# บทที่ 3

# โปรแกรมช่วยวิเคราะห์และออกแบบระบบควบคุมแมทแลบ

แมทแลบ (Matlab) คือ โปรแกรมที่ใช้ในงานการคำนวนทางคณิตศาสตร์ขั้นสูงที่สามารถแสดงผลการคำนวนออกมาเป็นภาพให้เห็นได้ แมทแลบจะประกอบไปด้วย

- การวิเคราะห์ทางคณิตศาสตร์ (Numerical analysis)
- การคำนวณทางแมททริกซ์ (Matrix compensation)
- สัญญาณของกระบวนการผลิต (Signal processing)
- กราฟฟิค (Graphic)

ซึ่งทั้ง 4 ส่วนนี้จะอยู่ในลักษณะที่ง่ายต่อการใช้ สมการทางคณิตศาสตร์ก็จะถูกจัดให้อยู่ในสภาพที่ง่ายต่อการใช้โดยไม่มีไวยากรณ์หรือกฎเกณฑ์มาเกี่ยวข้องให้ยุ่งยากเหมือนโปรแกรมทั่วๆไป

## 3.1 การพัฒนาของแมทแลบ

แมทแลบ (MATLAB) เป็นชื่อย่อของโปรแกรม "MATrix LABoratory" แมทแลบได้ถูกสร้างขึ้นจากการพยายามที่จะพัฒนาโปรแกรมที่ใช้ในการคำนวณแมททริกซ์ภายใต้โครงการ ที่ชื่อว่า "LINPACK and EISPACK"

แมทแลบจะเป็นโปรแกรมที่สามารถโต้ตอบกับผู้ใช้งานได้ในทันทีทันใด (Interactive) โดยข้อมูลหรือโจทก์ทางคณิตศาสตร์ ที่จะใช้แมทแลบคำนวณจะต้องอยู่ในรูปของแมททริกซ์และไม่ จำเป็นต้องมีการแสดงหน่วยของข้อมูลนั้น

แมทแลบได้มีการพัฒนามาอย่างต่อเนื่อง ในการศึกษาแมทแลบจะถูกใช้ไปในการสอนทางคณิตศาสตร์,การวิจัย ฯลฯ ในทางอุตสาหกรรมแมทแลบจะใช้การวิจัยทางวิศวกรรม และการแก้ไขปัญหาทางคณิตศาสตร์เช่น การควบคุมกระบวนการแบบอัตโนมัติ การวิจัยสัญญาณของกระบวนการผลิตในรูปดิจิตัล

การทำงานของแมทแลบจะเป็นฟังก์ชันของคำสั่งที่อยู่ในรูปของโปรแกรม "M-file" ที่ถูกบรรจุอยู่ในพื้นที่ที่เรียกว่า "Toolboxes" ซึ่งคำสั่งเหล่านี้สามารถนำมาใช้แก้ปัญหาทางคณิตศาสตร์ คือ

ก. การคำนวนเกี่ยวกับแมทริกซ์

โปรแกรมแมทแลบสามารถทำแมททริกซ์ทรานโพส การคูณแมททริกซ์ การหาดีเทอร์มิแนนท์ การทำอินเวอร์สแมททริกซ์ หาค่าไอเกน ตลอดจนการแก้สมการเชิงเส้น การประมาณค่า เป็นต้น

ข. การคำนวนโพลิโนเมียล

โปรแกรมแมทแลบสามารถใช้ในการคำนวนเกี่ยวกับโพลิโนเมียล เช่น การหารากของโพลิโนเมียล การหาคอนโวลูชั่น (Convolution) และดีคอนโวลูชั่น (Deconvolution) การหาร โพลิโนเมียล การหาสมการถดถอยแบบโพลิโนเมียล เป็นต้น

ค. การจัดการเกี่ยวกับเวคเตอร์

โปรแกรมแมทแลบสามารถนำมาคำนวนผลรวมแบบเวคเตอร์ การหาค่าเฉลี่ยและการหาค่าเบี่ยงเบนมาตรฐาน เป็นต้น

ง. การจัดการเกี่ยวกับการแสดงผลกราฟ

โปรแกรมแมทแลบมีการแสดงผลเป็นกราฟให้เลือกทั้งสิ้น 7 ชนิด คือ

- การพลอต x-y บนสเกลเส้นตรง
- การพลอต x-y บนสเกลล็อก-ล็อก

- การพลอต x-y บนสเกลกึ่งล็อกบนแกน x
- การพลอต x-y บนสเกลกึ่งล็อกบนแกน y
- การพลอตแบบโพลาร์
- การพลอตแบบตะแกรง 3 มิติ
- การพลอตแบบคอนทัวร์

จ. การจัดการต่างๆ ของกระบวนการส่งสัญญาณ

ฉ. การวิเคราะห์ไม่เป็นเชิงเส้น

ช. การประเมินและวิเคราะห์ผลทางสถิติ

นอกจากคำสั่งพื้นฐานเฉพาะในโปรแกรมแมทแลบที่เหมือนสมการคณิตศาสตร์แล้ว โปรแกรมแมทแลบยังสามารถใช้งานหรือพัฒนางานร่วมกับโปรแกรมพัฒนาระบบอื่นๆ ได้อีก เช่น โปรแกรมภาษาซี โปรแกรมภาษาฟอร์แทรน เป็นต้น

แมทแลบถูกเขียนขึ้นครั้งแรกโดยใช้ภาษาฟอร์แทรน (Fortran) โดยคลีฟ โมลเลอร์ (Cleve Moller) จากนั้นก็ได้รับการพัฒนาจากโปรแกรมเมอร์อีกหลายท่านในโครงการ "LINPACK and EISPACK" ปัจจุบันแมทแลบถูกเขียนขึ้นโดยใช้ภาษาซี (C language)

นอกจากนี้ในโปรแกรมแมทแลบจะมีโปรแกรมซิมูลิงค์ (Simuink) ติดตั้งอยู่ด้วย ซึ่งโปร แกรมซิมูลิงค์นี้จะใช้ในการซิมูเลทระบบที่มีการเปลี่ยนแปลงไปตามเวลาและจะสามารถแสดงผล ออกมาในรูปของกราฟฟิคโดยจะใช้เมาส์ในการทำงาน ผู้ใช้สามารถสร้างแบบจำลองกระบวนการ ผลิตได้ด้วยการสร้างบล็อคไดอะแกรม (Block diagram) ลงในจอคอมพิวเตอร์ บล็อคไดอะแกรม เหล่านี้สามารถปรับเปลี่ยนค่าได้ตามต้องการและจะถูกควบคุมด้วยโปรแกรม "M-file" ของแมทแลบ อีกต่อหนึ่ง

## 3.2 โปรแกรมซิมูลิงค์ (Simulink)

ซิมูลิงค์ คือโปรแกรมที่ใช้สำหรับการทำซิมูเลทระบบที่มี การเปลี่ยนแปลงโดยซิมูลิงค์จะเป็นโปรแกรมที่ขยายออกมาจากโปรแกรมแมทแลบ ซึ่งโปรแกรมซิมูลิงค์นี้จะประกอบไปด้วยฟังก์-ชันต่างๆมากมายและฟังก์ชันเหล่านี้สามารถนำมาประกอบกันเพื่อให้ได้ระบบที่มีความแตกต่างซึ่งจะ สามารถปรับเปลี่ยนได้ตามต้องการ

ซิมมูลิงค์จะมีทิศทางในการใช้ 2 แนวทางคือ

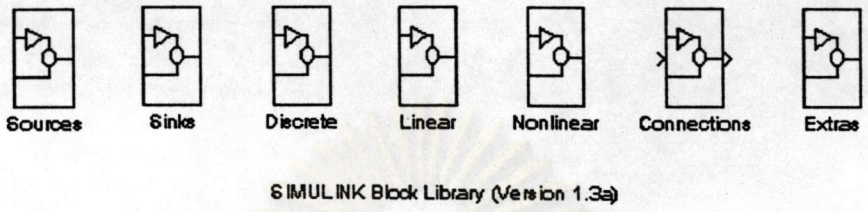
- ใช้ในการจำลองแบบโมเดลหรือออกแบบโมเดล
- ใช้ในการวิเคราะห์โมเดลในรูปแบบต่างๆกัน

โดยทั่วไปแล้วเราจะใช้ทั้ง 2 วิธีการมาทำงานร่วมกัน คือ อันดับแรกจะต้องทำการจำลองแบบหรือออกแบบโมเดลให้ได้ตามที่ต้องการ โดยการนำบล็อกไดอะแกรมต่างๆที่มีอยู่ในฟังก์ชันมาต่อเข้าด้วยกันโดยใช้เมาส์ จากนั้นก็จะทำการวิเคราะห์โมเดลที่ได้ด้วยการเลือกการปรับเปลี่ยนค่าในบล็อกไดอะแกรมต่างๆที่ปรากฏอยู่บนเมนูหน้าบนวินโดว์และเมื่อทำการปรับเปลี่ยนเสร็จเรียบร้อยก็จะทำการซิมูเลทโปรแกรมโดยเราจะทราบถึงผลการซิมูเลทได้จาก การดูผลของผลการตอบสนอง (Responses) ที่อยู่ในรูปของกราฟที่ปรากฏอยู่บนจอวินโดว์ และสามารถที่จะทำการพิมพ์ผลการตอบสนองดังกล่าวออกมาได้ทางเครื่องพิมพ์

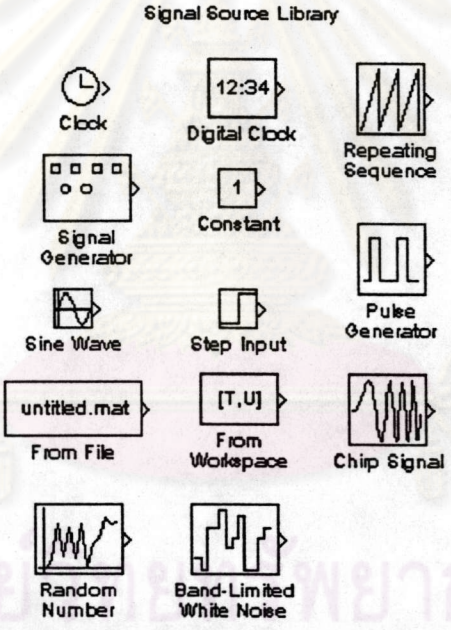
### 3.2.1. การจำลองแบบโมเดลหรือออกแบบโมเดล

การใช้งานโปรแกรมซิมูลิงค์ในลักษณะนี้ จะเป็นการสร้างบล็อคไดอะแกรมบนหน้าจอแสดงผลของคอมพิวเตอร์ซึ่งองค์ประกอบต่างๆของบล็อคไดอะแกรม จะสามารถนำมาใช้ได้จากห้องสมุดความจำรูปบล็อคไดอะแกรม (Block Diagram Library) ดังรูป 3.1 ถึงรูป 3.8 โดยการใช้เมาส์ลากลงมาใส่ในเวอร์คชีต (Work Sheet) และใช้เมาส์ลากเส้นเชื่อมบล็อคแต่ละบล็อคเข้าด้วยกันตามวิธีการสร้างแบบจำลองกระบวนการผลิตแบบต่างๆในโปรแกรมซิมูลิงค์ที่จะเสนอดังต่อไปนี้คือ

1.ห้องสมุดความจำบล็อคไดอะแกรมจะประกอบไปด้วย บล็อคแหล่งกำเนิดสัญญาณควบคุมบล็อคเชิงเส้น บล็อคไม่เป็นเชิงเส้น แบบดีสครีต แบบดิจิตอลและแบบต่อเนื่อง ฯลฯ

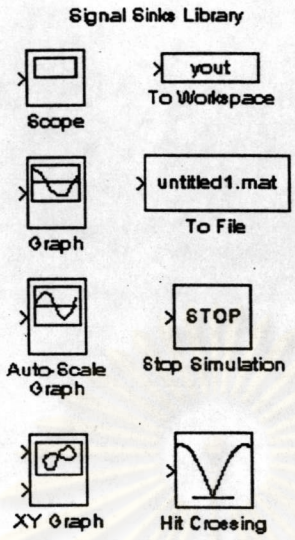



รูปที่ 3.1 ห้องสมุดของบล็อคไดอะแกรมของซิมูสิงค์

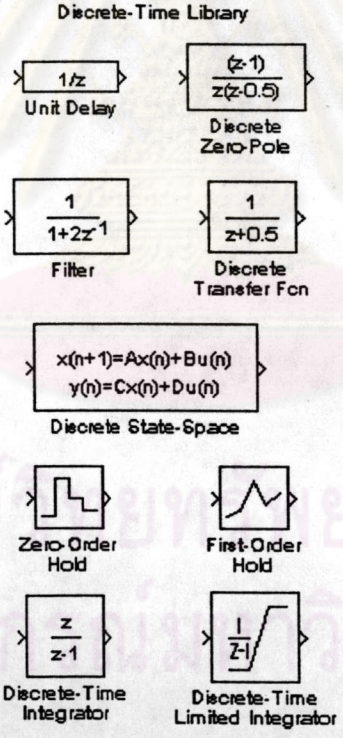



รูปที่ 3.2 บล็อคที่เป็นตัวกำเนิดสัญญาณแบบต่างๆ

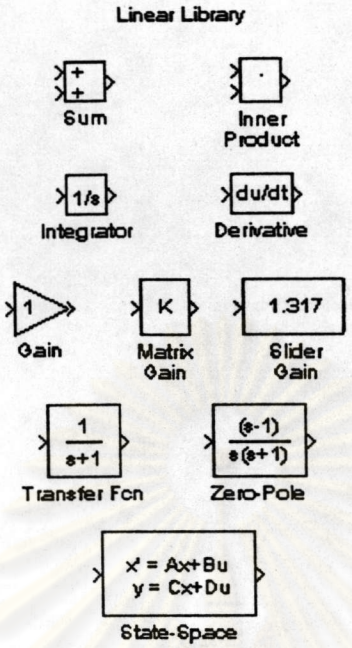
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



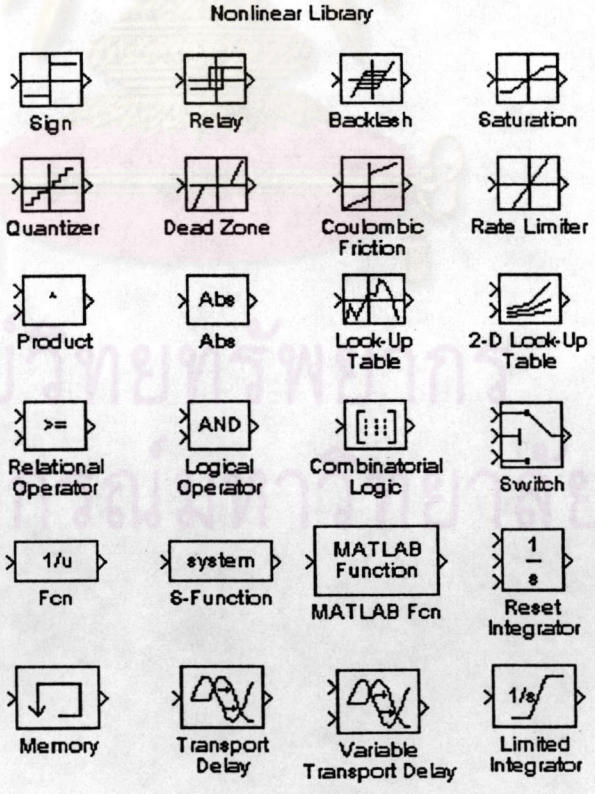

รูปที่ 3.3 รูปแบบของสัญญาณส่งออกแบบต่างๆ



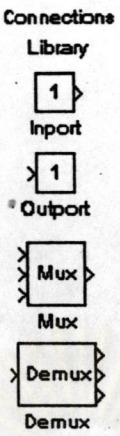

รูปที่ 3.4 บล็อคไดอะแกรมของดีสครีตแบบต่างๆ



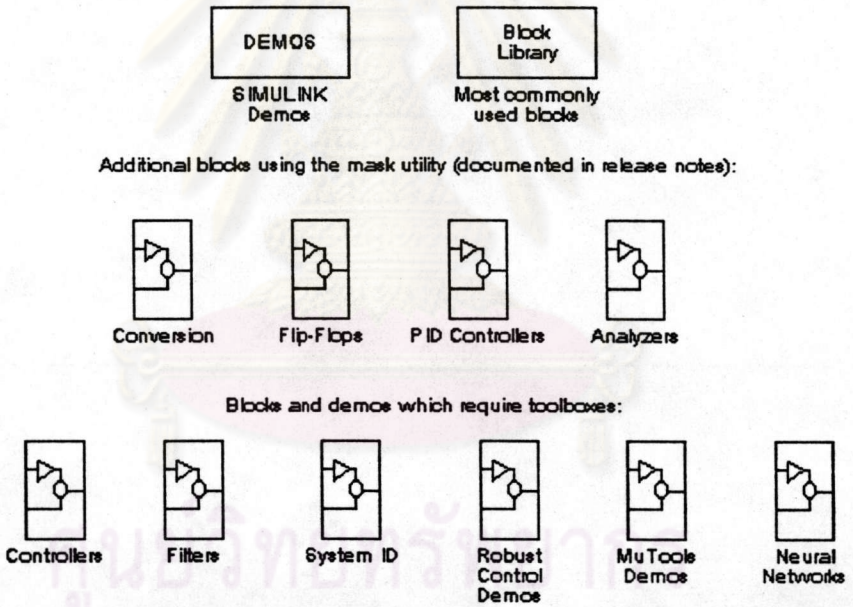

รูปที่ 3.5 บล็อคไดอะแกรมและเกนแบบต่างๆ ที่เป็นแบบเชิงเส้น




รูปที่ 3.6 บล็อคไดอะแกรมของและเกนแบบต่างๆ ที่เป็นแบบไม่เชิงเส้น




รูปที่ 3.7 บล็อคไดอะแกรมของตัวเชื่อมสัญญาณแบบต่างๆ




รูปที่ 3.8 บล็อคไดอะแกรมของอุปกรณ์ควบคุมเพิ่มเติม

2.การเชื่อมต่อบล็อคทั้งแบบสกาลาร์และเวคเตอร์ โดยสามารถทำแบบมีอินพุทค่าเดียว เอาท์พุทเดียวหรือแบบอินพุทหมู่ เอาท์พุทหมู่ โดยโปรแกรมสามารถแปลงสัญญาณให้เป็นเวคเตอร์หรือสกาลาร์ได้ โดยมัลติเพล็กเซอร์ และ ดีมัลติเพล็กเซอร์

3.การแสดงสัญญาณอินพุทและเอาท์พุท สัญญาณอินพุทสามารถกำหนดให้เป็นแบบสเต็พ แบบซายน์ หรือแบบสัญญาณรบกวน (White Noise) ส่วนสัญญาณเอาท์พุทอาจแสดงผลในรูปของกราฟ หรือข้อมูลของเวอร์คชีตบนพื้นฐานของเวลาจริง

4.การจัดแบบจำลองออกเป็นระดับชั้นต่างๆกัน (Hierachical Model) อาจจะแบ่งเป็นจากบนสู่ล่างหรือจัดแบบเป็นกลุ่มๆ โดยไม่มีข้อจำกัดของกลุ่มหรือจำนวนบล็อค

5.การขยายห้องสมุดบล็อคไดอะแกรม ผู้ใช้งานสามารถสร้างหรือขยายลักษณะของบล็อคที่ใช้งานในด้านต่างๆ และเก็บไว้เป็นห้องสมุดบล็อคไดอะแกรมได้ 4 รูปแบบ คือ

- กำหนดหรือสร้างในระบบย่อย
- สร้างจากเอ็มไฟล์ (M-File) ในโปรแกรมแมทแลบ
- ใช้โปรแกรมภาษาซี หรือ ภาษาฟอร์แทรน
- โดยการเปลี่ยนแปลงจากบล็อคไดอะแกรมที่มีอยู่แล้ว

### 3.2.2. การวิเคราะห์แบบจำลอง

ภายหลังจากที่ผู้ใช้ได้สร้างหรือกำหนดแบบจำลองเสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปของ ผู้ใช้งานคือ ต้องเลือกวิธีการในการวิเคราะห์ ซึ่งในโปรแกรมซิมูลิงค์มีให้ เลือกทั้งสิ้น 3 วิธี คือ การทำการเลียนแบบ การวิเคราะห์เชิงเส้น และการทดสอบหาจุดสมดุลย์

1. การทำการเลียนแบบ (Simulation)

ขั้นตอนอันดับแรกในการทำเลียนแบบ จะเป็นขั้นตอนการแก้สมการดิฟเฟอเรนเชียลที่มีอยู่ทั้งหมดในแบบจำลองเปรียบเทียบกับเวลา โดยโปรแกรมซิมูลิงค์จะมีเครื่องมือในการแก้สมการอยู่ทั้งหมด 6 แบบ คือ

ก. รังจ์-กัตตา อันดับที่ 3 (Runge-Kutta Third Order)

ข. รังจ์-กัตตา-เฟลเบอร์ก อันดับที่ 5 (Runge-Kutta-Fehlberg Fifth Order)

ค. การทำนายและการแก้ค่าของอดัมส์ (Adams Predict-Corrector)

ง. วิธีการของเกียร์สำหรับระบบแบบสติฟ (Gear's Method for Stiff Systems)

จ. การแยกไดนามิคเชิงเส้น (Linear Dynamics Extraction)

ฉ. วิธีการของออยเลอร์ (Euler Method)

2. การวิเคราะห์เชิงเส้น

การวิเคราะห์ในโปรแกรมซิมูลิงค์จะใช้สมการหลัก 2 สมการ คือ

$$\overset{o}{X} = Ax + Bu \qquad (3.1)$$

$$\overset{o}{Y} = Cx + Du \qquad (3.2)$$

ทั้งสมการ 3.1 และ 3.2 จะใช้คำนวนวิเคราะห์ได้ทั้งแบบตัวแปรต่อเนื่องและแบบดีสครีต

3. การทดสอบหาจุดสมดุลย์

เป็นเครื่องมือชุดที่สามที่ใช้หาจุดสมดุลย์ ซึ่งผู้ใช้โปรแกรมซิมูลิงค์สามารถกำหนดค่าต่างๆ เช่น อินพุท เอาท์พุท และค่าของสภาวะให้แก่โปรแกรม เพื่อให้โปรแกรมคำนวนหาจุดสมดุลย์ให้

## 3.3 เครื่องมือในการวิเคราะห์อื่นๆ

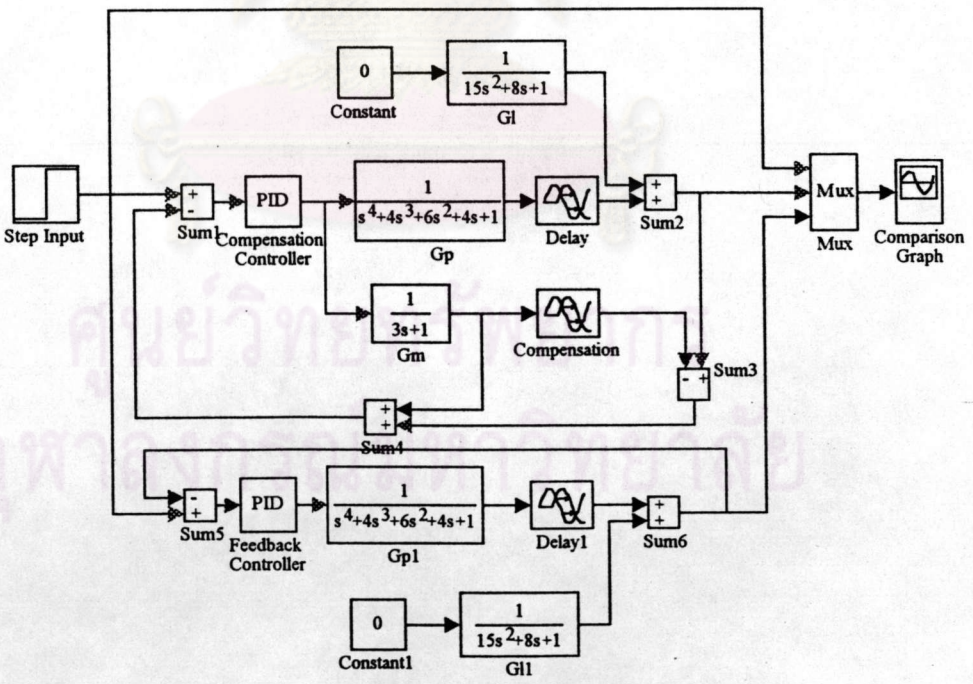
โปรแกรมแมทแลบได้จัดเครื่องมือต่างๆไว้หลายชุด เพื่อให้เหมาะในการใช้งานในลักษณะต่างๆได้แก่

- การจัดการด้านกระบวนการสัญญาณ
- การควบคุมรอบัสต์ของระบบ
- นิวรัลเนทเวอร์ค
- คณิตศาสตร์แบบสัญญลักษณ์
- การวิเคราะห์ทางสถิติ
- การจำแนกระบบ
- การระดับจุลภาค (Micro Analysis)
- การทำออพติไมซ์
- ระบบควบคุม
- เป็นต้น

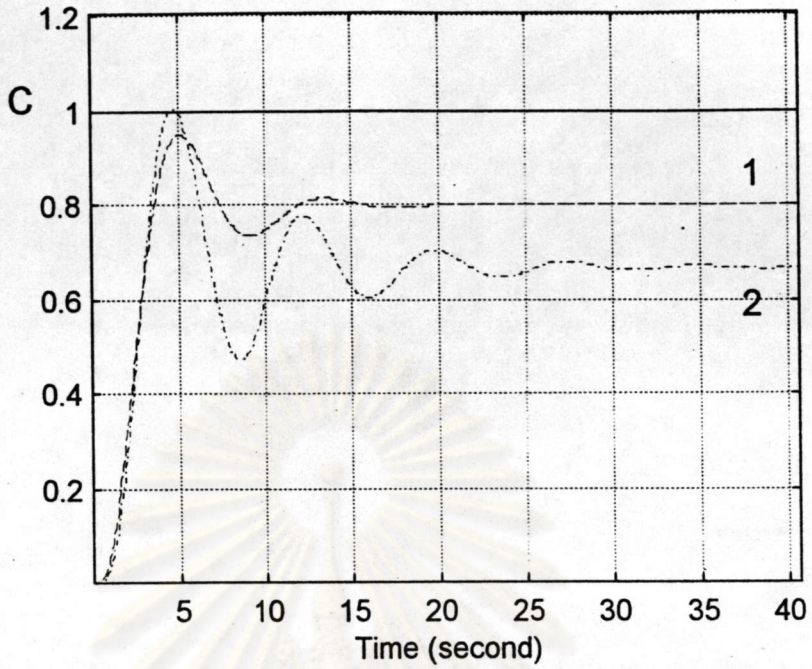
## 3.4 ตัวอย่างการใช้งาน

รูปที่ 3.9 เป็นบล็อคไดอะแกรมที่นำมาจากหนังสือ Process Systems Analysis and Control 2nd Edition แต่งโดย Donald R. Coughanowr ดังตัวอย่างที่ 18.4 รูปที่ 18.25 หน้า 270 แสดงถึงกระบวนการที่ทรานส์เฟอร์ฟังก์ชันมีอันดับเท่ากับ 4 ดัง สมการ 3.3 โดยจะเป็นการเปรียบเทียบให้เห็นถึงความแตกต่างของการควบคุมแบบป้อนกลับที่มีตัวชดเชยเดดไทม์ของสมิธและการควบคุมแบบป้อนกลับที่เป็นการควบคุมแบบสัดส่วนธรรมดาและเมื่อนำบล็อคไดอะแกรมทั้ง 2 นี้มาทำการการเลียนแบบพร้อมกันโดยใช้โปรแกรมซิมูลิงค์และใช้การจูน โดยวิธี ของ"ซิเกลอร์-นิโคลส์" การตอบสนองที่ได้จะปรากฏ ดังรูปที่ 3.10 ซึ่งจะตรงกับการตอบ สนองที่ได้จากตัวอย่าง ที่ 18.4 ดัง รูปที่ 18.26 หน้า 271

$$G_p = \frac{e^{-6s}}{s^4+4s^3+6s^2+4s+1} \qquad (3.3)$$




รูปที่ 3.9 บล็อคไดอะแกรมของกระบวนการควบคุมแบบป้อนกลับที่ติดตั้งตัวชดเชยเดดไทม์ของสมิธและกระบวนการควบคุมแบบป้อนกลับ

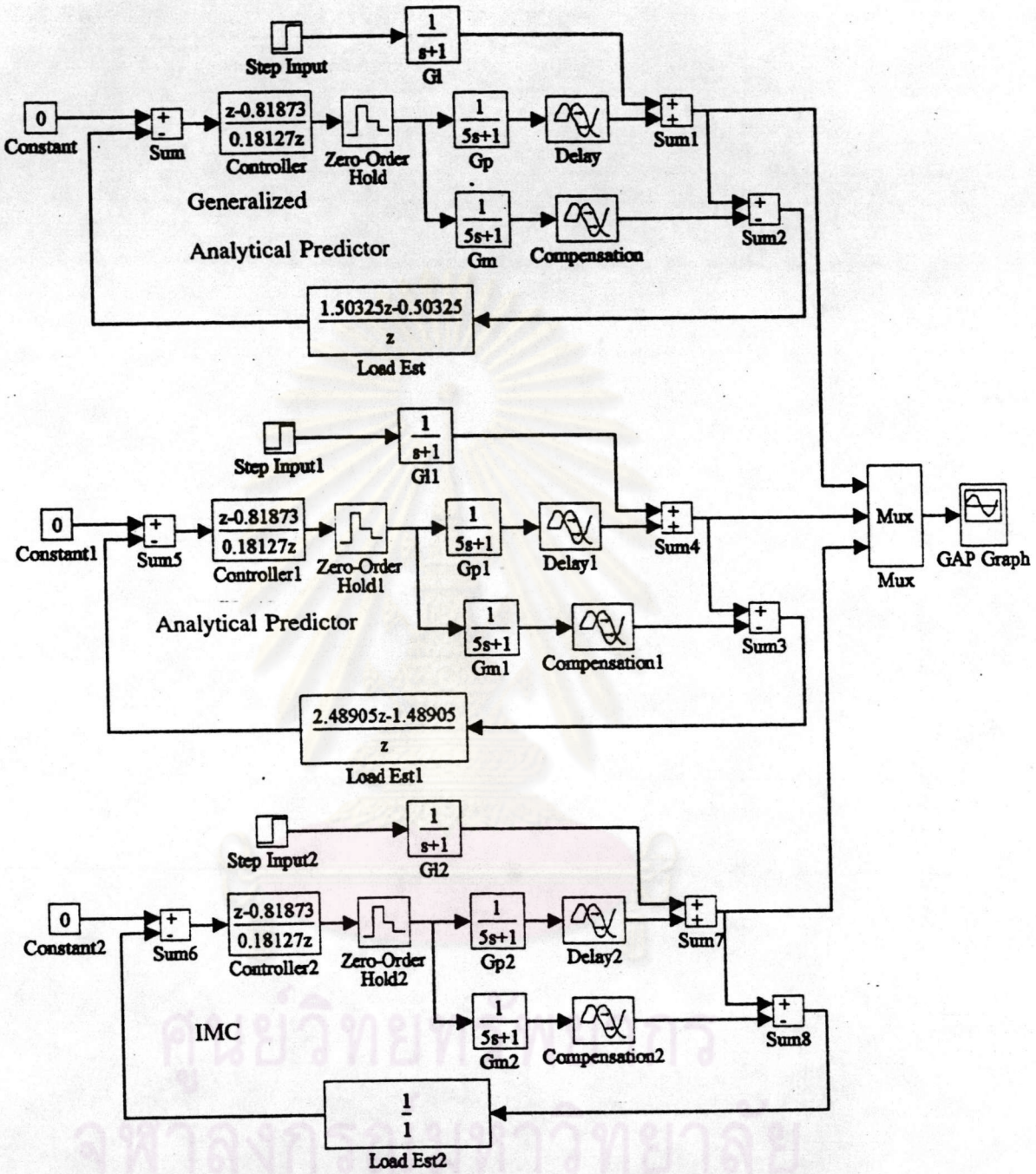



รูปที่ 3.10 แสดงผลการตอบสนอง (Responses) ที่ได้จากการทำซิมูเลทบล็อคไดอะแกรมของกระบวนการ ดังรูปที่ 3.9

จากรูปที่ 3.10 ผลการตอบสนองที่ได้จะพบว่า

- ผลการตอบสนองหมายเลข 1 คือ ผลจากการทำการเลียนแบบตัวทำนายของสมิธ
- ผลการตอบสนองหมายเลข 2 คือ ผลจากการทำการเลียนแบบกระบวนการควบคุมแบบป้อนกลับแบบพีไอดี

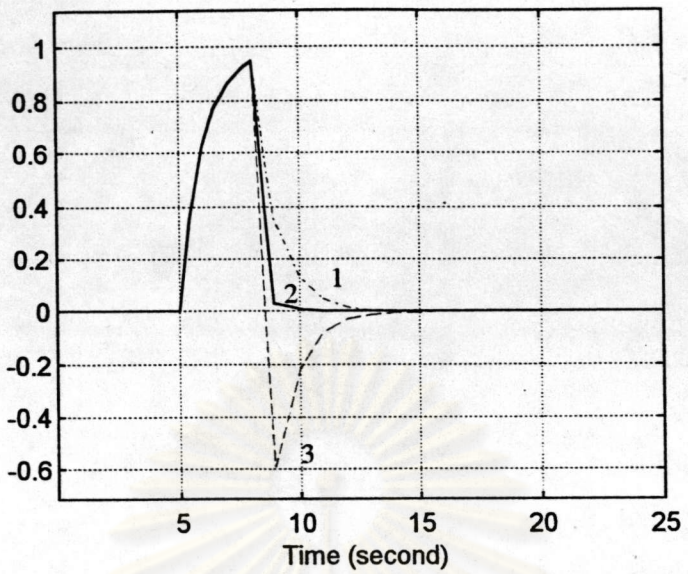
รูปที่ 3.11 เป็นบล็อคไดอะแกรมที่นำมาจากหนังสือ Process Dynamic and Control แต่งโดย Dale E. Seborg และคณะ ดังตัวอย่างที่ 26.7 รูปที่ 26.14 หน้า 641 จะเป็นการแสดงถึงการชดเชยเดดไทม์ของกระบวนการป้อนกลับด้วยวิธีของตัวทำนายเชิงวิเคราะห์โดยใช้นำวิธีการควบคุมโมเดลภายใน มาประยุกต์ใช้และเมื่อนำบล็อคไดอะแกรมนี้มาทำการเลียนแบบ โดยใช้โปรแกรมซิมูลิงค์ การตอบสนองที่ได้จะปรากฏดังรูปที่ 3.12




รูปที่ 3.11 บล็อคไดอะแกรมของกระบวนการควบคุมแบบป้อนกลับด้วยวิธีของตัวทำนาย

เชิงวิเคราะห์โดยวิธีการควบคุมโมเดลภายใน

I16884401




รูปที่ 3.12 ผลการตอบสนอง (Responses) ที่ได้จากการทำซิมูเลทบล็อคไดอะแกรม ของกระบวนการ ดังรูปที่ 3.11

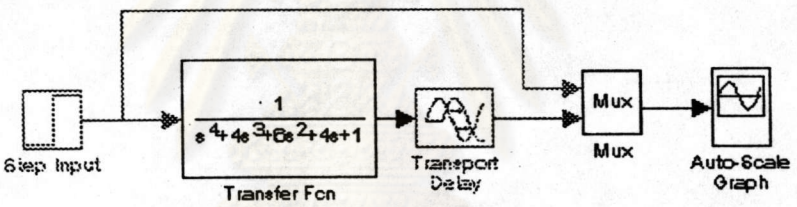
จากรูปที่ 3.12 ผลการตอบสนองที่ได้จะพบว่า

- ผลการตอบสนองหมายเลข 1 คือ ผลการตอบสนองจากการทำการเลียนแบบ (Simulation) ตัวทำนายเชิงวิเคราะห์โดยสมมุติให้ A= 1
- ผลการตอบสนองหมายเลข 2 คือ ผลการตอบสนองจากการทำการเลียนแบบ (Simulation) ตัวทำนายเชิงวิเคราะห์แบบทั่วไป
- ผลการตอบสนองหมายเลข 3 คือ ผลการตอบสนองจากการทำการเลียนแบบ (Simulation) ตัวทำนายเชิงวิเคราะห์

## 3.5 การจัดบล็อคไดอะแกรมที่มีอันดับที่มากกว่าหนึ่งโดยวิธีการเปลี่ยนแปลงแบบสเต็พ (Fitting Higher Order Model using Step Change)

จากการที่กระบวนการผลิตของโรงงานอุตสาหกรรมทั่วไปมักจะมีอันดับที่มากกว่าหนึ่งขึ้นไปเสมอ อันจะมีผลทำให้การปรับแต่งตัวควบคุมกระบวนการทำได้ลำบากมากโดยเฉพาะการทำ

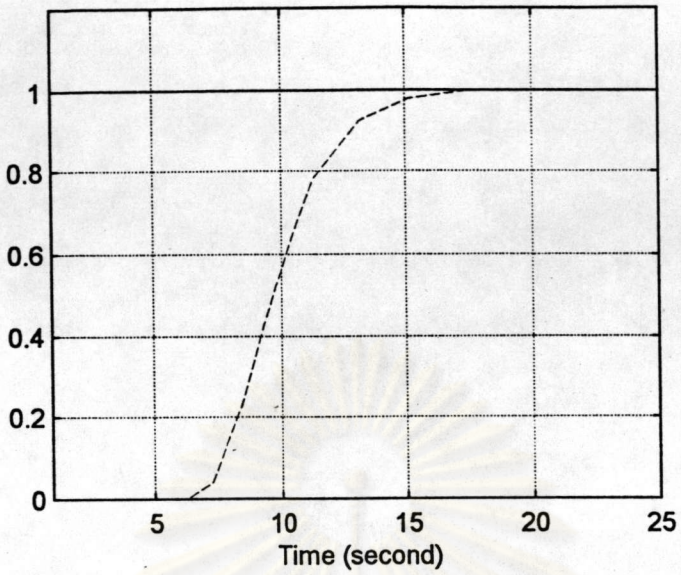
การชดเชยเดดไทม์ เพราะเราจะไม่ทราบเลยว่ากระบวนการจะมีอันดับเป็นเท่าไร ดังนั้นในการแก้ปัญหาที่เกิดขึ้นนี้จะทำได้โดยการสร้างโมเดลของกระบวนการดังกล่าว ให้มีอันดับเท่ากับหนึ่งเสียก่อน โดยการนำทรานส์เฟอร์ฟังก์ชันของกระบวนการ (Process Transfers Functions) มาทำการเปลี่ยนแปลงแบบสเต็พ (Step change) แล้วนำการตอบสนอง (Response) ที่ได้มาหาค่า $\tau$ และค่า $\theta$ (เดดไทม์) จากบล็อคไดอะแกรมของรูป 3.9 จะพบว่าทรานส์เฟอร์ฟังก์ชันของกระบวนการมีอันดับเท่ากับ 4 ดังนั้นจะของนำทรานส์เฟอร์ฟังก์ชันของกระบวนการนี้มาทำการเปลี่ยนแปลงแบบสเต็พเพื่อทำการเปลี่ยนกระบวนการดังกล่าวให้มีอันดับเท่ากับหนึ่ง ดังรูป 3.13 จะเป็นลักษณะของบล็อคไดอะแกรมที่จะทำการเปลี่ยนแปลงแบบสเต็พ




Deadtime = Transport delay = 6 Sec

รูปที่ 3.13 แสดงลักษณะของบล็อคไดอะแกรมที่จะทำการเปลี่ยนแปลงแบบสเต็พ

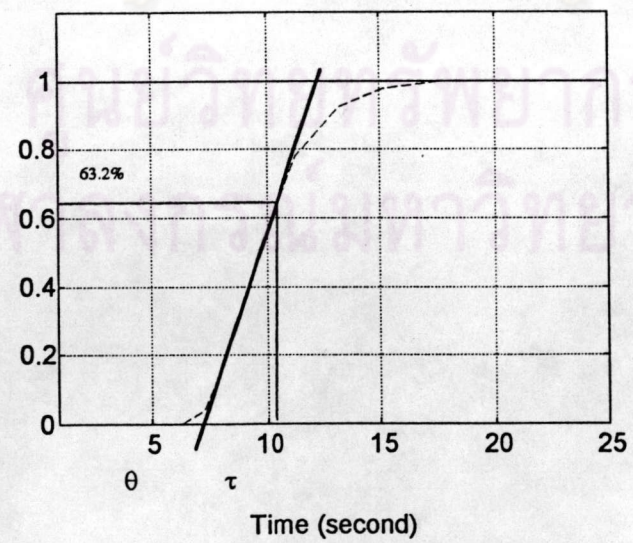
การตอบสนองของการทำการเปลี่ยนแปลงแบบสเต็พของบล็อคไดอะแกรมดังรูปที่ 3.13 จะมีลักษณะดังรูปที่ 3.14




รูปที่ 3.14 แสดงการเปลี่ยนแปลงแบบสเต็พของกระบวนการที่อันดับมากกว่าหนึ่ง

จากการรูป 3.14 ค่า $\tau+\theta$ จะได้เท่ากับ 63.2% ของการตอบสนอง (Response) ค่า $\theta$ หาได้จากจุดตัดแกน x ของเส้นสัมผัสกราฟ ดังรูป 3.15 ด้วยความสัมพันธ์ดังกล่าวทำให้สามารถเขียนทรานส์เฟอร์ฟังก์ชัน ของกระบวนการให้อยู่ในอันดับที่หนึ่ง (1st order) ได้คือ

$$G_m(s) = \frac{e^{-\theta s}}{\tau s+1} \qquad (3.4)$$

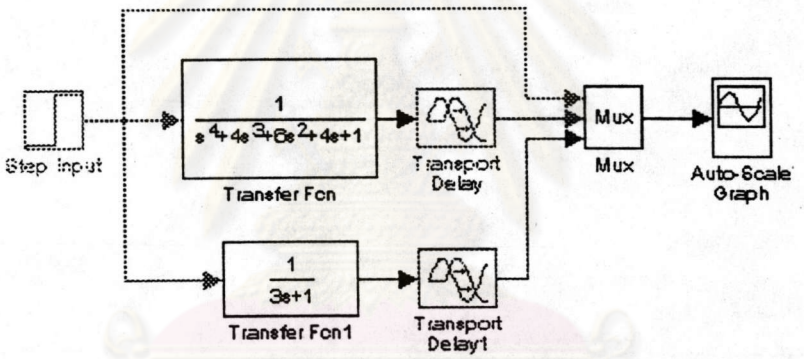



รูปที่ 3.15 แสดงการหาค่า $\theta_m$ และค่า $\tau_m$ ของ $G_m$

จากวิธีการดังกล่าวข้างต้นทำให้สามารถหาค่า $\theta$ ได้เท่ากับ 7.3 และหาค่า $\tau$ ได้เท่ากับ 3 ดังนั้นจะได้ $G_m$ 'คือ

$$G_m(s) = \frac{e^{-7.3s}}{3s+1} \qquad (3.5)$$

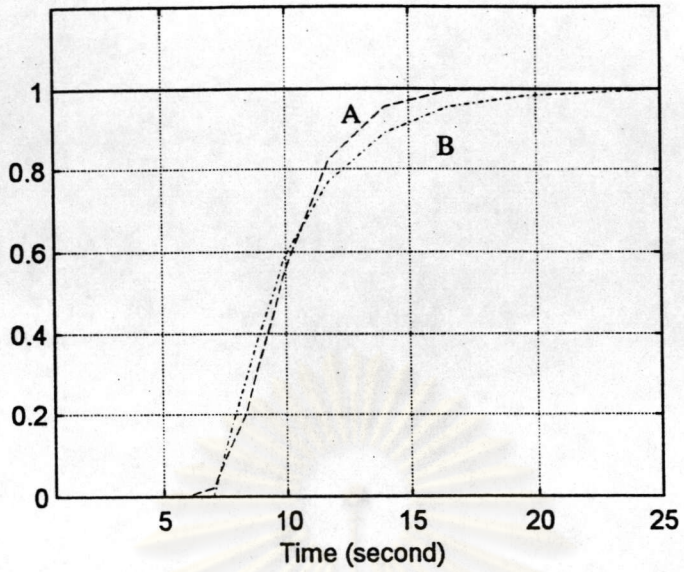
$G_m$ ที่ได้จากสมการ 3.5 จะสามารถทดสอบนี้ได้โดยการนำมา ทำการเปลี่ยนแปลงแบบสเต็พเปรียบเทียบกับ $G_p$ ดังรูป 3.16 จากนั้นนำการตอบสนอง (Response) ที่ได้มาวาดกราฟลงในแกนเดียวกัน แล้วสังเกตถึงการเปลี่ยนแปลงดังรูปที่ 3.17 ถ้า $G_m$ ที่ได้ถูกต้องการตอบสนองก็ควรจะมีลักษณะคล้ายคลึงกัน




Deadtime = Transport Delay = 6

New Deadtime = Transport Delay1 = 7.3

รูปที่ 3.16 แสดงลักษณะของบล็อคไดอะแกรมที่จะทำการเปลี่ยนแปลงแบบสเต็พเมื่อติดตั้ง$G_m$




รูปที่ 3.17 แสดงการตอบสนองที่มีการเปลี่ยนแปลงแบบสเต็พของกระบวนการที่ทรานส์เฟอร์ฟังก์ชันมีอันดับมากกว่าหนึ่งและกระบวนการที่ทรานส์เฟอร์ฟังก์ชันได้รับการเปลี่ยนเป็นอันดับหนึ่ง โดยวิธีการแบบสเต็พ

$A = G_p$

$B = G_m'$

สัญญาณการตอบสนองที่ได้ควรจะมีลักษณะที่ใกล้เคียงกันดังรูป 3.17 ซึ่งหมายถึง $A = G_p$ ที่มีอันดับมากกว่าหนึ่ง และ$B = G_m$ ที่มีอันดับเท่ากับหนึ่งที่มีค่าที่ใกล้เคียงกัน

จากการที่สามารถหาโมเดลของกระบวนการผลิตได้ในรูปของทรานส์เฟอร์ฟังก์ชันที่มีอันดับเท่ากับหนึ่ง จะมีประโยชน์ในการทำการเลียนแบบ (Simulation) ของตัวชดเชยเดดไทม์ ซึ่งจะต้องสร้างโมเดลของกระบวนการผลิตให้ใกล้เคียงกับกระบวนการผลิตจริงมากที่สุด อีกทั้งจะต้องทำการเลียนแบบ (Simulation) ในรูปแบบของการกำหนดโมเดลผิดพลาด (Modeling Error) ซึ่งในการทำวิทยานิพนธ์ฉบับนี้จะมีการปรับเปลี่ยนค่า $\tau$ และค่า $\theta$ ของโมเดลให้มีค่าต่างๆกันไป โดยจะกล่าวในรายละเอียดในบทต่อไป

## 3.6 การจูนตัวควบคุม (Controller Tunning)

ในการจูนค่า พีไอดี ของตัวควบคุมกระบวนการจะแบ่งตามรูปแบบของตัวชดเชยเดดไทม์ คือ จะใช้วิธีของ "ซีเกลอร์-นิโคลส์" (Ziegler-Nichols) ในรูปแบบของวิธีการไซเคิลแบบต่อเนื่อง (Continuous Cycling Method) ซึ่งการควบคุมชนิดนี้มีพื้นฐานมาจากการจูนแบบลองผิดลองถูก (Trial and Error Tunning) ในการจูนตัวควบคุมกระบวนการนี้ จะทำในโปรแกรมซิมูลิงค์ที่ได้จัดบล็อคไดอะแกรม ของกระบวนการไว้แล้วและสามารถแบ่งออกเป็นขั้นตอนต่างๆคือ

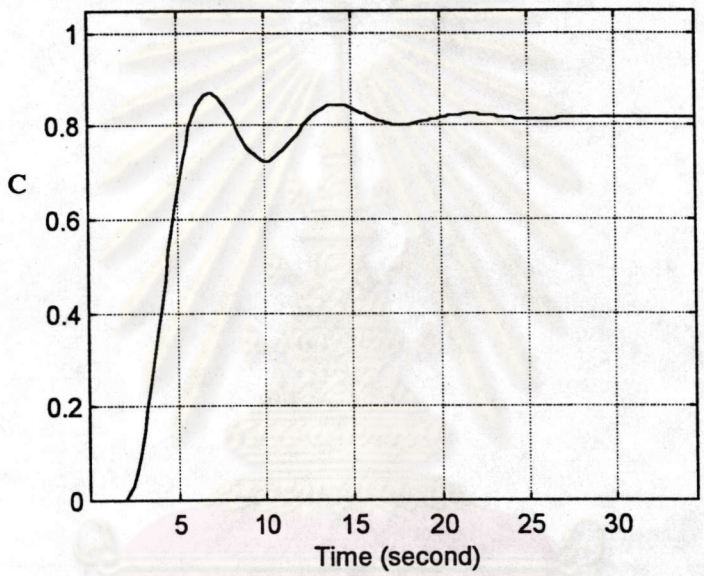
**ขั้นที่ 1 การจูนตัวควบคุมแบบลองผิดลองถูก (Trial and Error Tunning)**

เป็นการจูนตัวควบคุมเพื่อหาค่า $K_c$ ซึ่งสามารถแบ่งเป็นขั้นตอนย่อยได้ดังนี้

1. กำจัดค่าอินทีกรัล (Integral) และดีไรเวทีฟ (Derivative) ออก โดยการตั้งค่า $\tau_I$ ให้มีค่าสูงที่สุด และค่า $\tau_D$ ให้มีค่าต่ำที่สุด
2. สุ่มค่า $K_c$ ขึ้นมาหนึ่งค่า (เริ่มที่ค่า $K_c$ ต่ำๆ ก่อน) จากนั้นก็ปล่อยให้โปรแกรมซิมูลิงค์ทำงานไปแบบอัตโนมัติ
3. สังเกตดูลักษณะของกราฟของการตอบสนอง (Response) ที่ได้ ถ้าเกิดโอเวอร์ชูต (Overshoot) แสดงว่าจะต้องลดค่า $K_c$ ลงและถ้ากราฟการตอบสนองมีแนวโน้มเข้าใกล้ค่าเซ็ทพอยท์ (Setpoint) แสดงว่าจะต้องเพิ่มค่า $K_c$ ให้มากขึ้น ทั้งนี้การเพิ่มหรือลดค่า $K_c$ จะทำจนกระทั่งกราฟของการตอบสนองที่ได้มีลักษณะเป็นกราฟคงที่ (Uniform) คือกราฟที่ได้จะมีแอมพลิจูด (Amplitude) และเวลา (Period) ที่เท่ากันโดยตลอด วิธีการตอบสนองแบบนี้เราจะเรียกว่า วิธีการไซเคิลแบบต่อเนื่อง (Continuous Cycling Method) ค่า $K_c$ ที่ได้จะมีค่าเท่ากับค่า $K_{cu}$ ซึ่งค่า $K_{cu}$ นี้เราจะนำไปใช้ในการหาค่าการควบคุมสัดส่วน (Proportional Control) ของค่าพีไอดี โดยวิธีของ"ซีเกลอร์-นิโคลส์" (Ziegler-Nichols)

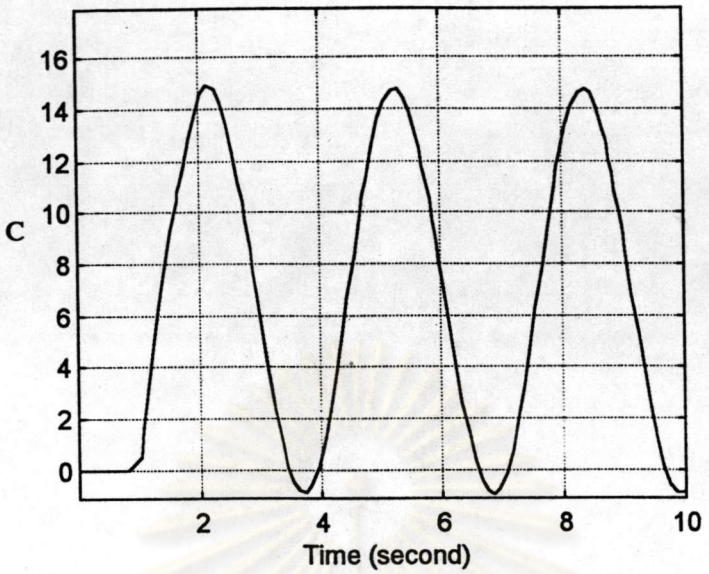
นิยามของ $K_{cu}$ ค่า $K_c$ หรือเรียกอีกชื่อว่า "Ultimate gain" คือ ค่า $K_c$ ที่มากที่สุดของตัวควบคุมกระบวนการที่ทำให้กระบวนการแบบวงจรปิด (Closed loop) มีเสถียรภาพที่ดีเมื่อใช้ค่าสัดส่วน (Proportional) ในการควบคุมกระบวนการเพียงอย่างเดียว

จากรูปจะแสดงถึงลักษณะของการตอบสนองที่ค่า $K_{cu}$ ต่างๆ กัน

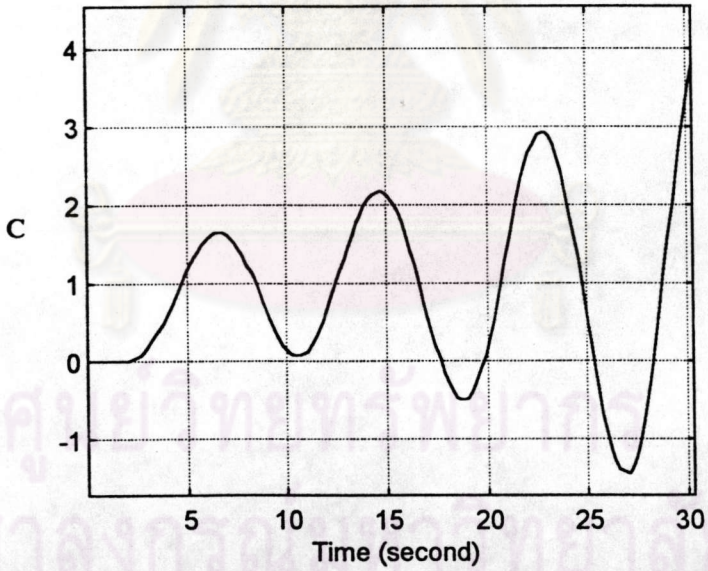



รูปที่ 3.18 $K_c > K_{cu}$

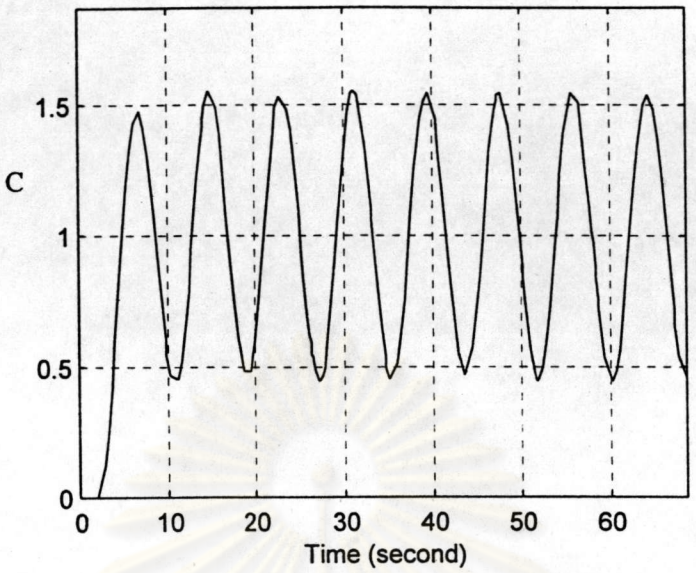
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย




รูปที่ 3.19 $K_c = K_{cu}$




รูปที่ 3.20 $K_c > K_{cu}$ (without saturation)




รูปที่ 3.21 $K_c > K_{cu}$ (with saturation)

**a)** $\mathbf{K_c < K_{cu}}$ ในกรณีที่การตอบสนอง (Response) เกิดแบบโอเวอร์แดม (Overdamp) และค่า $K_c$ น้อยกว่าค่า $K_{cu}$ มากๆ การตอบสนองจะเกิดการแกว่ง (Oscillatory) แก้ไขโดยเพิ่มค่า $K_c$ จนได้การตอบสนองเป็นแบบไซเคิลแบบต่อเนื่อง (Continuous Cycling)

**b)** $\mathbf{K_c = K_{cu}}$ ในกรณีนี้การตอบสนอง (Response) จะมีค่าแอมพลิจูด (Amplitude) และเวลา (Period) ที่คงที่ ในลักษณะนี้เราเรียกว่าการไซเคิลแบบต่อเนื่อง (Continuous Cycling ) ค่า $K_c$ ที่ได้จะมีค่าสูงสุด

**c)** $\mathbf{K_c > K_{cu}}$ **(w/o saturation)** ในกรณีนี้กระบวนการจะไม่มีเสถียรภาพ เนื่องจากไม่ได้คำนึงถึงค่า saturation การตอบสนองที่ได้จะเป็นอันเดอร์แดม (Underdamp)

**d)** $\mathbf{K_c > K_{cu}}$ **(with saturation)** ในกรณีนี้กระบวนการจะมีเสถียรภาพดีกว่าแต่ค่า $K_{cu}$ ที่ได้จะมีค่า มากจนเกินไป จนมีผลทำให้การควบคุมกระบวนการทำได้ไม่ดีนัก

**ขั้นที่ 2 วิธีการควบคุมแบบการไซเคิลแบบต่อเนื่อง (Continuous Cycling Method)**

วิธีการควบคุมแบบการไซเคิลแบบต่อเนื่องนี้ซีเกลอร์และนิโคลส์ ได้เป็นผู้นำเสนอครั้งแรก ในปี ค.ศ.1942 โดยการนำค่า $K_{cu}$ ที่ได้จากการทำลองผิดลองถูก (Trial and Error) มาหาค่าพีไอดี ได้ดังตารางที่ 3.1 จากหนังสือ Process Dynamic and Control

โดย Seborg (ตารางที่ 13.2 หน้า 298)

**ตารางที่ 3.1 การหาค่า PID**

| Controller | $K_c$ | $\tau_I$ | $\tau_D$ |
|---|---|---|---|
| P | $0.5K_{cu}$ | - | - |
| PI | $0.45K_{cu}$ | $P_u/1.2$ | - |
| PID | $0.6K_{cu}$ | $P_u/2$ | $P_u/8$ |

ค่าพีไอดี ที่ได้ก็จะนำไปแทนค่าลงในตัวควบคุมกระบวนการต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย