่าคงที่จากแบบจำลองที่เวลา k
B_i	ค่าคงที่ในรูปเมตริกซ์บนโพลีโทพจุดที่ i
B_o	ค่าคงที่จากแบบจำลองที่สภาวะคงตัวในรูปเมตริกซ์
$B(k)$	ค่าคงที่จากแบบจำลองที่เวลา k
C	ค่าคงที่ในรูปเมตริกซ์
$C_{r,i}$	ค่าสัมประสิทธิ์การไหลของวาล์วได้ถึงที่ i
Co	คอนเวกซ์ฮูล(Convex hull)
g	ค่าความเร่งเนื่องจากแรงโน้มถ่วงของโลก
G_f	ค่าความถ่วงจำเพาะของของเหลว
m_i	ค่าพารามิเตอร์การไหลของวาล์วได้ถึงที่ i
P^-	แสดงถึงระบบสี่ถังภายใต้สภาวะเฟสต่ำสุด
P^+	แสดงถึงระบบสี่ถังภายใต้สภาวะไม่มีเฟสต่ำสุด
h_i	ระดับในถังที่ i
k_p	ค่าคงที่การไหลของป้ม
K	ค่าอัตราขยายแบบป้อนกลับ
K_s	ค่าอัตราขยายแบบป้อนกลับของตัวควบคุมลูปรอง

$K(\rho(k))$	ค่าอัตราขยายแบบป้อนกลับจากการประยุกต์ใช้การประมาณค่า อัตราขยายแบบป้อนกลับ
$q_{in,i}$	อัตราการไหลของน้ำจากปั๊มไปยังถังที่ i
$q_{out,i}$	อัตราการไหลของน้ำออกจากถังที่ i ผ่านวาล์วได้ถัง
$q_{pump,i}$	อัตราการไหลของน้ำจากปั๊มตัวที่ i
R	เมตริกซ์ถ่วงน้ำหนัก
L_p	ค่าอัตราการขยายของตัวประมาณค่าตัวแปรสถานะ
u	ตัวแปรสัญญาณขาเข้า
v_i	ค่าแรงดันไฟฟ้าซึ่งป้อนให้แก่ปั๊มตัวที่ i
V_i	ปริมาตรน้ำในถังที่ i
x	ตัวแปรสถานะ (State variable)
\hat{x}	ตัวแปรสถานะจากตัวประมาณค่าตัวแปรสถานะ
y	ตัวแปรสัญญาณขาออก

ตัวอักษรกรีก

Ω	เซตขอบเขตโพลีโทป
Θ	เมตริกซ์ถ่วงน้ำหนัก
φ	Estimator design parameter
γ_i	สัดส่วนในการปรับวาล์วเพื่อจ่ายน้ำจากปั๊มที่ i ไปยังถังด้านล่าง
ρ	ค่าความหนาแน่นของน้ำ
τ	ค่าคงตัวของเวลา

ตัวอักษรย่อ

PDLF	Dependent Lyapunov Function
IAE	Integral Absolute Error
LMI	Linear Matrix Inequalities
MPC	Model Predictive Control
RMPC	Robust Model Predictive Control
SLF	Single Lyapunov Function

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันกระบวนการผลิตทางอุตสาหกรรมมีความซับซ้อนขึ้น มีจำนวนตัวแปรควบคุมและตัวแปรปรับจำนวนมาก รวมทั้งมีเงื่อนไขทางกระบวนการผลิตที่ซับซ้อนเพิ่มขึ้น จึงมีความจำเป็นในการพัฒนาเทคโนโลยีการควบคุมเชิงทำนายแบบจำลอง (Model Predictive Control, MPC) เพื่อควบคุมตัวแปรจำนวนมากรวมทั้งจัดการกับระบบที่มีเงื่อนไขทางการผลิตให้มีประสิทธิภาพ ซึ่งเทคโนโลยีการควบคุมดังกล่าวเป็นการควบคุมที่อาศัยแบบจำลองทางกระบวนการ โดยอาศัยหลักของการออปติไมซ์เพื่อคำนวณหาตัวแปรปรับที่เหมาะสมในการควบคุม หลักการทำงานของ การควบคุมชนิดนี้ จะทำการออปติไมซ์เพื่อคำนวณค่าตัวแปรปรับที่เหมาะสมตามเงื่อนไขของระบบจำนวนหนึ่งชุดแล้วเลือกใช้ตัวแปรปรับเพียงค่าเดียวในการควบคุม หลังจากนั้นจะวัดค่าตัวแปรสถานะ (State variable) ใหม่เพื่อคำนวณค่าตัวแปรปรับที่เหมาะสมต่อไป

จากการควบคุมเชิงทำนายแบบจำลองที่กล่าวไว้ข้างต้น เป็นการควบคุมที่ใช้แบบจำลองทางคณิตศาสตร์ของระบบ เมื่อแบบจำลองที่ใช้มีความคลาดเคลื่อนไปจากระบบจริง จึงทำให้การควบคุมนั้นไม่มีประสิทธิภาพ แสดงถึงความไม่คงทนของตัวควบคุม จึงมีการเสนอเทคโนโลยีการควบคุมที่เรียกว่า การควบคุมเชิงทำนายแบบจำลองคงทน (Robust Model Predictive Control, RMPC) เพื่อคำนึงผลจากความไม่แน่นอนของแบบจำลองที่ใช้ในการคำนวณ ทำให้ตัวควบคุมที่ใช้วิธีการควบคุมดังกล่าว มีความคงทนมากยิ่งขึ้น สามารถทำงานได้ในสถานะที่แบบจำลองทางคณิตศาสตร์มีความผิดพลาดจากระบบจริง เทคนิคการควบคุมดังกล่าวได้ถูกพัฒนามาอย่างต่อเนื่องด้วยวิธีการต่างๆ เช่น การออกแบบตัวควบคุมเชิงทำนายแบบจำลองคงทนใช้ฟังก์ชันเลียปูนอฟเดี่ยว (Single Lyapunov Function, SLF) หรือใช้ฟังก์ชันเลียปูนอฟซึ่งขึ้นกับตัวแปรเสริม (Parameter Dependent Lyapunov Function, PDLF) เป็นต้น นอกจากนี้ยังมีการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะ (State estimator) ขึ้นเพื่อความถูกต้องของแบบจำลองทางคณิตศาสตร์ เนื่องจากสมการแบบจำลองจะขึ้นกับตัวแปรต่างๆ รวมทั้งตัวแปรสถานะซึ่งโดยปกติแล้วเราจะทราบค่าตัวแปรสถานะผ่านเครื่องมือวัดเรียกว่า ตัวแปรสัญญาณขาออก (Output variable) ซึ่งค่าที่ได้อาจมีความคลาดเคลื่อนจากค่าตัวแปรสถานะจริง ดังนั้นการประมาณค่าตัวแปรสถานะด้วยตัวประมาณค่าตัวแปรสถานะ จึงทำให้แบบจำลองทางคณิตศาสตร์ที่ใช้ในการควบคุมมีความถูกต้องยิ่งขึ้น ส่งผลให้ประสิทธิภาพในการควบคุมดีขึ้น ในระบบควบคุมที่มีการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะซึ่งมีการใช้ค่าสัญญาณขาออกในการคำนวณหาตัวแปรสถานะจากการประมาณ (Estimated state) ในขั้นตอนการ

คำนวณค่าตัวแปรปรับ สามารถเรียกการควบคุมดังกล่าวได้ว่า การควบคุมแบบป้อนกลับค่าสัญญาณขาออก (Output Feedback Control)

ในการควบคุมระบบ ปัญหาอีกหนึ่งประการที่สำคัญคือเวลาในการคำนวณค่าตัวแปรปรับ ซึ่งเป็นปัญหาที่สำคัญของการควบคุมประเภทออนไลน์(Online-control) จึงมีการพัฒนาการควบคุมประเภทออนไลน์มาเป็นการควบคุมแบบออฟไลน์(Offline-control) โดยแนวคิดการควบคุมแบบออฟไลน์ถูกนำมาประยุกต์ใช้กับการควบคุมเชิงทำนายแบบจำลองคงทน เพื่อลดปัญหาในการควบคุมกรณีที่ต้องการควบคุมระบบซึ่งระยะเวลาในการคำนวณตัวแปรปรับมีความสำคัญ เช่น ระบบความปลอดภัยที่ต้องการการควบคุมอย่างเร่งด่วน ระบบการควบคุมทางการบิน เป็นต้น นอกจากนี้ความเร็วในการควบคุมเพื่อให้ได้ค่าตามเป้าหมายที่กำหนดก็เป็นปัจจัยหนึ่งของประสิทธิภาพของการควบคุมจึงมีการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับระหว่างเซตยีนยงทรงรีในการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ภายใต้ความไม่แน่นอนของโพลีโทปิก (Interpolation-based offline Robust MPC for uncertain polytopic) เพื่อลดระยะเวลาในการควบคุม โดยการประยุกต์วิธีดังกล่าวนี้ระบบยังคงมีความเสถียรภาพ

เป้าหมายหลักของงานวิจัยนี้ต้องการเสนอการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ภายใต้ขอบเขตความไม่แน่นอนของโพลีโทป โดยประยุกต์การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีและประยุกต์ใช้ตัวประมาณค่าตัวแปรสภาวะ เพื่อใช้ควบคุมระดับน้ำภายในถังของระบบสี่ถัง โดยระบบสี่ถังที่เลือกใช้ในงานวิจัยสามารถเป็นตัวแทนของระบบชนิดหลายตัวแปร และมีลักษณะของระบบซึ่งมีความไม่เป็นเชิงเส้น ตัวแปรปรับของระบบคือปริมาณแรงดันไฟฟ้าซึ่งป้อนให้แก่ปั๊มตัวที่ 1 และ 2 ส่วนตัวแปรควบคุมของระบบคือระดับน้ำในถังทั้ง 4 ใบ

1.2 วัตถุประสงค์ของงานวิจัย

- 1) พัฒนาระเบียบวิธีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ภายใต้ขอบเขตความไม่แน่นอนของโพลีโทป โดยการประยุกต์ใช้ตัวประมาณค่าตัวแปรสภาวะ และประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีเพื่อใช้ควบคุมระดับน้ำภายในระบบสี่ถัง
- 2) ประยุกต์ใช้การควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ภายใต้ขอบเขตความไม่แน่นอนของโพลีโทป โดยการประยุกต์ใช้ตัวประมาณค่าตัวแปรสภาวะ และประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีเพื่อใช้ควบคุมระดับน้ำภายในระบบสี่ถัง

1.3 ขอบเขตงานวิจัย

- 1) แสดงแบบจำลองของระบบสี่ถึงในรูปแบบของระบบเชิงเส้นซึ่งมีการเปลี่ยนแปลงตามเวลาในเทอมของสมการปริภูมิสถานะดังต่อไปนี้

$$x(k+1) = A(k)x(k) + B(k)u(k)$$

$$y(k) = Cx(k)$$

$$[A(k) \ B(k)] \in \Omega$$

โดยที่ $x(k) \in R^n$ เป็นค่าตัวแปรสถานะ ณ เวลาที่ k , $u(k) \in R^m$ เป็นค่าตัวแปรปรับ ณ เวลาที่ k และ $y(k) \in R^p$ เป็นค่าตัวแปรสัญญาณขาออก ณ เวลาที่ k ตามลำดับ นอกจากนี้ในระบบที่เป็นโพลีโทพ (polytope) ได้นิยาม Ω เป็นเซตขอบเขตของโพลีโทพตามสมการดังต่อไปนี้ $\Omega = \text{Co}\{[A_1 \ B_1], [A_2 \ B_2], \dots, [A_L \ B_L]\}$

- 2) ใช้ชุดคำสั่ง YALMIP และ SEDUMI ร่วมกับโปรแกรม MATLAB ในการสร้างชุดคำสั่งในการควบคุม
- 3) สมการตัวประมาณค่าตัวแปรสถานะซึ่งใช้ในการควบคุมดังนี้

$$\hat{x}(k+1) = A_0 \hat{x}(k) + B_0 u(k) + L_p (y(k) - C\hat{x}(k))$$

โดยที่ L_p คือค่าอัตราขยายของตัวประมาณ และ ตัวแปร $[A_0 \ B_0]$ นั้นอ้างอิงตามระบบที่สภาวะคงตัว (steady state)

- 4) ระเบียบวิธีการควบคุมที่พัฒนาขึ้นเป็นวิธีการควบคุมตัวเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ภายใต้ขอบเขตความไม่แน่นอนของโพลีโทพ โดยการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะ และประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีเพื่อใช้ควบคุมระดับน้ำภายในระบบสี่ถึง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ระเบียบวิธีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ภายใต้ขอบเขตความไม่แน่นอนของโพลีโทพ โดยการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะ และประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง

ในสภาวะปัจจุบัน กระบวนการทางอุตสาหกรรมมีความซับซ้อนมากยิ่งขึ้นเนื่องด้วยจำนวนของตัวแปรปรับ ตัวแปรควบคุม และเงื่อนไขข้อจำกัดต่างๆ ทำให้มีการพัฒนาตัวควบคุมเชิงทำนายแบบจำลอง (Model Predictive Control, MPC) ขึ้น หลักการทำงานของตัวควบคุมชนิดนี้ได้นำหลักของการออปติไมซ์เข้ามาประยุกต์ใช้ โดยทำการวัดค่าตัวแปรสถานะ ณ เวลา t จากนั้นคำนวณหาค่าตัวแปรปรับจำนวนหนึ่งชุดจากการแก้ปัญหาออปติไมซ์ โดยจะใช้ค่าตัวแปรปรับเพียงค่าเดียวเท่านั้น ณ เวลา $t+1$ ต่อจากนั้นจะทำการคำนวณหาค่าตัวแปรปรับชุดใหม่โดยอาศัยข้อมูลจากค่าตรวจรู้ ณ เวลาใหม่ เนื่องจากตัวควบคุมเชิงทำนายแบบจำลอง เป็นเทคนิคการควบคุมซึ่งอาศัยแบบจำลอง ดังนั้นผลจากความไม่แน่นอนอันเนื่องมาจากตัวแปรรบกวนหรือความไม่แน่นอนของค่าพารามิเตอร์ในแบบจำลอง ทำให้ประสิทธิภาพในการควบคุมลดลง จึงมีการพัฒนาตัวควบคุมเชิงทำนายแบบจำลองคงทน ขึ้นเพื่อรองรับระบบที่มีความไม่แน่นอนทางแบบจำลอง

จากการศึกษางานวิจัยเกี่ยวกับตัวควบคุมเชิงทำนายแบบจำลองคงทนพบว่า สามารถแบ่งตัวควบคุมออกเป็น 2 กลุ่มใหญ่ คือตัวควบคุมเชิงทำนายแบบจำลองคงทนแบบออนไลน์และตัวควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ในส่วนของการควบคุมแบบออนไลน์ งานวิจัยเริ่มแรกของ Kothare และคณะ [1] ได้นำทฤษฎีเลียบูโนฟเดียวมาใช้ในการออกแบบตัวควบคุมเพื่อประกันเสถียรภาพของระบบ และประยุกต์ใช้ปัญหาออปติไมซ์ให้อยู่ในรูปอสมการเมตริกซ์เชิงเส้น (Linear Matrix Inequalities) โดยมีเป้าหมายเพื่อออกแบบตัวควบคุมตามกฎการป้อนกลับสถานะ (State feedback control law) ณ ช่วงเวลาที่ทำนายนั้นกำหนดให้ค่าน้อยที่สุดของฟังก์ชันวัตถุประสงค์แบบสถานการณ์เลวร้ายที่สุด (minimized worst case objective function) ภายใต้เงื่อนไขข้อจำกัดของตัวแปรควบคุมและตัวแปรสัญญาณขาออก การควบคุมเชิงทำนายแบบจำลองคงทนแบบออนไลน์ จะมีประสิทธิภาพในการควบคุมดีกว่าการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ เนื่องมาจาก การควบคุมเชิงทำนายแบบจำลองคงทนแบบออนไลน์ใช้ค่าตัวแปรสถานะ ณ เวลาจริงในแต่ละรอบของการออปติไมซ์เพื่อมาคำนวณค่าตัวแปรปรับในการควบคุมระบบ แต่ก็มีข้อเสียในบางเวลาในการคำนวณตัวแปรควบคุมซึ่งมากกว่าการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ดังนั้นการควบคุมเชิงทำนายแบบจำลองคงทนแบบออนไลน์จึงไม่เหมาะกับระบบซึ่งผลของเวลาในการควบคุมเป็นสิ่งสำคัญ ทำให้เกิดการพัฒนากการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์

จากงานวิจัยของ Wan และคณะ [2] ได้พัฒนาการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ โดยแบ่งขั้นตอนออกเป็น 2 ส่วนใหญ่ ในส่วนแรก จะอาศัยหลักการของการสร้างเซตยีนยงทรงรีเสถียรภาพเชิงเส้นกำกับ (An asymptotically stable invariance ellipsoid) โดยสร้างเซตยีนยงทรงรีตามทฤษฎีเลียบูโนฟเดียวเพื่อประกันเสถียรภาพของระบบ ข้อมูลดังกล่าวจะถูกเก็บไว้เพื่อใช้

ในการหาค่าตัวแปรปรับ ส่วนที่สอง จะเป็นการคำนวณค่าตัวแปรปรับแบบออนไลน์ โดยวัดค่าตัวแปรสถานะ ณ ปัจจุบัน จากนั้นคำนวณค่าตัวแปรปรับจากข้อมูลในส่วนแรกซึ่งได้คำนวณไว้ก่อนหน้านี้ ทำให้ลดเวลาในการคำนวณลง และจากงานวิจัยดังกล่าวได้เปรียบเทียบประสิทธิภาพการควบคุมระหว่างการควบคุมเชิงทำนายแบบจำลองคงทนแบบออนไลน์ และการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ พบว่าประสิทธิภาพของการควบคุมทั้งสองแบบมีประสิทธิภาพใกล้เคียงกัน แต่การควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์สามารถช่วยลดระยะเวลาในการควบคุม

อย่างไรก็ตามยังได้มีการพัฒนาการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ขึ้นในขั้นตอนการคำนวณค่าตัวแปรปรับ ตามงานวิจัยของ Bumroongsri และคณะ [3] ได้เสนอวิธีประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับระหว่างเซตยีนยงทรงรีร่วมกับการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ภายใต้ความไม่แน่นอนโพลีโทปิก โดยเสนอ 3 วิธีด้วยกัน โดยวิธีแรก ใช้หลักการประมาณเซตยีนยงทรงรีเล็กที่สุดซึ่งค่าตัวแปรสถานะที่วัดมานั้นยังคงเป็นสมาชิกในเซตนั้น วิธีที่สองใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับ(State feedback gain) เพื่อให้ได้ค่ามากที่สุด วิธีที่สามใช้การประมาณโดยหาค่าอัตราขยายของตัวควบคุมแบบป้อนกลับที่ทำให้ค่าความคลาดเคลื่อนของเงื่อนไขต่อเซตยีนยงทรงรีวงถัดไปน้อยที่สุด จากทั้ง 3 วิธีพบว่าวิธีที่ 2 และ 3 ค่าอัตราขยายของตัวควบคุมแบบป้อนกลับที่คำนวณนั้นมีค่ามากกว่าวิธีที่ 1 มีผลทำให้ตัวแปรที่สนใจสามารถเข้าสู่ค่าเป้าหมายได้เร็วขึ้น โดยที่การควบคุมยังคงเสถียรภาพอยู่ และเมื่อเปรียบเทียบเวลาในการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับพบว่าวิธีที่ 2 ใช้เวลาในการคำนวณน้อยกว่าวิธีที่ 3

นอกจากนี้ในการควบคุมระบบโรงงานอุตสาหกรรมในสภาพจริงนั้น พบว่าเราสามารถทราบค่าตัวแปรสถานะจากระบบผ่านเครื่องมือวัดซึ่งค่าตัวแปรสถานะที่ได้รับจะมีความถูกต้องนั้นขึ้นกับความแม่นยำของเครื่องมือวัดและปัญหาอีกประการที่พบนั้นคือ ความขัดข้องของเครื่องมือวัดซึ่งอาจมีการส่งสัญญาณผิดพลาดไปจากความเป็นจริงส่งผลให้เกิดผลกระทบต่อระบบ จึงมีการพัฒนาการควบคุมแบบป้อนกลับค่าสัญญาณขาออกขึ้น จากงานวิจัยของ Wan และคณะ[4] ได้ประยุกต์ใช้ตัวประมาณตัวแปรสถานะร่วมกับการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ โดยเพิ่มอัลกอริทึมในการประกันเสถียรภาพของระบบกรณีที่ระบบมีการใช้ตัวประมาณค่าตัวแปรสถานะร่วมด้วย

ในส่วนของงานวิจัยที่เกี่ยวข้องกับการประยุกต์ใช้ตัวควบคุมเชิงทำนายแบบจำลองในการควบคุมระดับน้ำของระบบสี่ถัง เช่น งานวิจัยของ Sukapradit [5] ได้นำเสนอการประยุกต์ใช้การควบคุมเชิงทำนายแบบจำลองแบบคงทนสำหรับระบบเชิงเส้นซึ่งพารามิเตอร์มีการเปลี่ยนแปลงในการควบคุมระบบสี่ถัง งานวิจัยของ Poontong [6] ได้เสนอการประยุกต์ใช้ตัวควบคุมเชิงทำนายแบบคงทนบนพื้นฐานต้นทุนสมรรถนะปกติโดยใช้ฟังก์ชันเลียปูนอฟเดียว ตัวควบคุมเชิงทำนายแบบคงทน

บนพื้นฐานต้นทุนสมรรถนะในสถานการณ์เลวร้ายที่สุดใช้ฟังก์ชันเลียปูนอฟเดียว และตัวควบคุมเชิง
ทำนายแบบคงทนบนพื้นฐานต้นทุนสมรรถนะในสถานการณ์เลวร้ายที่สุดใช้ฟังก์ชันเลียปูนอฟเสริม ซึ่ง
พิจารณาให้ค่าสัมประสิทธิ์การไหลของวาล์วได้ถึงทั้งสี่ใบมีความไม่แน่นอน



บทที่ 3 ทฤษฎีทางการควบคุม

ในการควบคุมกระบวนการต่าง ๆ นั้นมีวิธีการควบคุมหลากหลายวิธีด้วยกัน ซึ่งในงานวิจัยนี้ได้นำเสนอการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ร่วมกับตัวประมาณค่าตัวแปรสถานะ โดยมีการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับระหว่างเซตยีนยงทรงรี โดยใช้หลักการการหาค่าอัตราขยายของตัวควบคุมแบบป้อนกลับที่มีค่ามากที่สุดแล้วยังสามารถทำให้ระบบมีความเสถียรภาพ ในหัวข้อที่ 3.1 และ 3.2 จะเป็นพื้นฐานทางการควบคุมและเทคนิคต่างๆ ในควบคุมซึ่งนำมาใช้ในงานวิจัย ส่วนหัวข้อที่ 3.3 จะเป็นส่วนระเบียบวิธีวิธีการควบคุมซึ่งงานวิจัยนี้ นำเสนอ

3.1 ทฤษฎีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออนไลน์

แสดงแบบจำลองในระบบเชิงเส้นซึ่งมีการเปลี่ยนแปลงตามเวลาในเทอมของสมการปริภูมิสถานะดังสมการที่ 1

$$x(k+1) = A(k)x(k) + B(k)u(k)$$

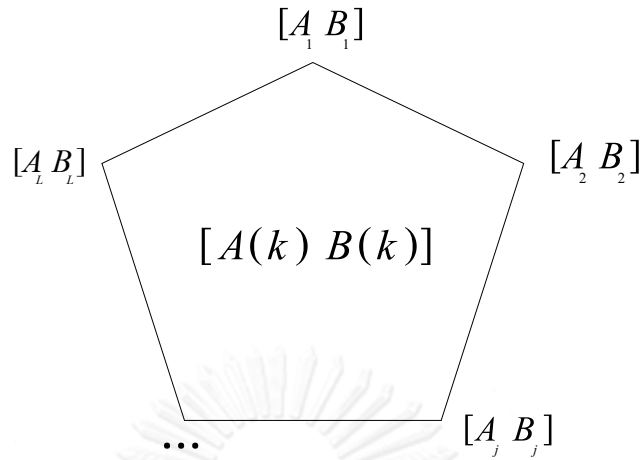
$$y(k) = Cx(k)$$

$$[A(k) \ B(k)] \in \Omega \quad (1)$$

โดยที่ $u(k) \in R^m$, $x(k) \in R^n$, $y(k) \in R^p$ เป็นค่าตัวแปรปรับ ตัวแปรสถานะ และตัวแปรสัญญาณขาออก ณ เวลาที่ k ตามลำดับ นอกจากนี้ในระบบที่มีความไม่แน่นอนเชิงพารามิเตอร์ จะมีการสร้างเซตขอบเขตความไม่แน่นอนหรือโพลีโทป (Polytope) ขึ้น โดยนิยาม Ω แทนเซตขอบเขตโพลีโทป ดังสมการที่ 2

$$\Omega = Co\{[A_1 \ B_1], [A_2 \ B_2], \dots, [A_L \ B_L]\} \quad (2)$$

โดยที่ Co หมายถึง คอนเวกซ์ฮัลล์ (Convex hull) ซึ่ง $[A_i \ B_i]$ แทนจุดยอดของเซตขอบเขตความไม่แน่นอนโพลีโทป ดังรูปที่



รูปที่ 3. 1 เซตขอบเขตความไม่แน่นอนแบบโพลีโทพ

และจากสมบัติของคอนเวกซ์เซต(Convex set) พบว่าถ้า $[A(k) B(k)] \in \Omega$ ดังนั้น $[A(k) B(k)]$ มีความสัมพันธ์กับจุดยอดของเซตขอบเขตโพลีโทพดังสมการที่ 3

$$A = \sum_{j=1}^L \alpha_j A_j, B = \sum_{j=1}^L \alpha_j B_j, \sum_{j=1}^L \alpha_j = 1, 0 \leq \alpha_j \leq 1 \quad (3)$$

ในระบบการควบคุมจะมีการกำหนดปัญหาในการควบคุมระบบซึ่งแสดงสมการดังนี้

$$\min_{u(k+i|k)} \max_{x(k+i|k), B(k+i|k) \in \Omega} J_\infty \quad (4)$$

โดยที่ $J_\infty(k) = \sum_{i=0}^{\infty} [x(k+i|k)^T \Theta x(k+i|k) + u(k+i|k)^T R u(k+i|k)] \quad (5)$

$$|u_r(k+i|k)| \leq u_{r,\max}, \quad i \geq 0, r = 1, 2, \dots, n_u \quad (6)$$

$$|y_s(k+i|k)| \leq y_{s,\max}, \quad i \geq 0, s = 1, 2, \dots, n_y \quad (7)$$

จากสมการ (4) เป็นสมการแสดงเป้าหมายของการควบคุมระบบตามสมการที่ (1) เพื่อหาชุดตัวแปรปรับตามกฎการป้อนกลับสถานะ $u(k+i|k) = K(k)x(k+i|k)$ โดยทำให้ฟังก์ชันวัตถุประสงค์ที่สภาวะเลวร้ายที่สุดมีค่าน้อยที่สุด ภายใต้ขอบเขตข้อจำกัดของตัวแปรปรับตามสมการที่ (6) และข้อจำกัดของตัวแปรสัญญาณขาออกตามสมการที่ (7) ในส่วนรายละเอียดของฟังก์ชันวัตถุประสงค์ตามสมการที่ (5) นั้น จะเป็นการทำนายและคำนวณชุดตัวแปรปรับจนถึงระยะเวลา

อนันต์ โดยมีเวกเตอร์ถ่วงน้ำหนักเพื่อแสดงการเลือกผลการตอบสนองของกระบวนการตามเงื่อนไข $\Theta > 0$ และ $R > 0$

ในงานวิจัยนี้จะสนใจการควบคุมเชิงทำนายแบบจำลองคงทนโดยประยุกต์ทฤษฎีเลียปูนอฟ เดียวมาใช้ในการออกแบบตัวควบคุมเพื่อรับประกันเสถียรภาพของระบบ โดยแสดงเป็นสมการในรูปของอสมการเมตริกซ์เชิงเส้น (Linear Matrix Inequalities, LMIs) [1] ดังนี้

$$\max_{u(k|k), \gamma, Q, Y} \gamma$$

Subject to

$$\begin{pmatrix} 1 & x(k|k)^T \\ x(k|k) & Q \end{pmatrix} \geq 0, Q > 0 \quad (8)$$

$$\begin{pmatrix} Q & QA_j^T + Y^T B_j^T & Q^T \Theta^{1/2} & Y^T R^{1/2} \\ A_j Q + B_j Y & Q & 0 & 0 \\ \Theta^{1/2} Q & 0 & \gamma I & 0 \\ R^{1/2} Y & 0 & 0 & \gamma I \end{pmatrix} \geq 0, j=1,2,\dots,L \quad (9)$$

$$\begin{pmatrix} X & Y \\ Y^T & Q \end{pmatrix} \geq 0, X_{rr} \leq u_{r,\max}^2, r=1,2,\dots,n_u \quad (10)$$

$$\begin{pmatrix} Z & C(A_j Q + B_j Y) \\ (A_j Q + B_j Y)^T C^T & Q \end{pmatrix} \geq 0, Z_{ss} \leq y_{s,\max}^2, s=1,2,\dots,n_y \quad (11)$$

โดยเงื่อนไขอสมการที่ (8) เป็นสมการที่ใช้สร้างเซตอิมมูนยวงรี (Invariant ellipsoid) ซึ่งแสดงเซตของตัวแปรสถานะที่สนใจในการควบคุมระบบโดยจะประกันการลู่เข้าสู่ศูนย์ แสดงถึงความมีเสถียรภาพของระบบ ในส่วนอสมการที่ (9) เป็นการประกันเสถียรภาพของระบบด้วยฟังก์ชันเลียปูนอฟเดียว ถัดมาในอสมการที่ (10) แสดงขอบเขตจำกัดของตัวแปรสถานะและสุดท้ายอสมการที่ (11) แสดงขอบเขตจำกัดของตัวแปรสัญญาณขาออก รายละเอียดการพิสูจน์สามารถศึกษาได้จาก [1]

อัลกอริทึม 1

อัลกอริทึมการควบคุมเชิงทำนายแบบจำลองคงทนพื้นฐานต้นทุนสมรรถนะในสถานการณ์เลวร้ายที่สุดโดยใช้ฟังก์ชันเลียปูนอฟเดียว

ขั้นตอนที่ 1: วัดค่า $x(k|k)$

ขั้นตอนที่ 2: คำนวณ $\max_{u(k), Y, Q, \mathcal{Y}} \mathcal{J}$ ภายใต้เงื่อนไขตามสมการที่ (8) ถึง (11) และคำนวณค่า

$$K(k) = YQ^{-1}$$

ขั้นตอนที่ 3: นำค่าตัวแปรปรับซึ่งคำนวณจาก $u(k) = K(k)x(k)$ ป้อนเข้าสู่ระบบที่ต้องการควบคุม

ขั้นตอนที่ 4: ณ เวลา $k+1$ ให้ $k = k+1$ แล้วกลับสู่ขั้นตอนที่ 1

3.2 ทฤษฎีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์

ในงานวิจัยของ Wan และคณะ[2] ได้เสนอการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์โดยแบ่งการควบคุมออกเป็น 2 ส่วน ส่วนแรกเป็นขั้นตอนในการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับเก็บไว้ในฐานข้อมูล และส่วนที่สองเป็นการควบคุมแบบออนไลน์โดยรับค่าตัวแปรสถานะจากระบบแล้วนำค่าอัตราขยายของตัวควบคุมแบบป้อนกลับที่คำนวณไว้ในขั้นตอนแรกมาใช้เพื่อคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับที่เหมาะสมต่อระบบ

อัลกอริทึม 2

อัลกอริทึมการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ในขั้นตอนการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับเก็บไว้ในฐานข้อมูล

ขั้นตอนที่ 1: กำหนดจำนวนค่าอัตราขยายของตัวควบคุมแบบป้อนกลับที่ต้องการคำนวณเก็บไว้ในฐานข้อมูล (N จำนวน)

ขั้นตอนที่ 2: กำหนดค่าตัวแปรสถานะ x_i ตามจำนวนค่าอัตราขยายของตัวควบคุมแบบป้อนกลับที่ต้องการคำนวณเก็บไว้ (x_1, x_2, \dots, x_N)

ขั้นตอนที่ 3: นำค่า x_i มาคำนวณ $\max_{u(k), Y, Q, \mathcal{Y}_i} \mathcal{J}_i$ ภายใต้เงื่อนไขตามสมการที่ (8) ถึง (11) และ

คำนวณค่า $K_i(k) = Y_i Q_i^{-1}$ โดยค่า Y_i, Q_i^{-1}, K_i จะถูกเก็บไว้ในฐานข้อมูล

ขั้นตอนที่ 4: ถ้า $i < N$ เลือก x_{i+1} โดยมีเงื่อนไขตามสมการ $\|x_{i+1}\|_{Q_i^{-1}}^2 < 1$ แล้วกลับสู่ขั้นตอนที่ 3

ในส่วนที่สองซึ่งจะกล่าวถึงนั้นเป็นการนำข้อมูลในส่วนแรกซึ่งมีการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับเก็บไว้ในฐานข้อมูลแล้วมาใช้ในการคำนวณค่าตัวแปรปรับที่เหมาะสมกับตัวแปรสถานะซึ่งวัดได้จากระบบ ตามอัลกอริทึมดังต่อไปนี้

อัลกอริทึม 3

การควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ร่วมกับระบบที่สถานะจริง

ขั้นตอนที่ 1: วัดค่าตัวแปรสถานะจากระบบ $x(k)$

ขั้นตอนที่ 2: หาค่าอัตราขยายของตัวควบคุมแบบป้อนกลับโดย ตรวจสอบตามเงื่อนไข $\|x\|_{e^{-1}}^2 < 1$

โดยเริ่มตรวจสอบจาก $i = N$ ถ้าไม่ตรงตามสมการเงื่อนไข ให้ตรวจสอบ ที่ $i = i - 1$ จนกระทั่งยอมรับสมการเงื่อนไข แล้วใช้ค่า $K(k) = K_i$

ขั้นตอนที่ 3: นำค่าตัวแปรปรับ $u(k) = K(k)x(k)$ ป้อนเข้าสู่ระบบที่ต้องการควบคุม

ขั้นตอนที่ 4: ณ เวลา $k + 1$ ให้ $k = k + 1$ แล้วกลับสู่ขั้นตอนที่ 1

3.3 ทฤษฎีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ด้วยการประยุกต์ใช้ตัวประมาณตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

ในระบบการควบคุมแบบป้อนกลับค่าสัญญาณขาออก(Output feedback control) จะมีการออกแบบการควบคุมโดยแยกเป็น 2 ส่วน ส่วนที่ 1 เป็นการออกแบบการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ และส่วนที่ 2 จะเป็นการออกแบบตัวประมาณตัวแปรสถานะ[4]

3.3.1 ตัวประมาณค่าตัวแปรสถานะ

$$\hat{x}(k+1) = A_0 \hat{x}(k) + B_0 u(k) + L_p (y(k) - C\hat{x}(k)) \quad (12)$$

จากสมการที่ 12 เป็นรูปแบบของสมการตัวประมาณค่าตัวแปรสถานะซึ่งใช้ในงานวิจัยนี้ โดยที่ L_p คือค่าอัตราขยายของตัวประมาณค่าตัวแปรสถานะ และตัวแปร $[A_0 \ B_0]$ นั้นอ้างอิงตามระบบที่สถานะคงตัว (Steady state)

3.3.2 การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

จากงานวิจัยของ Bumroongsri และคณะ [3] เสนอเทคนิคการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับตัวแปรสถานะ(State feedback control) ระหว่างเซตยีนยงทรงรี 3

เทคนิคด้วยกัน ซึ่งงานวิจัยนี้สนใจในเทคนิคการประมาณค่าอัตราขยายของตัวควบคุมเพื่อให้ได้ค่าอัตราขยายของตัวควบคุมแบบป้อนกลับมากที่สุด โดยนำมาประยุกต์ใช้กับการควบคุมแบบป้อนกลับค่าสัญญาณขาออก ซึ่งใช้ตัวแปรสถานะจากการประมาณ (\hat{x}) แทนการใช้ตัวแปรสถานะ (x) โดยใช้สมการที่ (12) ในการประมาณค่า

$$\min \rho(k) \quad (13)$$

Subject to

$$\begin{pmatrix} 1 & ((A_j + B_j K(\rho(k)))\hat{x}(k))^T \\ (A_j + B_j K(\rho(k)))\hat{x}(k) & Q_{i,j} \end{pmatrix} \geq 0, j=1,2,\dots,L \quad (14)$$

$$\begin{pmatrix} u_{h,\max}^2 & (K(\rho(k))\hat{x}(k)_h)^T \\ K(\rho(k))\hat{x}(k)_h & 1 \end{pmatrix} \geq 0 \quad (15)$$

$$0 \leq \rho(k) \leq 1 \quad (16)$$

ค่าอัตราขยายของตัวควบคุมแบบป้อนกลับมากที่สุด ณ เวลา k ($K(\rho(k))$) สามารถคำนวณได้จาก $K(\rho(k)) = \rho(k)K_i + (1 - \rho(k))K_{i+1}$ ซึ่งค่า $\rho(k)$ คำนวณจากสมการ (13) ภายใต้เงื่อนไขสมการ (14) - (16) โดยสมการที่ (14) แสดงเงื่อนไขเพื่อประกันให้ตัวแปรสถานะที่ถูกทำนายค่าถัดไปยังอยู่ภายในเซตยีนยงทรงรีซึ่งเป็นการประกันเสถียรภาพของการควบคุม และสมการที่ (15) แสดงเงื่อนไขข้อจำกัดตัวแปรปรับ

$$Q_i^{-1} - (A_j + B_j K_{i+1})^T Q_i^{-1} (A_j + B_j K_{i+1}) > 0, \forall j, \forall i=1,\dots,N-1 \quad (17)$$

นอกจากนี้ในการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับนั้น จำเป็นต้องยอมรับเงื่อนไขตามสมการที่ (17) ก่อนจึงจะสามารถประยุกต์เทคนิคดังกล่าวได้

3.3.3 อัลกอริทึมในการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ต่อระบบในกรณีประยุกต์ใช้ตัวประมาณตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

ในระบบการควบคุมแบบป้อนกลับค่าสัญญาณขาออก(Output feedback control) จะมีการออกแบบการควบคุมโดยแยกการออกแบบเป็น 2 ส่วน ส่วนที่ 1 เป็นการออกแบบตัวควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ส่วนที่ 2 จะเป็นการออกแบบตัวประมาณตัวแปรสถานะ [4] โดยจำเป็นต้องมีอัลกอริทึมในการควบคุมเพิ่มเติมในกรณีที่มีการประยุกต์ใช้ตัวประมาณตัวแปรสถานะเพื่อประกันเสถียรภาพของระบบ

อัลกอริทึม 4

อัลกอริทึมในการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ในกรณีที่มีการประยุกต์ใช้ตัวประมาณตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

ขั้นตอนที่ 1: ออกแบบตัวควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ตามอัลกอริทึม 2 และระบบ Θ และ R

ขั้นตอนที่ 2: ออกแบบตัวประมาณค่าตัวแปรสถานะโดยการหาค่าอัตราขยายของตัวประมาณค่าตัวแปรสถานะเพื่อให้ค่าความผิดพลาดระหว่างค่าตัวแปรสถานะที่ถูกประมาณกับค่าตัวแปรสถานะจริง มีค่าลดลงเมื่อเวลาผ่านไป ตามเงื่อนไขข้อสมการที่ (18) ด้วยการกำหนดค่า φ (Estimator design parameter) ในช่วง $0 < \varphi < 1$

$$\begin{pmatrix} \varphi^2 G & GA_0 - EC \\ A_0^T G - C^T E^T & G \end{pmatrix} \geq 0, G > 0 \quad (18)$$

โดย $L_p = G^{-1}E$ รายละเอียดการพิสูจน์สามารถศึกษาได้จาก [2]

ขั้นตอนที่ 3: ตรวจสอบความเสถียรภาพของระบบกรณีตัวควบคุมมีการประยุกต์ใช้ตัวประมาณตัวแปรสถานะ ตามเงื่อนไขข้อสมการที่ (19)

$$\begin{pmatrix} F & FA_{p,i,j}^T \\ A_{p,i,j} F & F \end{pmatrix} > 0 \quad (19)$$

$$\text{โดยที่ } A_{p,i,j} = \begin{pmatrix} A_j & B_j F_i \\ L_p C & A_0 + B_0 F_i - L_p C \end{pmatrix}, j=1, \dots, L, i=1, \dots, N$$

ระบบจะมีความเสถียรภาพเมื่อ $F > 0$

ถ้า $F \leq 0$ ให้ กลับสู่ขั้นตอน 1 และ 2

อัลกอริทึม 5

อัลกอริทึมในการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ต่อระบบในกรณีที่มีการ
ประยุกต์ใช้ตัวประมาณตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบ
ป้อนกลับค่าสัญญาณขาออกระหว่างเซตวงรียืนยันร่วมกับระบบที่สถานะจริง

ขั้นตอนที่ 1: รับค่าสัญญาณขาออกจากระบบ $y(k)$ และที่ $k=0$ ให้ $\hat{x}(0)=y(0)$

ขั้นตอนที่ 2: หาค่าอัตราขยายของตัวควบคุมแบบป้อนกลับโดย ตรวจสอบตามเงื่อนไข $\|\hat{x}\|_{Q^{-1}}^2 < 1$

โดยเริ่มตรวจสอบจาก $i=N$ ถ้าไม่ตรงตามสมการเงื่อนไข ให้ตรวจสอบ ที่ $i=i-1$ จนกระทั่ง
ยอมรับสมการเงื่อนไข แล้วใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณ
ขาออกระหว่างเซตยืนยันทรงรี โดย $\min \rho(k)$ ภายใต้เงื่อนไข สมการที่ (14)-(16) ได้ค่า

อัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออก ตามสมการ

$$K(\rho(k)) = \rho(k)K_i + (1 - \rho(k))K_{i+1}$$

ขั้นตอนที่ 3: นำค่าตัวแปรปรับซึ่งหาตามกฎ $u(k) = K(\rho(k))\hat{x}(k)$ สู่อุปกรณ์ที่ต้องการควบคุม

ขั้นตอนที่ 4: คำนวณค่าตัวแปรสถานะจากการประมาณในเวลา $k+1$ จากสมการ

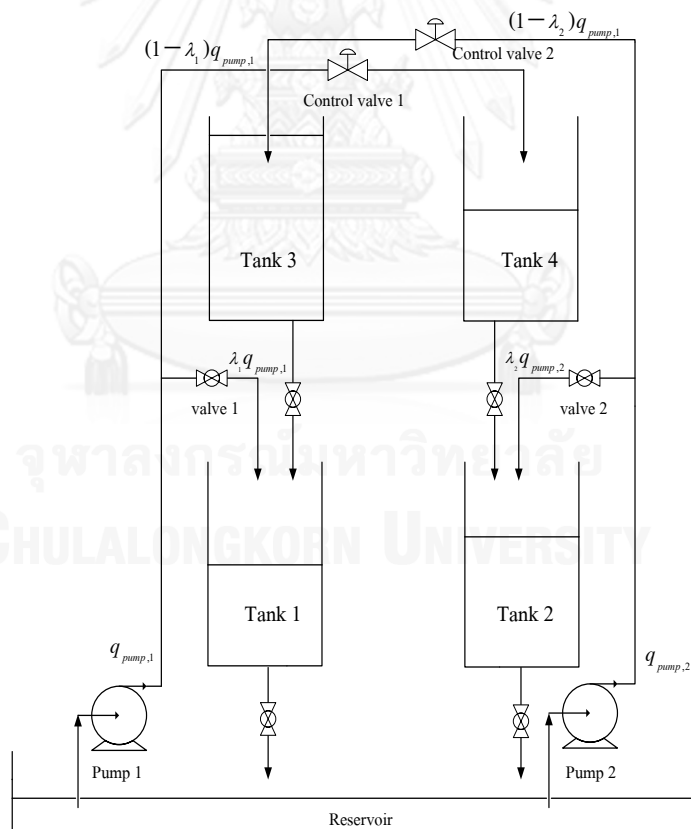
$$\hat{x}(k+1) = A_0\hat{x}(k) + B_0u(k) + L_p(y(k) - C\hat{x}(k))$$

ขั้นตอนที่ 5: ณ เวลา $k+1$ ให้ $k = k+1$ แล้วกลับไปสู่ขั้นตอนที่ 1

บทที่ 4 ระบบสี่ถังและตัวควบคุม

4.1 ชุดทดลองระบบสี่ถัง

ระบบสี่ถังเป็นระบบซึ่งมีการนำมาใช้เป็นตัวแทนของระบบการผลิตทางอุตสาหกรรม เนื่องจากระบบสี่ถังเป็นระบบซึ่งมีหลายตัวแปร เป็นระบบที่มีความไม่เป็นเชิงเส้น และสามารถระบุข้อจำกัดของตัวแปรปรับและสัญญาณขาออกของระบบ จากรูปที่ 4.1 แสดงภาพรวมโครงสร้างของระบบสี่ถังซึ่งประกอบไปด้วยถังน้ำจำนวน 4 ใบ โดยได้ถังแต่ละใบจะมีวาล์วเพื่อปรับสัดส่วนของน้ำ ในส่วนของระบบส่งน้ำจะประกอบไปด้วยปั๊ม 2 ตัว ปั๊มตัวที่ 1 จะทำหน้าที่ส่งน้ำไปยังถังที่ 1 และ 4 โดยวาล์วตัวที่ 1 ทำหน้าที่ปรับสัดส่วนของน้ำจากปั๊มตัวที่ 1 ไปยังถังแต่ละถัง ส่วนปั๊มตัวที่ 2 ทำหน้าที่ส่งน้ำไปยังถังที่ 2 และ 3 โดยจะมีวาล์วตัวที่ 2 ในการปรับสัดส่วนของน้ำจากปั๊มตัวที่ 2 ไปยังถังแต่ละใบ การปรับสัดส่วนของวาล์วจะใช้สัญลักษณ์แทนด้วย λ ซึ่งเป็นไปตามเงื่อนไข $0 \leq \lambda \leq 1$ และอาศัยแรงดันไฟฟ้าซึ่งจ่ายให้แก่ปั๊มเป็นตัวแปรปรับของระบบควบคุม

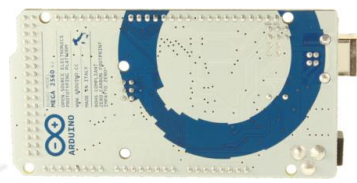


รูปที่ 4. 1 ภาพรวมของระบบสี่ถังและตัวแปรที่เกี่ยวข้อง

ในส่วนของการควบคุมจะรับส่งข้อมูลตัวแปรในการควบคุมระหว่างคอมพิวเตอร์ซึ่งใช้โปรแกรม MATLAB ในการสร้างชุดคำสั่งควบคุมกับชุดควบคุมของระบบสี่ถังผ่านอุปกรณ์ “ อาร์ดูโน้ (arduino) ” ดังรูปที่ 4.2 อาร์ดูโน้เป็นไมโครคอนโทรลเลอร์ประเภทหนึ่ง มีหน่วยประมวลผลและความจำขนาดเล็กภายในตัวเอง สามารถรับ-ส่ง ข้อมูลได้ทั้งแบบดิจิตอลและอนาล็อก การเชื่อมโยงระหว่าง คอมพิวเตอร์-อาร์ดูโน้-ระบบสี่ถัง แสดงได้ดังรูปที่ 4.3

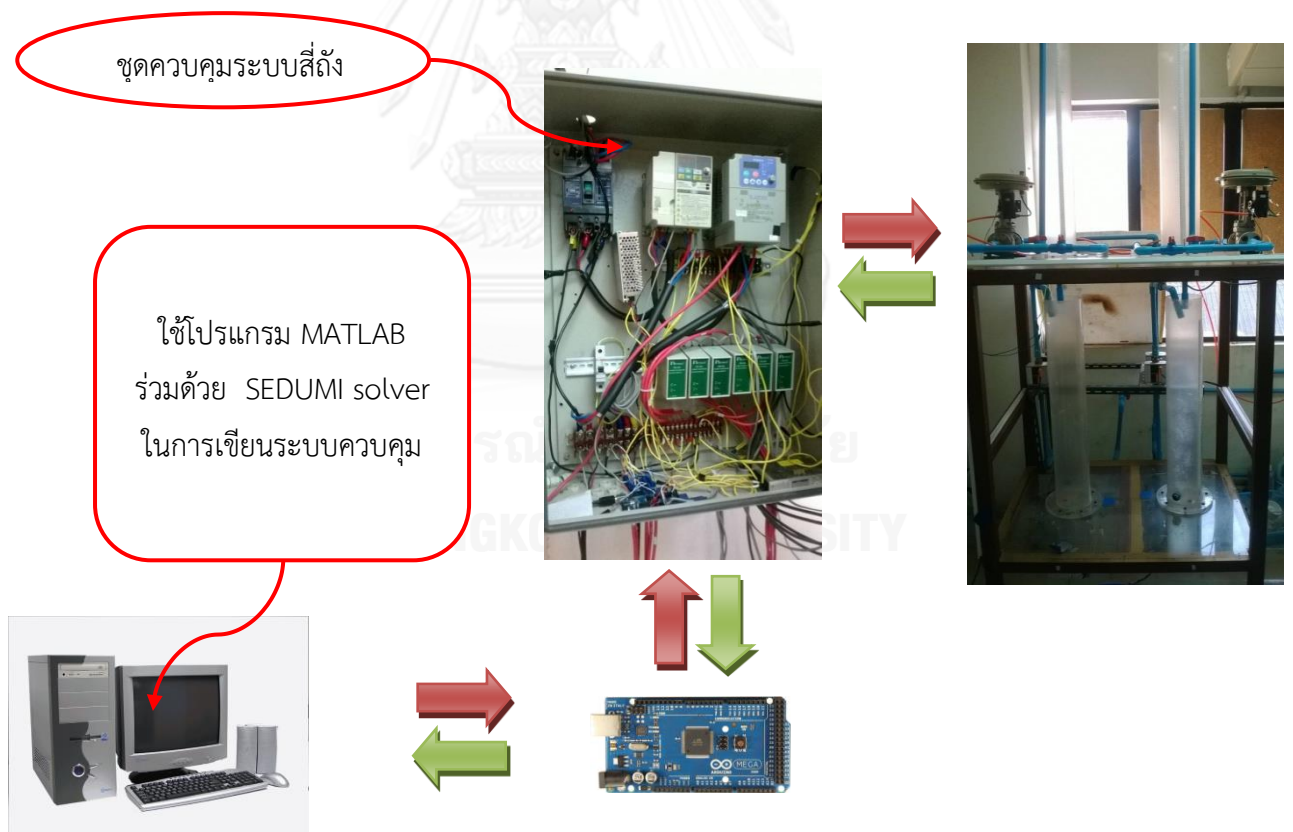


ก) ด้านหน้าของแผงวงจร



ข) ด้านหน้าของแผงวงจร

รูปที่ 4.2 อาร์ดูโน้



รูปที่ 4.3 แสดงภาพรวมการรับ-ส่ง สัญญาณ ระหว่าง คอมพิวเตอร์-อาร์ดูโน้-ระบบสี่ถัง

4.2 อุปกรณ์ในระบบสี่ถัง

อุปกรณ์ในระบบสี่ถังสามารถแบ่งได้เป็น 2 ส่วนด้วยกัน ดังนี้

ส่วนที่ 1 ชุดสี่ถัง

- 1) ถังทรงกระบอกจำนวน 4 ใบ วัสดุทำจากอะคริลิก เส้นผ่านศูนย์กลาง 14.3 เซนติเมตร สูง 90 เซนติเมตร
- 2) ปั้มน้ำแบบแรงเหวี่ยงหนีศูนย์กลาง (Centrifugal pump) ของ Bruno จำนวน 2 ตัว ขนาด ½ แรงม้า
- 3) วาล์วควบคุม (Control valve) ของ Samson จำนวน 2 ตัว ขนาด ½ นิ้ว และ ¾ นิ้ว ซึ่งมีการตอบสนองแบบปกติปิด ในงานวิจัยนี้ได้ทำการเปิดวาล์วควบคุม คงที่ไว้ที่ 100 เปอร์เซ็นต์
- 4) ปั้มนม 1 ตัว ใช้เพื่อจ่ายลมให้กับวาล์วควบคุม
- 5) อุปกรณ์ตรวจวัดแรงดัน (Pressure transmitter) ของ ADZ Nagono มีย่านการวัดอยู่ในช่วง 0 ถึง 100 mbar และส่งสัญญาณขาออกในช่วง 4-20 mA ค่าสัญญาณจากอุปกรณ์สามารถนำไปเทียบเป็นค่าความสูงของระดับน้ำในถัง
- 6) อินเวอร์เตอร์ (Inverter) จำนวน 2 ตัว ของ HITACHI รุ่น SJ200 และ OMRON ใช้สำหรับปรับความถี่ของแรงดันไฟฟ้าที่จ่ายสู่มอเตอร์ทั้งสอง
- 7) อุปกรณ์วัดอัตราการไหล (Flow meter) ของ ifm electron จำนวน 4 ตัว สำหรับวัดอัตราการไหลของน้ำที่จะเข้าสู่ถังแต่ละใบ
- 8) บอลวาล์ว (Ball valve) จำนวน 6 ตัว วัสดุทำจากพีวีซี
- 9) ท่อพีวีซี ขนาด ½ นิ้ว ¾ นิ้ว และ 1 นิ้ว โดยส่วนใหญ่ท่อขนาด ½ นิ้ว จะถูกใช้ในในระบบ ในส่วนท่อขนาด ¾ นิ้ว จะเป็นท่อซึ่งต่อออกจากวาล์วควบคุม และท่อขนาด 1 นิ้ว เป็นท่อซึ่งต่อออกจากปั้ม
- 10) อ่างพักน้ำ วัสดุทำจากอะคริลิก ขนาดกว้าง 180 เซนติเมตร ยาว 45 เซนติเมตร สูง 20 เซนติเมตร

ส่วนที่ 2 ชุดควบคุมระบบสี่ถัง

- 1) อุปกรณ์แปลงกระแสไฟฟ้า (Current transmitter) มีหน้าที่แปลงสัญญาณไฟฟ้า 4-20 mA จากอุปกรณ์ตรวจวัดแรงดันเพื่อแปลงเป็นค่าแรงดันไฟฟ้าสำหรับส่งเข้าอาร์ดูโน
- 2) อุปกรณ์แปลงแรงดันไฟฟ้า (Voltage transmitter) มีหน้าที่แปลงแรงดันไฟฟ้าจากอาร์ดูโนเป็นค่ากระแสไฟฟ้า 4-20 mA ไปควบคุมวาล์วควบคุม

- 3) อาร์ดูโน้ รุ่น Mega 2560
- 4) เครื่องคอมพิวเตอร์พร้อมโปรแกรม MATLAB R2008b สำหรับสร้างชุดคำสั่งการควบคุมและประมวลผลตัวควบคุมสำหรับระบบสี่ถัง

ภาพอุปกรณ์ในการทดลองแสดงไว้ในภาพที่ 4.4 ตั้งแต่ภาพ ก ถึง ฎ

รูปที่ 4. 4 อุปกรณ์ในการทดลอง



ก. ชุดสี่ถัง



ข. วาล์วควบคุม



ค. อุปกรณ์ตรวจวัดแรงดัน



ง. บั้มลม



จ. บั้มน้ำ



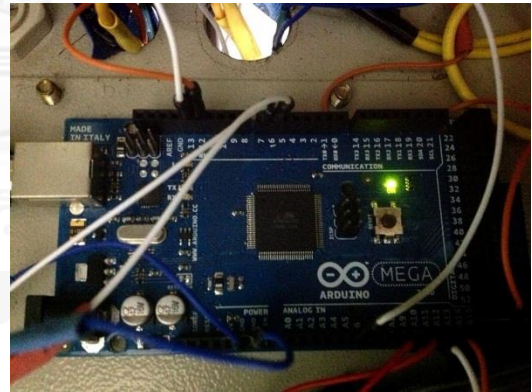
ฉ. อุปกรณ์แปลงกระแสไฟฟ้าและแรงดันไฟฟ้า



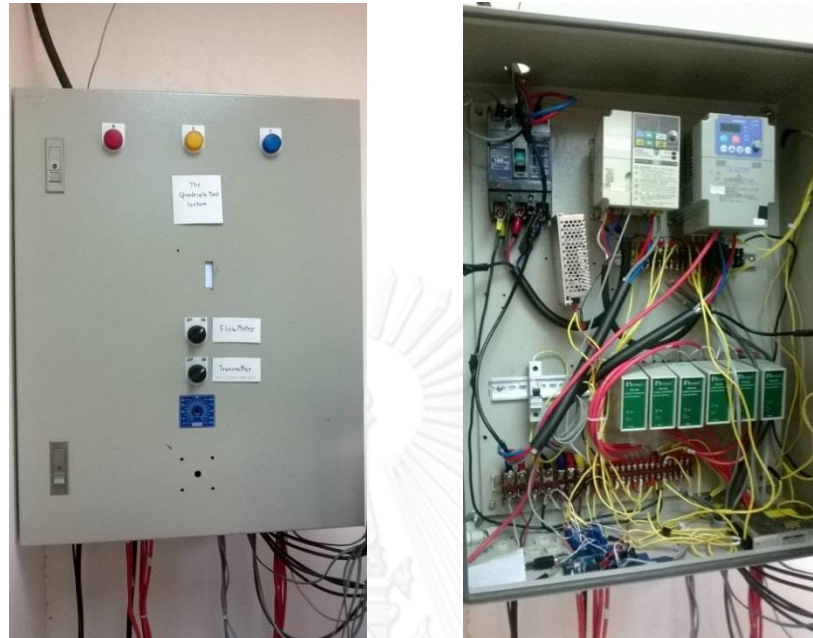
ช. อินเวอร์เตอร์



ช. อุปกรณ์วัดอัตราการไหล



ณ. อาร์ดูโน้



ญ. ตู้คอนโทรล



ฎ. คอมพิวเตอร์

4.3 แบบจำลองทางคณิตศาสตร์ของระบบสี่ถังอุปกรณ์ในระบบสี่ถัง

ระบบสี่ถังประกอบไปด้วยถังน้ำจำนวนสี่ใบซึ่งสามารถสร้างแบบจำลองทางคณิตศาสตร์ของแต่ละถังโดยการทำสมดุลมวลสารแต่ละถัง i ได้ดังต่อไปนี้

$$\frac{d(\rho V_i)}{dt} = \rho q_{in,i} - \rho q_{out,i} \quad (20)$$

โดยที่ $q_{in,i}$ แทนอัตราการไหลของน้ำที่ไหลเข้าถังที่ i และ $q_{out,i}$ แทนอัตราการไหลของน้ำที่ไหลออกจากถังที่ i

กำหนดให้การสร้างแบบจำลองทางคณิตศาสตร์ของระบบสี่ถังอยู่ภายใต้สมมุติฐานดังต่อไปนี้

1. ความหนาแน่นของน้ำมีค่าคงที่ เท่ากับ 1000 กิโลกรัมต่อลูกบาศก์เมตร
2. อุณหภูมิของระบบไม่มีการเปลี่ยนแปลง เนื่องจากการทดลองทำที่อุณหภูมิห้องในห้องปฏิบัติการ
3. วาล์วควบคุมถูกปรับให้เปิด 100 เปอร์เซ็นต์ ตลอดการทดลอง
4. ความแรงเนื่องจากแรงโน้มถ่วงของโลกมีค่าคงที่ เท่ากับ 9.81 ตารางเซนติเมตรต่อวินาที

จากความสัมพันธ์ทางรูปทรงของถัง $V = Ah$ สามารถแทนค่าความสัมพันธ์ดังกล่าวลงในสมการที่ (20) จัดรูปสมการใหม่ได้สมการที่ 21

$$A \frac{d(h_i)}{dt} = q_{in,i} - q_{out,i} \quad (21)$$

โดยที่อัตราการไหลของน้ำขาออกผ่านวาล์วได้ถังต่างๆเป็นไปตามสมการที่ 22

$$q_{out,i} = C_{v,i} \sqrt{\frac{\rho g h_i}{G_f}} \quad (22)$$

จากสมการข้างต้นสามารถจัดรูปสมการอัตราการไหลของน้ำขาออกผ่านวาล์วได้ถังดังนี้

$$q_{out,i} = m_i \sqrt{h_i} \quad , \quad m_i = C_{v,i} \sqrt{\frac{\rho g}{G_f}} \quad (23)$$

โดยที่ m_i คือค่าพารามิเตอร์การไหลของวาล์วได้ถังที่ i

ในงานวิจัยต้องการควบคุมระดับน้ำในถังที่1และถังที่2 โดยการปรับค่าแรงดันไฟฟ้าซึ่งป้อนเข้าสู่ปั๊มผ่านอินเวอร์เตอร์เพื่อปรับความถี่ของกระแสไฟฟ้าที่ไหลเข้าสู่ปั๊ม ตามสมการ $q_{out,i} = k_p v_i$ ซึ่งใช้ในการคำนวณหาอัตราการไหลของน้ำจากปั๊มที่ป้อนเข้าสู่ระบบ

ตารางที่ 4. 1 แสดงอัตราการไหลของน้ำที่ป้อนจากปั๊มตัวที่ 1 และ 2 เข้าสู่ระบบ

	ถังใบที่ 1	ถังใบที่ 2	ถังใบที่ 3	ถังใบที่ 4
ปั๊มน้ำตัวที่ 1	$\gamma_1 k_p v_1$	-	-	$(1-\gamma_1)k_p v_1$
ปั๊มน้ำตัวที่ 2	-	$\gamma_2 k_p v_2$	$(1-\gamma_2)k_p v_2$	-

จากข้อมูลในตารางที่ 4.1 และสมการที่ (23) ซึ่งถูกนำไปใช้ในสมการ (21) นั้น สามารถเขียนสมการสมดุลมวลสารรอบถังแต่ละใบได้ดังสมการที่ 24-31

สำหรับถังใบที่ 1

$$A \frac{d(h_1)}{dt} = \gamma_1 q_{pump,1} + q_{out,3} - q_{out,1} \quad (24)$$

$$\dot{h}_1 = \left(\frac{1}{A}\right) \times (\gamma_1 k_p v_1 + m_3 \sqrt{h_3} - m_1 \sqrt{h_1}) \quad (25)$$

สำหรับถังใบที่ 2

$$A \frac{d(h_2)}{dt} = \gamma_2 q_{pump,2} + q_{out,4} - q_{out,2} \quad (26)$$

$$\dot{h}_2 = \left(\frac{1}{A}\right) \times (\gamma_2 k_p v_2 + m_4 \sqrt{h_4} - m_2 \sqrt{h_2}) \quad (27)$$

สำหรับถังใบที่ 3

$$A \frac{d(h_3)}{dt} = (1-\gamma_2) q_{pump,2} - q_{out,3} \quad (28)$$

$$\dot{h}_3 = \left(\frac{1}{A}\right) \times ((1-\gamma_2) k_p v_2 - m_3 \sqrt{h_3}) \quad (29)$$

สำหรับถังใบที่ 4

$$A \frac{d(h_4)}{dt} = (1 - \gamma_1) q_{pump,1} - q_{out,4} \quad (30)$$

$$\dot{h}_2 = \left(\frac{1}{A}\right) \times ((1 - \gamma_1) k_p v_1 - m_4 \sqrt{h_4}) \quad (31)$$

โดยที่	A	คือ พื้นที่หน้าตัดของถังที่ i
	V_i	คือ ปริมาตรน้ำในถังที่ i
	h_i	คือ ระดับความสูงของน้ำในถังที่ i
	g	คือ ค่าความเร่งเนื่องจากแรงโน้มถ่วงของโลก
	k_p	คือ ค่าคงที่การไหลของปั๊ม
	γ_i	คือ สัดส่วนในการปรับวาล์วเพื่อจ่ายน้ำจากปั๊มที่ i ไปยังถังด้านล่าง
	v_i	คือ แรงดันไฟฟ้าที่ป้อนให้กับปั๊มตัวที่ i
	G_f	คือ ค่าความถ่วงจำเพาะของของเหลว
	ρ	คือ ค่าความหนาแน่นของน้ำ
	$c_{v,i}$	คือ ค่าสัมประสิทธิ์การไหลของวาล์วตัวที่ i
	$q_{pump,i}$	คือ อัตราการไหลของน้ำจากปั๊มตัวที่ i

ตารางที่ 4. 2 แสดงค่าคงที่ในแบบจำลองทางคณิตศาสตร์ของระบบสี่ถัง

	สัญลักษณ์	ค่าคงที่	หน่วย
พื้นที่หน้าตัดของถัง	A	160.6	cm^2
ค่าความหนาแน่นของน้ำ	ρ	0.1	kg / cm^3
ค่าความถ่วงจำเพาะของของเหลว	G_f	1	-
ความเร่งเนื่องจากแรงโน้มถ่วงของโลก	g	981	cm / s^2

4.4 วิธีการสร้างแบบจำลองเชิงเส้นซึ่งมีการเปลี่ยนแปลงตามเวลาของระบบสี่ถึง

จากแบบจำลองที่มีความไม่เป็นเชิงเส้นของระบบสี่ถึงตามสมการ (25), (27), (29), (31) สามารถจัดรูปใหม่ให้อยู่ในรูปสมการปริภูมิสถานะตามแบบจำลองเชิงเส้นซึ่งมีการเปลี่ยนแปลงตามเวลาดังนี้

$$\begin{pmatrix} \dot{h}_1 \\ \dot{h}_2 \\ \dot{h}_3 \\ \dot{h}_4 \end{pmatrix} = \begin{pmatrix} -\frac{m_1}{A\sqrt{h_1}} & 0 & \frac{m_3}{A\sqrt{h_3}} & 0 \\ 0 & -\frac{m_2}{A\sqrt{h_2}} & 0 & \frac{m_4}{A\sqrt{h_4}} \\ 0 & 0 & -\frac{m_3}{A\sqrt{h_3}} & 0 \\ 0 & 0 & 0 & -\frac{m_4}{A\sqrt{h_4}} \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} + \begin{pmatrix} \frac{k_p \gamma_1}{A} & 0 \\ 0 & \frac{k_p \gamma_2}{A} \\ 0 & \frac{k_p (1-\gamma_2)}{A} \\ \frac{k_p (1-\gamma_1)}{A} & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (32)$$

จากนั้นอาศัยวิธีการ Euler's Forward Differential ในการจัดรูปแบบจำลองเชิงเส้นซึ่งเปลี่ยนแปลงตามเวลาของระบบสี่ถึงให้อยู่ในรูปของสมการปริภูมิสถานะแบบวิฤต ซึ่งมีรูปแบบสมการดังนี้

$$\dot{x}(k) \approx \frac{x(k+1) - x(k)}{\Delta t} \quad (33)$$

ใช้หลักการในสมการ (33) จัดรูปสมการ (32) ใหม่ดังนี้

$$\begin{pmatrix} h_1(k+1) \\ h_2(k+1) \\ h_3(k+1) \\ h_4(k+1) \end{pmatrix} = \begin{pmatrix} 1 - \frac{m_1 \Delta t}{A\sqrt{h_1(k)}} & 0 & \frac{m_3 \Delta t}{A\sqrt{h_3(k)}} & 0 \\ 0 & 1 - \frac{m_2 \Delta t}{A\sqrt{h_2(k)}} & 0 & \frac{m_4 \Delta t}{A\sqrt{h_4(k)}} \\ 0 & 0 & 1 - \frac{m_3 \Delta t}{A\sqrt{h_3(k)}} & 0 \\ 0 & 0 & 0 & 1 - \frac{m_4 \Delta t}{A\sqrt{h_4(k)}} \end{pmatrix} \begin{pmatrix} h_1(k) \\ h_2(k) \\ h_3(k) \\ h_4(k) \end{pmatrix} + \begin{pmatrix} \frac{k_p \gamma_1}{A} \\ 0 \\ 0 \\ \frac{k_p (1-\gamma_1)}{A} \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ \frac{k_p \gamma_2}{A} \\ \frac{k_p (1-\gamma_2)}{A} \\ 0 \end{pmatrix} v_2$$

$$+ \begin{pmatrix} \frac{\Delta tk_p \gamma_1}{A} & 0 \\ A & \frac{\Delta tk_p \gamma_2}{A} \\ 0 & A \\ 0 & \frac{\Delta tk_p (1-\gamma_2)}{A} \\ \frac{\Delta tk_p (1-\gamma_1)}{A} & A \\ A & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (34)$$

จากสมการ (34) ทำให้อยู่ในรูปของตัวแปรเบี่ยงเบน (Deviation form) โดยกำหนดให้ $\bar{h}_i = h_i - h_{ss,i}$, $\bar{v}_j = v_j - v_{ss,j}$ โดยที่ \bar{h}_i และ \bar{v}_j เป็นตัวแปรที่อยู่ในรูปค่าเบี่ยงเบน ส่วน $h_{ss,i}$ และ $v_{ss,j}$ คือตัวแปรในสภาวะคงตัว จากสมการ (34) กำหนดให้

$$h(k+1) = \begin{pmatrix} h_1(k+1) \\ h_2(k+1) \\ h_3(k+1) \\ h_4(k+1) \end{pmatrix}, a = \begin{pmatrix} 1 - \frac{m_1 \Delta t}{A \sqrt{h_1(k)}} & 0 & \frac{m_3 \Delta t}{A \sqrt{h_3(k)}} & 0 \\ 0 & 1 - \frac{m_2 \Delta t}{A \sqrt{h_2(k)}} & 0 & \frac{m_4 \Delta t}{A \sqrt{h_4(k)}} \\ 0 & 0 & 1 - \frac{m_3 \Delta t}{A \sqrt{h_3(k)}} & 0 \\ 0 & 0 & 0 & 1 - \frac{m_4 \Delta t}{A \sqrt{h_4(k)}} \end{pmatrix}$$

$$h(k) = \begin{pmatrix} h_1(k) \\ h_2(k) \\ h_3(k) \\ h_4(k) \end{pmatrix}, b = \begin{pmatrix} \frac{\Delta tk_p \gamma_1}{A} & 0 \\ A & \frac{\Delta tk_p \gamma_2}{A} \\ 0 & A \\ \frac{\Delta tk_p (1-\gamma_2)}{A} & A \\ \frac{\Delta tk_p (1-\gamma_1)}{A} & A \\ A & 0 \end{pmatrix}, v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

สมการที่ 35 แสดงสมการที่สภาวะใดๆ ดังนี้

$$h(k+1) = ah(k) + bv(k) \quad (35)$$

สมการที่ 36 แสดงสมการที่สภาวะคงตัว ดังนี้

$$h_{ss}(k+1) = a_{ss} h_{ss}(k) + b_{ss} v_{ss}(k) \quad (36)$$

จัดรูปสมการให้อยู่ในรูปสมการตัวแปรเบี่ยงเบนจากสภาวะคงตัว

$$\bar{h}(k+1) = ah(k) - a h_{ss}(k) + bv(k) - b v_{ss}(k) \quad (35)-(36)$$

$$\bar{h}(k+1) = a(\bar{h}(k) + h_{ss}(k)) - a h_{ss}(k) + b(\bar{v}(k) + v_{ss}(k)) - b v_{ss}(k)$$

$$\bar{h}(k+1) = a\bar{h}(k) + (a - a_{ss})h_{ss}(k) + b\bar{v}(k) + (b - b_{ss})v_{ss}(k)$$

$$\bar{h}(k+1) = a\bar{h}(k) + (a - a_{ss})h_{ss}(k)\bar{h}(k)^{-1}\bar{h}(k) + b\bar{h}(k) + (b - b_{ss})v_{ss}(k)\bar{v}(k)^{-1}\bar{v}(k)$$

$$\bar{h}(k+1) = \left[a + (a - a_{ss})h_{ss}(k)\bar{h}(k)^{-1} \right] \bar{h}(k) + \left[b + (b - b_{ss})v_{ss}(k)\bar{v}(k)^{-1} \right] \bar{v}(k) \quad (37)$$

สมการที่ (37) อยู่ในรูปของตัวแปรเบี่ยงเบนจากสภาวะคงตัว โดยที่ $a_{ss}, h_{ss}, b_{ss}, v_{ss}$ คือ เมตริกซ์ a, h, b, v ที่สภาวะคงตัว เมื่อเทียบกับแบบจำลองเชิงเส้นซึ่งมีการเปลี่ยนแปลงตามเวลาดังสมการต่อไปนี้ $x(k+1) = A(k)x(k) + B(k)u(k)$ พบว่าตัวแปรต่างๆมีความสัมพันธ์กันดังสมการที่ 38-41

$$x(k) = \bar{h}(k) \quad (38)$$

$$u(k) = \bar{v}(k) \quad (39)$$

$$A(k) = \left[a + (a - a_{ss})h_{ss}(k)\bar{h}(k)^{-1} \right] \quad (40)$$

$$B(k) = \left[b + (b - b_{ss})v_{ss}(k)\bar{v}(k)^{-1} \right] \quad (41)$$

4.5 ระบบสี่ถังภายใต้สภาวะเฟสต่ำสุดและสภาวะไม่มีเฟสต่ำสุด

การควบคุมระดับน้ำของระบบสี่ถังในงานวิจัยนี้ จะสนใจควบคุมระบบภายใต้สภาวะเฟสต่ำสุดและสภาวะไม่มีเฟสต่ำสุด โดยทั้งสองสภาวะจะเกิดจากการปรับทิศทางของวาล์วซึ่งปรับสัดส่วนการไหลของน้ำจากบ่ิมเข้าสู่ถังด้านล่าง จากงานวิจัยของ [7], [8] และ [9] ได้อธิบายเกี่ยวกับความสัมพันธ์ในการปรับค่าสัดส่วนของวาล์วต่อสภาวะต่างๆ โดยการปรับค่าสัดส่วนวาล์วให้อยู่ในช่วง $1 < \gamma_1 + \gamma_2 < 2$ การปรับทิศทางการไหลของน้ำจากบ่ิมทั้งสองในลักษณะนี้จะทำให้น้ำไหลเข้าสู่ถังด้านล่างมากกว่าถังด้านบน ส่งผลให้ระบบสี่ถังอยู่ภายใต้สภาวะเฟสต่ำสุด ในกรณีการปรับค่าสัดส่วนวาล์วอยู่ในช่วง $0 < \gamma_1 + \gamma_2 < 1$ การปรับทิศทางการไหลของน้ำจากบ่ิมทั้งสองในลักษณะนี้จะทำให้น้ำไหลเข้าสู่ถังด้านล่างมากกว่าถังด้านล่าง ส่งผลให้ระบบสี่ถังอยู่ภายใต้สภาวะไม่มีเฟสต่ำสุด ซึ่งในสภาวะไม่มีเฟสต่ำสุดมักจะเกิดพฤติกรรมการตอบสนองแบบตรงกันข้าม (Inverse response) ขึ้น

4.6 ขอบเขตความไม่แน่นอนของแบบจำลองระบบสี่ถัง

จากสมการแบบจำลองเชิงเส้นซึ่งมีการเปลี่ยนแปลงตามเวลาของระบบสี่ถัง $x(k+1) = A(k)x(k) + B(k)u(k)$ พบว่าในสภาวะจริงของระบบสี่ถังนั้น ตัวแปรในเมตริกซ์ $A(k)$ และ $B(k)$ จะมีค่าไม่แน่นอน ส่งผลให้แบบจำลองของระบบสี่ถังมีความไม่แน่นอนเกิดขึ้นดังนั้นก็จึงต้องมีการกำหนดขอบเขตความไม่แน่นอนของแบบจำลองขึ้น โดยจะมีการกำหนดขอบเขตทั้งในเมตริกซ์ $A(k)$ และ $B(k)$

ในการกำหนดขอบเขตของเมตริกซ์ $A(k)$ จากสมการที่ 40 พบว่าเมตริกซ์ $A(k)$ นั้นจะมีค่าเปลี่ยนไปตามความสูงของระดับน้ำ ดังนั้น ขอบเขตความไม่แน่นอนของ เมตริกซ์ $A(k)$ จะเกิดจากขอบเขตของระดับน้ำในถังทั้ง 4 ใบ โดยกำหนดให้ ระดับน้ำต่ำสุดของแต่ละถังซึ่งใช้ในการทดลองมีค่า เท่ากับ 4 เซนติเมตร และระดับน้ำสูงสุดของแต่ละถังซึ่งใช้ในการทดลองมีค่า เท่ากับ 50 เซนติเมตร นอกจากนี้ ยังมีตัวแปร m_1, m_2, m_3, m_4 ซึ่งเป็นตัวแปรที่มีค่าไม่แน่นอน อันเป็นผลมาจากค่าสัมประสิทธิ์ของวาล์วได้ถังทั้ง 4 ใบ ซึ่งในงานวิจัยนี้ได้กำหนดขอบเขตความไม่แน่นอนของ ค่าสัมประสิทธิ์ของวาล์วได้ถังแต่ละถังให้มีค่าอยู่ในช่วง +30 เปอร์เซ็นต์ และ -30 เปอร์เซ็นต์ จากค่าสัมประสิทธิ์ของวาล์วได้ถังที่สภาวะคงตัว

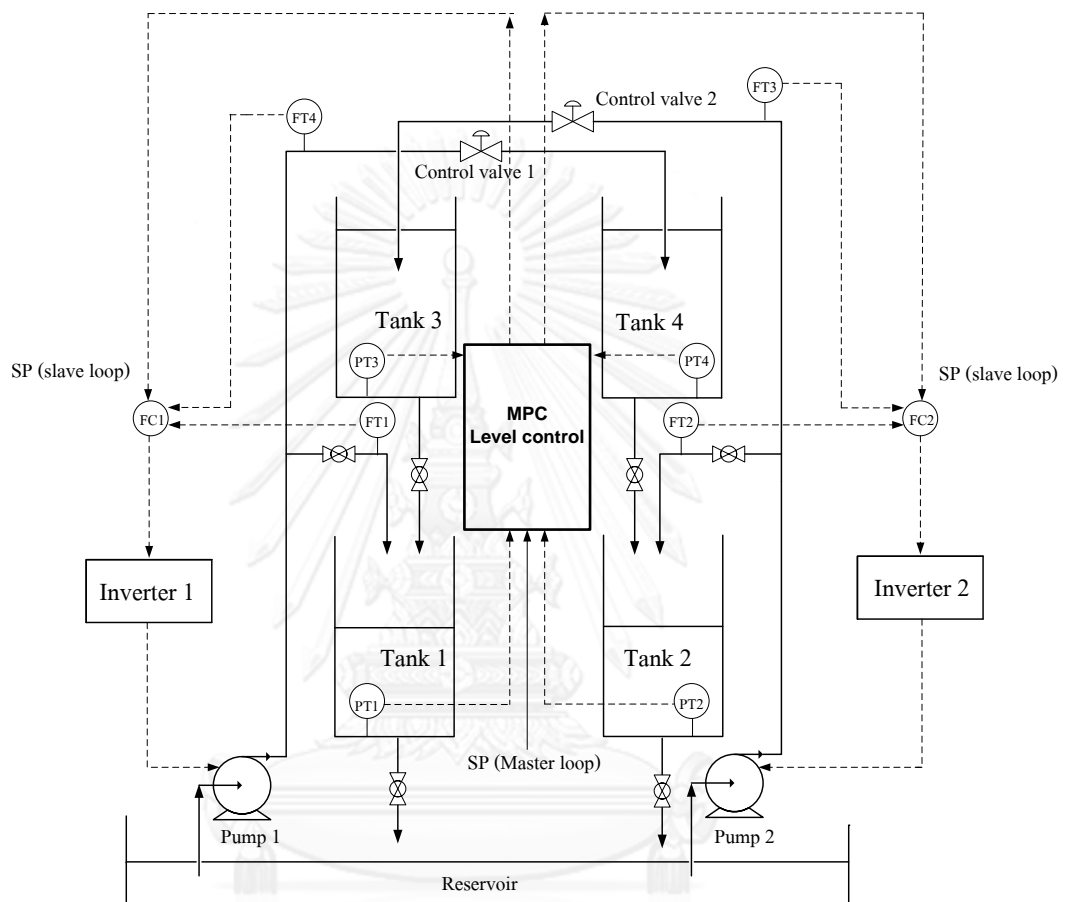
ในส่วนการกำหนดขอบเขตของเมตริกซ์ $B(k)$ จากสมการที่ 39 พบว่าเมตริกซ์ $B(k)$ นั้น มีความสัมพันธ์กับค่าคงที่ของปั๊ม ($k_{p,1}, k_{p,2}$) สัดส่วนการปรับวาล์วเพื่อแบ่งน้ำจากปั๊ม (γ_1, γ_2) และกำลังไฟฟ้าที่ป้อนให้แก่ปั๊ม (v_1, v_2) ซึ่งในระบบสี่ถังจริงนั้นตัวแปรทั้งสามมีค่าไม่แน่นอน ดังนั้น จึงมีการกำหนดขอบเขตของเมตริกซ์ $B(k)$ ขึ้น ค่าของเขตขอบตัวแปรต่างๆในระบบสี่ถังภายใต้สภาวะที่มีเฟสต่ำสุดและไม่มีเฟสต่ำสุดจะแสดงไว้ในบทที่ 6

ข้อมูลค่าพารามิเตอร์และขอบเขตของตัวแปรต่างๆในสมการแบบจำลองระบบสี่ถังจะแสดงไว้ในบทที่ 6 ในส่วนของการทดสอบสมรรถนะของปั๊ม

4.7 ตัวควบคุมระบบสี่ถัง

ในการควบคุมระบบสี่ถังจริงนั้น นอกจากปัญหาของตัวแปรในแบบจำลองที่มีความคลาดเคลื่อนแล้วนั้น ยังพบว่าอัตราการไหลของน้ำจากปั๊มทั้ง 2 ตัวมีค่าไม่คงที่ ดังนั้นการทดลองกับระบบสี่ถังจริงจะทำการควบคุมแบบคาสเคด (Cascade control) โดยให้ลู่วควบคุมหลักเป็นการควบคุมระดับน้ำในถัง ส่วนในลู่วควบคุมรองจะควบคุมอัตราการไหลของน้ำให้คงที่ตามการปรับแรงดันไฟฟ้าแก่ปั๊มทั้ง 2 ตัว ลักษณะความสัมพันธ์ของระบบควบคุมกับระบบสี่ถังแสดงดังรูปที่ 4.5 โดยลู่วหลักของการควบคุมแบบคาสเคดจะตรวจรู้ค่าความสูงของระดับในแต่ละถังผ่านอุปกรณ์ตรวจวัดแรงดัน โดยสัญญาณขาออกจากลู่วหลักจะถูกส่งมาเป็นค่าเป้าหมาย (Set point) ของลู่วรอง

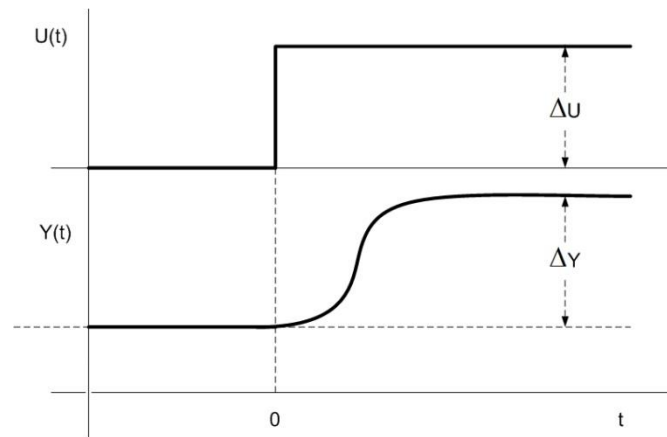
ซึ่งเป็นรูปในการควบคุมอัตราการไหลของน้ำ โดยสัญญาณขาออกจากลูประงก็ถูกส่งไปยังอินเวอร์เตอร์เพื่อปรับความถี่ของมอเตอร์ที่เหมาะสมเพื่อให้อัตราการไหลของน้ำที่จ่ายจากปั๊มคงที่ตามค่าที่ต้องการ



รูปที่ 4. 5 แสดงความสัมพันธ์ระหว่างระบบควบคุมกับระบบส่ถึง

ในส่วนการออกแบบตัวควบคุมนั้น ตัวควบคุมหลักซึ่งเป็นรูปในการควบคุมระดับน้ำในถัง จะใช้การควบคุมตามที่งานวิจัยนี้นำเสนอ ส่วนตัวควบคุมลูประงซึ่งเป็นรูปในการควบคุมอัตราการไหลของน้ำให้คงที่ซึ่งจะเลือกใช้วิธีการควบคุมเชิงทำนายแบบจำลองคงแบบออฟไลน์ตามหัวข้อที่ 3.2 ซึ่งการควบคุมดังกล่าวก็จะมีอัลกอริทึมในการประกันเสถียรภาพของระบบจึงสามารถนำมาใช้ในการสร้างชุดคำสั่งควบคุม โดยจะสร้างแบบจำลองระหว่างอัตราไหลของน้ำและแรงดันไฟฟ้าซึ่งจ่ายแก่ปั๊มด้วยการทดสอบการตอบสนองของกระบวนการหรือเรียกว่าการทดสอบแบบสเต็ป (Step testing) วิธีการดังกล่าวจะทดสอบการเปลี่ยนแปลงแบบสเต็ปของตัวควบคุมในกรณีระบบเปิดที่ยังไม่มีตัว

ควบคุม โดยจะบันทึกค่าอัตราการไหลของน้ำที่เปลี่ยนแปลงไปในช่วงของการปรับค่าแรงดันไฟฟ้าแบบสแต็ป แสดงตัวอย่างของการทดสอบแบบสแต็ปดังรูปที่ 4.6



รูปที่ 4. 6 ผลการตอบสนองของระบบแบบสแต็ป

จากรูปที่ 4.6 U คือ แรงดันไฟฟ้าซึ่งมีขนาดเปลี่ยนแปลงไป ΔU และ Y คือ อัตราการไหลของน้ำซึ่งมีขนาดเปลี่ยนแปลงไป ΔY ในงานวิจัยนี้สนใจประยุกต์ใช้การตอบสนองแบบสแต็ปโดยใช้ความสัมพันธ์ระหว่างค่าแรงดันไฟฟ้าซึ่งจ่ายให้แก่ปั๊มและค่าอัตราการไหลของน้ำในรูปแบบของระบบสมการอันดับหนึ่งดังสมการที่ 42

$$\frac{Y(s)}{U(s)} = \frac{K_s}{\tau s + 1} \quad (42)$$

จัดรูปสมการที่ 42 ได้

$$sY(s)\tau + Y(s) = K_s \Delta U \quad (43)$$

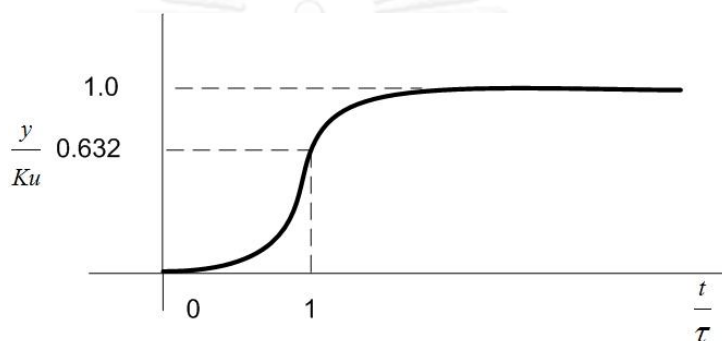
จากสมการที่ 43 แปลง Laplace transform จะได้

$$\tau \frac{dy(t)}{dt} + Y(t) = K_s u(t) \quad (44)$$

จัดรูปสมการ ที่ 44 โดยให้ $\frac{dy(t)}{dt} \approx \frac{y(t+1) - y(t)}{\Delta t}$ จะได้

$$y(t+1) = \left(1 - \frac{\Delta t}{\tau}\right)y(t) + \left(\frac{K \Delta t}{\tau}\right)u(t) \quad (45)$$

ในสมการที่ 45 เป็นสมการแบบจำลองซึ่งแสดงความสัมพันธ์ระหว่างแรงดันไฟฟ้า $u(t)$ ต่อ อัตราการไหลของน้ำ $y(t)$ ซึ่งแบบจำลองนี้จะนำไปใช้ในการควบคุมเชิงทำนายแบบจำลองคงแบบ ออฟไลน์ ส่วนตัวแปร τ (ค่าคงตัวของเวลา) สามารถหาได้จากการสร้างกราฟดังรูปที่ 4.7 จากกราฟ ดังกล่าว ณ ค่า $\frac{y}{Ku}$ เท่ากับ 0.632 ค่าคงตัวของเวลาจะเท่ากับค่าเวลา ณ ตำแหน่งนั้น[10]



รูปที่ 4. 7 เส้นโค้งการตอบสนองสำหรับการคำนวณค่าคงตัวของเวลา

จากสมการแบบจำลองของตัวควบคุมลูปรองดังสมการที่ 45 พบว่า ค่าคงตัวของเวลา (τ) และค่าอัตราขยายของตัวควบคุมลูปรอง (K_s) เป็นตัวแปรซึ่งส่งผลต่อความไม่แน่นอนของแบบจำลอง เพื่อให้สมการแบบจำลองสามารถใช้กับค่าแรงดันไฟฟ้าทุกค่าที่อยู่ในช่วงของการทดสอบระบบสี่ถึง จึงมีการจำกัดขอบเขตของตัวแปร τ และ K_s โดยทำการทดสอบผลการตอบสนองแบบสเต็ปภายใน ช่วงของแรงดันไฟฟ้าที่ใช้ในการทดสอบระบบสี่ถึง เพื่อดูค่า τ และ K_s ที่มีการเปลี่ยนแปลงในช่วง ของการปรับค่าแรงดันไฟฟ้า 1.86 ถึง 2.06 โวลต์ โดยข้อมูลค่าพารามิเตอร์และขอบเขตของตัวแปร ต่างๆในสมการที่ 45 จะแสดงไว้ในบทที่ 6 ในส่วนของการทดสอบสมรรถนะของปั๊ม

บทที่ 5 วิธีการดำเนินงานวิจัย

5.1 การหาสภาวะคงตัวของระบบภายใต้พฤติกรรมแบบเฟสต่ำสุด

- 1) เปิดวาล์วควบคุมคงที่ ที่ 100 เปอร์เซ็นต์ ปรับสัดส่วนของวาล์วตัวที่ 1 และ 2 ไว้ที่ 0.7
- 2) ใช้คำสั่งจากคอมพิวเตอร์ในการป้อนค่าแรงดันไฟฟ้า 1.96 โวลต์ จากอาร์ดูโนไปยังอินเวอร์เตอร์ของปั๊มทั้งสองตัว
- 3) ดูแนวโน้มของระดับน้ำที่สภาวะคงตัว พร้อมทั้งปรับวาล์วได้ถึงให้ได้ระดับน้ำในแต่ละถังตามต้องการ
- 4) หยุดระบบ
- 5) เริ่มทดสอบหาสภาวะคงตัวจากถังเปล่า โดยใช้ชุดคำสั่งตัวควบคุมลูปรองเพื่อควบคุมอัตราการไหลของปั๊มตัวที่หนึ่งให้มีค่าเท่ากับ 15.5 L/min และปั๊มตัวที่สองให้มีค่าเท่ากับ 16 L/min ในขณะที่ทดสอบจะมีการตรวจวัดค่าระดับน้ำในถังทุกๆ 0.36 วินาที เพื่อดูพฤติกรรมแบบเฟสต่ำสุดของระบบ
- 6) รอให้ระบบเข้าสู่สภาวะคงตัว แล้วบันทึกค่าระดับน้ำที่สภาวะคงตัวในแต่ละถัง

5.2 การหาสภาวะคงตัวของระบบภายใต้พฤติกรรมแบบไม่มีเฟสต่ำสุด

- 1) เปิดวาล์วควบคุมคงที่ ที่ 100 เปอร์เซ็นต์ ปรับสัดส่วนของวาล์วตัวที่ 1 ที่ 0.13 และ ปรับสัดส่วนของวาล์วตัวที่ 2 ที่ 0.165
- 2) ใช้คำสั่งจากคอมพิวเตอร์ในการป้อนค่าแรงดันไฟฟ้า 1.96 โวลต์ จากอาร์ดูโนไปยังอินเวอร์เตอร์ของปั๊มทั้งสองตัว
- 3) ที่เวลา 1.05 นาที วัดระดับน้ำทั้ง 4 ถัง เก็บไว้เป็นระดับน้ำเริ่มต้นตอนควบคุมระบบจริงด้วยชุดคำสั่งที่งานวิจัยเสนอ
- 4) ดูแนวโน้มของระดับน้ำที่สภาวะคงตัว พร้อมทั้งปรับวาล์วได้ถึงให้ได้ระดับน้ำในแต่ละถังตามต้องการ
- 5) หยุดระบบ
- 6) เริ่มระบบจากถังเปล่าโดยป้อนค่าแรงดันไฟฟ้า 1.96 โวลต์ จากอาร์ดูโนไปยังอินเวอร์เตอร์ของปั๊มทั้งสองตัว

- 7) ณ เวลา ที่ 1.05 นาที เริ่มทดสอบหาสภาวะคงตัวโดยใช้ลูบควบคุมรองเพื่อควบคุมอัตราการไหลของปั๊มตัวที่หนึ่งให้มีค่าเท่ากับ 11.79 L/min และปั๊มตัวที่สองให้มีค่าเท่ากับ 10.66 L/min ในขณะที่ทดสอบจะมีตรววัดค่าระดับน้ำในถังทุกๆ 0.36 วินาที เพื่อดูพฤติกรรมแบบไม่มีเฟสต่ำสุดของระบบ
- 8) รอให้ระบบเข้าสู่สภาวะคงตัว แล้วบันทึกค่าระดับน้ำที่สภาวะคงตัวในแต่ละถัง

5.3 ขั้นตอนการควบคุมระบบ

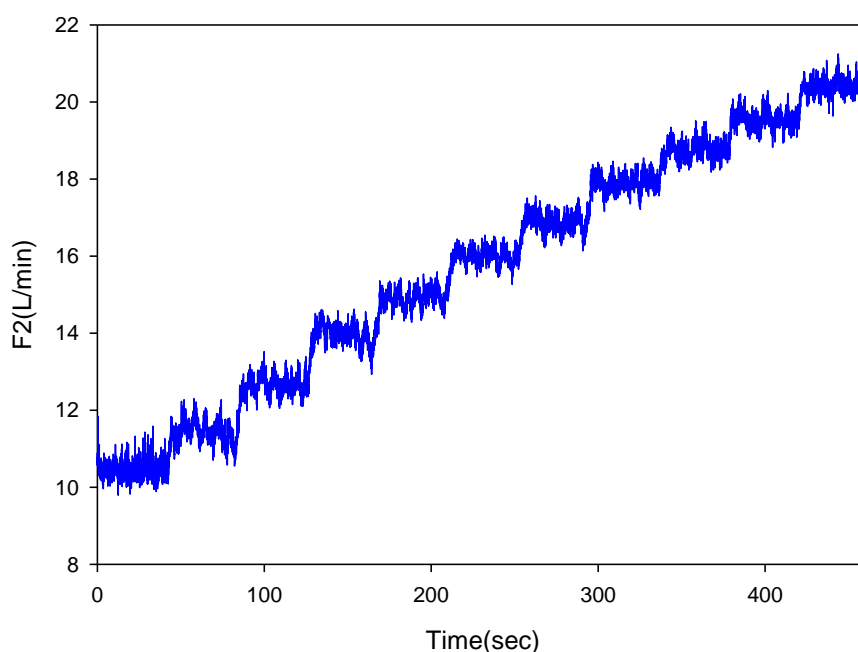
- 1) เตรียมระบบสี่ถังภายใต้สภาวะเฟสต่ำสุดเพื่อทดสอบการควบคุมซึ่งใช้ระเบียบวิธีการควบคุมที่งานวิจัยเสนอ
- 2) สร้างแบบจำลองของระบบสี่ถังภายใต้สภาวะเฟสต่ำสุดโดยใช้ทฤษฎีสมดุลมวลสาร
- 3) สร้างแบบจำลองความสัมพันธ์ระหว่างค่าแรงดันไฟฟ้าที่จ่ายให้แก่ปั๊มต่ออัตราการไหลด้วยการทดสอบการตอบสนองแบบสเต็ป
- 4) ทดสอบสมรรถนะการทำงานของปั๊มทั้งสองตัวเพื่อหาค่าขอบเขตของตัวแปรและค่าพารามิเตอร์ในสมการแบบจำลอง
- 5) สร้างชุดคำสั่งการควบคุมตามระเบียบวิธีการควบคุมที่งานวิจัยเสนอ ได้ชุดคำสั่งลูบหลักของระบบคาสเคดในเพื่อควบคุมระดับน้ำ
- 6) สร้างชุดคำสั่งการควบคุมตามการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ได้ชุดคำสั่งลูบรองของระบบคาสเคดในการควบคุมอัตราการไหลของน้ำ
- 7) ควบคุมระบบสี่ถังภายใต้สภาวะเฟสต่ำสุด บันทึกผลการควบคุมระบบ
- 8) เตรียมระบบสี่ถังในสภาวะไม่มีเฟสต่ำสุดเพื่อทดสอบการควบคุมซึ่งใช้ระเบียบวิธีการควบคุมที่งานวิจัยเสนอ
- 9) ทำตามขั้นตอนตั้งแต่ขั้นตอนที่สองถึงขั้นตอนที่หก
- 10) ควบคุมระบบสี่ถังภายใต้สภาวะไม่มีเฟสต่ำสุด บันทึกผลการควบคุมระบบ

บทที่ 6 ผลและวิเคราะห์ผลการทดลอง

ในบทนี้จะเป็นส่วนของการเสนอผลการหาค่าพารามิเตอร์ในแบบจำลองรวมถึงผลการทดสอบชุดคำสั่งที่งานวิจัยเสนอกับระบบสั่งและวิเคราะห์ผลการทดลอง ในการทดลองระบบสั่งนี้ ตัวแปรสถานะ คือระดับน้ำทั้ง 4 ถัง โดยจะทำการควบคุมระดับน้ำในถังที่ 1 และ 2 โดยให้ค่าเป้าหมาย(set point) ของระดับน้ำทั้งสองถัง คือ ค่าระดับน้ำที่สถานะคง

6.1 ผลการทดสอบสมรรถนะการทำงานของปั๊ม

การทดสอบสมรรถนะของปั๊มทั้งสองทำขึ้นเพื่อหาค่าพารามิเตอร์บางค่าในแบบจำลองเช่น ค่าคงตัวของปั๊ม ค่าสัดส่วนของวาล์ว ค่าอัตราการไหลของปั๊ม เป็นต้น ตัวอย่างการทดสอบสมรรถนะของปั๊มตัวที่สองในสถานะเฟสต่ำสุดแสดงไว้ดังรูปที่ 6.1

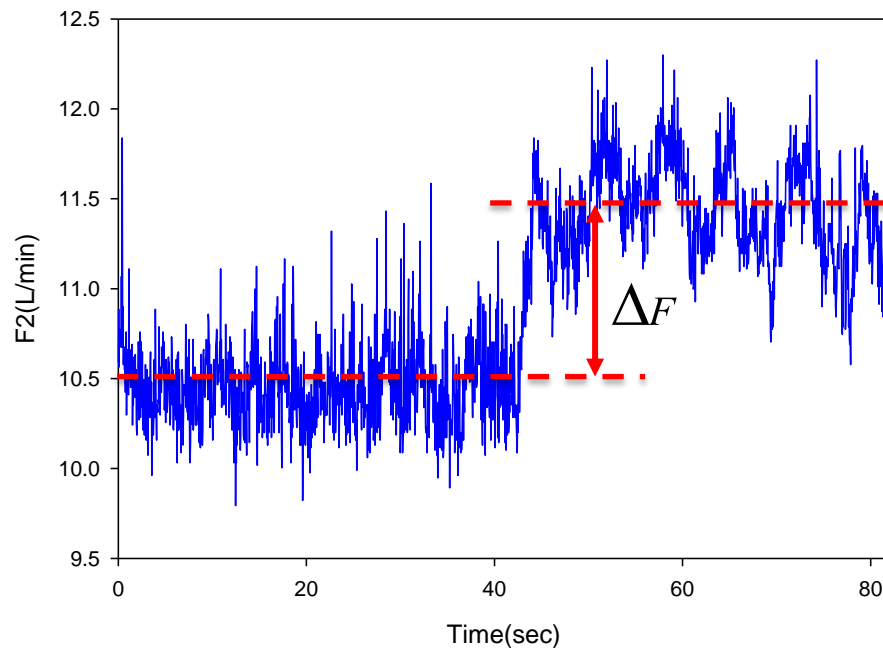


รูปที่ 6. 1 ผลการทดสอบสมรรถนะปั๊มตัวที่สองในสถานะเฟสต่ำสุด

จากรูปที่ 6.1 เป็นตัวอย่างการทดสอบสมรรถนะปั๊มโดยแสดงผลการทดสอบกับปั๊มตัวที่สองในสถานะเฟสต่ำสุดโดยจ่ายค่าแรงดันไฟฟ้าแก่ปั๊มค่าหนึ่งคงที่ โดยจะเปลี่ยนค่าแรงดันไฟฟ้าเพิ่มขึ้นทุกๆ 42 วินาที การทดสอบดังกล่าวจะเขียนชุดคำสั่งในโปรแกรมแมทแล็บ (MATLAB) ขึ้น โดยเริ่มจากค่าแรงดันไฟฟ้าต่ำสุด 1.86 โวลต์ จนถึงค่าแรงดันไฟฟ้าสูง 2.06 โวลต์ โดยค่าแรงดันไฟฟ้าที่เปลี่ยนไปจะมีค่าต่างกันประมาณ 0.02 โวลต์ เหตุผลของการทดสอบแรงดันไฟฟ้าที่ค่าแรงดันไฟฟ้า

ต่างกันน้อยสุด 0.02 โวลต์ เนื่องจากข้อจำกัดของอุปกรณ์อาร์ดูโนที่สามารถส่งค่าแรงดันไฟฟ้าที่ต่างกันได้น้อยที่สุดแค่ 0.02 โวลต์

ในส่วนการหาค่าตัวพารามิเตอร์ต่างๆ จะแสดงตัวอย่างการคำนวณจากกราฟในรูปที่ 6.1 ในช่วงเวลาตั้งแต่เริ่มต้นจนถึงวินาทีที่ 84 แสดงกราฟดังรูปที่ 6.2



รูปที่ 6.2 ผลการทดสอบสมรรถนะปั๊มตัวที่สองในสภาวะเฟสต่ำสุดในช่วงเวลาเริ่มต้นจนถึงวินาทีที่ 84

จากกราฟในรูปที่ 6.2 แสดงช่วงการเปลี่ยนค่าแรงดันไฟฟ้าโดยเริ่มจากค่าแรงดันไฟฟ้าเริ่มต้น 1.86 โวลต์ หลังจากเวลาผ่านไป 42 วินาที จะเป็นช่วงการเปลี่ยนค่าแรงดันไฟฟ้าเพิ่มขึ้นเป็น 1.88 โวลต์ โดยในแต่ละช่วงแรงดันไฟฟ้าที่จ่ายให้แก่ปั๊มนั้นสามารถหาค่าคงที่ของปั๊มได้จากสมการที่ 46

$$K_p = \frac{F}{v} \quad (46)$$

โดยที่ F คือ ค่าเฉลี่ยของอัตราการไหลน้ำจากปั๊มที่ค่าแรงดันไฟฟ้า v โวลต์

นอกจากนี้ในช่วงการเปลี่ยนค่าอัตราการไหลของน้ำอันเนื่องมาจากการเปลี่ยนเพิ่มค่าแรงดันไฟฟ้าไป 0.02 โวลต์ สามารถหาค่าคงตัวของเวลา (τ) และค่าอัตราขยายของตัวควบคุมลูปรอง (K_s) โดยค่าคงตัวของเวลาจะเท่ากับระยะเวลาในการเปลี่ยนแปลงค่าอัตราการไหลของน้ำจาก 0 เปอร์เซ็นต์ เป็น 63.2 เปอร์เซ็นต์ ในส่วนของค่าอัตราขยายของตัวควบคุมลูปรองสามารถหาได้จากสมการที่ 47

$$K_s = \frac{\Delta F}{\Delta v} \quad (47)$$

โดยที่ ΔF คือ ค่าอัตราการไหลของน้ำที่เปลี่ยนแปลงไปอันเนื่องมาจากการเปลี่ยนแปลงค่าแรงดันไฟฟ้าเท่ากับ Δv โวลต์

ในการหาค่าขอบเขตของค่าพารามิเตอร์ทั้งค่าคงตัวของปั๊ม ค่าสัดส่วนของวาล์ว ค่าอัตราการไหลของปั๊ม จะทำการคำนวณค่าของพารามิเตอร์ต่างๆช่วงการเปลี่ยนค่าแรงดันไฟฟ้า ตั้งแต่ค่าแรงดันไฟฟ้าเริ่มต้น 1.86 ถึง 2.06 โวลต์ เพื่อดูขอบเขตของค่าพารามิเตอร์และข้อมูลที่สภาวะคงตัว ซึ่งคำนวณได้จากการทดสอบสมรรถนะของปั๊มทั้งสองตัวแสดงไว้ในตารางที่ 6.1-6.3

โดย กำหนดให้ P^- แสดงถึงระบบสี่ถังภายใต้สภาวะเฟสต่ำสุดและ P^+ แสดงถึงระบบสี่ถังภายใต้สภาวะไม่มีเฟสต่ำสุด

ตารางที่ 6. 1 ข้อมูลค่าพารามิเตอร์ที่สภาวะคงตัว

	สัญลักษณ์	P^-	P^+	หน่วย
แรงดันไฟฟ้า	$v_{1,ss}, v_{2,ss}$	1.96	1.96	V
ค่าคงที่ปั๊มตัวที่ 1	$k_{pss,1}$	136.05	100.125	cm^3 / sV
ค่าคงที่ปั๊มตัวที่ 2	$k_{pss,2}$	140.31	90.16	cm^3 / sV
ค่าสัดส่วนการปรับวาล์วตัวที่ 1	$\gamma_{1,ss}$	0.7	0.131	-
ค่าสัดส่วนการปรับวาล์วตัวที่ 2	$\gamma_{2,ss}$	0.7	0.165	-
อัตราการไหลของปั๊มตัวที่ 1	$F_{1,ss}$	15.55	11.79	L / min
อัตราการไหลของปั๊มตัวที่ 2	$F_{2,ss}$	16	10.66	L / min
อัตราการไหลจากปั๊มที่ 1 สู่อ่างที่ 1	$f_{1,ss}$	10.85	1.55	L / min
อัตราการไหลจากปั๊มที่ 1 สู่อ่างที่ 2	$f_{2,ss}$	11.2	1.76	L / min
อัตราการไหลจากปั๊มที่ 1 สู่อ่างที่ 3	$f_{3,ss}$	4.65	8.9	L / min
อัตราการไหลจากปั๊มที่ 1 สู่อ่างที่ 4	$f_{4,ss}$	4.7	10.24	L / min
ค่าพารามิเตอร์การไหลวาล์วได้อ่างที่ 1	$m_{1,ss}$	40.6837	35.6707	$cm^{2.5} / s$
ค่าพารามิเตอร์การไหลวาล์วได้อ่างที่ 2	$m_{2,ss}$	40.8456	41.5586	$cm^{2.5} / s$
ค่าพารามิเตอร์การไหลวาล์วได้อ่างที่ 3	$m_{3,ss}$	19.0045	24.0534	$cm^{2.5} / s$
ค่าพารามิเตอร์การไหลวาล์วได้อ่างที่ 4	$m_{4,ss}$	16.1042	27.8921	$cm^{2.5} / s$

ตารางที่ 6. 2 ข้อมูลขอบเขตค่าพารามิเตอร์ในแบบจำลองของตัวควบคุมหลัก

	สัญลักษณ์	P^-	P^+
แรงดันไฟฟ้าต่ำสุด	vn	1.86	1.86
แรงดันไฟฟ้าสูงสุด	vx	2.06	2.06
ค่าคงที่ปั๊มตัวที่ 1 ต่ำสุด	$k_{p,1} n$	83	86
ค่าคงที่ปั๊มตัวที่ 1 สูงสุด	$k_{p,1} x$	142	113
ค่าคงที่ปั๊มตัวที่ 2 ต่ำสุด	$k_{p,2} x$	86	80
ค่าคงที่ปั๊มตัวที่ 2 สูงสุด	$k_{p,2} n$	171	113
ค่าสัดส่วนการปรับวาล์วตัวที่ 1 ต่ำสุด	$\gamma_1 n$	0.66	0.125
ค่าสัดส่วนการปรับวาล์วตัวที่ 1 สูงสุด	$\gamma_1 x$	0.94	0.142
ค่าสัดส่วนการปรับวาล์วตัวที่ 2 ต่ำสุด	$\gamma_2 n$	0.62	0.146
ค่าสัดส่วนการปรับวาล์วตัวที่ 2 สูงสุด	$\gamma_2 x$	0.99	0.222

ตารางที่ 6. 3 ข้อมูลขอบเขตค่าพารามิเตอร์ในแบบจำลองของตัวควบคุมประกอบ

	สัญลักษณ์	ตัวควบคุมที่ 1	ตัวควบคุมที่ 2	หน่วย
ระยะเวลาในการสุ่มรับค่า	Δt	0.042	0.042	sec
ค่าคงตัวของเวลาน้อยที่สุด	τ_n	0.0466	0.105	sec
ค่าคงตัวของเวลามากที่สุด	τ_x	2.238	1.836	sec
ค่าอัตราขยายของตัวควบคุมน้อยที่สุด	K_x	1	100	$L / (\text{min} \cdot V)$
ค่าอัตราขยายของตัวควบคุมมากที่สุด	K_n	1	100	$L / (\text{min} \cdot V)$

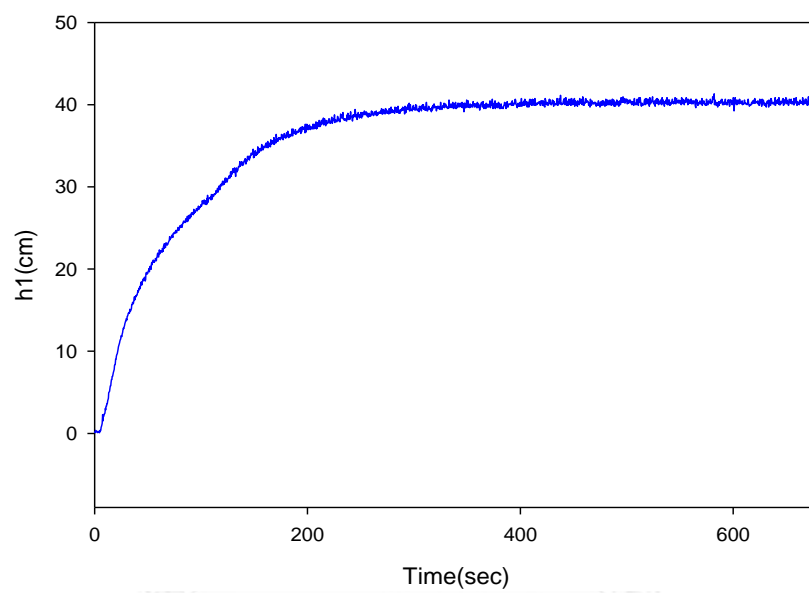
6.2 พฤติกรรมของระบบสี่ถังภายใต้สภาวะเฟสต่ำสุดและไม่มีเฟสต่ำสุด

ในหัวข้อนี้จะแสดงพฤติกรรมของระบบสี่ถังภายใต้สภาวะเฟสต่ำสุดและไม่มีเฟสต่ำสุดกรณีที่ระบบสี่ถังไม่มีการควบคุมระดับน้ำแต่จะมีการใช้ตัวควบคุมลูประองในการควบคุมอัตราการไหลของน้ำจากปั๊มทั้งสองให้คงที่ตามค่าที่ระบุในตารางที่ 6.4 โดยค่าอัตราไหลของน้ำในตารางที่ 6.4 คือค่าอัตราการไหลเฉลี่ยที่ค่าแรงดันไฟฟ้า 1.96 โวลต์ ซึ่งเป็นค่าแรงดันไฟฟ้าที่สภาวะคงตัว โดยข้อมูลในตารางทราบได้จากการทดสอบสมรรถนะของปั๊มทั้งสองตัว

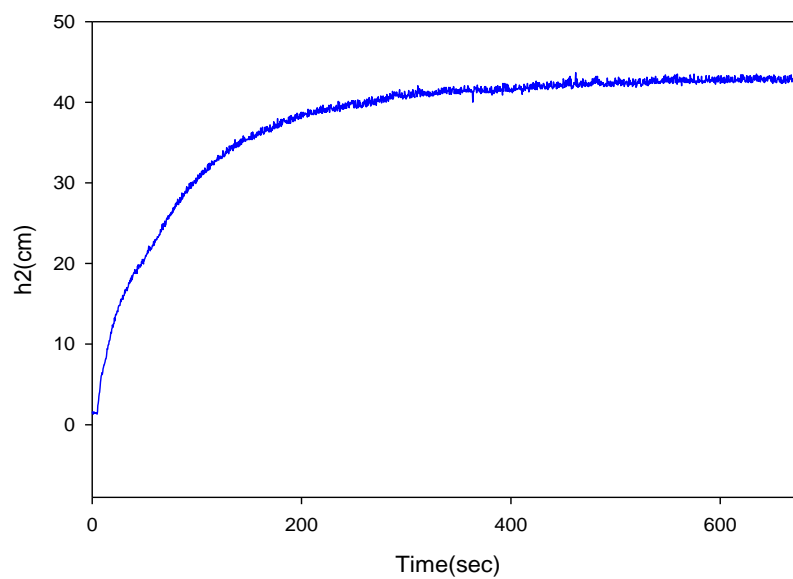
ตารางที่ 6. 4 ค่าอัตราการไหลของปั๊มในระบบสี่ถังที่สภาวะคงตัวภายใต้สภาวะเฟสต่ำสุดและไม่มีเฟสต่ำสุด

	P^-	P^+	หน่วย
อัตราการไหลปั๊มตัวที่1	15.5	11.79	L / min
อัตราการไหลปั๊มตัวที่2	16	10.66	L / min

ในส่วนพฤติกรรมแบบเฟสต่ำสุดของถังใบที่ 1 และ 2 แสดงดังรูปที่ 6.3 และ 6.4



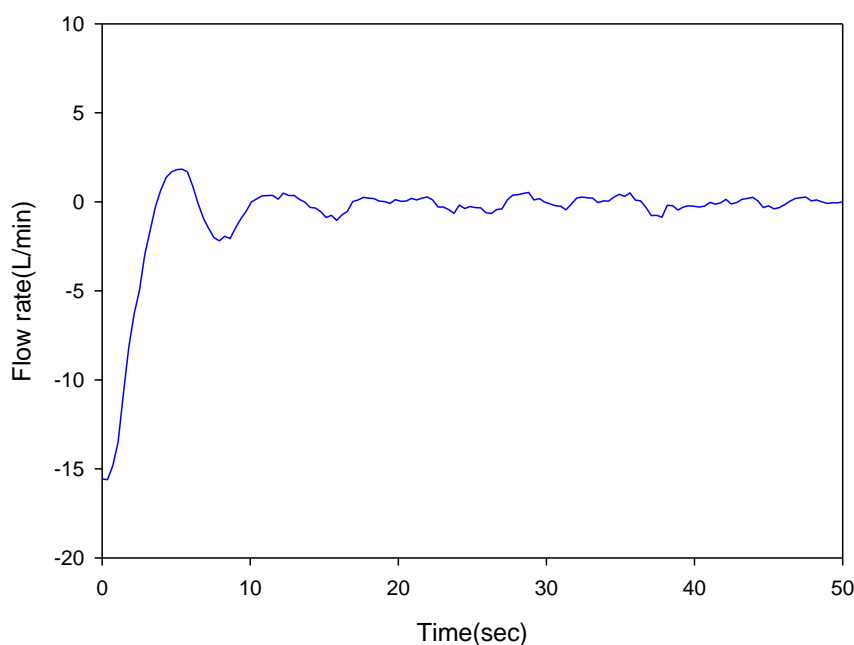
รูปที่ 6. 3 พฤติกรรมเฟสต่ำสุดของถังใบที่1ในระบบสี่ถัง



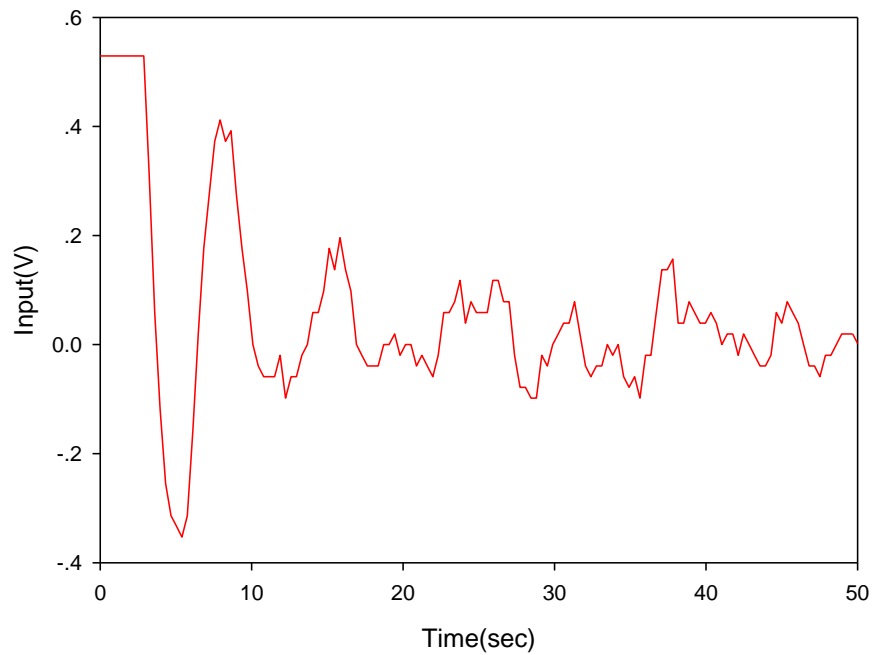
รูปที่ 6. 4 พฤติกรรมเฟสต่ำสุดของถังใบที่2ในระบบสี่ถัง

พฤติกรรมเฟสต่ำสุดของระบบสี่ถังดังรูปที่ 6.3 และ 6.4 แสดงระดับน้ำของถังใบที่ 1 และ 2 ซึ่งเป็นถังด้านล่างในระบบสี่ถัง โดยพฤติกรรมเฟสต่ำสุดดังกล่าวเป็นผลมาจากการปรับสัดส่วนวาล์วตัวที่ 1 และ 2 (γ_1, γ_2) ตามสมการ $1 < \gamma_1 + \gamma_2 < 2$ ซึ่งในการทดลองนี้ได้ปรับสัดส่วนวาล์วทั้งสองตัวไว้ที่ 0.7 ลักษณะปรับวาล์วดังกล่าว ส่งผลให้ทิศทางการไหลของน้ำเข้าสู่ถังล่างมากกว่าถังด้านบน ดังนั้นอัตราการไหลของน้ำจากบ่ยมจึงมีผลต่อถังคู่ล่างมากกว่าคู่บน ระดับน้ำในถังทั้ง 2 จึงเปลี่ยนแปลงทันทีโดยเพิ่มขึ้นเรื่อยๆจนระดับน้ำเข้าสู่สภาวะคงตัว ในการทดลองนี้ระดับน้ำในถังที่ 1 และ 2 เริ่มต้นที่ 0.098 และ 1.4 เซนติเมตร และปล่อยให้ระบบเข้าสู่สภาวะคงตัวที่อัตราการไหลของบ่ยมตัวที่ 1 เท่ากับ 15.5 L/min และบ่ยมตัวที่ 2 เท่ากับ 16 L/min โดยพบว่าระดับน้ำที่สภาวะคงตัวของถังใบที่ 1 เท่ากับ 40.32 เซนติเมตร และถังใบที่ 2 เท่ากับ 42.89 เซนติเมตร

ในส่วนสมรรถนะของตัวควบคุมลูปรองในการควบคุมอัตราการไหลของน้ำแสดงได้ดังรูปที่ 6.5 และ 6.6 จากรูปที่ 6.5ก) และ 6.6ก) สังเกตได้ว่าผลต่างระหว่างค่าอัตราการไหลจริงและค่าอัตราการไหลเป้าหมายมีค่าเข้าสู่ศูนย์ แสดงให้เห็นความสามารถของตัวควบคุมในการควบคุมให้ค่าอัตราการไหลเข้าสู่ค่าเป้าหมายได้ ส่วนรูปที่ 6.6ข) และ 6.6ข) แสดงลักษณะของตัวแปรปรับในรูปตัวแปรเบี่ยงเบน เมื่อมีค่าอัตราการไหลเบี่ยงเบนจากค่าเป้าหมาย ตัวควบคุมจะพยายามคำนวณค่าตัวแปรปรับที่เหมาะสมเพื่อควบคุมให้อัตราการไหลมีค่าใกล้เคียงกับค่าเป้าหมาย

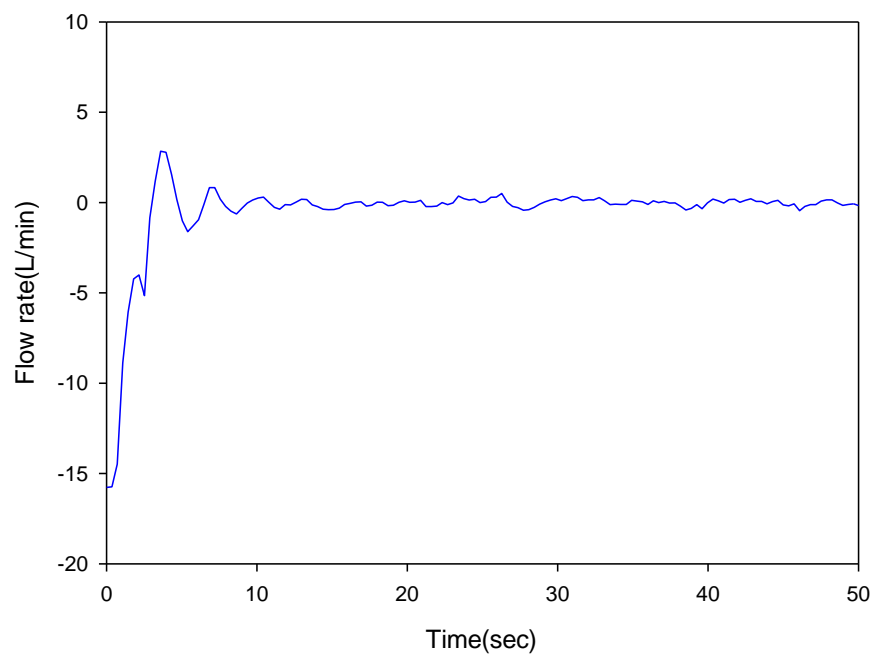


ก) ความสัมพันธ์ระหว่างอัตราการไหลของน้ำจากบ่ยมตัวที่ 1 และเวลาในรูปตัวแปรเบี่ยงเบน

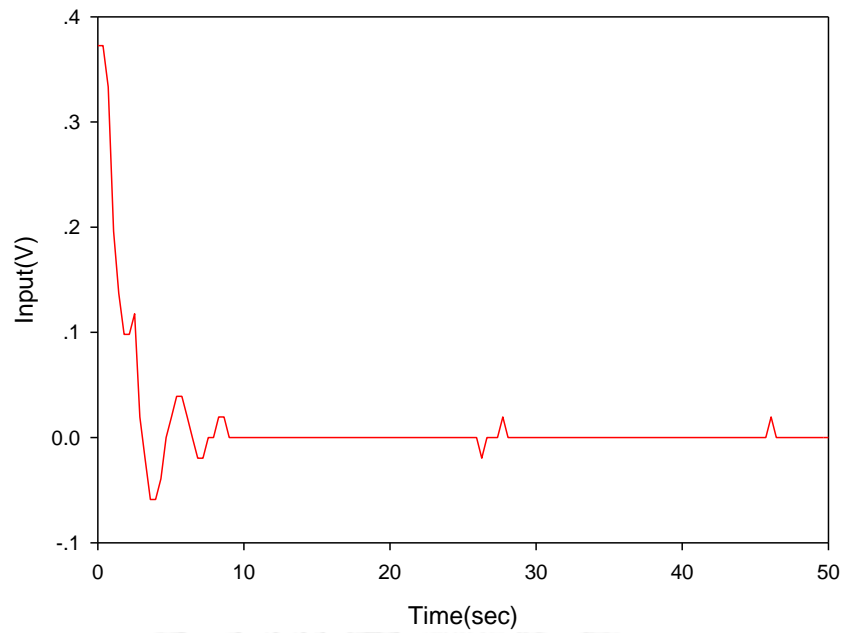


ข) ความสัมพันธ์ระหว่างตัวแปรปรับและเวลาในรูปตัวแปรเบี่ยงเบน

รูปที่ 6.5 ลักษณะการควบคุมอัตราไหลของตัวควบคุมลูปรองที่ 1



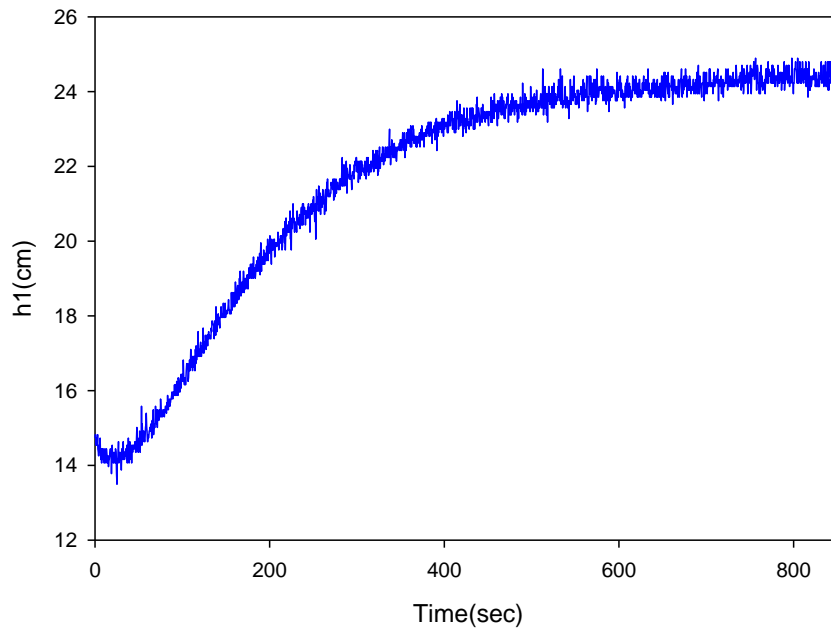
ก) ความสัมพันธ์ระหว่างอัตราไหลของน้ำจากปั๊มตัวที่ 2 และเวลาในรูปตัวแปรเบี่ยงเบน



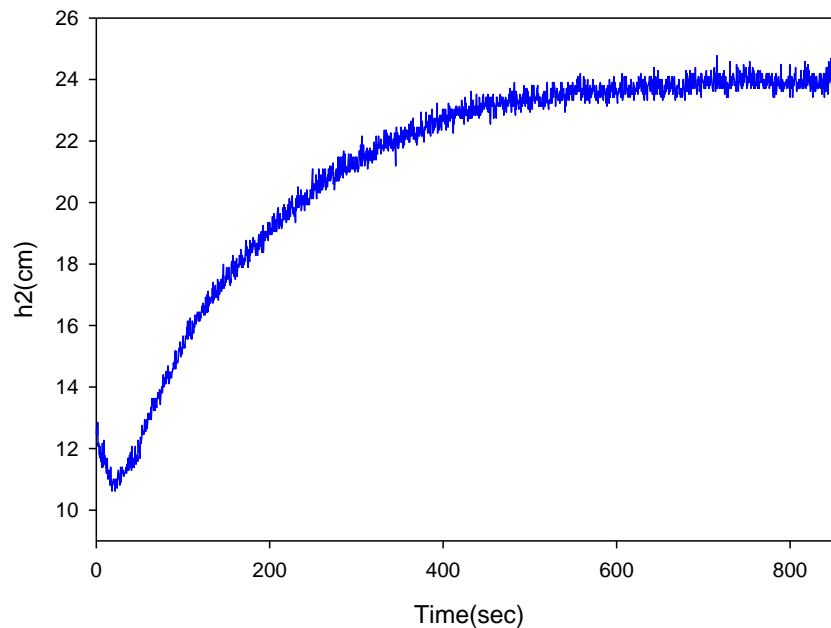
ข) ความสัมพันธ์ระหว่างตัวแปรปรับและเวลาในรูปตัวแปรเชิงเบน

รูปที่ 6. 6 ลักษณะการควบคุมอัตราการใช้ของตัวควบคุมลูปรองที่ 2

ในส่วนพฤติกรรมแบบไม่มีเฟสต่ำสุดของถังใบที่ 1 และ 2 แสดงดังรูปที่ 6.6 และ 6.7



รูปที่ 6. 7 พฤติกรรมไม่มีเฟสต่ำสุดของถังใบที่ 1 ในระบบสี่ถัง



รูปที่ 6. 8 พฤติกรรมไม่มีเฟสต่ำสุดของถังใบที่ 2 ในระบบสี่ถัง

พฤติกรรมไม่มีเฟสต่ำสุดของระบบสี่ถังดังรูปที่ 6.7 และ 6.8 แสดงระดับน้ำของถังใบที่ 1 และ 2 ซึ่งเป็นถังด้านล่างในระบบสี่ถัง โดยพฤติกรรมไม่มีเฟสต่ำสุดดังกล่าวเป็นผลมาจากการปรับสัดส่วนวาล์วตัวที่ 1 และ 2 (γ_1, γ_2) ตามสมการ $0 < \gamma_1 + \gamma_2 < 1$ ซึ่งในการทดลองนี้ได้รับสัดส่วนวาล์วตัวที่ 1 ที่ 0.131 และปรับสัดส่วนวาล์วตัวที่ 2 ที่ 0.165 ลักษณะการปรับวาล์วดังกล่าวส่งผลให้ทิศทางการไหลของน้ำเข้าสู่ถังบนมากกว่าถังด้านล่าง เมื่อสังเกตลักษณะของระดับน้ำในช่วงแรกของถังทั้ง 2 ใบ พบว่า ระดับน้ำจะค่อยๆ ลดลงก่อนที่จะเพิ่มขึ้นเรื่อยๆ เข้าสู่สภาวะคงตัว ลักษณะดังกล่าวเป็นผลมาจากทิศทางการไหลของน้ำจากปั๊มซึ่งไหลเข้าถังด้านบนมากกว่าด้านล่าง ดังนั้นระดับน้ำของถังด้านล่างจึงขึ้นกับระดับน้ำในถังด้านบนเป็นส่วนใหญ่ ในช่วงแรกระดับน้ำในถังด้านบนยังสูงไม่มากนัก ส่งผลให้อัตราการไหลของน้ำจากถังด้านบนลงมาถึงด้านล่างรวมทั้งสัดส่วนในการไหลของน้ำจากปั๊มเข้าถังล่างน้อยเช่นกัน ทำให้ระดับน้ำในถังล่างลดลง แต่เมื่อระยะเวลาผ่านไป ระดับน้ำในถังด้านบนสูงขึ้นเรื่อยๆ เป็นไปตามสมการ $q = m\sqrt{h}$ ส่งผลให้อัตราการไหลของน้ำจากถังด้านบนลงสู่ด้านล่างมากขึ้น ระดับน้ำในถังด้านล่างจึงค่อยเพิ่มสูงขึ้น จนระดับน้ำเข้าสู่สภาวะคงตัว พฤติกรรมที่เกิดขึ้นกับระดับน้ำในถังด้านล่างในช่วงแรกเรียกว่า การตอบสนองตรงกันข้าม (Inverse response) ซึ่งมักจะพบในระบบที่พฤติกรรมแบบไม่มีเฟสต่ำสุด ในการทดลองนี้ระดับน้ำในถังที่ 1 และ 2 เริ่มต้นที่ 14.173 และ 12.46 เซนติเมตร และปล่อยให้ระบบเข้าสู่สภาวะคงตัว ที่อัตราการไหลของปั๊มตัวที่ 1 เท่ากับ 11.79 L/min และปั๊มตัวที่ 2 เท่ากับ 10.66 L/min โดยพบว่าระดับน้ำที่สภาวะคงตัวของถังที่ 1 เท่ากับ 14.728 เซนติเมตร และถังที่สอง เท่ากับ 12.46 เซนติเมตร

6.3 ผลการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกและค่าอัตราขยายของตัวประมาณตัวแปรสถานะเก็บไว้ในฐานข้อมูล

ตามทฤษฎีการควบคุมที่งานวิจัยนำเสนอจะมีขั้นตอนในการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับและค่าอัตราขยายของตัวประมาณตัวแปรสถานะเก็บไว้ในฐานข้อมูล ซึ่งข้อมูลของค่าทั้งสองจะถูกนำไปใช้ในระหว่างการควบคุมระบบสปีด

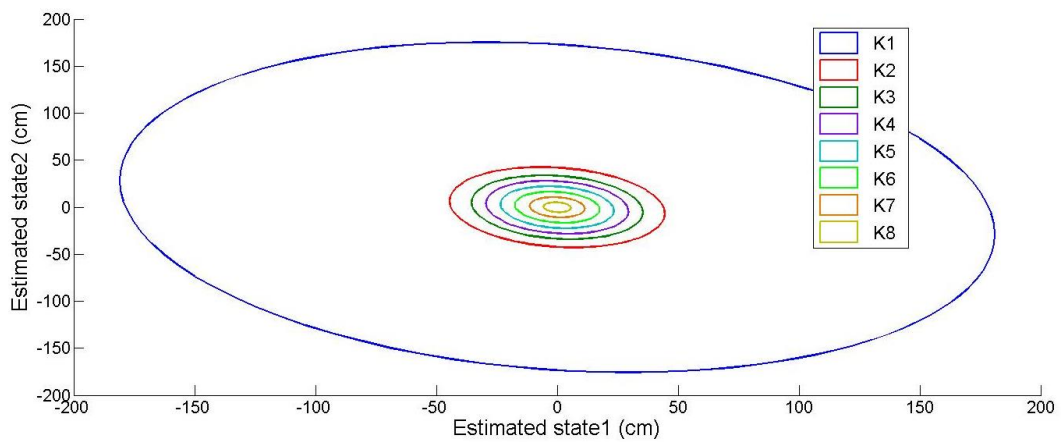
6.3.1 ผลการคำนวณของระบบในสภาวะเฟสต่ำสุด

ในการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกนั้นสามารถคำนวณได้จากชุดสมการออฟติไมซ์ $\max_{u(k), y_Q, Y}$ ภายใต้เงื่อนไขข้อสมการเมตริกซ์เชิงเส้นที่ 8 ถึง 11 โดยในงานวิจัยนี้จะคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกเก็บไว้จำนวน 8 ค่า โดยเลือกค่าตัวแปรสถานะจากการประมาณเพื่อคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกดังตารางที่ 6.5

ตารางที่ 6. 5 ค่าตัวแปรสถานะจากการประมาณเพื่อคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกดังนี้

ค่าอัตราขยายของตัวควบคุมแบบป้อนกลับ	ค่าตัวแปรสถานะจากการประมาณ ($\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4$)
K1	60
K2	15
K3	12
K4	10
K5	8
K6	6
K7	4
K8	2

จากรูปที่ 6.9 แสดงเซตยีนยงทรงรีที่สร้างขึ้นจากเงื่อนไขข้อสมการเมตริกซ์เชิงเส้นที่ 8 จำนวน 8 เซต โดยแต่ละเซตจะมีค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกเฉพาะเซตนั้น ซึ่งสามารถควบคุมตัวแปรสถานะจากการประมาณในเซตดังกล่าวให้สามารถลู่เข้าสู่ค่าเป้าหมายได้และระบบยังคงมีเสถียรภาพ



รูปที่ 6.9 เซตยืนยันทรงรีจากการคำนวณของระบบในสถานะเฟสต่ำสุด

ส่วนค่าอัตราขยายของตัวประมาณตัวแปรสถานะสามารถคำนวณได้จากอัลกอริทึม 4 ตามเงื่อนไขข้อสมการ ที่ 18 เพื่อให้ค่าตัวแปรสถานะจากการประมาณมีความแม่นยำขึ้นเมื่อเวลาผ่านไป และสมการที่ 19 เพื่อประกันเสถียรภาพของระบบกรณีประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะ ผลการคำนวณค่าอัตราขยายของตัวประมาณตัวแปรสถานะได้ค่าดังนี้

$$Lp = \begin{bmatrix} 0.6021 & 0 & 0.1455 & 0 \\ 0 & 0.6126 & 0 & 0.1039 \\ 0.1388 & 0 & 0.7087 & 0 \\ 0 & 0.1001 & 0 & 0.7929 \end{bmatrix}$$

ค่าอัตราขยายของตัวประมาณค่าตัวแปรสถานะจะถูกนำมาใช้หาค่าตัวแปรสถานะจากการประมาณ ณ เวลาถัดไป ตามสมการ $\hat{x}(k+1) = A_0\hat{x}(k) + B_0u(k) + L_p(y(k) - C\hat{x}(k))$

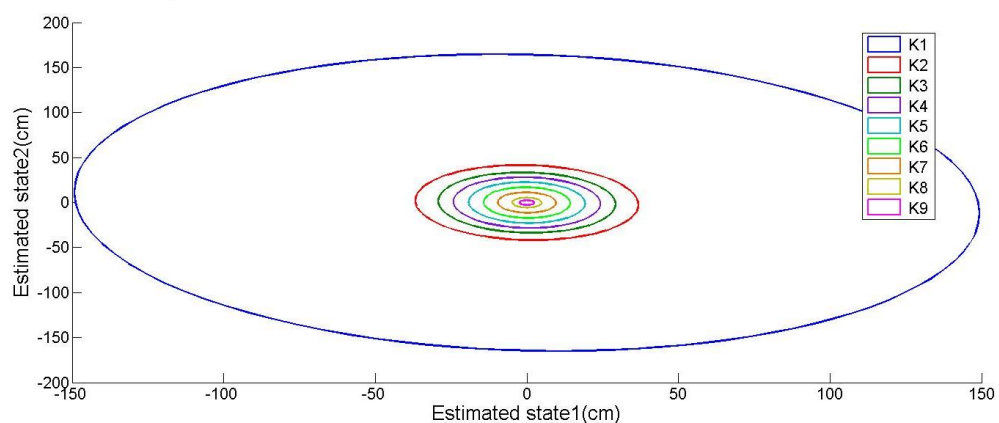
6.3.2 ผลการคำนวณของระบบในสถานะไม่มีเฟสต่ำสุด

ในการคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกนั้นสามารถคำนวณได้เช่นเดียวกับกรณีที่ระบบมีเฟสต่ำสุด โดยงานวิจัยนี้จะคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกเก็บไว้จำนวน 9 ค่า โดยเลือกค่าตัวแปรสถานะจากการประมาณเพื่อคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกดังตารางที่ 6.6

ตารางที่ 6. 6 ค่าตัวแปรสถานะจากการประมาณเพื่อคำนวณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับสัญญาณขาออกดังนี้

ค่าอัตราขยายของตัวควบคุมแบบป้อนกลับ	ค่าตัวแปรสถานะจากการประมาณ ($\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4$)
K1	60
K2	15
K3	12
K4	10
K5	8
K6	6
K7	4
K8	2
K9	1

จากรูปที่ 6.10 แสดงเซตยีนยงทรงรีจำนวน 9 เซต โดยแต่ละเซตจะมีค่าอัตราขยายของตัวควบคุมแบบป้อนกลับเฉพาะเซตนั้นซึ่งสามารถควบคุมตัวแปรสถานะจากการประมาณซึ่งอยู่ในเซตดังกล่าวให้ระบบมีเสถียรภาพและสามารถลู่เข้าสู่ค่าเป้าหมายได้ สังเกตว่าขนาดของเซตยีนยงทรงรีที่ 1 มีขนาดใหญ่มากสืบเนื่องมาจากความต้องการในการออกแบบตัวควบคุมกรณีมีตัวแปรรบกวนซึ่งมั่นใจว่าตัวแปรรบกวนที่จะส่งผลกระทบต่อระบบนั้นมีผลให้ค่าตัวแปรสถานะถดถอยยังคงอยู่ในเซตยีนยงทรงรีที่สร้างขึ้น แม้ในงานวิจัยนี้จะได้ทดสอบตัวรบกวน แต่เป็นการออกแบบเพื่อให้ครอบคลุมและเป็นแนวทางในการนำไปประยุกต์ใช้กับระบบอื่นๆต่อไป



รูปที่ 6. 10 เซตยีนยงทรงรีจากการคำนวณของระบบในสถานะไม่มีเฟสต่ำสุด

ส่วนผลการคำนวณค่าอัตราขยายของตัวประมาณตัวแปรสถานะแสดงค่าดังนี้

$$Lp = \begin{bmatrix} 0.4796 & 0 & 0.1431 & 0 \\ 0 & 0.1403 & 0 & 0.1683 \\ 0.1339 & 0 & 0.7154 & 0 \\ 0 & 0.1541 & 0 & 0.6659 \end{bmatrix}$$

6.4 ผลการควบคุมระบบสี่ถึงด้วยระเบียบวิธีการควบคุมตามทฤษฎีงานวิจัยเสนอ

ในงานวิจัยนี้ได้ทดสอบระเบียบวิธีการควบคุมที่เสนอกับระบบสี่ถึงภายใต้สภาวะเพสต่ำสุด และไม่มีเพสต่ำสุดโดยทดสอบความสามารถในการลู่เข้าสู่ค่าเป้าหมายของค่าตัวแปรสถานะจากการประมาณและค่าตัวแปรสถานะจากเครื่องมือวัด ทดสอบความมีเสถียรภาพของระบบ รวมทั้งทดสอบสมรรถนะในการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีซึ่งเป็นส่วนหนึ่งในระเบียบการควบคุมที่งานวิจัยนำเสนอ

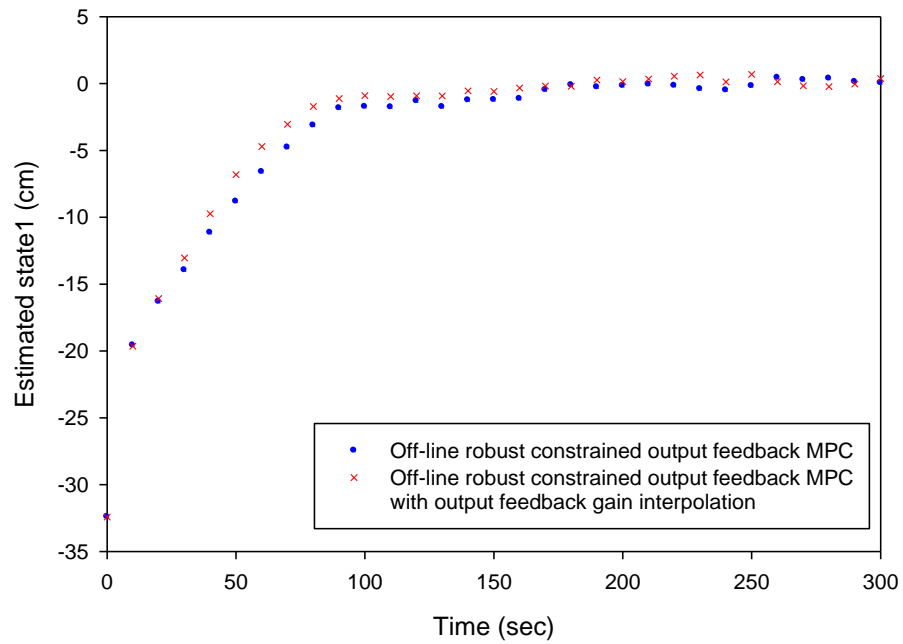
6.4.1 ผลการควบคุมระบบสี่ถึงภายใต้สภาวะเพสต่ำสุด

ผลการควบคุมระดับน้ำในถังที่ 1 และ 2 ภายใต้สภาวะเพสต่ำสุดแสดงดังรูปที่ 6.1 และ 6.12 ในการทดสอบดังกล่าวจะใช้ชุดคำสั่ง 2 ชุดด้วยกัน ชุดคำสั่งแรกใช้การควบคุมตามวิธีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ ภายใต้ขอบเขตความไม่แน่นอนของโพลีโทพ โดยประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะ ส่วนชุดคำสั่งที่สอง จะใช้วิธีการควบคุมตามชุดคำสั่งแบบแรกแต่มีการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีร่วมด้วย จากกราฟทั้งสองแสดงลักษณะของค่าตัวแปรสถานะจากการประมาณ(Estimated state) ต่อช่วงเวลาในการควบคุม โดยค่าตัวแปรสถานะจากการประมาณทั้ง 2 กราฟนั้น คือ ระดับน้ำของถังที่ 1 และ 2 ซึ่งคำนวณจากตัวประมาณค่าตัวแปรสถานะแทนการใช้เครื่องมือวัด

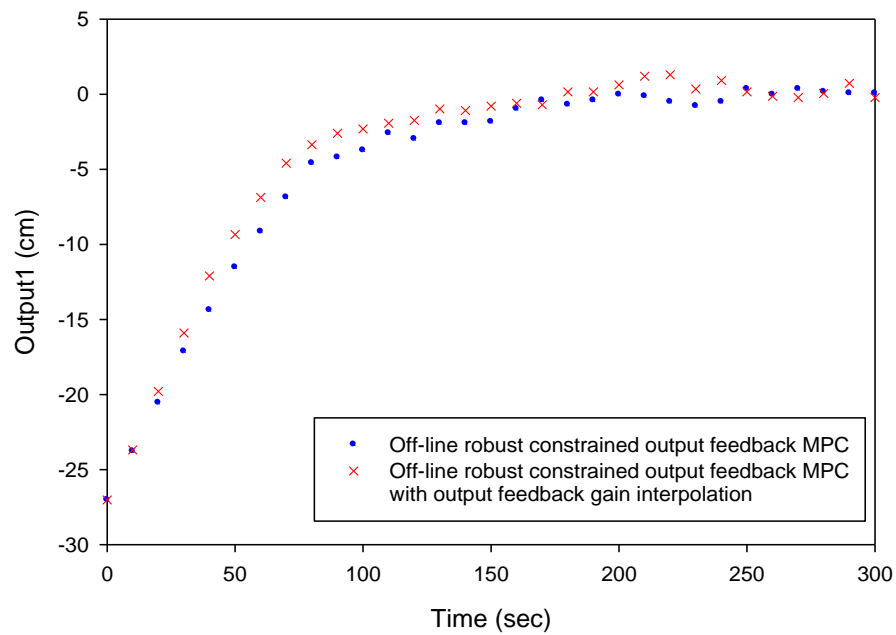
จากรูปที่ 6.11ก) และ 6.12ก) แสดงความสัมพันธ์ระหว่างค่าตัวแปรสถานะจากการประมาณในรูปตัวแปรเบี่ยงเบนกับเวลา จากรูปทั้งสองเมื่อพิจารณาถึงสมรรถนะในการควบคุมจะพบว่าชุดคำสั่งที่สอง มีสมรรถนะในการควบคุมดีกว่าชุดคำสั่งแรก ตัวแปรสถานะจากการประมาณมีการปรับเข้าสู่ค่าเป้าหมายได้รวดเร็วกว่าชุดคำสั่งแรก ผลดังกล่าวยังสามารถแสดงในรูปของค่า IAE ดังตารางที่ 6.7 ซึ่งพบว่าชุดคำสั่งที่สองมีค่า IAE ต่ำกว่า ผลรวมค่าสัมบูรณ์ของผลต่างระหว่างค่าตัวแปรควบคุมกับค่าเป้าหมายในช่วงเวลาของการควบคุมระบบสี่ถึงมีค่าต่ำกว่า จากผลของสมรรถนะที่ได้กล่าวมาข้างต้นทั้งหมดสืบเนื่องมาจากการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี โดยการหาค่าน้อยที่สุดของ $\rho(k)$ ดังสมการที่

13-16 เพื่อหาค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออก ตามสมการ $K(\rho(k)) = \rho(k)K_i + (1 - \rho(k))K_{i+1}$ ค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกที่คำนวณได้จะมีค่าเพิ่มขึ้นจากชุดคำสั่งแรก ส่งผลให้ค่าตัวแปรปรับที่คำนวณได้มีค่าเพิ่มขึ้นเช่นกันตามกฎการควบคุมแบบป้อนกลับค่าสัญญาณขาออก $u(k) = K(k)\hat{x}(k)$ โดยค่าตัวแปรปรับของปั๊มทั้งสองตัวตลอดช่วงเวลาในการควบคุมระบบ แสดงไว้ดังรูปที่ 6.13 และ 6.14 ผลของสมรรถนะดังกล่าวยังแสดงต่อค่าตัวแปรสถานะจากเครื่องมือวัดของทั้ง 2 ถึง ดังรูปที่ 6.11ข) และ 6.12ข) โดยพบว่าค่าตัวแปรสถานะจากเครื่องมือวัดของทั้ง 2 ถึง สามารถลู่เข้าสู่ค่าเป้าหมายได้และมีความเร็วในการลู่เข้ามากขึ้นจากการควบคุมด้วยชุดคำสั่งที่สองและเมื่อพิจารณาในด้านการทำงานของตัวประมาณค่าตัวแปรสถานะ จากการพิจารณาระหว่างค่าตัวประมาณค่าตัวแปรสถานะจากการประมาณ (\hat{x}) จากกราฟในรูปที่ 6.11ก) และ 6.12ก) และค่าตัวแปรสถานะจากเครื่องมือวัด (y) จากกราฟในรูปที่ 6.11ข) และ 6.12ข) พบว่าจะมีค่าใกล้เคียงกันเมื่อระบบสเถียรเริ่มเข้าสู่สถานะคงตัวเป็นไปตามสมการตัวประมาณค่าตัวแปรสถานะ $\hat{x}(k+1) = A_0\hat{x}(k) + B_0u(k) + L_p(y(k) - C\hat{x}(k))$ เนื่องจากค่าเมตริกซ์ A_0 และ B_0 เป็นค่าเมตริกซ์ของแบบจำลองกรณีที่แบบจำลองของระบบอยู่ในสถานะคงตัว จากสมการเมื่อค่าตัวประมาณค่าตัวแปรสถานะจากการประมาณมีค่าใกล้เคียงกับค่าตัวแปรสถานะจากเครื่องมือวัดจะทำให้พจน์ $y(k) - C\hat{x}(k)$ ในสมการมีค่าเข้าใกล้ศูนย์ ส่งผลให้รูปแบบสมการตัวประมาณค่าตัวแปรสถานะใกล้เคียงกับรูปสมการ $\hat{x}(k+1) = A_0\hat{x}(k) + B_0u(k)$ ซึ่งเมื่อเปรียบเทียบกับสมการแบบจำลองของระบบในกรณีที่ระบบอยู่ภายใต้สถานะคงตัว $x(k+1) = A_0x(k) + B_0u(k)$ จะพบว่าค่าตัวแปรสถานะจากการประมาณ (\hat{x}) จะมีค่าเท่ากับค่าตัวแปรสถานะจริงของระบบ (x) ดังนั้นตัวประมาณค่าตัวแปรสถานะจึงมักถูกนำไปใช้กับระบบที่มีตัวแปรสถานะบางตัวซึ่งไม่สามารถวัดได้โดยจะใช้ตัวแปรค่าสัญญาณขาออกในการทำนายค่าตัวแปรสถานะนั้นผ่านตัวประมาณค่าตัวแปรสถานะหรือมักถูกนำไปใช้ในบางระบบที่เครื่องมือวัดมีความคลาดเคลื่อนสูงเพื่อลดค่าความคลาดเคลื่อนจากการตรวจวัดค่าของเครื่องมือวัด

ในส่วนการพิจารณาเสถียรภาพระบบของการควบคุมด้วยตัวควบคุมที่ใช้ระเบียบวิธีการควบคุมตามที่งานวิจัยเสนอนั้นพบว่า ระบบยังคงมีเสถียรภาพในการควบคุม ค่าตัวแปรสถานะจากการประมาณและค่าตัวแปรสถานะจากเครื่องมือวัดสามารถเข้าสู่ค่าเป้าหมายได้ ผลดังกล่าวสืบเนื่องจากเงื่อนไขในการประกันเสถียรภาพของระบบตามสมการที่ 9 สมการที่ 14 และสมการที่ 18 โดยสมการที่ 9 เป็นการประกันเสถียรภาพในการใช้การควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ สมการที่ 14 เป็นการประกันเสถียรภาพในกรณีมีการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะและสมการที่ 18 เป็นการประกันเสถียรภาพในกรณีมีการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

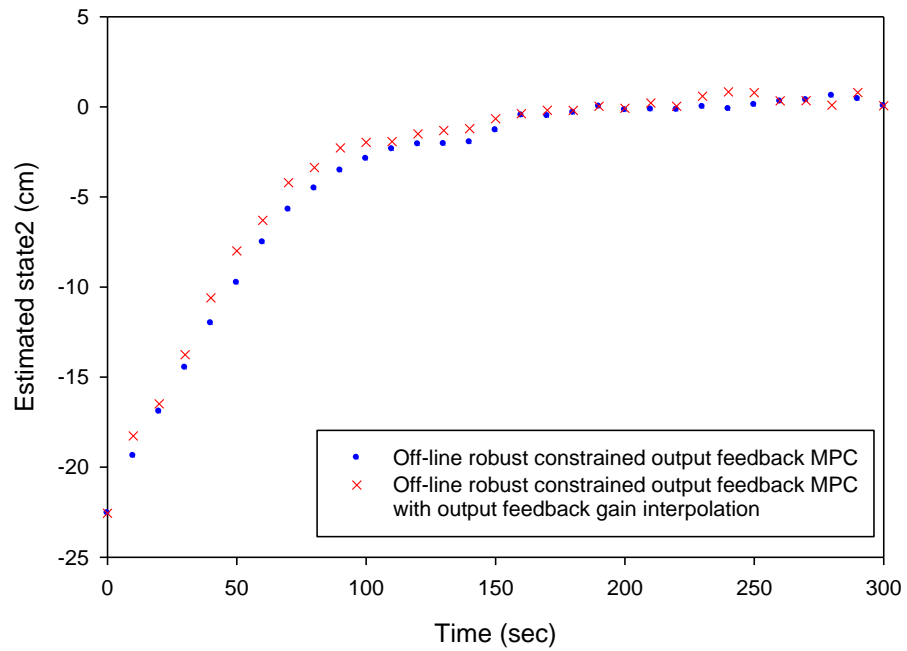


ก) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะที่ 1 จากการประมาณ
ในรูปตัวแปรเบี่ยงเบนกับเวลา

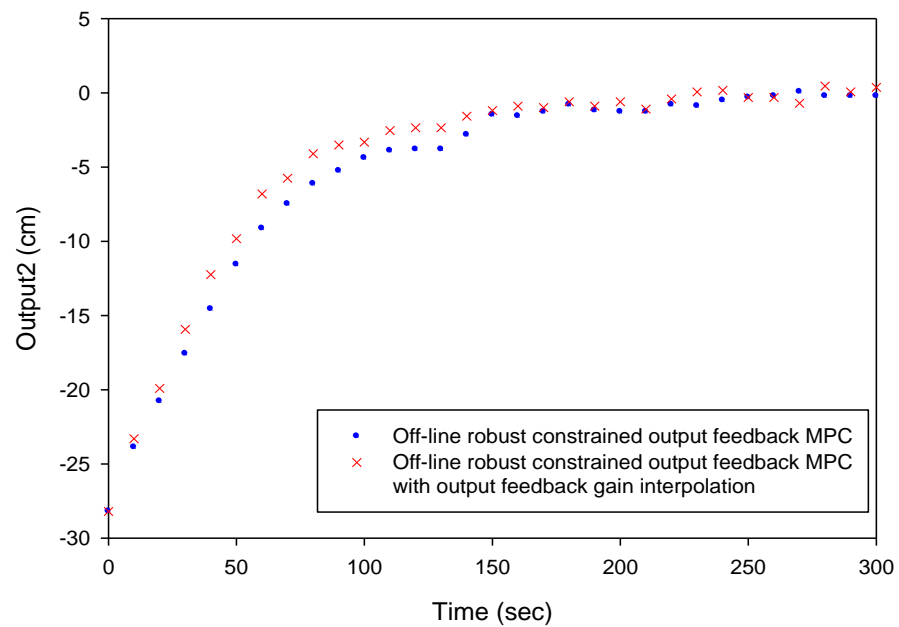


ข) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะถึงที่ 1 จากเครื่องมือวัดในรูปตัวแปรเบี่ยงเบนกับเวลา

รูปที่ 6. 11 ผลการควบคุมระดับน้ำในถังใบที่ 1 ในรูปตัวแปรเบี่ยงเบน

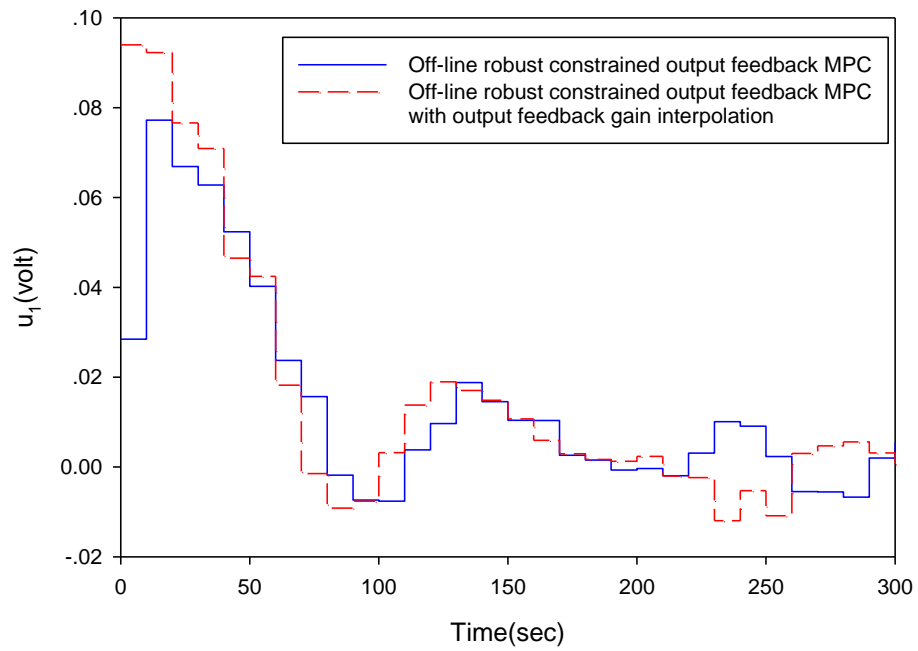


ก) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะที่ 2 จากการประมาณ
ในรูปตัวแปรเบี่ยงเบนกับเวลา

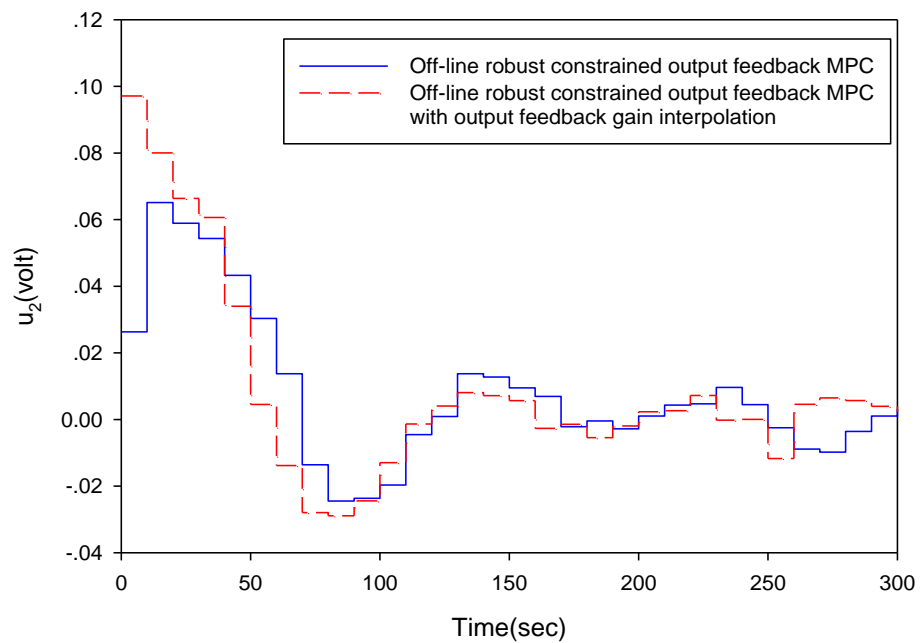


ข) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะในถังที่ 2 จากเครื่องมือวัดในรูปตัวแปรเบี่ยงเบนกับเวลา

รูปที่ 6. 12 ผลการควบคุมระดับน้ำในถังใบที่ 2 ในรูปตัวแปรเบี่ยงเบน

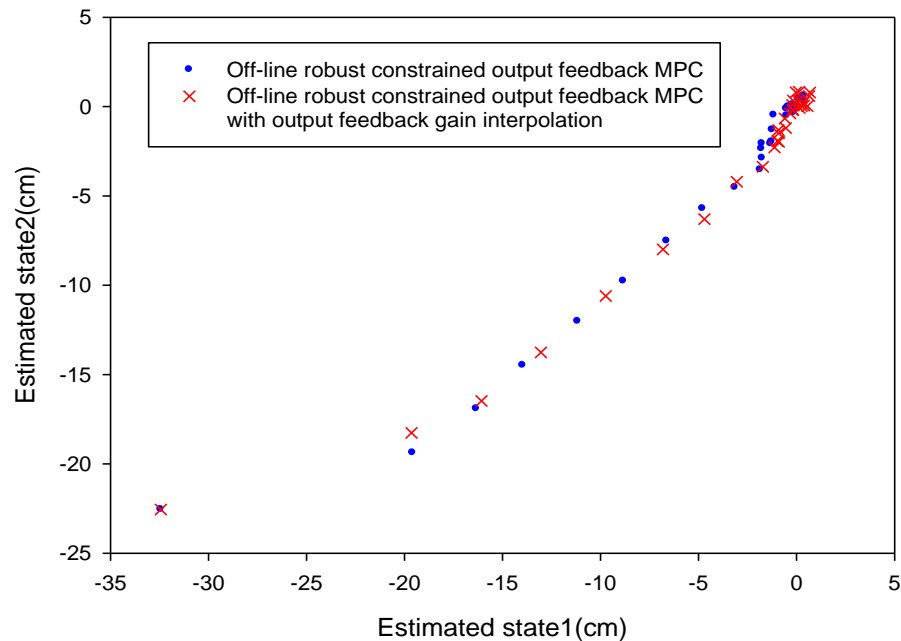


รูปที่ 6. 13 ค่าตัวแปรปรับที่ป้อนให้แก่ปั๊มตัวที่ 1 ในรูปตัวแปรเบี่ยงเบน



รูปที่ 6. 14 ค่าตัวแปรปรับที่ป้อนให้แก่ปั๊มตัวที่ 2 ในรูปตัวแปรเบี่ยงเบน

ในรูปที่ 6.15 จะแสดงภาพรวมเส้นทางการเปลี่ยนแปลงค่าตัวแปรสถานะจากการประมาณ ซึ่งแสดงในรูปของตัวแปรเบี่ยงเบน จากรูปดังกล่าวแสดงให้เห็นว่า ณ เวลาเดียวกัน ชุดคำสั่งที่สองซึ่งมีการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีสามารถควบคุมให้ค่าตัวแปรสถานะจากการประมาณทั้งสองค่าใกล้เคียงกับค่าเป้าหมายมากกว่าโดยที่ระบบยังคงมีเสถียรภาพในการควบคุม



รูปที่ 6. 15 เส้นทางการเปลี่ยนแปลงค่าตัวแปรสถานะจากการประมาณ
ในรูปของตัวแปรเบี่ยงเบน

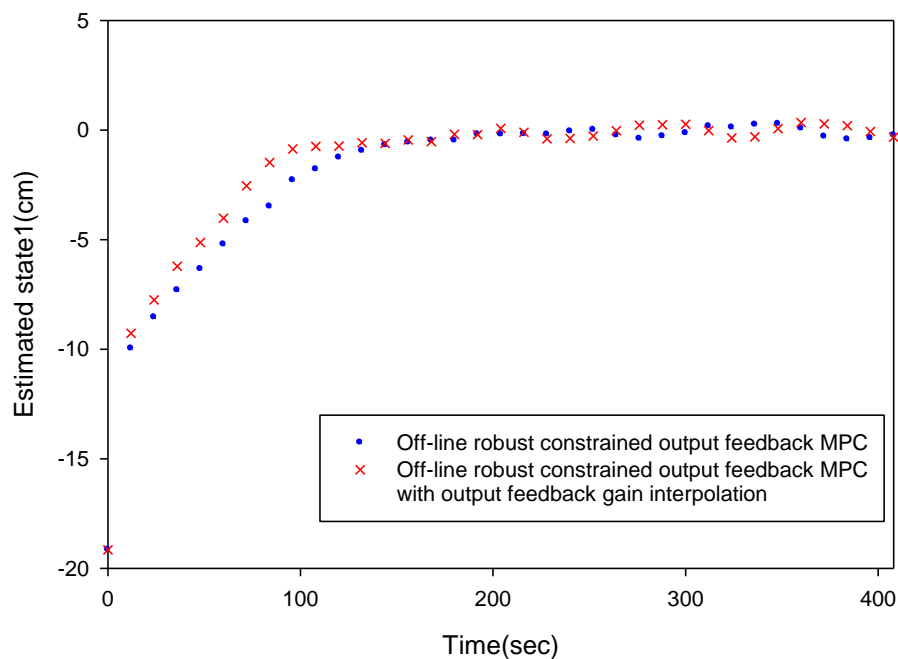
6.4.2 ผลการควบคุมระบบสี่ถึงภายใต้สถานะไม่มีเฟสต่ำสุด

ผลการควบคุมระดับน้ำในถังที่ 1 และ 2 ภายใต้สถานะไม่มีเฟสต่ำสุดแสดงดังรูปที่ 6.16 และ 6.17 ในการทดสอบดังกล่าวจะใช้ชุดคำสั่ง 2 ชุด เหมือนกับการควบคุมระบบสี่ถึงภายใต้สถานะเฟสต่ำสุดเพื่อทดสอบสมรรถนะในการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรง

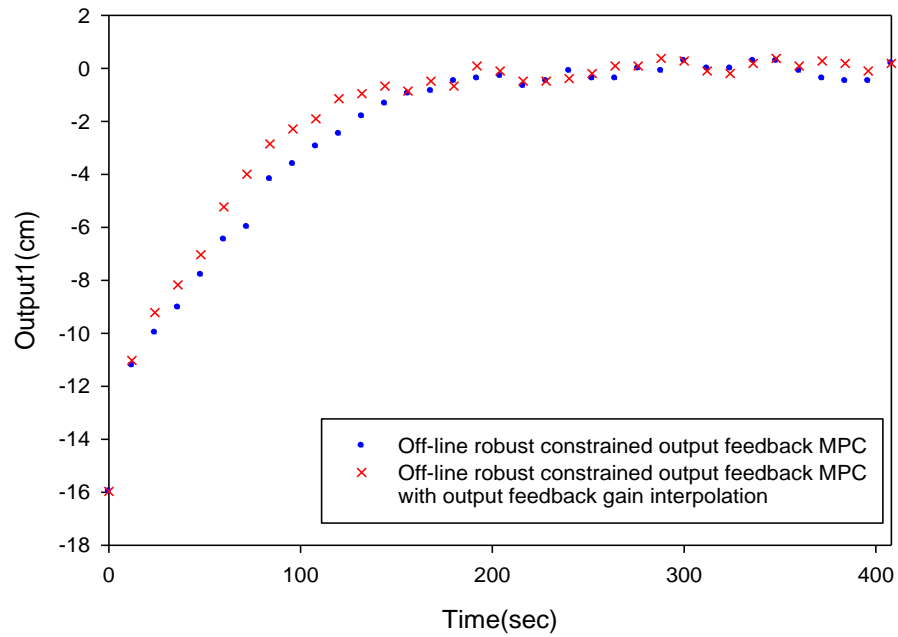
จากรูปที่ 6.16ก) และ 6.17ก) เมื่อพิจารณาถึงสมรรถนะในการควบคุมจะพบว่าชุดคำสั่งที่สอง มีสมรรถนะในการควบคุมดีกว่าชุดคำสั่งแรก ตัวแปรสถานะจากการประมาณมีการปรับเข้าสู่ค่าเป้าหมายได้เร็วกว่า ผลดังกล่าวยังสามารถแสดงในรูปของค่า IAE ดังตารางที่ 6.7 ซึ่งพบว่าชุดคำสั่งที่สองมีค่า IAE ต่ำกว่า ผลรวมค่าสัมบูรณ์ของผลต่างระหว่างค่าตัวแปรควบคุมกับค่าเป้าหมายในช่วงเวลาของการควบคุมระบบสี่ถึงมีค่าต่ำกว่า นอกจากนี้ผลของสมรรถนะดังกล่าวยังแสดงต่อค่าตัวแปรสถานะจากเครื่องมือวัดของทั้ง 2 ถัง ดังรูปที่ 6.11ข) และ 6.12ข) โดยพบว่าค่าตัวแปรสถานะ

จากเครื่องมือวัดของทั้ง 2 ถึง สามารถลู่เข้าสู่ค่าเป้าหมายได้และมีความเร็วในการลู่เข้ามากขึ้นจากการควบคุมด้วยชุดคำสั่งที่สอง ลักษณะผลที่กล่าวมาข้างต้นดังแสดงเช่นเดียวกับการควบคุมระบบสี่ถึงภายใต้สภาวะเฟสต่ำสุด ซึ่งเมื่อพิจารณาจากระบบสี่ถึงภายใต้สภาวะไม่มีเฟสต่ำสุดนั้นจะมีความยุ่งยากในการควบคุมมากกว่าเนื่องจากมักเกิดพฤติกรรมการตอบสนองแบบตรงข้ามซึ่งเป็นพฤติกรรมของการเปลี่ยนแปลงของตัวแปรควบคุมในสองทิศทาง ซึ่งเมื่อพิจารณาจากกราฟในรูปที่ 6.12ก) และ 6.12ข) พบว่าลักษณะการเปลี่ยนแปลงของตัวแปรทั้งตัวแปรสถานะจากการประมาณและตัวแปรสถานะจากเครื่องมือวัด มีทิศทางการเปลี่ยนแปลงไปในทิศทางเดียวกัน ไม่แสดงพฤติกรรมการตอบสนองแบบตรงข้าม ผลดังกล่าวแสดงถึงความสามารถของระเบียบวิธีควบคุมที่งานวิจัยเสนอนั้นสามารถจัดการกับพฤติกรรมการตอบสนองแบบตรงกันข้ามได้ ทำให้การควบคุมระดับน้ำเป็นไปในทิศทางเดียวซึ่งเป็นทิศทางที่จะทำให้ตัวแปรควบคุมนั้นสามารถลู่เข้าสู่ค่าเป้าหมาย

ในส่วนการพิจารณาเสถียรภาพระบบของการควบคุมด้วยตัวควบคุมที่ใช้ระเบียบวิธีการควบคุมตามที่งานวิจัยเสนอนั้นพบว่า ระบบยังคงมีเสถียรภาพในการควบคุม ค่าตัวแปรสถานะจากการประมาณและค่าตัวแปรสถานะจากเครื่องมือวัดสามารถเข้าสู่ค่าเป้าหมายได้ ผลเป็นไปตามสมการเงื่อนไขในการประกันเสถียรภาพซึ่งถูกสร้างขึ้นภายในระเบียบวิธีการควบคุมที่งานวิจัยเสนอ

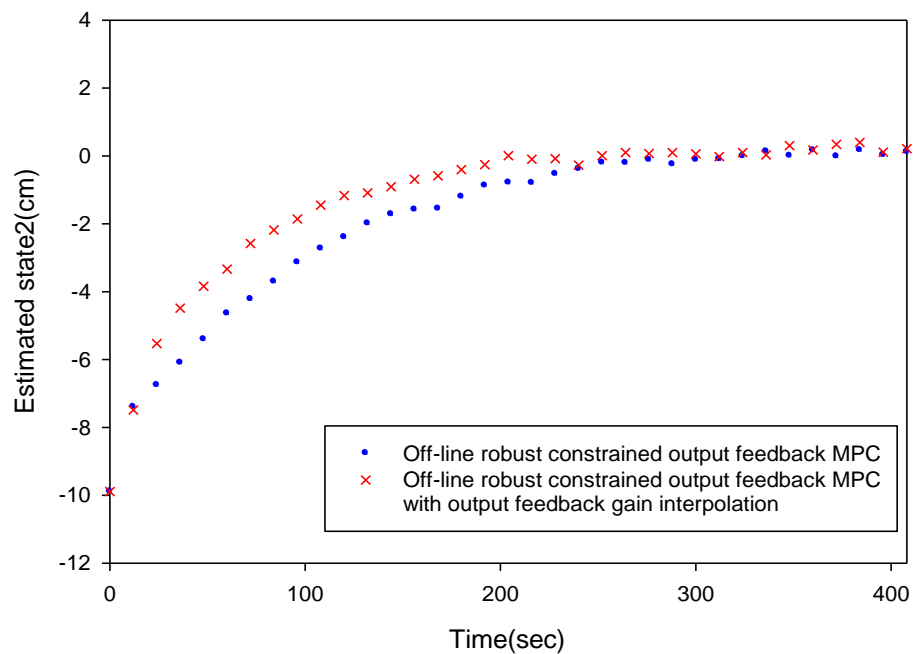


- ก) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะที่ 1 จากการประมาณ
ในรูปตัวแปรเบี่ยงเบนกับเวลา

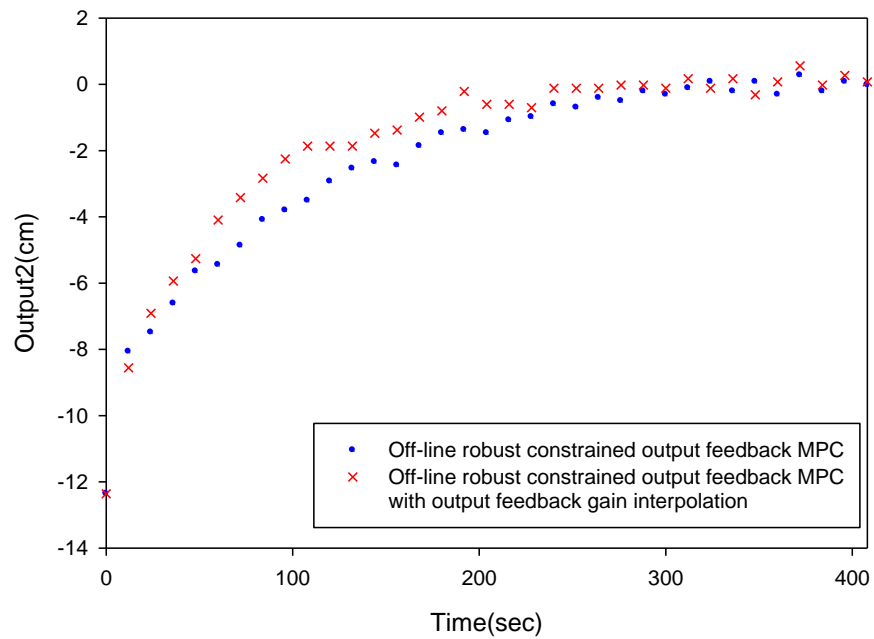


ข) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะในถังที่1จากเครื่องมือวัดในรูปตัวแปรเบี่ยงเบนกับเวลา

รูปที่ 6. 16 ผลการควบคุมระดับน้ำในถังใบที่ 1

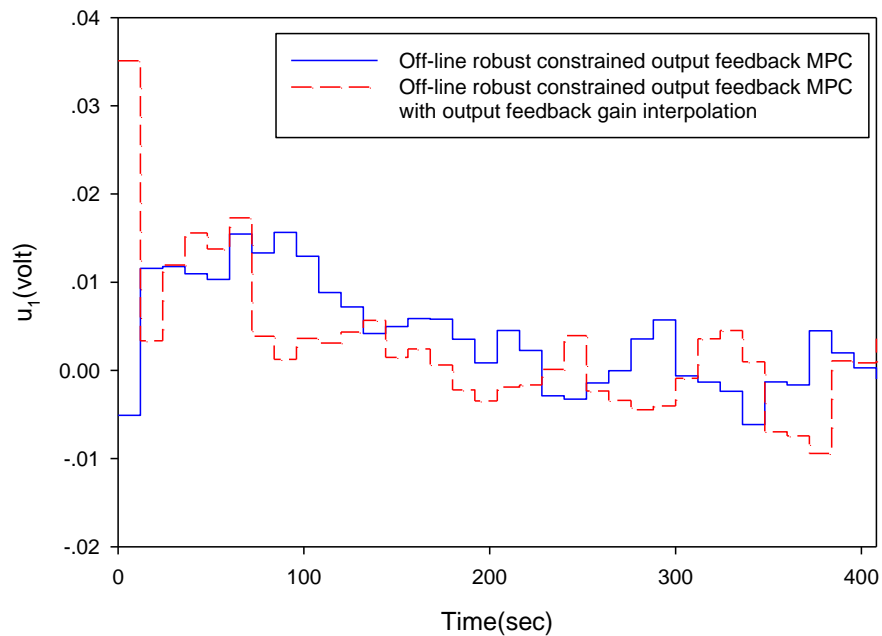


ก) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะที่2จากการประมาณ
ในรูปตัวแปรเบี่ยงเบนกับเวลา

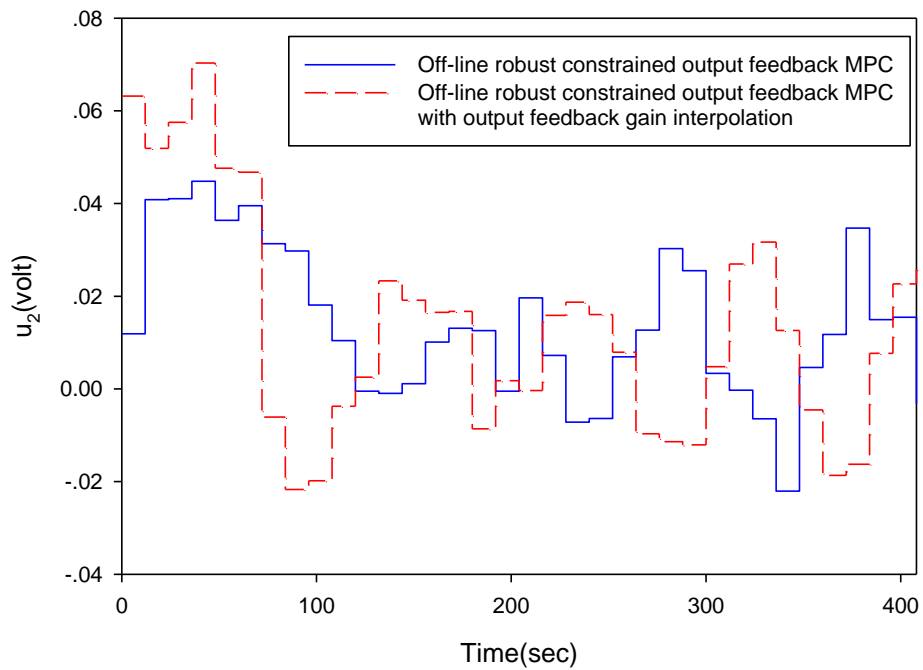


ข) ความสัมพันธ์ระหว่างค่าตัวแปรสถานะในถังที่ 2 จากเครื่องมือวัดในรูปตัวแปรเบี่ยงเบนกับเวลา

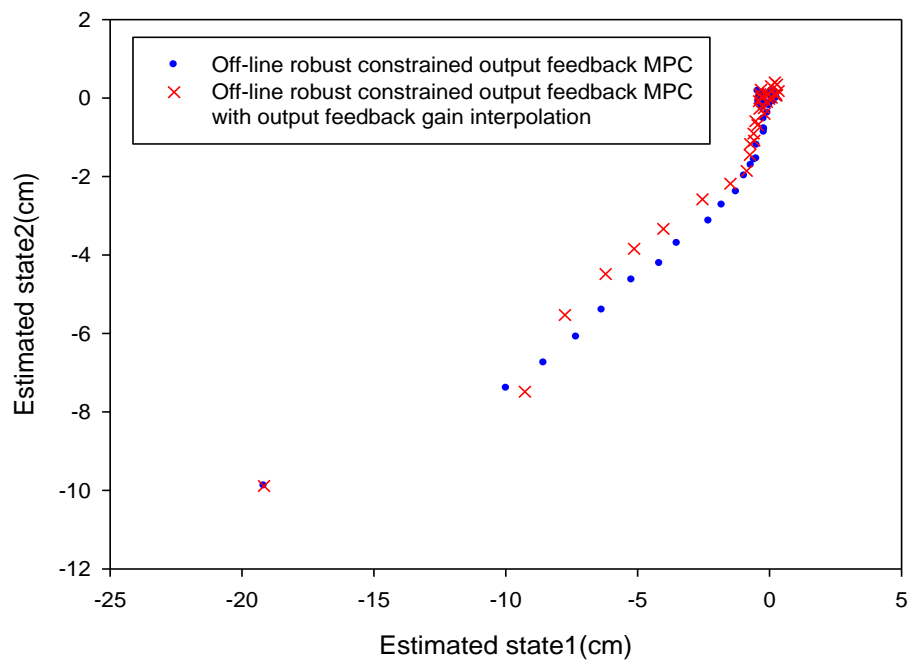
รูปที่ 6.17 ผลการควบคุมระดับน้ำในถังใบที่ 2



รูปที่ 6.18 ค่าตัวแปรปรับที่ป้อนให้แก่ปั๊มตัวที่ 1 ในรูปตัวแปรเบี่ยงเบน



รูปที่ 6. 19 ค่าตัวแปรปรับที่ป้อนให้แก่บีมตัวที่ 1 ในรูปตัวแปรเชิงเบน



รูปที่ 6. 20 เส้นทางการเปลี่ยนแปลงค่าตัวแปรสถานะจากการประมาณ
ในรูปของตัวแปรเชิงเบน

ในรูปที่ 6.20 จะแสดงภาพรวมเส้นทางการเปลี่ยนแปลงค่าตัวแปรสถานะจากการประมาณ ซึ่งแสดงในรูปของตัวแปรเบี่ยงเบน จากรูปดังกล่าวแสดงให้เห็นว่า ณ เวลาเดียวกัน ชุดคำสั่งที่สองซึ่งมีการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่าง เซตยีนยงทรงรีสามารถควบคุมให้ค่าตัวแปรสถานะจากการประมาณทั้งสองค่าใกล้เคียงกับค่า เป้าหมายมากกว่าโดยที่ระบบยังคงมีเสถียรภาพในการควบคุม

ตารางที่ 6. 7 ข้อมูลค่า IAE

	ชุดคำสั่งที่ 1	ชุดคำสั่งที่ 2 (งานวิจัยเสนอ)
ระบบสี่ถังภายใต้สภาวะเฟสต่ำสุด		
ข้อมูลค่าตัวแปรสถานะที่1จากการประมาณ	1332.68	1187.48
ข้อมูลค่าตัวแปรสถานะที่2จากการประมาณ	1352.91	1233.88
ระบบสี่ถังภายใต้สภาวะไม่มีเฟสต่ำสุด		
ข้อมูลค่าตัวแปรสถานะที่1จากการประมาณ	923.66	773.64
ข้อมูลค่าตัวแปรสถานะที่2จากการประมาณ	883.08	602.68

บทที่ 7 สรุปผลการวิจัย

ในงานวิจัยนี้ได้เสนอระเบียบวิธีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ภายใต้ขอบเขตความไม่แน่นอนของโพลีโทพ โดยการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะ และประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีเพื่อใช้ควบคุมระบบสี่ถึงซึ่งเป็นระบบที่สามารถใช้เป็นตัวแทนของกระบวนการผลิตในอุตสาหกรรมได้เนื่องจากเป็นระบบซึ่งมีลักษณะของสมการแบบจำลองไม่เป็นเชิงเส้น มีจำนวนตัวแปรควบคุมและตัวแปรปรับมากกว่าหนึ่งตัว รวมทั้งเป็นระบบที่สามารถศึกษาเงื่อนไขต่างๆภายในระบบได้เช่น เงื่อนไขของตัวแปรปรับ เงื่อนไขของตัวแปรสัญญาณขาออก เป็นต้น นอกจากนี้ระบบสี่ถึงยังสามารถดำเนินการภายใต้สถานะสองแบบด้วยกัน คือ สถานะเฟสต่ำสุดและไม่มีเฟสต่ำสุดโดยอาศัยการปรับวาล์วตัวที่1และ2 ในระบบ ในการทดลองการควบคุมกับระบบสี่ถึง จะควบคุมระดับของถังที่ 1 และ 2 โดยศึกษาการควบคุมระบบสี่ถึงทั้งสองสถานะ เพื่อศึกษาลักษณะการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ เมื่อมีการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี รวมทั้งทดสอบสมรรถนะของการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี ซึ่งผลการควบคุมระบบสี่ถึงสามารถสรุปได้ตามกรณีดังนี้

7.1 ลักษณะการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ เมื่อมีการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

จากผลการทดลองพบว่าการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์ซึ่งประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะและการประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี สามารถใช้ควบคุมระบบสี่ถึงภายใต้สถานะเฟสต่ำสุดและไม่มีเฟสต่ำสุดได้โดยตัวแปรควบคุมสามารถเข้าสู่ค่าเป้าหมายได้โดยที่ระบบยังคงมีความเสถียรภาพ

7.2 ระเบียบวิธีการควบคุมที่งานวิจัยเสนอต่อพฤติกรรมการตอบสนองแบบตรงข้าม

จากผลการทดลองพบว่าระเบียบวิธีการควบคุมที่งานวิจัยเสนอสามารถจัดการกับพฤติกรรมการตอบสนองแบบตรงข้ามโดยมักเกิดขึ้นกับระบบซึ่งดำเนินการภายใต้สถานะไม่มีเฟสต่ำสุด การจัดการดังกล่าวทำให้ทิศทางการเปลี่ยนแปลงของค่าตัวแปรควบคุมมีทิศทางเดียวซึ่งเป็นทิศทางในการเข้าสู่ค่าเป้าหมาย ผลดังกล่าวทำให้การควบคุมระบบมีประสิทธิภาพมากกว่ากรณีที่ตัวควบคุมนั้นไม่มีความสามารถในการจัดการกับพฤติกรรมการตอบสนองแบบตรงข้าม

7.3 สมรรถนะในการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรี

จากผลการทดลองพบว่าการประยุกต์ใช้การประมาณค่าอัตราขยายของตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกระหว่างเซตยีนยงทรงรีร่วมกับวิธีการควบคุมเชิงทำนายแบบจำลองคงทนแบบออฟไลน์เมื่อมีการประยุกต์ใช้ตัวประมาณค่าตัวแปรสถานะ ช่วยเพิ่มสมรรถนะในการควบคุมโดยตัวแปรควบคุมสามารถเข้าสู่ค่าเป้าหมายได้เร็วขึ้นโดยที่ระบบยังคงมีความเสถียรภาพ จากผลดังกล่าวแสดงให้เห็นว่าระเบียบวิธีการควบคุมที่งานวิจัยเสนอมีสมรรถนะที่ดีกว่าการควบคุมเชิงทำนายแบบจำลองคงทนที่มีเงื่อนไขบังคับคงทนแบบออฟไลน์ปกติ



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

รายการอ้างอิง

- [1] Kothare, M.V., Balakrishnan, V., and Morari, M. Robust constrained model predictive control using linear matrix inequalities. *Automatica* 39(2003) : 837-846.
- [2] Wan, Z., Kothare, M.V. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica* 39(2003) : 837-846.
- [3] Bumroongsri, P., and Kheawhom, Interpolation-based Off-line Robust MPC for Uncertain Polytopic Discrete-time Systems. *Engineering Journal* 18(2014) : 87-104.
- [4] Zhaoyang Wan, Mayuresh V.Kothare, Robust output feedback model predictive control using off-line linear matrix inequalities. *Journal of Process Control* 12 (2002): 763–774.
- [5] Sukapradit Araya, LINEAR PARAMETER VARYING MODEL PREDICTIVE CONTROL OF QUADRUPLE-TANK SYSTEM, Master's Thesis, Department of electrical engineering, Faculty of engineering, Chulalongkorn University, 2009.
- [6] Poontong Pramesak, Robust MPC based on nominal performance cost with application in quadruple tank system, Master's Thesis, Department of electrical engineering, Faculty of engineering, Chulalongkorn University, 2011.
- [7] Johansson, K.H. The quadruple-tank process: A multivariable laboratory process with adjustable zero. *IEEE Transactions on Control Systems Technology*, 2000. 8(3):456-465
- [8] Rusli, E., S. Ang, and R.D. Braatz, A Quadruple-tank process control experiment. *Chemical Engineering Education*, 38(2004) : 174-18.
- [9] Morari, M., and Lygeros, J. IfA Fachpraktikum Quad Tanks. *Automatic Control Laboratory* 18(2010) : 22-35.

- [10] Dale, E. Seborg, Thomas, F. Edgar, Duncan A. Mellichamp and Francis, J. Doyle III. (2011). Process Dynamics and Control (3rd ed.). John Wiley & Sons (Asia) Pte Ltd: John Wiley & Sons, Inc.





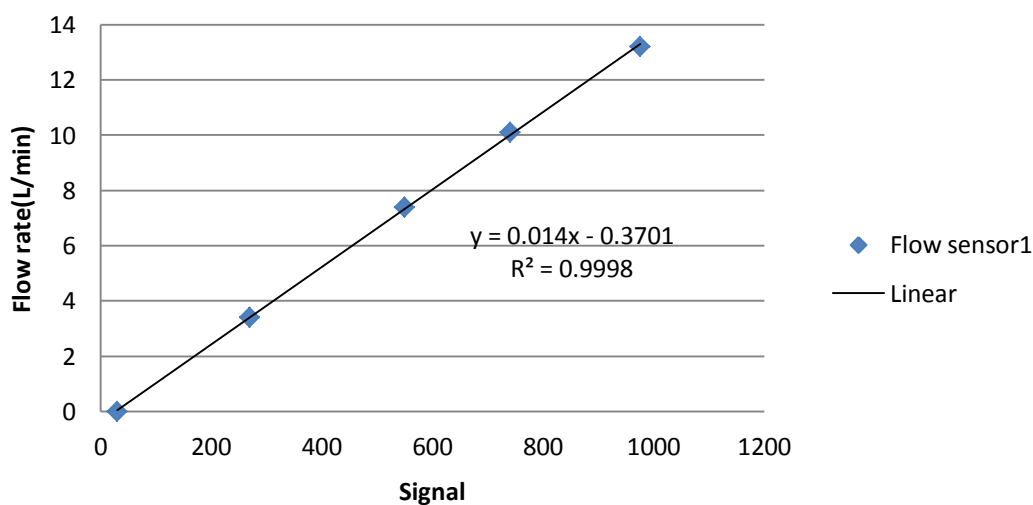
ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

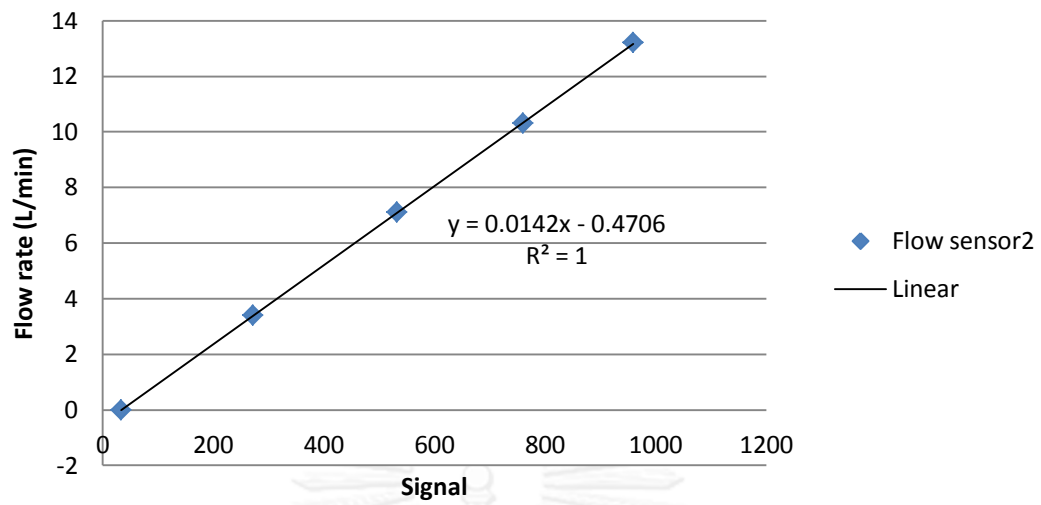
ภาคผนวก ก

การสอบเทียบอุปกรณ์ตรวจวัดอัตราการไหล

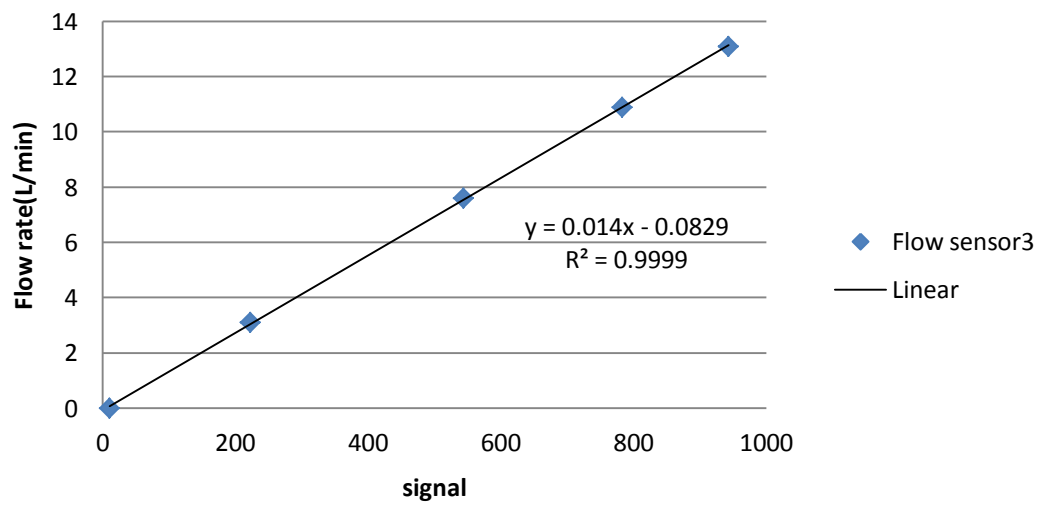
ในการตรวจวัดอัตราการไหลด้วยเครื่องมือตรวจวัดนั้นจำเป็นต้องมีการสอบเทียบอุปกรณ์ก่อนทุกครั้งเพื่อให้ค่าที่อ่านได้จากเครื่องมือวัดมีความถูกต้องที่สุด อุปกรณ์ตรวจวัดอัตราการไหลในระบบสี่ถังจะมีด้วยกัน 4 ตัว โดยใช้วัดอัตราการไหลของน้ำจากปั๊มก่อนเข้าแต่ละถัง ตัวอุปกรณ์จะส่งสัญญาณทางไฟฟ้าในช่วง 4-20 มิลลิแอมป์ หรือ 0-10 โวลต์ โดยค่าสัญญาณจะถูกส่งไปยังอาร์ดูโนซึ่งรับสัญญาณในหน่วยโวลต์ อาร์ดูโนจะแปลงค่าแรงดันไฟฟ้าเป็นค่าสัญญาณในช่วง 0-1023 ในการสอบเทียบอุปกรณ์นั้นจะสุ่มเก็บค่าอัตราการไหลจากหน้าปัดเครื่องมือวัดและสัญญาณที่อาร์ดูโนแปลงมาจำนวน 5 จุด แล้วใช้สมการเส้นตรงเป็นตัวแทนข้อมูล ผลการสอบเทียบของอุปกรณ์ตรวจวัดการไหลทั้ง 4 ตัวแสดงดังรูปที่ ก.1-ก.4 จากกราฟทั้ง 4 พบว่าสามารถใช้สมการเส้นตรงแทนข้อมูลได้เนื่องจาก ค่าสัมประสิทธิ์ของการตัดสีนใจ (R square) มีค่าเข้าใกล้หนึ่ง



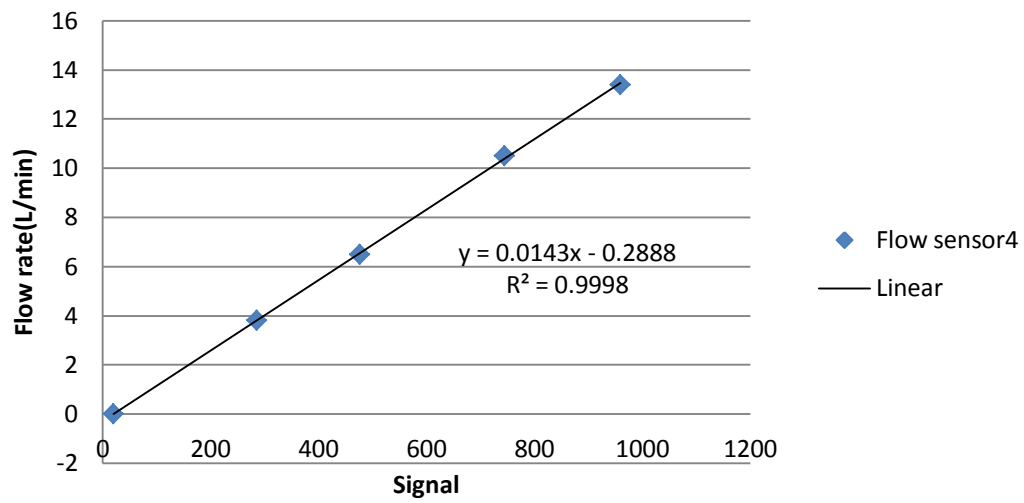
รูปที่ ก.1 ความสัมพันธ์ระหว่างค่าอัตราการไหลจากหน้าปัดเครื่องมือวัดที่ 1 กับค่าสัญญาณจากอาร์ดูโน



รูปที่ ก.2 ความสัมพันธ์ระหว่างค่าอัตราการไหลจากหน้าปัดเครื่องมือวัดที่2
กับค่าสัญญาณจากอาร์ดูโน้



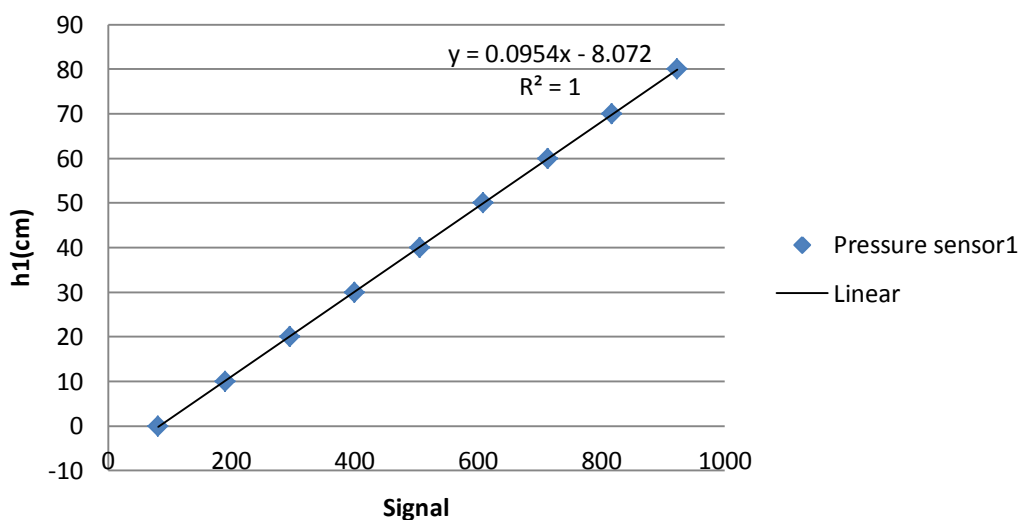
รูปที่ ก.3 ความสัมพันธ์ระหว่างค่าอัตราการไหลจากหน้าปัดเครื่องมือวัดที่3
กับค่าสัญญาณจากอาร์ดูโน้



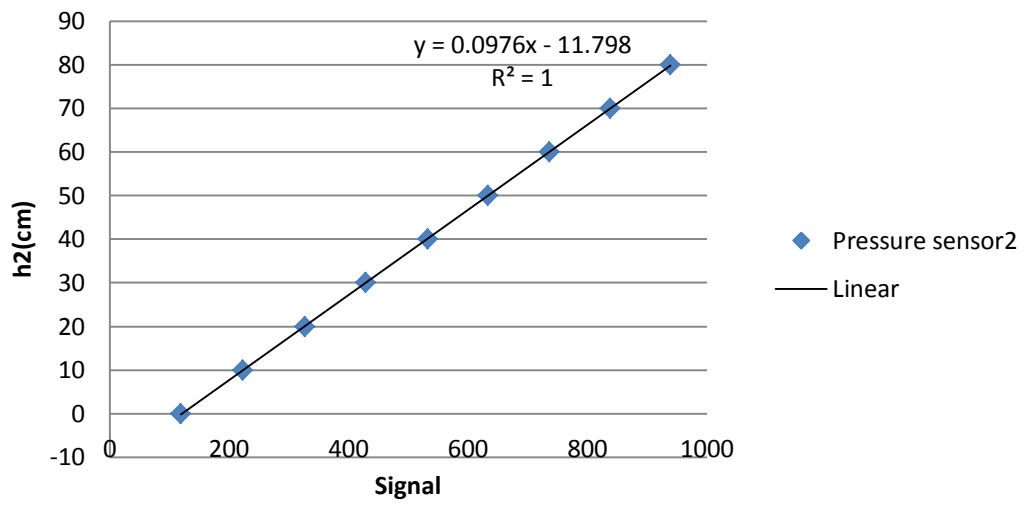
รูปที่ ก.4 ความสัมพันธ์ระหว่างค่าอัตราการไหลจากหน้าปัดเครื่องมือวัดที่4
กับค่าสัญญาณจากอาร์ดูโน้

ภาคผนวก ข
การสอบเทียบอุปกรณ์ตรวจวัดแรงดัน

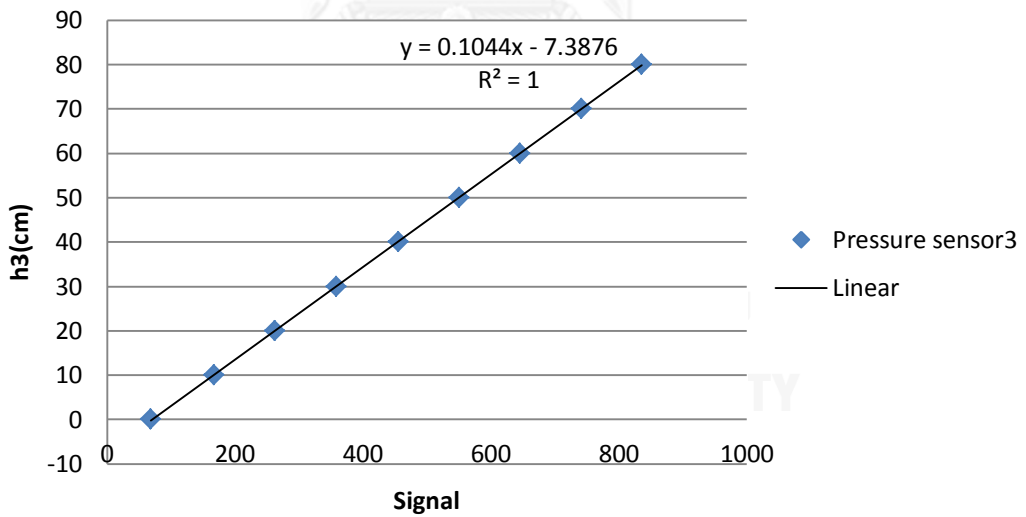
เช่นเดียวกับภาคผนวก ก ในการใช้งานเครื่องมือตรวจวัดนั้นจำเป็นต้องมีการสอบเทียบอุปกรณ์ก่อนทุกครั้ง อุปกรณ์ตรวจวัดแรงดันในระบบสีถังจะมีด้วยกัน 4 ตัว ถูกติดตั้งไว้ใต้ถังแต่ละใบ โดยใช้วัดแรงดันที่ก้นถัง ตัวอุปกรณ์จะส่งสัญญาณทางไฟฟ้าในช่วง 4-20 มิลลิแอมป์ ผ่านอุปกรณ์แปลงกระแสไฟฟ้า ซึ่งจะแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้าก่อนส่งเข้าอาร์ดูโน้ ในการสอบเทียบอุปกรณ์แต่ละตัวนั้นจะอ่านค่าระดับน้ำและสัญญาณที่อาร์ดูโน้แปลงมาจำนวน 9 จุด แล้วใช้สมการเส้นตรงเป็นตัวแทนข้อมูล ผลการสอบเทียบของอุปกรณ์ตรวจวัดแรงดันทั้ง 4 ตัวแสดงดังรูปที่ ข.1-ข.4 จากกราฟทั้ง 4 พบว่าสามารถใช้สมการเส้นตรงแทนข้อมูลได้เนื่องจาก ค่าสัมประสิทธิ์ของการตัดสลิใจ (R square) มีค่าเข้าใกล้หนึ่ง



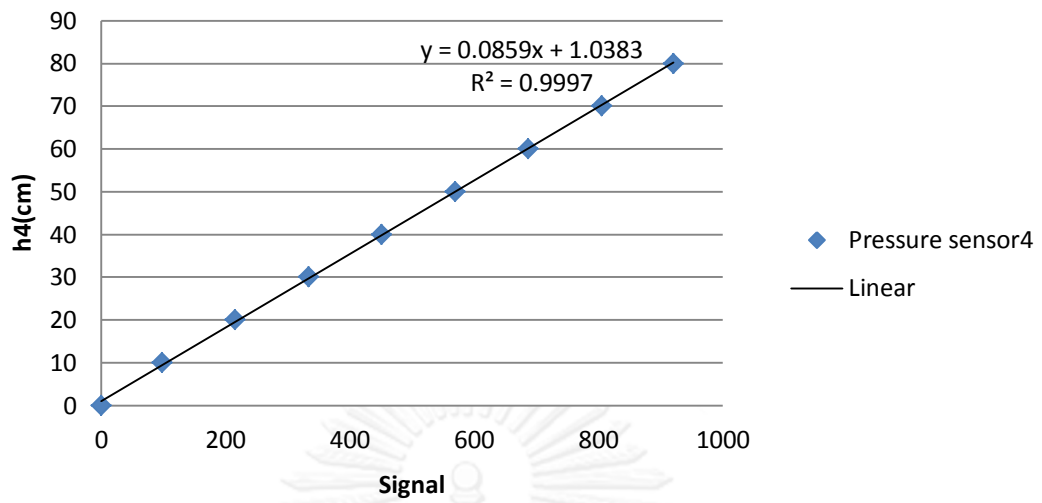
รูปที่ ข.1 ความสัมพันธ์ระหว่างค่าระดับน้ำภายในถังที่1กับค่าสัญญาณจากอาร์ดูโน้



รูปที่ ข.2 ความสัมพันธ์ระหว่างค่าระดับน้ำภายในถังที่2กับค่าสัญญาณจากอาร์ดูโน้



รูปที่ ข.3 ความสัมพันธ์ระหว่างค่าระดับน้ำภายในถังที่3กับค่าสัญญาณจากอาร์ดูโน้



รูปที่ ข.4 ความสัมพันธ์ระหว่างค่าระดับน้ำภายในถังที่4กับค่าสัญญาณจากอาร์ดูโน้

ภาคผนวก ค
ชุดคำสั่งที่ใช้ในการทดสอบระบบสี่ถัง

ในหัวข้อนี้แสดงชุดคำสั่งในส่วนของการทดสอบสมรรถนะของปั๊ม การทดสอบระบบสี่ถังที่สภาวะคงตัวและชุดคำสั่งในการควบคุมระบบสี่ถังภายใต้สภาวะเฟสต่ำสุดและไม่มีเฟสต่ำสุด

ข.1 ชุดคำสั่งในการทดสอบสมรรถนะของปั๊มทั้งสองตัว

```

%%%%%%%%%%%%%% Test performance of pumps %%%%%%%%%%%%%%%

round=1000; % Estimate 42 second
n=round*11; % Number of input step

% Create zero matrixes to collect data
F1_4=zeros(n,2); % Flow rate of pump1
F2_3=zeros(n,2); % Flow rate of pump2
time=zeros(n,1);
%

for i=1:n
    i
    tt=clock;
    time(i,1)=tt(5)+tt(6)/60; % Collect time data (min)

    if i>=round*10+1
        a.analogWrite(7,105); % Send 2.06 volt to the inverter of a pump1
        a.analogWrite(6,105); % Send 2.06 volt to the inverter of a pump2
    elseif i>=round*9+1
        a.analogWrite(7,104); % Send 2.04 volt to the inverter of a pump1
        a.analogWrite(6,104); % Send 2.04 volt to the inverter of a pump2
    elseif i>=round*8+1
        a.analogWrite(7,103); % Send 2.02 volt to the inverter of a pump1
        a.analogWrite(6,103); % Send 2.02 volt to the inverter of a pump2
    elseif i>=round*7+1
        a.analogWrite(7,102); % Send 2 volt to the inverter of a pump1
        a.analogWrite(6,102); % Send 2 volt to the inverter of a pump2
    elseif i>=round*6+1
        a.analogWrite(7,101); % Send 1.98 volt to the inverter of a pump1
        a.analogWrite(6,101); % Send 1.98 volt to the inverter of a pump2
    end
end

```

```

elseif i>=round*5+1
    a.analogWrite(7,100); % Send 1.96 volt to the inverter of a pump1
    a.analogWrite(6,100); % Send 1.96 volt to the inverter of a pump2
elseif i>=round*4+1
    a.analogWrite(7,99); % Send 1.94 volt to the inverter of a pump1
    a.analogWrite(6,99); % Send 1.94 volt to the inverter of a pump2
elseif i>=round*3+1
    a.analogWrite(7,98); % Send 1.92 volt to the inverter of a pump1
    a.analogWrite(6,98); % Send 1.92 volt to the inverter of a pump2
elseif i>=round*2+1
    a.analogWrite(7,97); % Send 1.9 volt to the inverter of a pump1
    a.analogWrite(6,97); % Send 1.9 volt to the inverter of a pump2
elseif i>=round*1+1
    a.analogWrite(7,96); % Send 1.88 volt to the inverter of a pump1
    a.analogWrite(6,96); % Send 1.88 volt to the inverter of a pump2
else
    a.analogWrite(7,95); % Send 1.86 volt to the inverter of a pump1
    a.analogWrite(6,95); % Send 1.86 volt to the inverter of a pump2
end

% Receive flow rate from flow meters
F1_4(i,1)=0.014*a.analogRead(11)-0.37; % Flow rate from a flow meter1 [L/min]
F1_4(i,2)=0.014*a.analogRead(10)-0.288; % Flow rate from a flow meter4 [L/min]
F2_3(i,1)=0.014*a.analogRead(8)-0.47; % Flow rate from a flow meter2 [L/min]
F2_3(i,2)=0.014*a.analogRead(9)-0.082; % Flow rate from a flow meter3 [L/min]

% Collect flow rate of pumps
F14=F1_4(i,1)+ F1_4(i,2); % Flow rate of pump1
F23=F2_3(i,1)+ F2_3(i,2); % Flow rate of pump2

end

% Save variables to file excels to plot graphs with SigmaPlot program
xlswrite('pump_minimumphase.xls',F1_4(1:n,1)+F1_4(1:n,2),1,'B5')
xlswrite('pump_minimumphase.xls',F2_3(1:n,1)+ F2_3(1:n,2),1,'C5')
%
```

ข.2 ชุดคำสั่งในของตัวควบคุมลูปรองที่หนึ่งในส่วนของการคำนวณค่าอัตราขยายแบบป้อนกลับเก็บไว้ในฐานข้อมูลก่อน

```

%% vertices of polytopic uncertainty
A1=[1-0.01];B1=[1*0.01];B2=[100*0.01];
A2=[1-0.1];B3=[1*0.1];B4=[100*0.1];

%% Find state feedback gain(Ks)

QQf=1; % weighting matrix
Rf=1; % weighting matrix
nu=1; % number of input variable
nx=1; % number of state variable
f=[30 15 10 6 3 2 1]; % initial flow rate to find pre-computed state feedback gains(Ks)
[rf rc]=size(f);
n=rc; % number of pre-computed state feedback gains

% Create zero matrix to collect data
k=zeros(1,1,n); % pre-computed state feedback gains matrix (k(i)=y(i)*inv(q(i)))
q=zeros(1,1,n);
y=zeros(1,1,n);
%

for i=1:n

yalmp('clear')
ops=sdpsettings('solver','sedumi');

% Declare variable
Y=sdpvar(nu,nx);
Q=sdpvar(nx,nx);
G=sdpvar(1,1);
%

l1=eye(nx); % state identity matrix
l2=eye(nu); % input identity matrix
F=f(i);

```

```
% Robust stability constraints
```

```
C=[Q>=0];
C=[C,[1 F';F Q]>=0]; % Invariant ellipsoid
C=[C,[Q (Q*A1'+Y'*B1') Q*(QQf^0.5) Y*(Rf^0.5);(A1*Q+B1*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];
C=[C,[Q (Q*A1'+Y'*B2') Q*(QQf^0.5) Y*(Rf^0.5);(A1*Q+B2*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];
C=[C,[Q (Q*A2'+Y'*B3') Q*(QQf^0.5) Y*(Rf^0.5);(A2*Q+B3*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];
C=[C,[Q (Q*A2'+Y'*B4') Q*(QQf^0.5) Y*(Rf^0.5);(A2*Q+B4*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];
```

```
% Input constraints
```

```
C=[C,[0.49^2 Y;Y' Q]>=0];
```

```
% Solve
```

```
solvesdp(C,G,ops);
```

```
% Collect state feedback gain
```

```
q(:,i)=double(Q);
y(:,i)=double(Y);
k(:,i)=double(Y)*inv(double(Q));
%
```

```
end
```

```
save('controller_gain_14','k','q') % save variable to controller_gain_14.mat
```

ข.3 ชุดคำสั่งในของตัวควบคุมลูปรองที่2ในส่วนของกรคำนวณค่าอัตราขยายแบบป้อนกลับเก็บไว้ในฐานข้อมูลก่อน

```
%% vertices of polytopic uncertainty
```

```
A1=[1-0.01];B1=[1*0.01];B2=[100*0.01];
```

```
A2=[1-0.6];B3=[1*0.6];B4=[100*0.6];
```

```
%% Find state feedback gain(Ks)
```

```
QQf=1; % weighting matrix
```

```
Rf=1; % weighting matrix
```

```

nu=1; % number of input variable
nx=1; % number of state variable
f=[30 15 10 6 3 2 1]; % initial flowrate to find state feedback gain(Ks)
[rf rc]=size(f);
n=rc; % number of pre-computed state feedback gains

% Create zero matrix to collect data
k=zeros(1,1,n); % pre-computed state feedback gains matrix (k(i)=y(i)*inv(q(i)))
q=zeros(1,1,n);
y=zeros(1,1,n);
%

for i=1:n

yalmip('clear')
ops=sdpssettings('solver','sedumi');

%Declare variable
X=sdpvar(nu,nu);
Y=sdpvar(nu,nx);
Q=sdpvar(nx,nx);
G=sdpvar(1,1);
%

I1=eye(nx); % state identity matrix
I2=eye(nu); % input identity matrix
F=f(i);

% Robust stability constraints

C=[Q>=0];
C=[C,[1 F';F Q]>=0]; % Invariant ellipsoid
C=[C,[Q (Q*A1'+Y'*B1') Q*(QQf^0.5) Y*(Rf^0.5);(A1*Q+B1*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];
C=[C,[Q (Q*A1'+Y'*B2') Q*(QQf^0.5) Y*(Rf^0.5);(A1*Q+B2*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];
C=[C,[Q (Q*A2'+Y'*B3') Q*(QQf^0.5) Y*(Rf^0.5);(A2*Q+B3*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];
C=[C,[Q (Q*A2'+Y'*B4') Q*(QQf^0.5) Y*(Rf^0.5);(A2*Q+B4*Y) Q zeros(nx,nx) zeros(nx,nu);(QQf^0.5)*Q
zeros(nx,nx) G*I1 zeros(nx,nu);(Rf^0.5)*Y zeros(nu,nx) zeros(nu,nx) G*I2]>=0];

```

```

% Input constraints
C=[C,[0.098^2 Y;Y' Q]>=0];
% Solve
    solvesdp(C,G,ops);
% Collect state feedback gain
q(:,i)=double(Q);
y(:,i)=double(Y);
k(:,i)=double(Y)*inv(double(Q));
%
end

save('controller_gain_23','k','q') % save variable to controller_gain_23.mat

```

ข.4 ชุดคำสั่งในการทดสอบระบบสี่ถังเพื่อหาค่าสภาวะคงตัว

```

nu=2; % Number of input variables (flow control loop)
nx=2; % Number of state variables (flow control loop)
fss=[15.5;16]; % Flow rate setpoint
d=2000; % Rounds of sampling

% Create zero matrixes to collect data
input=zeros(d,nx); % Deviation of input variable (volt)
input_pump=zeros(d,nx); % Signal from Arduino to inverters
input_uC=zeros(d,nx); % Input (volt)
flow=zeros(d,nx); % Deviation of flow rate (L/min)
f=zeros(d,nx); % Flow rate (L/min)
time=zeros(d,1); % Time (min)
output=zeros(d,nx); % Level of water in each a tank (cm)
%

it=1;
clear sum_er1;
clear sum_er2;

for i=1:d

    tt=clock;
    time(i,1)=tt(5)+tt(6)/60; % Collect time data

```

```

% Receive signal from pressure sensors pass a Arduino
for j=1:5 % Random 5 samples of each a pressure sensor
    h1_sig(j)=a.analogRead(3); % pressure sensor1
    h2_sig(j)=a.analogRead(5); % pressure sensor1
    h3_sig(j)=a.analogRead(2); % pressure sensor1
    h4_sig(j)=a.analogRead(4); % pressure sensor1
end
% Level of water in each a tank (cm)
h1=0.095*median(h1_sig)-8.072 % tank1(cm)
h2=0.097*median(h2_sig)-11.79 % tank2(cm)
h3=0.104*median(h3_sig)-7.387 % tank3(cm)
h4=0.085*median(h4_sig)+1.038 % tank4(cm)
% Collect level of water in each a tank (cm)
output(i,1)=h1; % tank1(cm)
output(i,2)=h2; % tank2(cm)
output(i,3)=h3; % tank3(cm)
output(i,4)=h4; % tank4(cm)
% Receive signal from flow meters pass a Arduino
for j=1:5 % Random 5 samples of each a flow meter
    f1(j)=0.014*a.analogRead(11)-0.37;
    f2(j)=0.014*a.analogRead(8)-0.47;
    f3(j)=0.014*a.analogRead(9)-0.082;
    f4(j)=0.014*a.analogRead(10)-0.288;
end
%
F=[median(f1)+median(f4)-fss(1);median(f2)+median(f3)-fss(2)]; % State variable of a flow
control loop
f(i,1)=median(f1)+median(f4); % Flow rate of a pump1
f(i,2)=median(f2)+median(f3); % Flow rate of a pump2

%% Find input variable for flow control loop1
load controller_gain_14
n=7; % Number of pre-computed state feedback gain
clear K;

if i==1
    x1=F(1);
else
    x1=f(i,1)-f(i-1,1)+x1; % x1(i)-x1(i-1)=f(i,1)-f(i-1,1)
end

```

% Choose k that state variables is member of ellipsoid invariant set

```

if x1*inv(q(:,n))*x1<=1;
    K=k(:,n);
elseif x1*inv(q(:,n-1))*x1<=1;
    K=k(:,n-1);
elseif x1*inv(q(:,n-2))*x1<=1;
    K=k(:,n-2);
elseif x1*inv(q(:,n-3))*x1<=1;
    K=k(:,n-3);
elseif x1*inv(q(:,n-4))*x1<=1;
    K=k(:,n-4);
elseif x1*inv(q(:,n-5))*x1<=1;
    K=k(:,n-5);
elseif x1*inv(q(:,n-6))*x1<=1;
    K=k(:,n-6);
end
%
u(1)=K*x1; % Deviation of input1 (volt)

```

```

%% Find input variable for flow control loop2
clear K;
load controller_gain_23
n=7; % Number of pre-computed state feedback gain
if i==1
    x2=F(2);
else
    x2=f(i,2)-f(i-1,2)+x2; % x2(i)-x2(i-1)=f(i,1)-f(i-1,1)
end

```

% Choose k that state variables is member of ellipsoid invariant set

```

if x2*inv(q(:,n))*x2<=1;
    K=k(:,n);
elseif x2*inv(q(:,n-1))*x2<=1;
    K=k(:,n-1);
elseif x2*inv(q(:,n-2))*x2<=1;
    K=k(:,n-2);
elseif x2*inv(q(:,n-3))*x2<=1;
    K=k(:,n-3);
elseif x2*inv(q(:,n-4))*x2<=1;
    K=k(:,n-4);

```



```

elseif x2*inv(q(:,n-5))*x2<=1;
    K=k(:,n-5);
elseif x2*inv(q(:,n-6))*x2<=1;
    K=k(:,n-6);
end
%
u(2)=K*x2; % Deviation of a input2 (volt)

% Collect deviations of input
input(i,1)=u(1); % deviation of input1
input(i,2)=u(2); % deviation of input2
% Input variable
    if i==1
        u1C=u(1)+(5/255)*100;
        u2C=u(2)+(5/255)*100;
    else
        u1C=input(i,1)-input(i-1,1)+input_uC(i-1,1);
        u2C=input(i,2)-input(i-1,2)+input_uC(i-1,2);
    end
% Collect input variables (volt)
input_uC(i,1)=u1C; % input1 (volt)
input_uC(i,2)=u2C; % input2 (volt)
% Convert input (volt) to signal [0-5v to 0-255]
    U1=int8(u1C*(255/5));
    U2=int8(u2C*(255/5));
% Collect signal
input_pump(i,1)=U1;
input_pump(i,2)=U2;
% Send signal to control a pump1 and a pump2
a.analogWrite(7,U1); %pump1
a.analogWrite(6,U2); %pump2
%
flow(i,:)=F'; % Collect deviation of flow rate (volt)
% Stop pump2 when i==final round of sampling
    if i==d
        a.analogWrite(6,0);
        a.analogWrite(7,0);
    end
end
end

```

%Save variables to excel file to plot graphs with SigmaPlot program

```
xlswrite('steady_state.xls',input_pump(:,1)-100,1,'B5')
```

```
xlswrite('steady_state.xls',flow(:,1),1,'C5')
```

```
xlswrite('steady_state.xls',input_pump(:,2)-100,1,'E5')
```

```
xlswrite('steady_state.xls',flow(:,2),1,'F5')
```

```
xlswrite('steady_state2.xls',output(:,1),1,'B5')
```

```
xlswrite('steady_state2.xls',output(:,2),1,'C5')
```

```
xlswrite('steady_state2.xls',output(:,3),1,'D5')
```

```
xlswrite('steady_state2.xls',output(:,4),1,'E5')
```

```
%
```

ข.5 ชุดคำสั่งในการควบคุมระดับน้ำในถังที่หนึ่งและถังที่สอง ในส่วนของการคำนวณค่าอัตราขยายตัวควบคุมแบบป้อนกลับสัญญาณขาออกและค่าอัตราขยายของตัวประมาณค่าตัวแปรสถานะเก็บไว้ในฐานข้อมูลก่อน

%%%%This file will find pre-computed output feedback gain(K) and estimator gain(Lp)%%%

```
yalmp('clear')
```

```
clear all
```

```
%% Constant value
```

```
at=160.6; %Cross-section area of each tank [cm^2] , Dtank=14.3cm
```

```
g=981; %Gravity force[cm/s^2]
```

```
% Minimumphase case
```

```
%h1s=40.32; %Water level of tank1 at steady state [cm]
```

```
%h2s=42.89; %Water level of tank2 at steady state [cm]
```

```
%h3s=16.63; %Water level of tank3 at steady state [cm]
```

```
%h4s=23.66; %Water level of tank4 at steady state [cm]
```

```
% Non-minimumphase case
```

```
%h1s=24.9; %Water level of tank1 at steady state [cm]
```

```
%h2s=24.82; %Water level of tank2 at steady state [cm]
```

```
%h3s=39.26; %Water level of tank3 at steady state [cm]
```

```
%h4s=38.56; %Water level of tank4 at steady state [cm]
```

```
%
```

```
h1s=40.32; %Water level of tank1 at steady state [cm]
```

```
h2s=42.89; %Water level of tank2 at steady state [cm]
```

```
h3s=16.63; %Water level of tank3 at steady state [cm]
```

```
h4s=23.66; %Water level of tank4 at steady state [cm]
```

```
% Minimumphase case
```

```
%hn=4; %Minimum level of water in each tank[cm]
```

```
%hx=50; %Maximum level of water in each tank[cm]
```

```

% Non-inimumphase case
    %hn=8; %Minimum level of water in each tank[cm]
    %hx=45; %Maximum level of water in each tank[cm]
%
    hn=4; %Minimum level of water in each tank[cm]
    hx=50; %Maximum level of water in each tank[cm]

% Minimumphase case
    %dt=10; %Sampling time
% Non-inimumphase case
    %dt=12; %Sampling time
%
    dt=10; %Sampling time

    nx=4; %Number of state;
    nu=2; %Number of input;

%% Vertices of matrix A

% Minimumphase case
    %f1ss=10.85; % flow from pump1 to tank1
    %f2ss=11.35; % flow from pump2 to tank2
    %f3ss=4.65; % flow from pump2 to tank3
    %f4ss=4.7; % flow from pump1 to tank4
% Non-inimumphase case
    %f1ss=1.55; % flow from pump1 to tank1
    %f2ss=1.76; % flow from pump2 to tank2
    %f3ss=8.9; % flow from pump2 to tank3
    %f4ss=10.24; % flow from pump1 to tank4
%

% flow at steady state (L/min)
f1ss=10.85; % flow from pump1 to tank1
f2ss=11.35; % flow from pump2 to tank2
f3ss=4.65; % flow from pump2 to tank3
f4ss=4.7; % flow from pump1 to tank4
% mi at steady state
m1ss=((f1ss+f3ss)*1000/60)/sqrt(h1s); % Qpipe1=m1*sqrt(h1) % FACTOR TO CONVERSE L/MIN TO
CM3/S IS 1000/60
m2ss=((f2ss+f4ss)*1000/60)/sqrt(h2s); % Qpipe2=m2*sqrt(h2)

```

```

m3ss=(f3ss*1000/60)/sqrt(h3s); % Qpipe2=m3*sqrt(h3)
m4ss=(f4ss*1000/60)/sqrt(h4s); % Qpipe2=m4*sqrt(h4)

v1ss=dt*m1ss/at;
v2ss=dt*m2ss/at;
v3ss=dt*m3ss/at;
v4ss=dt*m4ss/at;

t1n=1/(sqrt(hn));
t2n=1/(sqrt(hn));
t3n=1/(sqrt(hn));
t4n=1/(sqrt(hn));

t1x=1/(sqrt(hx));
t2x=1/(sqrt(hx));
t3x=1/(sqrt(hx));
t4x=1/(sqrt(hx));

hss=[h1s;h2s;h3s;h4s]; %Water level of at steady state [cm]
hnb=[hn-h1s;hn-h2s;hn-h3s;hn-h4s]; %Minimum deviation value of each tank
hxb=[hx-h1s;hx-h2s;hx-h3s;hx-h4s]; %Maximum deviation value of each tank

t1ss=1/(sqrt(h1s));
t2ss=1/(sqrt(h2s));
t3ss=1/(sqrt(h3s));
t4ss=1/(sqrt(h4s));

% Matrix A at steady state
Ass=[1-v1ss*t1ss 0 v3ss*t3ss 0;0 1-v2ss*t2ss 0 v4ss*t4ss;0 0 1-v3ss*t3ss 0;0 0 0 1-v4ss*t4ss];
%
vt_A=2^5; % Number of vertices
ex=1.3; % Maximum deviation from CV at steady state is CV +30%
en=0.7; % Minimum deviation from CV at steady state is CV +30%

%This state, every tank have minimum level of water
hb1=hnb;
% +30% error of CV from CV at steady state
a1=[1-v1ss*t1n*ex 0 v3ss*t3n*ex 0;0 1-v2ss*t2n*ex 0 v4ss*t4n*ex;0 0 1-v3ss*t3n*ex 0;0 0
0 1-v4ss*t4n*ex];
A1=[a1+(a1-Ass)*hss*pinv(hb1)];

```

```

% -30% error of CV from CV at steady state
a2=[1-v1ss*t1n*en 0 v3ss*t3n*en 0; 0 1-v2ss*t2n*en 0 v4ss*t4n*en; 0 0 1-v3ss*t3n*en 0; 0 0
0 1-v4ss*t4n*en];
A2=[a2+(a2-Ass)*hss*pinv(hb1)];

%This state, tank 1 has maximum level of water
hb2=[hxb(1);hnb(2:4)];
% +30% error of CV from CV at steady state
a3=[1-v1ss*t1x*ex 0 v3ss*t3n*ex 0; 0 1-v2ss*t2n*ex 0 v4ss*t4n*ex; 0 0 1-v3ss*t3n*ex 0; 0 0 0 1-
v4ss*t4n*ex];
A3=[a3+(a3-Ass)*hss*pinv(hb2)];
% -30% error of CV from CV at steady state
a4=[1-v1ss*t1x*en 0 v3ss*t3n*en 0; 0 1-v2ss*t2n*en 0 v4ss*t4n*en; 0 0 1-v3ss*t3n*en 0; 0 0 0
1-v4ss*t4n*en];
A4=[a4+(a2-Ass)*hss*pinv(hb2)];

%This state, tank 2 has maximum level of water
hb3=[hnb(1);hxb(2);hnb(3:4)];
% +30% error of CV from CV at steady state
a5=[1-v1ss*t1n*ex 0 v3ss*t3n*ex 0; 0 1-v2ss*t2x*ex 0 v4ss*t4n*ex; 0 0 1-v3ss*t3n*ex 0; 0 0 0 1-
v4ss*t4n*ex];
A5=[a5+(a5-Ass)*hss*pinv(hb3)];
% -30% error of CV from CV at steady state
a6=[1-v1ss*t1n*en 0 v3ss*t3n*en 0; 0 1-v2ss*t2x*en 0 v4ss*t4n*en; 0 0 1-v3ss*t3n*en 0; 0 0 0
1-v4ss*t4n*en];
A6=[a6+(a6-Ass)*hss*pinv(hb3)];

%This state, tank 3 has maximum level of water
hb4=[hnb(1:2);hxb(3);hnb(4)];
% +30% error of CV from CV at steady state
a7=[1-v1ss*t1n*ex 0 v3ss*t3x*ex 0; 0 1-v2ss*t2n*ex 0 v4ss*t4n*ex; 0 0 1-v3ss*t3x*ex 0; 0 0 0 1-
v4ss*t4n*ex];
A7=[a7+(a7-Ass)*hss*pinv(hb4)];
% -30% error of CV from CV at steady state
a8=[1-v1ss*t1n*en 0 v3ss*t3x*en 0; 0 1-v2ss*t2n*en 0 v4ss*t4n*en; 0 0 1-v3ss*t3x*en 0; 0 0 0
1-v4ss*t4n*en];
A8=[a8+(a8-Ass)*hss*pinv(hb4)];

```

%This state, tank 4 has maximum level of water

hb5=[hnb(1:3);hxb(4)];

% +30% error of CV from CV at steady state

a9=[1-v1ss*t1n*ex 0 v3ss*t3n*ex 0;0 1-v2ss*t2n*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3n*ex 0;0 0 1-v4ss*t4x*ex];

A9=[a9+(a9-Ass)*hss*pinv(hb5)];

% -30% error of CV from CV at steady state

a10=[1-v1ss*t1n*en 0 v3ss*t3n*en 0;0 1-v2ss*t2n*en 0 v4ss*t4x*en;0 0 1-v3ss*t3n*en 0;0 0 1-v4ss*t4x*en];

A10=[a10+(a10-Ass)*hss*pinv(hb5)];

%This state, tank 1 and tank 2 have maximum level of water

hb6=[hxb(1:2);hnb(3:4)];

% +30% error of CV from CV at steady state

a11=[1-v1ss*t1x*ex 0 v3ss*t3n*ex 0;0 1-v2ss*t2x*ex 0 v4ss*t4n*ex;0 0 1-v3ss*t3n*ex 0;0 0 1-v4ss*t4n*ex];

A11=[a11+(a11-Ass)*hss*pinv(hb6)];

% -30% error of CV from CV at steady state

a12=[1-v1ss*t1x*en 0 v3ss*t3n*en 0;0 1-v2ss*t2x*en 0 v4ss*t4n*en;0 0 1-v3ss*t3n*en 0;0 0 1-v4ss*t4n*en];

A12=[a12+(a12-Ass)*hss*pinv(hb6)];

%This state, tank 1 and tank 3 have maximum level of water

hb7=[hxb(1);hnb(2);hxb(3);hnb(4)];

% +30% error of CV from CV at steady state

a13=[1-v1ss*t1x*ex 0 v3ss*t3x*ex 0;0 1-v2ss*t2n*ex 0 v4ss*t4n*ex;0 0 1-v3ss*t3x*ex 0;0 0 1-v4ss*t4n*ex];

A13=[a13+(a13-Ass)*hss*pinv(hb7)];

% -30% error of CV from CV at steady state

a14=[1-v1ss*t1x*en 0 v3ss*t3x*en 0;0 1-v2ss*t2n*en 0 v4ss*t4n*en;0 0 1-v3ss*t3x*en 0;0 0 1-v4ss*t4n*en];

A14=[a14+(a14-Ass)*hss*pinv(hb7)];

%This state, tank 1 and tank 4 have maximum level of water

hb8=[hxb(1);hnb(2:3);hxb(4)];

% +30% error of CV from CV at steady state

a15=[1-v1ss*t1x*ex 0 v3ss*t3n*ex 0;0 1-v2ss*t2n*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3n*ex 0;0 0 1-v4ss*t4x*ex];

A15=[a15+(a15-Ass)*hss*pinv(hb8)];

```

% -30% error of CV from CV at steady state
a16=[1-v1ss*t1x*en 0 v3ss*t3n*en 0;0 1-v2ss*t2n*en 0 v4ss*t4x*en;0 0 1-v3ss*t3n*en 0;0 0 0
1-v4ss*t4x*en];
A16=[a16+(a16-Ass)*hss*pinv(hb8)];

%This state, tank 2 and tank 3 have maximum level of water
hb9=[hnb(1);hxb(2:3);hnb(4)];
% +30% error of CV from CV at steady state
a17=[1-v1ss*t1n*ex 0 v3ss*t3x*ex 0;0 1-v2ss*t2x*ex 0 v4ss*t4n*ex;0 0 1-v3ss*t3x*ex 0;0 0 0
1-v4ss*t4n*ex];
A17=[a17+(a17-Ass)*hss*pinv(hb9)];
% -30% error of CV from CV at steady state
a18=[1-v1ss*t1n*en 0 v3ss*t3x*en 0;0 1-v2ss*t2x*en 0 v4ss*t4n*en;0 0 1-v3ss*t3x*en 0;0 0 0
1-v4ss*t4n*en];
A18=[a18+(a18-Ass)*hss*pinv(hb9)];

%This state, tank 2 and tank 4 have maximum level of water
hb10=[hnb(1);hxb(2);hnb(3);hxb(4)];
% +30% error of CV from CV at steady state
a19=[1-v1ss*t1n*ex 0 v3ss*t3n*ex 0;0 1-v2ss*t2x*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3n*ex 0;0 0 0
1-v4ss*t4x*ex];
A19=[a19+(a19-Ass)*hss*pinv(hb10)];
% -30% error of CV from CV at steady state
a20=[1-v1ss*t1n*en 0 v3ss*t3n*en 0;0 1-v2ss*t2x*en 0 v4ss*t4x*en;0 0 1-v3ss*t3n*en 0;0 0 0
1-v4ss*t4x*en];
A20=[a20+(a20-Ass)*hss*pinv(hb10)];

%This state, tank 3 and tank 4 have maximum level of water
hb11=[hnb(1:2);hxb(3:4)];
% +30% error of CV from CV at steady state
a21=[1-v1ss*t1n*ex 0 v3ss*t3x*ex 0;0 1-v2ss*t2n*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3x*ex 0;0 0 0
1-v4ss*t4x*ex];
A21=[a21+(a21-Ass)*hss*pinv(hb11)];
% -30% error of CV from CV at steady state
a22=[1-v1ss*t1n*en 0 v3ss*t3x*en 0;0 1-v2ss*t2n*en 0 v4ss*t4x*en;0 0 1-v3ss*t3x*en 0;0 0 0
1-v4ss*t4x*en];
A22=[a22+(a22-Ass)*hss*pinv(hb11)];

```

%This state, tank 1, tank 2 and tank 3 have maximum level of water

```

hb12=[hxb(1:3);hnb(4)];
% +30% error of CV from CV at steady state
a23=[1-v1ss*t1x*ex 0 v3ss*t3x*ex 0;0 1-v2ss*t2x*ex 0 v4ss*t4n*ex;0 0 1-v3ss*t3x*ex 0;0 0 0
1-v4ss*t4n*ex];
A23=[a23+(a23-Ass)*hss*pinv(hb12)];
% -30% error of CV from CV at steady state
a24=[1-v1ss*t1x*en 0 v3ss*t3x*en 0;0 1-v2ss*t2x*en 0 v4ss*t4n*en;0 0 1-v3ss*t3x*en 0;0 0 0
1-v4ss*t4n*en];
A24=[a24+(a24-Ass)*hss*pinv(hb12)];

```

%This state, tank 1, tank 2 and tank 4 have maximum level of water

```

hb13=[hxb(1:2);hnb(3);hxb(4)];
% +30% error of CV from CV at steady state
a25=[1-v1ss*t1x*ex 0 v3ss*t3n*ex 0;0 1-v2ss*t2x*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3n*ex 0;0 0 0
1-v4ss*t4x*ex];
A25=[a25+(a25-Ass)*hss*pinv(hb13)];
% -30% error of CV from CV at steady state
a26=[1-v1ss*t1x*en 0 v3ss*t3n*en 0;0 1-v2ss*t2x*en 0 v4ss*t4x*en;0 0 1-v3ss*t3n*en 0;0 0 0
1-v4ss*t4x*en];
A26=[a26+(a26-Ass)*hss*pinv(hb13)];

```

%This state, tank 2, tank 3 and tank 4 have maximum level of water

```

hb14=[hnb(1);hxb(2:4)];
% +30% error of CV from CV at steady state
a27=[1-v1ss*t1n*ex 0 v3ss*t3x*ex 0;0 1-v2ss*t2x*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3x*ex 0;0 0 0
1-v4ss*t4x*ex];
A27=[a27+(a27-Ass)*hss*pinv(hb14)];
% -30% error of CV from CV at steady state
a28=[1-v1ss*t1n*en 0 v3ss*t3x*en 0;0 1-v2ss*t2x*en 0 v4ss*t4x*en;0 0 1-v3ss*t3x*en 0;0 0 0
1-v4ss*t4x*en];
A28=[a28+(a28-Ass)*hss*pinv(hb14)];

```

%This state,tank 1,tank 3 and tank4 have maximum level of water

```

hb15=[hxb(1);hnb(2);hxb(3:4)];
% +30% error of CV from CV at steady state
a29=[1-v1ss*t1x*ex 0 v3ss*t3x*ex 0;0 1-v2ss*t2n*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3x*ex 0;0 0 0
1-v4ss*t4x*ex];
A29=[a29+(a29-Ass)*hss*pinv(hb15)];

```



```

% -30% error of CV from CV at steady state
a30=[1-v1ss*t1x*en 0 v3ss*t3x*en 0;0 1-v2ss*t2n*en 0 v4ss*t4x*en;0 0 1-v3ss*t3x*en 0;0 0 0 1-
v4ss*t4x*en];
A30=[a30+(a30-Ass)*hss*pinv(hb15)];

%This state, every tank have maximum level of water
hb16=hxb;
% +30% error of CV from CV at steady state
a31=[1-v1ss*t1x*ex 0 v3ss*t3x*ex 0;0 1-v2ss*t2x*ex 0 v4ss*t4x*ex;0 0 1-v3ss*t3x*ex 0;0 0 0 1-
v4ss*t4x*ex];
A31=[a31+(a31-Ass)*hss*pinv(hb16)];
% -30% error of CV from CV at steady state
a32=[1-v1ss*t1x*en 0 v3ss*t3x*en 0;0 1-v2ss*t2x*en 0 v4ss*t4x*en;0 0 1-v3ss*t3x*en 0;0 0 0 1-
v4ss*t4x*en];
A32=[a32+(a32-Ass)*hss*pinv(hb16)];

A=zeros(nx,nx,vt_A); % Arrey of matrix A

A(:,,1)=A1;    A(:,,17)=A17;
A(:,,2)=A2;    A(:,,18)=A18;
A(:,,3)=A3;    A(:,,19)=A19;
A(:,,4)=A4;    A(:,,20)=A20;
A(:,,5)=A5;    A(:,,21)=A21;
A(:,,6)=A6;    A(:,,22)=A22;
A(:,,7)=A7;    A(:,,23)=A23;
A(:,,8)=A8;    A(:,,24)=A24;
A(:,,9)=A9;    A(:,,25)=A25;
A(:,,10)=A10;  A(:,,26)=A26;
A(:,,11)=A11;  A(:,,27)=A27;
A(:,,12)=A12;  A(:,,28)=A28;
A(:,,13)=A13;  A(:,,29)=A29;
A(:,,14)=A14;  A(:,,30)=A30;
A(:,,15)=A15;  A(:,,31)=A31;
A(:,,16)=A16;  A(:,,32)=A32;

%% Vertices of matrix B
vt_B=2^2; % Number of vertices
% Minimumphase case
%kp1x=142; %Maximum pump1 constant [cm^3/sV]
%kp1n=83; %Minimum pump1 constant [cm^3/sV]
%kp2x=171; %Maximum pump2 constant [cm^3/sV]

```

```

%kp2n=86;      %Minimum pump2 constant [cm^3/sV]
%kp1ss=136.05; %Pump1 constant at steady state [cm^3/sV]
%kp2ss=140.31; %Pump2 constant at steady state [cm^3/sV]

%u1x=2.06;     %Maximum pump1 voltage at steady state [V]
%u1n=1.86;     %Minimum pump1 voltage at steady state [V]
%u2x=2.06;     %Maximum pump1 voltage at steady state [V]
%u2n=1.86;     %Minimum pump1 voltage at steady state [V]
%uss=[1.96;1.96]; %Input variable (pump voltage) at steady state[V]

%gamma1x=0.66; % Minimum ratio of valve1 at maximum pump voltage
%gamma1n=0.94; % Maximum ratio of valve1 at minimum pump voltage
%gamma2x=0.62; % Minimum ratio of valve2 at maximum pump voltage
%gamma2n=0.99; % Maximum ratio of valve2 at minimum pump voltage
%gamma1ss=0.7; % Ratio of valve1 at steady state pump voltage
%gamma2ss=0.7; % Ratio of valve2 at steady state pump voltage
% Non-minimumphase case
%kp1x=113;     %Maximum pump1 constant [cm^3/sV]
%kp1n=86;     %Minimum pump1 constant [cm^3/sV]
%kp2x=113;     %Maximum pump2 constant [cm^3/sV]
%kp2n=60;     %Minimum pump2 constant [cm^3/sV]
%kp1ss=100.215; %Pump1 constant at steady state [cm^3/sV]
%kp2ss=90.16;  %Pump2 constant at steady state [cm^3/sV]

%u1x=2.06;     %Maximum pump1 voltage at steady state [V]
%u1n=1.86;     %Minimum pump1 voltage at steady state [V]
%u2x=2.06;     %Maximum pump1 voltage at steady state [V]
%u2n=1.86;     %Minimum pump1 voltage at steady state [V]
%uss=[1.96;1.96]; %Input variable (pump voltage) at steady state[V]

%gamma1x=0.142; % Minimum ratio of valve1 at maximum pump voltage
%gamma1n=0.125; % Maximum ratio of valve1 at minimum pump voltage
%gamma2x=0.222; % Minimum ratio of valve2 at maximum pump voltage
%gamma2n=0.146; % Maximum ratio of valve2 at minimum pump voltage
%gamma1ss=0.131; % Ratio of valve1 at steady state pump voltage
%gamma2ss=0.165; % Ratio of valve2 at steady state pump voltage
%
kp1x=142; %Maximum pump1 constant [cm^3/sV]
kp1n=83; %Minimum pump1 constant [cm^3/sV]
kp2x=171; %Maximum pump2 constant [cm^3/sV]
kp2n=86; %Minimum pump2 constant [cm^3/sV]

```

```

kp1ss=136.05; %Pump1 constant at steady state [cm^3/sV]
kp2ss=140.31; %Pump2 constant at steady state [cm^3/sV]

u1x=2.06;      %Maximum pump1 voltage at steady state [V]
u1n=1.86;      %Minimum pump1 voltage at steady state [V]
u2x=2.06;      %Maximum pump1 voltage at steady state [V]
u2n=1.86;      %Minimum pump1 voltage at steady state [V]
uss=[1.96;1.96]; %Input variable (pump voltage) at steady state[V]

gamma1x=0.66; % Minimum ratio of valve1 at maximum pump voltage
gamma1n=0.94; % Maximum ratio of valve1 at minimum pump voltage
gamma2x=0.62; % Minimum ratio of valve2 at maximum pump voltage
gamma2n=0.99; % Maximum ratio of valve2 at minimum pump voltage
gamma1ss=0.7; % Ratio of valve1 at steady state pump voltage
gamma2ss=0.7; % Ratio of valve2 at steady state pump voltage

bss=(eye(4)-Ass)*hss; % From hss=Ass*hss+Bss*uss (eye(4)-Ass)*hss=Bss*uss=bss
Bss=[bss(1)/uss(1) 0;0 bss(2)/uss(2);0 bss(3)/uss(2);bss(4)/uss(1) 0]; % Matrix B at steady state

% u1x,u2x,gamma1x,gamma2x
ub1=[u1x-uss(1);u2x-uss(2)];
b1=[(gamma1x*kp1x)*dt/at 0;0 (gamma2x*kp2x)*dt/at;0 (1-gamma2x)*kp2x*dt/at;(1-
gamma1x)*kp1x*dt/at 0];
B1=[b1+(b1-Bss)*uss*pinv(ub1)];
% u1x,u2n,gamma1x,gamma2n
ub2=[u1x-uss(1);u2n-uss(2)];
b2=[(gamma1x*kp1x)*dt/at 0;0 (gamma2n*kp2n)*dt/at;0 (1-gamma2n)*kp2n*dt/at;(1-
gamma1x)*kp1x*dt/at 0];
B2=[b2+(b2-Bss)*uss*pinv(ub2)];
% u1n,u2x,gamma1n,gamma2x
ub3=[u1n-uss(1);u2x-uss(2)];
b3=[(gamma1n*kp1n)*dt/at 0;0 (gamma2x*kp2x)*dt/(at);0 (1-gamma2x)*kp2x*dt/at;(1-
gamma1n)*kp1n*dt/at 0];
B3=[b3+(b3-Bss)*uss*pinv(ub3)];
% u1n,u2n,gamma1n,gamma2n
ub4=[u1n-uss(1);u2n-uss(2)];
b4=[(gamma1n*kp1n)*dt/at 0;0 (gamma2n*kp2n)*dt/(at);0 (1-gamma2n)*kp2n*dt/at;(1-
gamma1n)*kp1n*dt/at 0];
B4=[b4+(b4-Bss)*uss*pinv(ub4)];

```

```

B=zeros(nx,nu,vt_B); % Arrey of matrix B
B(:,1)=B1; B(:,3)=B3;
B(:,2)=B2; B(:,4)=B4;

%%
QQ=[1 0 0 0;0 1 0 0;0 0 0 0;0 0 0 0]; % Weighting matrix
R=[1 0;0 1]; % Weighting matrix
x0=[60 15 12 10 8 6 4 2;60 15 12 10 8 6 4 2;60 15 12 10 8 6 4 2;60 15 12 10 8 6 4 2]; % Initial
state to solve pre-computed output feedback gain
[xr xc]=size(x0);
n=xc; % Number of initial state or number of pre-computed output feedback gain
k=zeros(2,4,n);
yy=zeros(2,4,n);
q=zeros(4,4,n);

%% Solve pre-computed output feedback gain

for i=1:n

x=x0(:,i);
ops=sdpssettings('solver','sedumi');
%Declare variable
X=sdpvar(nu,nu);
Y=sdpvar(nu,nx);
Q=sdpvar(nx,nx);
G=sdpvar(1,1);
%
I1=eye(nx); % state identity matrix
I2=eye(nu); % input identity matrix
%Robust stability constraint
C=[Q>0];
C=[C,[1 x';x Q]>=0]; % Invariant ellipsoid

for j=1:vt_A
for jj=1:vt_B
C=[C,[Q (Q*A(:,j)'+Y*B(:,jj)) Q*(QQ^0.5) Y*(R^0.5);(A(:,j)*Q+B(:,jj)*Y) Q zeros(nx,nx)
zeros(nx,nu);((Q^0.5)*Q) zeros(nx,nx) G*I1 zeros(nx,nu);((R^0.5)*Y) zeros(nu,nx) zeros(nu,nx)
G*I2]>=0];
end
end
end

```

```

% Input constraints
C=[C,[X Y;Y' Q]>=0];
C=[C,X(1,1)<=0.098^2];
C=[C,X(2,2)<=0.098^2];
% Solve
solvesdp(C,G,ops);
% Collect state feedback gain
yy(:,i)=double(Y);
q(:,i)=double(Q);
k(:,i)=double(Y)*inv(double(Q));
%
end

%% Check ability in iteration between ellipsoid
o=zeros(1,n-1);
r=zeros(vt_A,n-1);

for i=1:n-1
    for j=1:vt_A
        for jj=1:vt_B
            r(j,i)=all(eig(inv(q(:,i))-(A(:,j)+B(:,jj))*k(:,i+1))* inv(q(:,i))*(A(:,j)+B(:,jj))*k(:,i+1)))>0);
        end
    end
    if all(r(:,i))==1
        o(1,i)=1;
    else
        o(1,i)=0;
    end
    fprintf('ring %d & %d',i,i+1)
    if o(i)~=1
        disp('failed')
    else
        disp('pass')
    end
end
end

%% Design of estimator to find estimator gain(Lp)
c=eye(4); % c from y=cx
[cc,cr]=size(c);
ny=cc; % Number of output variable

```

```

    stop=1; % Initial command to start while loop
    r=0.999; % Initial r
while stop
    % Declare variable
    E=sdpvar(nx,ny);
    J=sdpvar(nx,nx);
    % Design estimator gain
    C=[];
    C=[C,0<r<1];
    C=[C,[(r^2)*J*Ass-E*c;Ass'*J-c'*E' J]>=0];
    solvesdp(C,[],ops);
    %
    Lp=inv(double(J))*double(E); % Estimator gain
    Cp=[];
    V=sdpvar(8,8); % Declare variable

    % Check stability for output feedback system
    for j=1:vt_A
        for jj=1:vt_B
            for kj=1:n
                Cp=[Cp,[V*V*[A(:,j) B(:,jj)*k(:,kj);Lp*c Ass+Bss*k(:,kj)-Lp*c]';[A(:,j) B(:,jj)*k(:,kj);Lp*c
Ass+Bss*k(:,kj)-Lp*c]*V V]>0];
            end
        end
    end
    solvesdp(Cp,[],ops);
    if all(eig(double(V))>0)
        stop=0;
        r;
    else
        % If test is not pass ,stop =1 , random new r by r(new)=r(old)-0.1
        stop=1;
        r=r-0.1;
    end
    %
end

% Save variable to controller_gain_minimumphase.mat
save('controller_gain_minimumphase','k','q','o','Lp','c','Ass','Bss','A','B','uss','vt_A','vt_B','h1s','h2s','h3s','h4s','dt')

```

```
% Save variable to controller_gain_non_minimumphase.mat
```

```
%save('controller_gain_non_minimumphase','k','q','Lp','c','Ass','Bss','A','B','uss','vt_A','vt_B','h1s','h2s','h3s','h4s','dt')
```

```
%
```

ในส่วนชุดคำสั่งในการควบคุมระบบสี่ถึงจริงจะแบ่งชุดคำสั่งออกเป็นสองส่วนคือ ชุดคำสั่งที่หนึ่ง จะทำหน้าที่เชื่อมต่อรับส่งข้อมูลกับอาร์ดูโน้ เพื่อรับค่าระดับน้ำในถังจากอุปกรณ์ตรวจวัดแรงดัน และรับค่าอัตราการไหลจากอุปกรณ์ตรวจวัดอัตราการไหล นอกจากนี้ภายในยังมีชุดคำสั่งของตัวควบคุมลูปรองซึ่งทำหน้าที่ในการควบคุมอัตราการไหลให้คงที่ ในส่วนชุดคำสั่งที่สอง จะทำหน้าที่คำนวณค่าอัตราขยายตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกและค่าตัวแปรสถานะจากการประมาณโดยใช้ค่าอัตราขยายตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกและค่าอัตราขยายของตัวประมาณค่าสถานะที่เก็บไว้ในฐานข้อมูล จากการคำนวณค่าอัตราขยายตัวควบคุมแบบป้อนกลับค่าสัญญาณขาออกสามารถคำนวณค่าตัวแปรปรับจากลูปรองควบคุมหลัก นำไปสู่การคำนวณค่าอัตราการไหลเป้าหมายเพื่อส่งเป็นค่าเป้าหมายไปยังลูปรองควบคุมอัตราการไหล

ในการรันชุดคำสั่งในการควบคุมจะทำการรันชุดคำสั่งที่หนึ่งและสองพร้อมกันโดยการเปิดโปรแกรมแมทแลปสองหน้าจอ ในส่วนการรับส่งข้อมูลตัวแปรระหว่างชุดคำสั่งทั้งสองจะทำการบันทึกค่าตัวแปรในรูปของไฟล์แมทแลป(.mat) เพื่อให้ชุดคำสั่งทั้งสองสามารถเชื่อมโยงค่าตัวแปรได้แม้จะเป็นการรันโปรแกรมแมทแลปสองหน้าจอก็ตาม

ข.6 ชุดคำสั่งที่หนึ่งในการควบคุมระดับน้ำในถังที่หนึ่งและถังที่สองในส่วนของควบคุมระบบสี่ถึงจริง

```
% Minimumphase
```

```
load('controller_gain_minimumphase','k','q','Lp','Ass','Bss','A','B','uss','c','vt_A','vt_B','h1s','h2s','h3s','h4s','dt','o') % Load variable from controller_gain_minimumphase.mat
```

```
% Non-minimumphase
```

```
load('controller_gain_non_minimumphase','k','q','Lp','Ass','Bss','A','B','uss','c','vt_A','vt_B','h1s','h2s','h3s','h4s','dt','o') % Load variable from controller_gain_non_minimumphase.mat
```

```
load
```

```
('controller_gain_minimumphase','k','q','Lp','Ass','Bss','A','B','uss','c','vt_A','vt_B','h1s','h2s','h3s','h4s','dt','o') % Load variable from controller_gain_minimumphase.mat
```

```
nu=2; % Number of input variable
```

```
nx=4; % Number of state variable
```

```
ny=4; % Number of output variable
```

```
ii=0;
```

```
load('Find_K_by_interpolation_run','run') % Number of input variable
```

```

while run
ii=ii+1
%% Recieve level of water in each from pressure sensor
% Random signal from arduino 5 sample
for j=1:5
h1_sig(j)=a.analogRead(3);
h2_sig(j)=a.analogRead(5);
h3_sig(j)=a.analogRead(2);
h4_sig(j)=a.analogRead(4);
end
% Convert arduino signal to level of water in each tank
h1=0.095*(median(h1_sig))-8.072; %(cm)
h2=0.097*(median(h2_sig))-11.79; %(cm)
h3=0.104*(median(h3_sig))-7.387; %(cm)
h4=0.085*(median(h4_sig))+1.038; %(cm)
%
x=[h1-h1s;h2-h2s;h3-h3s;h4-h4s] % State variable
save('state_variable','x'); % Save state variable to state_variable.mat

%% Find input variable of flow control
load('Find_K_by_interpolation','u1','u2'); % Load input variable from
Find_K_by_interpolation.mat
input1(ii)=u1;
input2(ii)=u2;
nu=2;
nx=2;
%Minimumphase case
f1ss=15.5; % Flowrate1 setpoint at steady state
f2ss=16; % Flowrate2 setpoint at steady state
%Non-minimumphase case
f1ss=15.5; % Flowrate1 setpoint at steady state
f2ss=16; % Flowrate2 setpoint at steady state

f1ss=15.5; % Flowrate1 setpoint at steady state
f2ss=16; % Flowrate2 setpoint at steady state

% Check deviation between input(ii+1) and input(ii)
% To decrease of computed time in case that input(ii+1) = input(ii) or input(ii+1)-input(ii)=0
if ii==1
ct1=1;
ct2=1;

```



```

else
    ct1=input1(ii)-input1(ii-1);
    ct2=input2(ii)-input2(ii-1);
end

% Recieve flowrate of water from flow meter
for j=1:5
    f1(j)=0.014*a.analogRead(11)-0.37; % Flow meter 1
    f2(j)=0.014*a.analogRead(8)-0.47; % Flow meter 2
    f3(j)=0.014*a.analogRead(9)-0.082; % Flow meter 3
    f4(j)=0.014*a.analogRead(10)-0.288; % Flow meter 4
end
%

load('flow_setpoint','f1s','f2s') % Load flow set point from flow_setpoint.mat

F=[median(f1)+median(f4)-f1s;median(f2)+median(f3)-f2s]; % Matrix of deviation flow rate (flow
rate-flow rate at steady state)
f(ii,1)=median(f1)+median(f4); % Flow rate of pump1
f(ii,2)=median(f2)+median(f3); % Flow rate of pump2
    % Find output for slave loop1
clear Kf1;
load ('controller_gain_14','k','q')
sk=size(k);
n=sk(3); % Number of pre-computed state feedback gain (slave loop1)
    F1=F(1);
if F1*inv(q(:,n))*F1<=1;
    Kf1=k(:,n);
elseif F1*inv(q(:,n-1))*F1<=1;
    Kf1=k(:,n-1);
elseif F1*inv(q(:,n-2))*F1<=1;
    Kf1=k(:,n-2);
elseif F1*inv(q(:,n-3))*F1<=1;
    Kf1=k(:,n-3);
elseif F1*inv(q(:,n-4))*F1<=1;
    Kf1=k(:,n-4);
elseif F1*inv(q(:,n-5))*F1<=1;
    Kf1=k(:,n-5);
elseif F1*inv(q(:,n-6))*F1<=1;
    Kf1=k(:,n-6);
end

```

```
U1=Kf1*F1; % Deviation input1 variable to control flow(F1)
```

```
% Find output for slave loop2
```

```
clear Kf2;
load ('controller_gain_23','k','q')
sk=size(k);
n=sk(3);
F2=F(2);
```

```
if F2*inv(q(:,n))*F2<=1;
```

```
    Kf2=k(:,n);
```

```
elseif F2*inv(q(:,n-1))*F2<=1;
```

```
    Kf2=k(:,n-1);
```

```
elseif F2*inv(q(:,n-2))*F2<=1;
```

```
    Kf2=k(:,n-2);
```

```
elseif F2*inv(q(:,n-3))*F2<=1;
```

```
    Kf2=k(:,n-3);
```

```
elseif F2*inv(q(:,n-4))*F2<=1;
```

```
    Kf2=k(:,n-4);
```

```
elseif F2*inv(q(:,n-5))*F2<=1;
```

```
    Kf2=k(:,n-5);
```

```
elseif F2*inv(q(:,n-6))*F2<=1;
```

```
    Kf2=k(:,n-6);
```

```
end
```

```
U2=Kf2*F2; % Deviation input2 variable to control flow(F2)
```

```
input_F(ii,1)=U1;
```

```
input_F(ii,2)=U2;
```

```
% Input pump1
```

```
if (ii==1)||(ct1~=0)
```

```
    u1C=U1+(u1+1.96);
```

```
else
```

```
    u1C=input_F(ii,1)-input_F(ii-1,1)+ u1C;
```

```
end
```

```
% Input pump1
```

```
if (ii==1)||(ct2~=0)
```

```
    u2C=U2+(u2+1.96);
```

```
else
```

```
    u2C=input_F(ii,2)-input_F(ii-1,2)+ u2C;
```

```
end
```

```

%
input_uC(ii,1)=u1C; % Input pump1(volt)
input_uC(ii,2)=u2C; % Input pump2(volt)
input_pump(ii,1)=int8(u1C*(255/5)); % Convert input pump1(volt) to arduino signal
input_pump(ii,2)=int8(u2C*(255/5)); % Convert input pump2(volt) to arduino signal

%% Send signal to control pump1 and pump2
a.analogWrite(7,input_pump(ii,1)); %Pump1
a.analogWrite(6,input_pump(ii,2)); %Pump2
%% collect input to control pump1 and pump2
if ii==1
input_signal_uC1=[];
input_signal_uC2=[];
else
input_signal_uC1=[input_signal_uC1;input_pump(ii,1)]; %pump1
input_signal_uC2=[input_signal_uC2;input_pump(ii,2)]; %pump2
end
%% collect signal to control pump1 and pump2
if ii==1
input_signal_pump1=[];
input_signal_pump2=[];
else
input_signal_pump1=[input_signal_pump1;round(u1C*(255/5))]; %pump1
input_signal_pump2=[input_signal_pump2;round(u2C*(255/5))]; %pump2
end
%% Collect flow rate set point
if ii==1
flow1=[];
flow2=[];
else
flow1=[flow1;f1s];
flow2=[flow2;f2s];
end
%% Collect flow rate
if ii==1
Flow1=[];
Flow2=[];
else
Flow1=[Flow1,f(ii,1)];
Flow2=[Flow2,f(ii,2)];
end
end

```

```

F
load('Find_K_by_interpolation_run','run') % Load run variable from
Find_K_by_interpolation_run.mat
end
%% Stop control when run==0
if run==0;
a.analogWrite(6,0); % Stop pump2
a.analogWrite(7,0); % Stop pump1
end

```

ข.7 ชุดคำสั่งที่สองในการควบคุมระดับน้ำในถังที่หนึ่งและถังที่สองในส่วนของ การควบคุมระบบสี่ถังจริงที่สถานะเฟสต่ำสุด โดยใช้ชุดคำสั่งการควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทนแบบป้อนกลับสัญญาณขาออกซึ่งไม่มีการประยุกต์ใช้การประมาณค่าอัตราขยายแบบป้อนกลับสัญญาณขาออก(Off-line robust constrained output feedback MPC)

```

TIME=1000; % 600s = 10min

load('controller_gain_minimumphase','k','q','Lp','Ass','Bss','A','B','uss','c','vt_A','vt_B','h1s','h2s','h3s','h4s','dt') % Load variable from controller_gain_minimumphase

l=TIME/dt; % Round of iteration
nu=2; % Number of input variable
nx=4; % Number of state variable
ny=nx; % Number of output variable
Ub=[0.098;0.098]; % Boundary of input variable

% Creat zero matrix to collect data
input=zeros(l,nu); % Input data
estate=zeros(l,nx); % Estimated state data
%
c=eye(4); % y=cx
[ck_ro,ck_col]=size(clock); % clock=[year month day hour minute seconds]
time=zeros(l,ck_col); %Creat zero matrix to collect time each a sampling time
run=1; % Initial command to start
save('Find_K_by_interpolation_run','run') % Save run=1 to Find_K_by_interpolation_run.mat
i=0;

while run
i=i+1
tt=clock; % Initial time duration sampling time

```

```

time(i,:)=tt % Collect time data
if i==1
    % Variable data at first iteration
    h1=13.3;
    h2=14.69;
    h3=5.09;
    h4=4.86;
    x=[h1-h1s;h2-h2s;h3-h3s;h4-h4s]; % Matrix of state variable at initial time
    xe=[(h1-h1s)*1.2;(h2-h2s)*0.8;(h3-h3s)*1.1;(h4-h4s)*0.9]; % Matrix of estimated state variable at
initial time
else
    load ('state_variable','x') % Receive state variable from ARDUINO
end

clear K;
clear al;

sk=size(k);
n=sk(3); % Number of pre-computed output feedback gains

%% Find output feedback gain from pre-computed state feedback gain

if xe'*inv(q(:,n))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n)]
    K=k(:,n);
elseif xe'*inv(q(:,n-1))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-1)]
    K=k(:,n-1);
elseif xe'*inv(q(:,n-2))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-2)]
    K=k(:,n-2);
elseif xe'*inv(q(:,n-3))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-3)]
    K=k(:,n-3);
elseif xe'*inv(q(:,n-4))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-4)]
    K=k(:,n-4);

elseif xe'*inv(q(:,n-5))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-5)]
    K=k(:,n-5);
elseif xe'*inv(q(:,n-6))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-6)]
    K=k(:,n-6);
elseif xe'*inv(q(:,n-7))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-7)]
    K=k(:,n-7);
end

```

```

%% Find input variables
u1=K(1,:)*xe; %Deviation of input1 from steady state (V)
u2=K(2,:)*xe; %Deviation of input2 from steady state (V)
%%
save('Find_K_by_interpolation','u1','u2') % Save input variables to Find_K_by_interpolation.mat

%% While loop to check sampling time (dt=10s)
st=1; % Initial command to start while loop
while st
    tf=clock; % Time duration sampling time ,
        % tt is initial time duration sampling time
        if (tf(4)*3600+tf(5)*60+tf(6))>=tt(4)*3600+tt(5)*60+tt(6)+dt % (Time duration sampling
time)-(Initial time duration sampling time) is 10 so sampling time=10
            run=1 % Start sample data at next iteration
            st=0 % Command to stop this while loop
        else
            st=1
        end
    end
end

%% Output variable collection
y=c*x;
for j=1:ny
    output(i,j)=y(j);
end

%% Estimated state variable collection (1 row is 1 iteration)
estate(i,1)=xe(1);
estate(i,2)=xe(2);
estate(i,3)=xe(3);
estate(i,4)=xe(4);

%% Input variable collection (1 row is 1 iteration)
input(i,1)=u1;
input(i,2)=u2;

%% Find estimated state at i+1
y=c*x;
xe=Ass*xe+Bss*K*xe+Lp*(y-c*x); %xe(i+1)
%% Convert input signal (0-5V) to computer signal (0-255)
input1_signal=(u1+1.96)*(255/5);
input2_signal=(u2+1.96)*(255/5);

```

```

%% Find flow set point of slave loop at 1
if input1_signal==95
    f1s=10.1;
elseif input1_signal<=96
    % At input1_signal=96 flow rate of pump1 =11.2
    rt=input1_signal-95;
    f1s=rt*(11.2-10.1)+10.1;
elseif input1_signal<=97
    % At input1_signal=97 flow rate of pump1 =12.5
    rt=input1_signal-96;
    f1s=rt*(12.5-11.2)+11.2;
elseif input1_signal<=98
    % At input1_signal=98 flow rate of pump1 =13.5
    rt=input1_signal-97;
    f1s=rt*(13.5-12.5)+12.5;
elseif input1_signal<=99
    % At input1_signal=99 flow rate of pump1 =14.5
    rt=input1_signal-98;
    f1s=rt*(14.5-13.5)+13.5;
elseif input1_signal<=100
    % At input1_signal=100 flow rate of pump1 =15.5
    rt=input1_signal-99;
    f1s=rt*(15.5-14.5)+14.5;
elseif input1_signal<=101
    % At input1_signal=101 flow rate of pump1 =16.5
    rt=input1_signal-100;
    f1s=rt*(16.5-15.5)+15.5;
elseif input1_signal<=102
    % at input1_signal=101 flow rate of pump1 =17.5
    rt=input1_signal-101;
    f1s=rt*(17.5-16.5)+16.5;
elseif input1_signal<=103
    % At input1_signal=103 flow rate of pump1 =18.5
    rt=input1_signal-102;
    f1s=rt*(18.5-17.5)+17.5;
elseif input1_signal<=104
    % At input1_signal=104 flow rate of pump1 =19.5
    rt=input1_signal-103;
    f1s=rt*(19.5-18.5)+18.5;
else

```

```

    % At input1_signal=104 flow rate of pump1 =20.5
    rt=input1_signal-104;
    f1s=rt*(20.5-19.5)+19.5;
end

```

```

%% Find flow set point of slave loop at 2

```

```

if input2_signal<=95
    f2s=10.5;
elseif input2_signal<=96
    % At input1_signal=96 flow rate of pump1 =11.4
    rt=input2_signal-95;
    f2s=rt*(11.4-10.5)+10.5;
elseif input2_signal<=97
    % At input1_signal=97 flow rate of pump1 =12.6
    rt=input2_signal-96;
    f2s=rt*(12.6-11.4)+11.4;
elseif input2_signal<=98
    % At input1_signal=98 flow rate of pump1 =14
    rt=input2_signal-97;
    f2s=rt*(14-12.6)+12.6;
elseif input2_signal<=99
    % At input1_signal=99 flow rate of pump1 =14.9
    rt=input2_signal-98;
    f2s=rt*(14.9-14)+14;
elseif input2_signal<=100
    % At input1_signal=100 flow rate of pump1 =16
    rt=input2_signal-99;
    f2s=rt*(16-14.9)+14.9;
elseif input2_signal<=101
    % At input1_signal=101 flow rate of pump1 =16.9
    rt=input2_signal-100;
    f2s=rt*(16.9-16)+16;
elseif input2_signal<=102
    % At input1_signal=102 flow rate of pump1 =17.9
    rt=input2_signal-101;
    f2s=rt*(17.9-16.9)+16.9;
elseif input2_signal<=103
    % At input1_signal=103 flow rate of pump1 =18.7
    rt=input2_signal-102;
    f2s=rt*(18.7-17.9)+17.9;
elseif input2_signal<=104

```



```

    % At input1_signal=104 flow rate of pump1 =19.6
    rt=input2_signal-103;
    f2s=rt*(19.6-18.7)+18.7;
else
    % At input1_signal=105 flow rate of pump1 =20.5
    rt=input2_signal-104;
    f2s=rt*(20.5-19.6)+19.6;
end
%%
save('flow_setpoint','f1s','f2s') % save flow setpoint to flow_setpoint.mat
f1s
f2s
%% Stop calculation when i=number round of iteration
    if i==l
        run=0;
    end
%%
save('Find_K_by_interpolation_run','run'); % Stop calculation when i=number round of iteration
end
%Save variables to excel file to plot graphs with SigmaPlot program
xlswrite('minimumphase.xls',input(1:35,1),1,'B5')
xlswrite('minimumphase.xls',input(1:35,2),1,'C5')
xlswrite('minimumphase.xls',output(1:35,1),1,'D5')
xlswrite('minimumphase.xls',output(1:35,2),1,'E5')
xlswrite('minimumphase.xls',output(1:35,3),1,'F5')
xlswrite('minimumphase.xls',output(1:35,4),1,'G5')
xlswrite('minimumphase.xls',estate(1:35,1),1,'H5')
xlswrite('minimumphase.xls',estate(1:35,2),1,'I5')
xlswrite('minimumphase.xls',estate(1:35,3),1,'J5')
xlswrite('minimumphase.xls',estate(1:35,4),1,'K5')
%
```

ข.8 ชุดคำสั่งที่สองในการควบคุมระดับน้ำในถังที่หนึ่งและถังที่สองในส่วนของการควบคุมระบบสี่ถังจริงที่สถานะเฟสต่ำสุด โดยใช้ชุดคำสั่งการควบคุมตามที่งานวิจัยเสนอซึ่งมีการประยุกต์ใช้การประมาณค่าอัตราขยายแบบป้อนกลับสัญญาณขาออก(Off-line robust constrained output feedback MPC with output feedback gain interpolation)

```
TIME=1000; % 600s = 10min
```

```
load('controller_gain_minimumphase','k','q','Lp','Ass','Bss','A','B','uss','c','vt_A','vt_B','h1s','h2s','h3s','h4s','dt','o')% Load variable from controller_gain_minimumphase.mat
```

```

l=TIME/dt; % Round of iteration
nu=2; % Number of input variable
nx=4; % Number of state variable
ny=nx; % Number of output variable
Ub=[0.098;0.098]; % Boundary of input variable

% Creat zero matrix to collect data
input=zeros(l,nu); % Input data
estate=zeros(l,nx); % Estimated state data
%
c=eye(4); % y=cx
[ck_ro,ck_col]=size(clock); %clock=[year month day hour minute seconds]
time=zeros(l,ck_col); %matrix of time to collect time each sampling time
AL=zeros(l,1); % Create zero matrix to collect al

%% find K and Lp
run=1; % Initial command to start
save('Find_K_by_interpolation_run','run') % Save run=1 to Find_K_by_interpolation_run.mat
i=0;

while run
i=i+1
    tt=clock; % Initial time duration sampling time
    time(i,:)=tt % Collect time data
    if i==1
        % Variable data at first iteration
        h1=13.3;
        h2=14.69;
        h3=5.09;
        h4=4.86;
        x=[h1-h1s;h2-h2s;h3-h3s;h4-h4s]; % Matrix of state variable at initial time
        xe=[(h1-h1s)*1.2;(h2-h2s)*0.8;(h3-h3s)*1.1;(h4-h4s)*0.9]; % Matrix of estimated state variable at
initial time
    else
        load ('state_variable','x') % Receive state variable from ARDUINO
    end
    clear K;
    clear al;
    sk=size(k);
    n=sk(3); % Number of pre-computed output feedback gains

```

```
%% Find output feedback gain from pre-computed state feedback gain with linear interpolation
```

```
if i==1 % At first iteration
    % Variable value at first iteration
    u1=0.0994;
    u2=0.0917;
    al=0.0495;
    AL(i,1)=al;
    K=[-0.0008 -0.0002 -0.0017 -0.0028;-0.0005 -0.0004 -0.0035 -0.0013]; % Initial k at
iteration 1
else
    if xe*inv(q(:,n))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n)]
        K=k(:,n);
        al=1;
        AL(i,1)=al;
    elseif xe*inv(q(:,n-1))*xe<=1; % Estimated state is member of ellipsoid invariant set
[q(:,n-1)]
        if o(7)==1 % Check ability in iteration between ellipsoid
            % Declare variables
            al=sdpvar(1,1);
            K=al*k(:,n-1)+(1-al)*k(:,n);
            %
            w=[];
            % Constraint of al
            w=[w,al>=0];
            w=[w,al<=1];
            %
            % Constraint to guarantee robust stability
            for j=1:vt_A
                for jj=1:vt_B
                    w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-1)]>=0];
                end
            end
            %
            % Input constraint
            w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
            w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
            %
            solvesdp(w,al); % Solve al
            al=double(al);
            AL(i,1)=al; % Collect al to matrix AL
```

```

K=al*k(:,n-1)+(1-al)*k(:,n); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-1);
    al=1;
end

elseif xe*inv(q(:,n-2))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-2)]

if o(6)==1 % Check ability in iteration between ellipsoid
    % Declare variables
    al=sdpvar(1,1);
    K=al*k(:,n-2)+(1-al)*k(:,n-1);
    %
    % Constraint of al
    w=[w,al>=0];
    w=[w,al<=1];
    %
    % Constraint to guarantee robust stability
    for j=1:vt_A
        for jj=1:vt_B
            w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe'); (A(:,j)+B(:,jj))*K)*xe q(:,n-2)]>=0];
        end
    end
    %
    % Input constraint
    w=[w,[Ub(1)^2 (K(1,:)*xe');K(1,:)*xe 1]>=0];
    w=[w,[Ub(2)^2 (K(2,:)*xe');K(2,:)*xe 1]>=0];
    %
    solvesdp(w,al); % Solve
    al=double(al);
    AL(i,1)=al; % Collect al to matrix AL
    K=al*k(:,n-2)+(1-al)*k(:,n-1); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-2);
    al=1;
end

elseif xe*inv(q(:,n-3))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-3)]
if o(5)==1
    % Declare variable

```

```

al=sdpvar(1,1);
K=al*k(:,n-3)+(1-al)*k(:,n-2);
%
% Constraint of al
w=[w,al>=0];
w=[w,al<=1];
%
% Constraint to guarantee robust stability
for j=1:vt_A
    for jj=1:vt_B
        w=[w,[1 ((A(:,j)+B(:,jj)*K)*xe)';(A(:,j)+B(:,jj)*K)*xe q(:,n-3)]>=0];
    end
end
% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe);K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe);K(2,:)*xe 1]>=0];
%
solvesdp(w,al); % Solve
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-3)+(1-al)*k(:,n-2); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-3);
    al=1;
end
elseif xe*inv(q(:,n-4))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-4)]
if o(4)==1 % Check ability in iteration between ellipsoid
    % Declare variables
    al=sdpvar(1,1);
    K=al*k(:,n-4)+(1-al)*k(:,n-3);
    %
    w=[];
    % Constraint of al
    w=[w,al>=0];
    w=[w,al<=1];

    % Constraint to guarantee robust stability

    for j=1:vt_A
        for jj=1:vt_B

```

```

        w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-4)]>=0];
    end
end
% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
%
solvesdp(w,al); % Solve al
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-4)+(1-al)*k(:,n-3); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-4);
    al=1;
end
elseif xe*inv(q(:,n-5))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-5)]
    if o(3)==1 % Check ability in iteration between ellipsoid
        % Declare variables
        al=sdpvar(1,1);
        K=al*k(:,n-5)+(1-al)*k(:,n-4);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        %
        % Constraint to guarantee robust stability
        for j=1:vt_A
            for jj=1:vt_B
                w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-5)]>=0];
            end
        end
        %
        % Input constraint
        w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
        w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
        %
        solvesdp(w,al); % Solve al
        al=double(al);

```

```

AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-5)+(1-al)*k(:,n-4); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-5);
    al=1;
end
elseif xe*inv(q(:,n-6))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-6)]
    if o(2)==1 % Check ability in iteration between ellipsoid
        % Declare variables
        al=sdpvar(1,1);
        K=al*k(:,n-6)+(1-al)*k(:,n-5);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        %
        % Constraint to guarantee robust stability
        for j=1:vt_A
            for jj=1:vt_B
                w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-6)]>=0];
            end
        end
        %
        % Input constraint
        w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
        w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
        %
        solvesdp(w,al); % Solve al
        al=double(al);
        AL(i,1)=al; % Collect al to matrix AL
        K=al*k(:,n-6)+(1-al)*k(:,n-5); % Output feedback gain
    else
        %No ability in iteration between ellipsoid
        K=k(:,n-6);
        al=1;
    end
elseif xe*inv(q(:,n-7))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-7)]
    if o(1)==1 % Check ability in iteration between ellipsoid

```

```

% Declare variables
al=sdpvar(1,1);
K=al*k(:,n-7)+(1-al)*k(:,n-6);
%
w=[];
% Constraint of al
w=[w,al>=0];
w=[w,al<=1];
%
% Constraint to guarantee robust stability
for j=1:vt_A
    for jj=1:vt_B
        w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-7)]>=0];
    end
end
%
% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
%
solvesdp(w,al); % Solve al
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-7)+(1-al)*k(:,n-6); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-7);
    al=1;
end
end

u1=K(1,:)*xe; %Deviation of input1 from steady state (V)
u2=K(2,:)*xe; %Deviation of input2 from steady state (V)

end

save('Find_K_by_interpolation','u1','u2')

%% While loop to check sampling time (dt=10s)
st=1; % Initial command to start while loop

```



```

while st
tf=clock; % Time duration sampling time ,
    % tt is initial time duration sampling time
if (tf(4)*3600+tf(5)*60+tf(6))>=tt(4)*3600+tt(5)*60+tt(6)+dt % (Time duration sampling time)-(Initial
time duration sampling time) is 10 so sampling time=10
    run=1 % Start sample data at next iteration
    st=0 % Command to stop this while loop
else
    st=1
end
end

%% output variable collection
y=c*x;
for j=1:ny
    output(i,j)=y(j);
end

%% Estimated state variable collection(1 row is 1 iteration)
estate(i,1)=xe(1);
estate(i,2)=xe(2);
estate(i,3)=xe(3);
estate(i,4)=xe(4);

%% input variable collection(1 row is 1 iteration)
input(i,1)=u1;
input(i,2)=u2;

%% Find estimated state at i+1
y=c*x;
xe=Ass*x+Bs*s*K*x+Lp*(y-c*x); %xe(i+1)

%% Convert input signal (0-5V) to computer signal(0-255)
input1_signal=(u1+1.96)*(255/5);
input2_signal=(u2+1.96)*(255/5);

%% Find flow set point of slave loop at 1
if input1_signal==95
    f1s=10.1;
elseif input1_signal<=96
    % At input1_signal=96 flow rate of pump1 =11.2
    rt=input1_signal-95;
    f1s=rt*(11.2-10.1)+10.1;
elseif input1_signal<=97
    % At input1_signal=97 flow rate of pump1 =12.5

```

```

rt=input1_signal-96;
f1s=rt*(12.5-11.2)+11.2;
elseif input1_signal<=98
    % At input1_signal=98 flow rate of pump1 =13.5
    rt=input1_signal-97;
    f1s=rt*(13.5-12.5)+12.5;
elseif input1_signal<=99
    % At input1_signal=99 flow rate of pump1 =14.5
    rt=input1_signal-98;
    f1s=rt*(14.5-13.5)+13.5;
elseif input1_signal<=100
    % At input1_signal=100 flow rate of pump1 =15.5
    rt=input1_signal-99;
    f1s=rt*(15.5-14.5)+14.5;
elseif input1_signal<=101
    % At input1_signal=101 flow rate of pump1 =16.5
    rt=input1_signal-100;
    f1s=rt*(16.5-15.5)+15.5;
elseif input1_signal<=102
    % at input1_signal=101 flow rate of pump1 =17.5
    rt=input1_signal-101;
    f1s=rt*(17.5-16.5)+16.5;
elseif input1_signal<=103
    % At input1_signal=103 flow rate of pump1 =18.5
    rt=input1_signal-102;
    f1s=rt*(18.5-17.5)+17.5;
elseif input1_signal<=104
    % At input1_signal=104 flow rate of pump1 =19.5
    rt=input1_signal-103;
    f1s=rt*(19.5-18.5)+18.5;
else
    % At input1_signal=104 flow rate of pump1 =20.5
    rt=input1_signal-104;
    f1s=rt*(20.5-19.5)+19.5;
end

```

```

%% Find flow set point of slave loop at 2

```

```

if input2_signal<=95
    f2s=10.5;
elseif input2_signal<=96

```

```

% At input2_signal=96 flow rate of pump2 =11.4
    rt=input2_signal-95;
    f2s=rt*(11.4-10.5)+10.5;
elseif input2_signal<=97
% At input2_signal=97 flow rate of pump2 =12.6
    rt=input2_signal-96;
    f2s=rt*(12.6-11.4)+11.4;
elseif input2_signal<=98
% At input2_signal=98 flow rate of pump2 =14
    rt=input2_signal-97;
    f2s=rt*(14-12.6)+12.6;
elseif input2_signal<=99
% At input2_signal=99 flow rate of pump2 =14.9
    rt=input2_signal-98;
    f2s=rt*(14.9-14)+14;
elseif input2_signal<=100
% At input2_signal=100 flow rate of pump2 =16
    rt=input2_signal-99;
    f2s=rt*(16-14.9)+14.9;
elseif input2_signal<=101
% At input2_signal=101 flow rate of pump2 =16.9
    rt=input2_signal-100;
    f2s=rt*(16.9-16)+16;
elseif input2_signal<=102
% At input2_signal=102 flow rate of pump2 =17.9
    rt=input2_signal-101;
    f2s=rt*(17.9-16.9)+16.9;
elseif input2_signal<=103
% At input2_signal=103 flow rate of pump2 =18.7
    rt=input2_signal-102;
    f2s=rt*(18.7-17.9)+17.9;
elseif input2_signal<=104
% At input2_signal=104 flow rate of pump2 =19.6
    rt=input2_signal-103;
    f2s=rt*(19.6-18.7)+18.7;
else
% At input2_signal=105 flow rate of pump2 =20.5
    rt=input2_signal-104;
    f2s=rt*(20.5-19.6)+19.6;
end

```

```

%%
save('flow_setpoint','f1s','f2s') % save flow set point to flow_setpoint.mat
f1s
f2s
%% Stop calculation when i=number round of iteration
    if i==l
        run=0;
    end

%%
save('Find_K_by_interpolation_run','run'); % Stop calculation when i=number round of iteration
%Save variables to excel file to plot graphs with SigmaPlot program
xlswrite('minimumphase.xls',input(1:35,1),1,'B48')
xlswrite('minimumphase.xls',input(1:35,2),1,'C48')
xlswrite('minimumphase.xls',output(1:35,1),1,'D48')
xlswrite('minimumphase.xls',output(1:35,2),1,'E48')
xlswrite('minimumphase.xls',output(1:35,3),1,'F48')
xlswrite('minimumphase.xls',output(1:35,4),1,'G48')
xlswrite('minimumphase.xls',estate(1:35,1),1,'H48')
xlswrite('minimumphase.xls',estate(1:35,2),1,'I48')
xlswrite('minimumphase.xls',estate(1:35,3),1,'J48')
xlswrite('minimumphase.xls',estate(1:35,4),1,'K48')
%
end

```

ข.9 ชุดคำสั่งที่สองในการควบคุมระดับน้ำในถังที่หนึ่งและถังที่สองในส่วนของ การควบคุมระบบสี่ถึงจริงที่สถานะไม่มีเฟสต่ำสุด โดยใช้ชุดคำสั่งการควบคุมเชิงทำนายแบบจำลองที่มีเงื่อนไขบังคับคงทนแบบป้อนกลับสัญญาณขาออกซึ่งไม่มีการประยุกต์ใช้การประมาณค่าอัตราขยายแบบป้อนกลับสัญญาณขาออก (Off-line robust constrained output feedback MPC)

```

TIME=720; % 600s = 10min
load
('controller_gain_non_minimumphase','k','q','Lp','Ass','Bss','A','B','uss','c','vt_A','vt_B','h1s','h2s','h3s','h4s','dt') % Load variable from controller_gain_non_minimumphase.mat

```

```

l=TIME/dt; % Round of iteration
nu=2; % Number of input variable
nx=4; % Number of state variable
ny=nx; % Number of output variable
Ub=[0.098;0.098]; % Boundary of input variable

```

```

% Creat zero matrix to collect data
input=zeros(l,nu); % Input data
estate=zeros(l,nx); % Estimated state data
%
c=eye(4); % y=cx
[ck_rol,ck_col]=size(clock); % clock=[year month day hour minute seconds]
time=zeros(l,ck_col); %Creat zero matrix to collect time each a sampling time
run=1; % Initial command to start
save('Find_K_by_interpolation_run','run') % Save run=1 to Find_K_by_interpolation_run.mat

i=0;
while run
i=i+1
    tt=clock; % Initial time duration sampling time
    time(i,:)=tt % Collect time data
    if i==1
        % Variable data at first iteration
        h1=9.03;
        h2=9.26;
        h3=20.9;
        h4=24.33;
        x=[h1-h1s;h2-h2s;h3-h3s;h4-h4s]; % Matrix of state variable at initial time
        xe=[(h1-h1s)*1.2;(h2-h2s)*0.8;(h3-h3s)*1.1;(h4-h4s)*0.9]; % Matrix of estimated state variable at
initial time
    else
        load ('state_variable','x') % Recieve state variable from ARDUINO
    end

clear K;
clear al;
sk=size(k);
n=sk(3); % Number of pre-computed output feedback gains

if i==1
    u1=-0.0068; %Deviation of input1 from steady state (V)
    u2=0.0081;
    K=[-0.0017  0.0051  -0.0035  -0.0069;0.0012  -0.0002  -0.0266  -0.0043]*10^(-3);
else
if  xe*inv(q(:,:,n))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,:,n)]
    K=k(:,:,n);
elseif  xe*inv(q(:,:,n-1))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,:,n-1)]

```

```

    K=k(:,n-1);
elseif xe*inv(q(:,n-2))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-2)]
    K=k(:,n-2);
elseif xe*inv(q(:,n-3))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-3)]
    K=k(:,n-3);
elseif xe*inv(q(:,n-4))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-4)]
    K=k(:,n-4);
elseif xe*inv(q(:,n-5))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-5)]
    K=k(:,n-5);
elseif xe*inv(q(:,n-6))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-6)]
    K=k(:,n-6);
elseif xe*inv(q(:,n-7))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-7)]
    K=k(:,n-7);
elseif xe*inv(q(:,n-8))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-8)]
    K=k(:,n-8);
end
end

%% Find input variables
u1=K(1,)*xe; %Deviation of input1 from steady state (V)
u2=K(2,)*xe; %Deviation of input2 from steady state (V)
%%
save('Find_K_by_interpolation','u1','u2') % Save input variables to Find_K_by_interpolation.mat
%% While loop to check sampling time (dt=12s)
st=1; % Initial command to start while loop
while st
    tf=clock; % Time duration sampling time ,
        % tt is initial time duration sampling time
if (tf(4)*3600+tf(5)*60+tf(6))>=tt(4)*3600+tt(5)*60+tt(6)+dt % (Time duration sampling time)-(Initial
time duration sampling time) is 12 so sampling time=12
    run=1 % Start sample data at next iteration
    st=0 % Command to stop this while loop
else
    st=1
end
end
end
%% output variable collection
y=c*x;
for j=1:ny
    output(i,j)=y(j);
end

```

```

%% Estimated state variable collection (1 row is 1 iteration)
estate(i,1)=xe(1);
estate(i,2)=xe(2);
estate(i,3)=xe(3);
estate(i,4)=xe(4);
%% Input variable collection (1 row is 1 iteration)
input(i,1)=u1;
input(i,2)=u2;
%% Find estimated state at i+1
y=c*x;
xe=Ass*x+Bs*s*K*x+Lp*(y-c*x); %xe(i+1)
%% Convert input signal(0-5V) to computer signal(0-255)
input1_signal=(u1+1.96)*(255/5);
input2_signal=(u2+1.96)*(255/5);
% find flow14 set point
if input1_signal==95
    f1s=9.69;
elseif input1_signal<=96
    % At input1_signal=96 flow rate of pump1 =9.86
    rt=input1_signal-95;
    f1s=rt*(9.86-9.69)+9.69;
elseif input1_signal<=97
    % At input1_signal=97 flow rate of pump1 =10.28
    rt=input1_signal-96;
    f1s=rt*(10.28-9.86)+9.86;
elseif input1_signal<=98
    % At input1_signal=98 flow rate of pump1 =10.8
    rt=input1_signal-97;
    f1s=rt*(10.8-10.28)+10.28;
elseif input1_signal<=99
    % At input1_signal=99 flow rate of pump1 =11.37
    rt=input1_signal-98;
    f1s=rt*(11.37-10.8)+10.8;
elseif input1_signal<=100
    % At input1_signal=100 flow rate of pump1 =11.79
    rt=input1_signal-99;
    f1s=rt*(11.79-11.37)+11.37;
elseif input1_signal<=101
    % At input1_signal=101 flow rate of pump1 =12.12
    rt=input1_signal-100;
    f1s=rt*(12.12-11.79)+11.79;

```

```

elseif input1_signal<=102
    % At input1_signal=102 flow rate of pump1 =12.63
    rt=input1_signal-101;
    f1s=rt*(12.63-12.12)+12.12;
elseif input1_signal<=103
    % At input1_signal=103 flow rate of pump1 =12.93
    rt=input1_signal-102;
    f1s=rt*(12.93-12.12)+12.12;
elseif input1_signal<=104
    % At input1_signal=104 flow rate of pump1 =13.42
    rt=input1_signal-103;
    f1s=rt*(13.42-12.93)+12.93;
else
    % At input1_signal=104 flow rate of pump1 =13.93
    rt=input1_signal-104;
    f1s=rt*(13.93-13.42)+13.42;
end

% find flow23 set point
if input2_signal<=95
    f2s=6.75;
elseif input2_signal<=96
    % At input2_signal=96 flow rate of pump2 =7.62
    rt=input2_signal-95;
    f2s=rt*(7.62-6.75)+6.75;
elseif input2_signal<=97
    % At input2_signal=97 flow rate of pump2 =8.42
    rt=input2_signal-96;
    f2s=rt*(8.42-7.62)+7.62;
elseif input2_signal<=98
    % At input2_signal=98 flow rate of pump2 =9.27
    rt=input2_signal-97;
    f2s=rt*(9.27-8.42)+8.42;
elseif input2_signal<=99
    % At input2_signal=99 flow rate of pump2 =9.91
    rt=input2_signal-98;
    f2s=rt*(9.91-9.27)+9.27;
elseif input2_signal<=100
    % At input2_signal=100 flow rate of pump2 =10.66
    rt=input2_signal-99;
    f2s=rt*(10.66-9.91)+9.91;

```



```

elseif input2_signal<=101
    % At input2_signal=101 flow rate of pump2 =11.36
    rt=input2_signal-100;
    f2s=rt*(11.36-10.66)+10.66;
elseif input2_signal<=102
    % At input2_signal=102 flow rate of pump2 =11.98
    rt=input2_signal-101;
    f2s=rt*(11.98-11.36)+11.36;
elseif input2_signal<=103
    % At input2_signal=103 flow rate of pump2 =12.66
    rt=input2_signal-102;
    f2s=rt*(12.66-11.98)+11.98;
elseif input2_signal<=104
    % At input2_signal=104 flow rate of pump2 =13.3
    rt=input2_signal-103;
    f2s=rt*(13.3-12.66)+12.66;
else
    % At input2_signal=104 flow rate of pump2 =13.91
    rt=input2_signal-104;
    f2s=rt*(13.91-13.3)+13.3;
end
end
%%%
save('flow_setpoint','f1s','f2s') % save flow setpoint to flow_setpoint.mat
f1s
f2s
%%% Stop calculation when i=number round of iteration
if i==l
    run=0;
end
%%%
save('Find_K_by_interpolation_run','run'); % Stop calculation when i=number round of iteration
end
%Save variables to excel file to plot graphs with SigmaPlot program
xlswrite('non_minimumphase.xls',input(1:35,1),1,'B5')
xlswrite('non_minimumphase.xls',input(1:35,2),1,'C5')
xlswrite('non_minimumphase.xls',output(1:35,1),1,'D5')
xlswrite('non_minimumphase.xls',output(1:35,2),1,'E5')
xlswrite('non_minimumphase.xls',output(1:35,3),1,'F5')
xlswrite('non_minimumphase.xls',output(1:35,4),1,'G5')
xlswrite('non_minimumphase.xls',estate(1:35,1),1,'H5')
xlswrite('non_minimumphase.xls',estate(1:35,2),1,'I5')

```

```

xlswrite('non_minimumphase.xls',estate(1:35,3),1,'J5')
xlswrite('non_minimumphase.xls',estate(1:35,4),1,'K5')
%
```

ช.10 ชุดคำสั่งที่สองในการควบคุมระดับน้ำในถังที่หนึ่งและถังที่สองในส่วนของ การควบคุมระบบสี่ถังจริงที่สถานะไม่มีเฟสต่ำสุด โดยใช้ชุดคำสั่งการควบคุมตามที่งานวิจัยเสนอซึ่งมีการประยุกต์ใช้การประมาณค่าอัตราขยายแบบป้อนกลับสัญญาณขาออก(Off-line robust constrained output feedback MPC with output feedback gain interpolation)

```

TIME=960;
load
('controller_gain_non_minimumphase','k','q','Lp','Ass','Bss','A','B','uss','c','vt_A','vt_B','h1s','h2s','h3s','h4s','dt','o') % Load variable from controller_gain_non_minimumphase.mat

l=TIME/dt; % Round of iteration
nu=2; % Number of input variable
nx=4; % Number of state variable
ny=nx; % Number of output variable
Ub=[0.098;0.098]; % Boundary of input variable

% Creat zero matrix to collect data
input=zeros(l,nu); % Input data
estate=zeros(l,nx); % Estimated state data
%

c=eye(4); % y=cx
[ck_rol,ck_col]=size(clock); %clock=[year month day hour minute seconds]
time=zeros(l,ck_col); %matrix of time to collect time each sampling time
AL=zeros(l,1); % Creat zero matrix to collect al
run=1; % Initial command to start
save('Find_K_by_interpolation_run','run') % Save run=1 to Find_K_by_interpolation_run.mat
i=0;
while run
i=i+1
tt=clock; % Initial time duration sampling time
time(i,:)=tt % Collect time data
if i==1
% initail measure at 30 second (send signal 100 to pump)
%1.05 min
h1=8.93;
h2=12.46;
```

```

h3=20.07;
h4=16.34;
x=[h1-h1s;h2-h2s;h3-h3s;h4-h4s];
xe=[(h1-h1s)*1.2;(h2-h2s)*0.8;(h3-h3s)*1.1;(h4-h4s)*0.9];
else
    load ('state_variable','x')
end

clear K;
clear al;
sk=size(k);
n=sk(3);

if i==1
    u1=0.0351; %Deviation of input1 from steady state (V)
    u2=0.0632;
    al=4.9089e-013;
    AL(i,1)=al;
    K=[-0.0002  0.0009  0.0006  -0.0027;0.0008  0.0006  -0.0030  -0.0010];
else
if xe*inv(q(:,n))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n)]
    K=k(:,n);
    al=1;
    AL(i,1)=al;
elseif xe*inv(q(:,n-1))*xe<=1; % Estimated state is member of ellipsoid invariant set [q(:,n-1)]
    if o(8)==1 % Check ability in iteration between ellipsoid
        % Declare variables
        al=sdpvar(1,1);
        K=al*k(:,n-1)+(1-al)*k(:,n);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        % Constraint to guarantee robust stability
        for j=1:vt_A
            for jj=1:vt_B
                w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-1)]>=0];
            end
        end
    end
end

```

```

% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe);K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe);K(2,:)*xe 1]>=0];
%
solvesdp(w,al); % Solve al
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-1)+(1-al)*k(:,n); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-1);
    al=1;
end
elseif xe*inv(q(:,n-2))*xe<=1;
if o(8)==1 % Check ability in iteration between ellipsoid
    % Declare variables
    al=sdpvar(1,1);
    K=al*k(:,n-2)+(1-al)*k(:,n-1);
    %
    w=[];
    % Constraint of al
    w=[w,al>=0];
    w=[w,al<=1];
    % Constraint to guarantee robust stability
    for j=1:vt_A
        for jj=1:vt_B
            w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-2)]>=0];
        end
    end
% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe);K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe);K(2,:)*xe 1]>=0];
%
solvesdp(w,al); % Solve al
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-2)+(1-al)*k(:,n-1); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-2);
    al=1;

```

```

end
elseif xe*inv(q(:,n-3))*xe<=1;
    if o(6)==1 % Check ability in iteration between ellipsoid
        % Declare variables
        al=sdpvar(1,1);
        K=al*k(:,n-3)+(1-al)*k(:,n-2);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        % Constraint to guarantee robust stability
        for j=1:vt_A
            for jj=1:vt_B
                w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-3)]>=0];
            end
        end
        % Input constraint
        w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
        w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
        %
        solvesdp(w,al); % Solve al
        al=double(al);
        AL(i,1)=al; % Collect al to matrix AL
        K=al*k(:,n-3)+(1-al)*k(:,n-2); % Output feedback gain
    else
        %No ability in iteration between ellipsoid
        K=k(:,n-3);
        al=1;
    end
elseif xe*inv(q(:,n-4))*xe<=1;
    if o(5)==1 % Check ability in iteration between ellipsoid
        % Declare variables
        al=sdpvar(1,1);
        K=al*k(:,n-4)+(1-al)*k(:,n-3);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        %

```

```

% Constraint to guarantee robust stability
for j=1:vt_A
    for jj=1:vt_B
        w=[w,[1 ((A(:,j)+B(:,jj)*K)*xe)';(A(:,j)+B(:,jj)*K)*xe q(:,n-4)]>=0];
    end
end
%
% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
%
solvesdp(w,al); % Solve al
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-4)+(1-al)*k(:,n-3); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-4);
    al=1;
end
elseif xe*inv(q(:,n-5))*xe<=1;
if o(4)==1 % Check ability in iteration between ellipsoid
    % Declare variable
    al=sdpvar(1,1);
    K=al*k(:,n-5)+(1-al)*k(:,n-4);
    %
    w=[];
    % Constraint of al
    w=[w,al>=0];
    w=[w,al<=1];
    % Constraint to guarantee robust stability
    for j=1:vt_A
        for jj=1:vt_B
            w=[w,[1 ((A(:,j)+B(:,jj)*K)*xe)';(A(:,j)+B(:,jj)*K)*xe q(:,n-5)]>=0];
        end
    end
    %
% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];

```

```

%
solvedp(w,al); % Solve al
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-5)+(1-al)*k(:,n-4); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-5);
    al=1;
end
elseif xe*inv(q(:,n-6))*xe<=1;
    if o(3)==1 % Check ability in iteration between ellipsoid
        % Declare variables
        al=sdpvar(1,1);
        K=al*k(:,n-6)+(1-al)*k(:,n-5);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        % Constraint to guarantee robust stability
        for j=1:vt_A
            for jj=1:vt_B
                w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-6)]>=0];
            end
        end
    end
    % Input constraint
    w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
    w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
    %
    solvedp(w,al); % Solve al
    al=double(al);
    AL(i,1)=al; % Collect al to matrix AL
    K=al*k(:,n-6)+(1-al)*k(:,n-5); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-6);
    al=1;
end

```

```

elseif xe*inv(q(:,n-7))*xe<=1;
    if o(2)==1 % Check ability in iteration between ellipsoid
        % Declare variables
        al=sdpvar(1,1);
        K=al*k(:,n-7)+(1-al)*k(:,n-6);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        % Constraint to guarantee robust stability
        for j=1:vt_A
            for jj=1:vt_B
                w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-7)]>=0];
            end
        end
        % Input constraint
        w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
        w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
        %
        solvesdp(w,al); % Solve al
        al=double(al);
        AL(i,1)=al; % Collect al to matrix AL
        K=al*k(:,n-7)+(1-al)*k(:,n-6); % Output feedback gain
    else
        %No ability in iteration between ellipsoid
        K=k(:,n-7);
        al=1;
    end
elseif xe*inv(q(:,n-8))*xe<=1;
    if o(1)==1 % Check ability in iteration between ellipsoid
        % Declare variable
        al=sdpvar(1,1);
        K=al*k(:,n-8)+(1-al)*k(:,n-7);
        %
        w=[];
        % Constraint of al
        w=[w,al>=0];
        w=[w,al<=1];
        %

```



```

% Constraint to guarantee robust stability
for j=1:vt_A
    for jj=1:vt_B
        w=[w,[1 ((A(:,j)+B(:,jj))*K)*xe)';(A(:,j)+B(:,jj))*K)*xe q(:,n-8)]>=0];
    end
end

% Input constraint
w=[w,[Ub(1)^2 (K(1,:)*xe)';K(1,:)*xe 1]>=0];
w=[w,[Ub(2)^2 (K(2,:)*xe)';K(2,:)*xe 1]>=0];
%
solvesdp(w,al); % Solve al
al=double(al);
AL(i,1)=al; % Collect al to matrix AL
K=al*k(:,n-8)+(1-al)*k(:,n-7); % Output feedback gain
else
    %No ability in iteration between ellipsoid
    K=k(:,n-8);
    al=1;
end
end
u1=K(1,:)*xe; %Deviation of input1 from steady state (V)
u2=K(2,:)*xe; %Deviation of input2 from steady state (V)
end

save('Find_K_by_interpolation','u1','u2')
st=1;

%% While loop to check sampling time (dt=12s)
st=1; % Initial command to start while loop
while st
    tf=clock; % Time duration sampling time ,
        % tt is initial time duration sampling time
    if (tf(4)*3600+tf(5)*60+tf(6))>=tt(4)*3600+tt(5)*60+tt(6)+dt % (Time duration sampling time)-(Initial
time duration sampling time) is 12 so sampling time=12
        run=1 % Start sample data at next iteration
        st=0 % Command to stop this while loop
    else
        st=1
    end
end
end

```

```

%% output variable collection
y=c*x;
for j=1:ny
    output(i,j)=y(j);
end
%% Estimated state variable collection (1 row is 1 iteration)
estate(i,1)=xe(1);
estate(i,2)=xe(2);
estate(i,3)=xe(3);
estate(i,4)=xe(4);
%% Input variables collection (1 row is 1 iteration)
input(i,1)=u1;
input(i,2)=u2;
%% Find estimated state at i+1
y=c*x;
xe=Ass*x+Bs*s*K*x+Lp*(y-c*x); %xe(i+1)
%% Convert input signal(0-5V) to computer signal(0-255)
input1_signal=(u1+1.96)*(255/5);
input2_signal=(u2+1.96)*(255/5);
% find flow14 set point
if input1_signal==95
    f1s=9.69;
elseif input1_signal<=96
    % At input1_signal=96 flow rate of pump1 =9.86
    rt=input1_signal-95;
    f1s=rt*(9.86-9.69)+9.69;
elseif input1_signal<=97
    % At input1_signal=97 flow rate of pump1 =10.28
    rt=input1_signal-96;
    f1s=rt*(10.28-9.86)+9.86;
elseif input1_signal<=98
    % At input1_signal=98 flow rate of pump1 =10.8
    rt=input1_signal-97;
    f1s=rt*(10.8-10.28)+10.28;
elseif input1_signal<=99
    % At input1_signal=99 flow rate of pump1 =11.37
    rt=input1_signal-98;
    f1s=rt*(11.37-10.8)+10.8;
elseif input1_signal<=100
    % At input1_signal=100 flow rate of pump1 =11.79

```

```

rt=input1_signal-99;
f1s=rt*(11.79-11.37)+11.37;
elseif input1_signal<=101
    % At input1_signal=101 flow rate of pump1 =12.12
    rt=input1_signal-100;
    f1s=rt*(12.12-11.79)+11.79;
elseif input1_signal<=102
    % At input1_signal=102 flow rate of pump1 =12.63
    rt=input1_signal-101;
    f1s=rt*(12.63-12.12)+12.12;
elseif input1_signal<=103
    % At input1_signal=103 flow rate of pump1 =12.93
    rt=input1_signal-102;
    f1s=rt*(12.93-12.12)+12.12;
elseif input1_signal<=104
    % At input1_signal=104 flow rate of pump1 =13.93
    rt=input1_signal-103;
    f1s=rt*(13.42-12.93)+12.93;
else
    % At input1_signal=104 flow rate of pump1 =13.93
    rt=input1_signal-104;
    f1s=rt*(13.93-13.42)+13.42;
end

% find flow23 set point
if input2_signal<=95
    f2s=6.75;
elseif input2_signal<=96
    % At input2_signal=96 flow rate of pump2 =7.62
    rt=input2_signal-95;
    f2s=rt*(7.62-6.75)+6.75;
elseif input2_signal<=97
    % At input2_signal=97 flow rate of pump2 =8.42
    rt=input2_signal-96;
    f2s=rt*(8.42-7.62)+7.62;
elseif input2_signal<=98
    % At input2_signal=98 flow rate of pump2 =9.27
    rt=input2_signal-97;
    f2s=rt*(9.27-8.42)+8.42;

```

```

elseif input2_signal<=99
    % At input2_signal=99 flow rate of pump2 =9.91
    rt=input2_signal-98;
    f2s=rt*(9.91-9.27)+9.27;
elseif input2_signal<=100
    % At input2_signal=100 flow rate of pump2 =10.66
    rt=input2_signal-99;
    f2s=rt*(10.66-9.91)+9.91;
elseif input2_signal<=101
    % At input2_signal=101 flow rate of pump2 =11.36
    rt=input2_signal-100;
    f2s=rt*(11.36-10.66)+10.66;
elseif input2_signal<=102
    % At input2_signal=102 flow rate of pump2 =11.98
    rt=input2_signal-101;
    f2s=rt*(11.98-11.36)+11.36;
elseif input2_signal<=103
    % At input2_signal=103 flow rate of pump2 =12.66
    rt=input2_signal-102;
    f2s=rt*(12.66-11.98)+11.98;
elseif input2_signal<=104
    % At input2_signal=104 flow rate of pump2 =13.3
    rt=input2_signal-103;
    f2s=rt*(13.3-12.66)+12.66;
else
    % At input2_signal=104 flow rate of pump2 =13.91
    rt=input2_signal-104;
    f2s=rt*(13.91-13.3)+13.3;
end

%%
save('flow_setpoint','f1s','f2s') % save flow setpoint to flow_setpoint.mat
f1s
f2s
%% Stop calculation when i=number round of iteration
if i==l
    run=0;
end

```

```
%%  
    save('Find_K_by_interpolation_run','run'); % Stop calculation when i=number round of iteration  
end  
  
%Save variables to excel file to plot graphs with SigmaPlot program  
xlswrite('non_minimumphase.xls',input(1:35,1),1,'B48')  
xlswrite('non_minimumphase.xls',input(1:35,2),1,'C48')  
xlswrite('non_minimumphase.xls',output(1:35,1),1,'D48')  
xlswrite('non_minimumphase.xls',output(1:35,2),1,'E48')  
xlswrite('non_minimumphase.xls',output(1:35,3),1,'F48')  
xlswrite('non_minimumphase.xls',output(1:35,4),1,'G48')  
xlswrite('non_minimumphase.xls',estate(1:35,1),1,'H48')  
xlswrite('non_minimumphase.xls',estate(1:35,2),1,'I48')  
xlswrite('non_minimumphase.xls',estate(1:35,3),1,'J48')  
xlswrite('non_minimumphase.xls',estate(1:35,4),1,'K48')  
%
```

ประวัติผู้เขียนวิทยานิพนธ์

นาย กรภัทร ไยฤทธิ เกิดวันที่ 9 เมษายน พ.ศ. 2533 มีภูมิลำเนาอยู่ในจังหวัด นครศรีธรรมราช สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนเบญจมราชูทิศ จากนั้นได้เข้าศึกษา ในระดับปริญญาบัณฑิต สาขาวิศวกรรมเคมี คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ โดย สำเร็จการในปีการศึกษา 2554 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชา วิศวกรรมเคมี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2556



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY