บทที่ 3
โปรแกรมช่วยวิเคราะห์และออกแบบระบบควบคุมแมทแล็บ

การออกแบบระบบควบคุมแบบป้อนกลับ การควบคุมแบบป้อนหน้า การควบคุมแบบคาสเคด หรือการควบคุมแบบอื่นๆ จะมีความเกี่ยวข้องเป็นอย่างยิ่งที่ต้องมีการแสดงผลการคำนวณด้านวิศวกรรมเหล่านั้นอย่างถูกต้องและรวดเร็ว โปรแกรมช่วยวิเคราะห์และออกแบบระบบควบคุมแมทแล็บเป็นโปรแกรมหนึ่งที่มีบทบาทในการคำนวณในลักษณะนี้ โดยเฉพาะอย่างยิ่ง แมทแล็บมีรูปลักษณะที่สามารถแก้ปัญหาแมทริกซ์และเวคเตอร์ได้เป็นอย่างดี มีการแสดงผลด้านกราฟิกที่ดีและมีความยืดหยุ่นในการใช้งานเป็นอย่างดี ภายใต้โปรแกรมแมทแล็บนี้ยังมีโปรแกรมอีกชุดหนึ่งซึ่งเรียกว่าซิมูลิงค์ ซึ่งเป็นโปรแกรมที่มีความสามารถในการทำการเลียนแบบได้ดีมากโปรแกรมหนึ่ง ทำให้โปรแกรมทั้งสองเป็นโปรแกรมที่มีความเหมาะสมในการนำมาใช้ในการออกแบบและวิเคราะห์ระบบควบคุม ตลอดจนเป็นโปรแกรมที่สามารถที่จะใช้งานร่วมกับโปรแกรมอื่นๆ ได้เป็นอย่างดี เช่น โปรแกรมภาษาซีและโปรแกรมภาษาโฟรแทรน เป็นต้น จากคุณสมบัติดังกล่าวทำให้โปรแกรมนี้มีความยืดหยุ่นในการใช้งานและมีการนำโปรแกรมนี้ไปใช้งานอย่างแพร่หลาย

แมทแล็บ (Matlab) คือ โปรแกรมที่ใช้ในงานการคำนวณทางคณิตศาสตร์ขั้นสูงที่สามารถแสดงผลการคำนวณออกมาเป็นภาพให้เห็นได้ แมทแล็บจะประกอบไปด้วย

- การวิเคราะห์ทางคณิตศาสตร์ (Numerical analysis)
- การคำนวณทางแมทริกซ์ (Matrix compensation)
- กระบวนการการสัญญาณ (Signal processing)

- กราฟฟิก (Graphic)

ทั้ง 4 ส่วนนี้จะอยู่ในลักษณะที่ง่ายต่อการใช้ สมการทางคณิตศาสตร์ก็จะถูกจัดให้อยู่ในสภาพที่ง่ายต่อการใช้โดยไม่มีไวยากรณ์หรือกฎเกณฑ์มาเกี่ยวข้องให้ยุ่งยากเหมือนโปรแกรมทั่วๆไป

3.1 คุณลักษณะของโปรแกรมแมทแล็บ

แมทแล็บ (Matlab) เป็นชื่อย่อของโปรแกรม "MATrix LABoratory" แมทแล็บได้ถูกสร้างขึ้นจากการพยายามที่จะพัฒนาโปรแกรมที่ใช้ในการคำนวณทางแมทริกซ์ภายใต้โครงการที่ชื่อว่า "LINPACK and EISPACK"

แมทแล็บจะเป็นโปรแกรมที่สามารถโต้ตอบกับผู้ใช้งานได้ในทันทีทันใด (Interactive) โดยข้อมูลหรือโจทก์ทางคณิตศาสตร์ ที่จะใช้แมทแล็บคำนวณจะต้องอยู่ในรูปของแมทริกซ์และไม่จำเป็นต้องมีการแสดงหน่วยของข้อมูลนั้น โครงสร้างของโปรแกรม แมทแล็บ จะอยู่ในรูปของตัวแปรแมทริกซ์ ซึ่งจะไม่มีหน่วย มีคำสั่งเฉพาะในการใช้งาน ซึ่งคำสั่งต่างๆ นั้นมีลักษณะการใช้งานหมือนการเขียนสมการทางคณิตศาสตร์ โปรแกรม แมทแล็บ มีความสามารถในการแก้ปัญหาทางด้านเรขาคณิตเชิงเส้น และแมทริกซ์ ได้เป็นอย่างดี

แมทแล็บได้มีการพัฒนามาอย่างต่อเนื่อง ในการศึกษาแมทแล็บจะถูกใช้ไปในการสอนทางคณิตศาสตร์, การวิจัย ฯลฯ ในทางอุตสาหกรรมแมทแล็บจะใช้ในการวิจัยทางวิศวกรรม และการแก้ไขปัญหาทางคณิตศาสตร์เช่น การควบคุมกระบวนการแบบอัตโนมัติ, การวิจัยสัญญาณของกระบวน และ การผลิตในรูปดิจิตัล

การทำงานของแมทแล็บจะเป็นฟังก์ชันของคำสั่งที่อยู่ในรูปของโปรแกรม "M-file" ที่ถูกบรรจุอยู่ในพื้นที่ที่เรียกว่า "Toolboxes" ซึ่งคำสั่งเหล่านี้สามารถนำมาใช้แก้ปัญหาทางคณิตศาสตร์ดังนี้

3.1.1 การคำนวณเกี่ยวกับแมทริกซ์

โปรแกรมแมทแล็บสามารถทำแมทริกซ์ทรานส์โพส, การคูณแมทริกซ์, การหาดีเทอร์มิแนนท์, การทำอินเวอร์ทแมทริกซ์, หาค่าไอเกน, ตลอดจนการแก้สมการเชิงเส้นการประมาณค่าเป็นต้น

3.1.2 การคำนวณโพลิโนเมียล

โปรแกรม แมทแล็บ สามารถใช้ในการคำนวณเกี่ยวกับโพลิโนเมียล เช่น การหารากของโพลิโนเมียล, การหาคอนโวลูชั่น (Convolution) และดีคอนโวลูชั่น (Deconvolution) การหารโพลิโนเมียล, การหาสมการถดถอยแบบโพลิโนเมียล เป็นต้น

3.1.3 การจัดการเกี่ยวกับเวคเตอร์

โปรแกรมแมทแล็บสามารถนำมาคำนวณผลรวมแบบเวคเตอร์, การหาค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน เป็นต้น

3.1.4 การจัดการเกี่ยวกับการแสดงผลกราฟ

โปรแกรม แมทแล็บ มีการแสดงผลเป็นกราฟให้เลือกทั้งสิ้น 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

3.1.5 การจัดการต่างๆ ของกระบวนการสัญญาณ

3.1.6 การวิเคราะห์ไม่เชิงเส้น

3.1.7 การประเมินและวิเคราะห์ผลทางสถิติ

นอกจากคำสั่งพื้นฐานเฉพาะในโปรแกรมแมทแล็บที่เหมือนสมการคณิตศาสตร์แล้ว โปรแกรมแมทแล็บยังสามารถใช้งานหรือพัฒนางานร่วมกับโปรแกรมพัฒนาระบบอื่นๆ ได้อีก เช่น โปรแกรมภาษาซี, โปรแกรมภาษาโฟรแทรน เป็นต้น

แมทแล็บถูกเขียนขึ้นครั้งแรกโดยใช้ภาษาฟอร์แทรน (Fortran) โดยคลีฟ โมลเลอร์ (Cleve Moller) จากนั้นก็ได้รับการพัฒนาจากโปรแกรมเมอร์อีกหลายท่านในโครงการ "LINPACK and EISPACK" ปัจจุบันแมทแล็บถูกเขียนขึ้นโดยใช้ภาษาซี (C language)

นอกจากนี้ในโปรแกรมแมทแล็บจะมีโปรแกรมซิมูลิงค์ติดตั้งอยู่ด้วย ซึ่งโปรแกรมซิมูลิงค์นี้จะใช้ในการเลียนแบบระบบที่มีการเปลี่ยนแปลงไปตามเวลาและจะสามารถแสดงผลออกมาในรูปของกราฟฟิกโดยจะใช้เมาส์ในการทำงาน ผู้ใช้สามารถสร้างแบบจำลองกระ บวนการผลิตได้ด้วยการสร้างบล็อกไดอะแกรม (Block diagram) ลงในจอคอมพิวเตอร์ บล็อกไดอะแกรมเหล่านี้สามารถปรับเปลี่ยนค่าได้ตามต้องการและจะถูกควบคุมด้วยโปรแกรม "M-file" ของแมทแล็บ อีกต่อหนึ่ง

3.2 โปรแกรมซิมูลิงค์ (Simulink)

ซิมมูลิงค์ คือโปรแกรมที่ใช้สำหรับการทำเลียนแบบระบบที่มีการเปลี่ยนแปลง โดยจะเป็นโปรแกรม ที่ขยายออกมาจากโปรแกรมแมทแล็บ โปรแกรมซิมูลิงค์นี้จะประกอบไปด้วย ฟังก์ชัน

ต่างๆ มากมายและฟังก์ชันเหล่านี้สามารถนำมาประกอบกัน เพื่อให้ได้ระบบที่มีความแตกต่างและสามารถปรับเปลี่ยนได้ตามต้องการ

ซิมูลิงค์จะมีทิศทางในการใช้ 2 แนวทางคือ

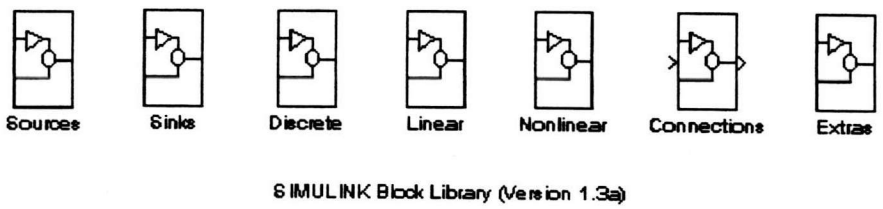
- ใช้ในการจำลองแบบจำลองหรือออกแบบจำลอง
- ใช้ในการวิเคราะห์แบบจำลองในรูปแบบต่างๆกัน

โดยทั่วไปแล้วเราจะใช้ทั้ง 2 แนวทางมาทำงานร่วมกัน คือ อันดับแรกจะต้องทำการจำลองแบบหรือออกแบบจำลองให้ได้ตามที่ต้องการ โดยการนำบล็อกไดอะแกรมต่างๆที่มีอยู่ในฟังก์ชันมาต่อเข้าด้วยกันโดยใช้เมาส์ จากนั้นก็จะทำการวิเคราะห์แบบจำลองที่ได้ด้วยการเลือกการปรับเปลี่ยนค่าโปรแกรมต่างๆที่ปรากฏอยู่บนเมนูหน้าจอ ซิมูลิงค์นี้เป็นโปรแกรมที่ใช้งานบนวินโดว์และเมื่อทำการปรับเปลี่ยนเสร็จเรียบร้อย ก็จะทำการเลียนแบบโปรแกรม โดยเราจะทราบถึงผลการเลียนแบบได้จาก การดูผลของกราฟที่ปรากฏอยู่บนจอ และสามารถที่จะทำการพิมพ์กราฟดังกล่าวออกมาได้ทางเครื่องพิมพ์

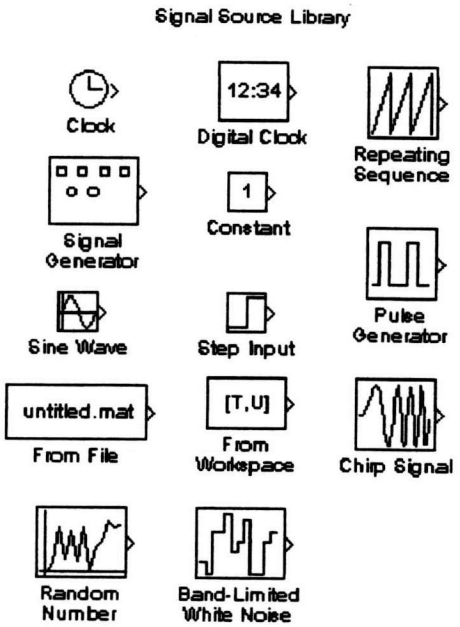
3.2.1. การจำลองแบบแบบจำลองหรือออกแบบแบบจำลอง

การใช้งานโปรแกรมซิมูลิงค์ในลักษณะนี้ จะเป็นการสร้างบล็อกไดอะแกรมบนหน้าจอแสดงผลของคอมพิวเตอร์ซึ่งองค์ประกอบต่างๆของบล็อกไดอะแกรม จะสามารถนำมาใช้ได้จากห้องสมุดความจำรูปบล็อกไดอะแกรม (Block Diagram Library) ดังรูป 3.1 ถึงรูป 3.8 โดยการใช้เมาส์ลากลงมาใส่ในเวอร์คชีต (Work Sheet) และใช้เมาส์ลากเส้นเชื่อมบล็อกแต่ละบล็อกเข้าด้วยกัน ตามวิธีการสร้างแบบจำลองกระบวนการผลิตแบบต่างๆในโปรแกรมซิมูลิงค์ดังนี้คือ

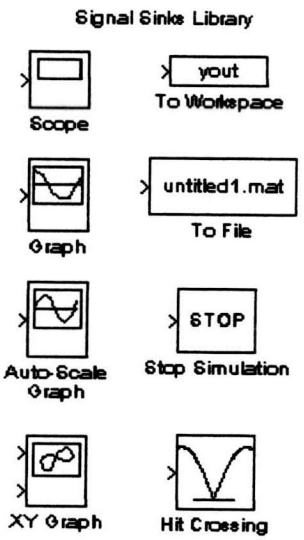
ก. ห้องสมุดความจำบล็อกไดอะแกรมจะประกอบไปด้วย บล็อกแหล่งกำเนิดสัญญาณควบคุมบล็อกเชิงเส้น บล็อกไม่เป็นเชิงเส้น แบบดีสครีต แบบดิจิตอลและแบบต่อเนื่อง ฯลฯ



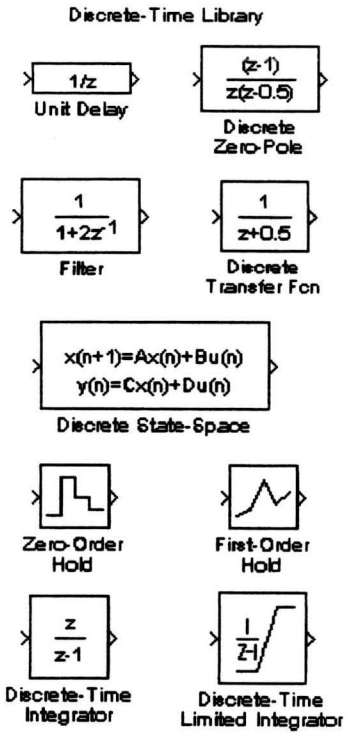
รูปที่ 3.1 แสดงถึงห้องสมุดของบล็อคไดอะแกรมของซิมูลิงค์



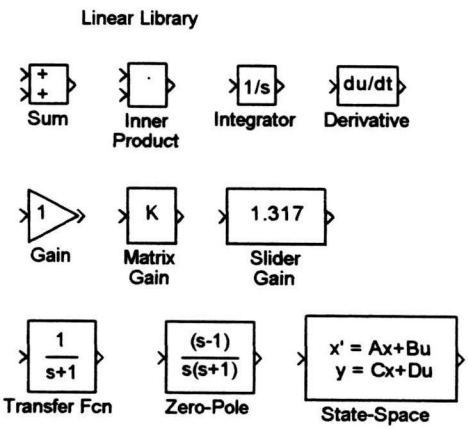
รูปที่ 3.2 แสดงถึงบล็อกที่เป็นตัวกำเนิดสัญญาณแบบต่างๆ



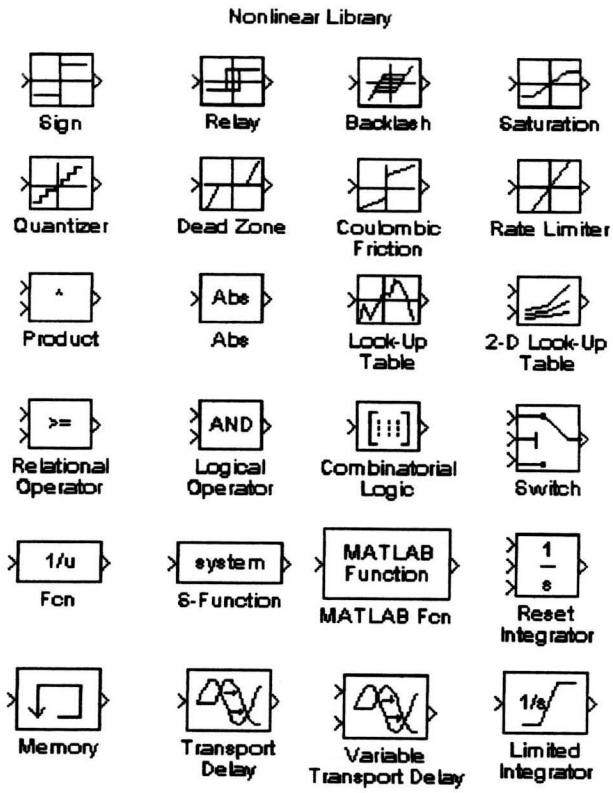
รูปที่ 3.3 แสดงถึงรูปแบบของสัญญาณส่งออกแบบต่างๆ



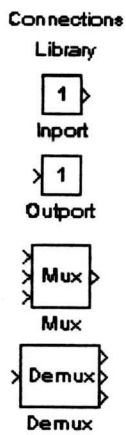
รูปที่ 3.4 แสดงถึงบล็อกไดอะแกรมของดีสครีตแบบต่างๆ



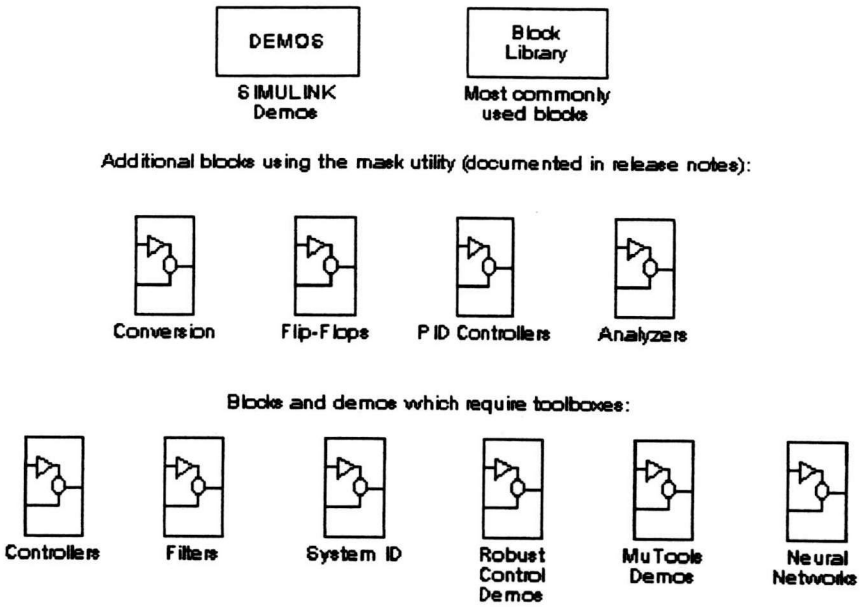
รูปที่ 3.5 แสดงถึงบล็อกไดอะแกรมและเกนแบบต่างๆ ที่เป็นแบบเชิงเส้น



รูปที่ 3.6 แสดงถึงบล็อกไดอะแกรมของแมทแล็บและค่าเกนแบบต่างๆ ที่เป็นแบบไม่เชิงเส้น



รูปที่ 3.7 แสดงถึงบล็อกไดอะแกรมของตัวเชื่อมสัญญาณแบบต่างๆ



รูปที่ 3.8 แสดงถึงบล็อกไดอะแกรมของอุปกรณ์ควบคุมเพิ่มเติม

ข. การเชื่อมต่อบล็อกทั้งแบบสกาลาร์และเวคเตอร์ โดยสามารถทำแบบมีอินพุทค่าเดียวเอาท์พุทเดียวหรือแบบอินพุทหมู่ เอาท์พุทหมู่ โดยโปรแกรมสามารถแปลงสัญญาณ ให้เป็นเวคเตอร์หรือสกาลาร์ได้ โดยมัลติเพล็กเซอร์ และ ดีมัลติเพล็กเซอร์

ค. การแสดงสัญญาณอินพุทและเอาท์พุท สัญญาณอินพุทสามารถกำหนดให้เป็นแบบสเต็พแบบซายน์ หรือแบบสัญญาณรบกวน (White Noise) ส่วนสัญญาณเอาท์พุทอาจแสดงผลในรูปของกราฟ หรือข้อมูลของเวอร์คชีตบนพื้นฐานของเวลาจริง

ง. การจัดแบบจำลองออกเป็นระดับชั้นต่างๆกัน (Hierachical Model) อาจจะแบ่งเป็นจากบนสู่ล่างหรือจัดแบบเป็นกลุ่มๆ โดยไม่มีข้อจำกัดของกลุ่มหรือจำนวนบล็อก

ฉ. การขยายห้องสมุดบล็อกไดอะแกรม ผู้ใช้งานสามารถสร้างหรือขยายลักษณะของบล็อกที่ใช้งานในด้านต่างๆ และเก็บไว้เป็นห้องสมุดบล็อกไดอะแกรมได้ 4 รูปแบบ คือ

- กำหนดหรือสร้างในระบบย่อย

- สร้างจากเอ็มไฟล์ (M-File) ในโปรแกรมแมทแล็บ

- ใช้โปรแกรมภาษาซี หรือ ภาษาฟอร์แทรน

- โดยการเปลี่ยนแปลงจากบล็อกไดอะแกรมที่มีอยู่แล้ว

3.2.2 การวิเคราะห์แบบจำลอง

ภายหลังจากที่ผู้ใช้ได้สร้างหรือกำหนดแบบจำลองเสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปของผู้ใช้งานคือ ต้องเลือกวิธีการในการวิเคราะห์ ซึ่งในโปรแกรมซิมูลิงค์ มีให้เลือกทั้งสิ้น 3 วิธี คือ การทำการเลียนแบบ การวิเคราะห์เชิงเส้น และการทดสอบหาจุดสมดุลย์

ก. การทำการเลียนแบบ

ขั้นตอนอันดับแรกในการทำเลียนแบบ จะเป็นขั้นตอนการแก้สมการดิฟเฟอเรนเชียลที่มีอยู่ทั้งหมดในแบบจำลองเปรียบเทียบกับเวลา โดยโปรแกรมซิมูลิงค์จะมีเครื่องมือในการแก้สมการอยู่ทั้งหมด 6 แบบ คือ

- รั้งจ์-กัตตา อันดับที่ 3 (Runge-Kutta Third Order)
- รั้งจ์-กัตตา-เฟลเบอร์ก อันดับที่ 5 (Runge-Kutta-Fehlberg Fifth Order)
- การทำนายและการแก้ค่าโดยอดัมส์ (Adams Predict-Corrector)
- วิธีการของเกียร์สำหรับระบบแบบสติฟ (Gear's Method for Stiff Systems)
- การแยกพลวัตเชิงเส้น (Linear Dynamics Extraction)
- วิธีการของออยเลอร์ (Euler Method)

ซึ่งข้อมูลต่างๆ ที่ได้ออกมาตามเวลาต่างๆ สามารถแสดงได้ด้วยกราฟในโปรแกรมซิมูลิงค์และข้อมูลแบบตารางในโปรแกรมแมทแล็บ ซึ่งสามารถนำไปใช้วิเคราะห์โดยโปรแกรมวิเคราะห์ผลอื่นๆได้ รวมทั้งสามารถเก็บเป็นแฟ้มข้อมูลที่นำมาใช้ประโยชน์ในภายหลังได้

ข. การวิเคราะห์เชิงเส้น

การวิเคราะห์ในโปรแกรมซิมูลิงค์ จะใช้สมการหลัก 2 สมการ คือ

$$\dot{X} = Ax + Bu \quad (3.1)$$

$$\dot{Y} = Cx + Du \quad (3.2)$$

ทั้งสมการ 3.1 และ 3.2 จะใช้คำนวณวิเคราะห์ได้ทั้งแบบตัวแปรต่อเนื่องและแบบดีสครีต

ค. การทดสอบหาจุดสมดุลย์

เป็นเครื่องมือชุดที่สามที่ใช้หาจุดสมดุลย์ ซึ่งผู้ใช้โปรแกรมซิมูลิงค์ สามารถกำหนดค่าต่างๆ เช่น อินพุท เอาท์พุท และค่าของสภาวะให้แก่โปรแกรม เพื่อให้โปรแกรมคำนวณหาจุดสมดุลย์ให้

3.3 เครื่องมือในการวิเคราะห์อื่นๆ

โปรแกรมแมทแล็บ ได้จัดเครื่องมือต่างๆ ไว้หลายชุดเพื่อให้เหมาะในการใช้งานในลักษณะต่างๆ ได้แก่

- การจัดการด้านกระบวนการสัญญาณ
- การจำแนกระบบ
- การทำออพติไมซ์
- ข่ายงานนิวรัล
- ระบบควบคุม
- การควบคุมรอบัสต์ของระบบ
- การวิเคราะห์ระดับจุลภาค (Micro Analysis)
- คณิตศาสตร์แบบสัญญลักษณ์

- การวิเคราะห์ทางสถิติ

3.4 ตัวอย่างการใช้งาน

จากความสามารถของโปรแกรมแมทแล็บ และโปรแกรมซิมูลิงค์ ในการวิเคราะห์แก้ไขปัญหาด้านวิทยาศาสตร์และวิศวกรรม ทำให้ผู้ใช้งานสามารถลดเวลาในการแก้สมการต่างๆ ลงไปได้มาก ตัวอย่างที่ 3.1 และตัวอย่างที่ 3.2 จะเป็นการนำโปรแกรมฯ มาใช้ในการควบคุมแบบชดเชยค่าเดดไทม์เปรียบเทียบกับการควบคุมแบบอื่นๆ

ตัวอย่างที่ 3.1 เป็นการเปรียบเทียบการควบคุมแบบป้อนกลับ และการควบคุมแบบชดเชยค่าเดดไทม์ โดยวิธีการทำนายค่าแบบสมิธ (Smith Predictor) ซึ่งการควบคุมในตัวอย่างนี้มีแบบจำลองของกระบวนการเป็นอันดับที่ 4 ซึ่งจะแสดงทรานส์เฟอร์แลกอย่างชัดเจนตามสมการที่ 3.3

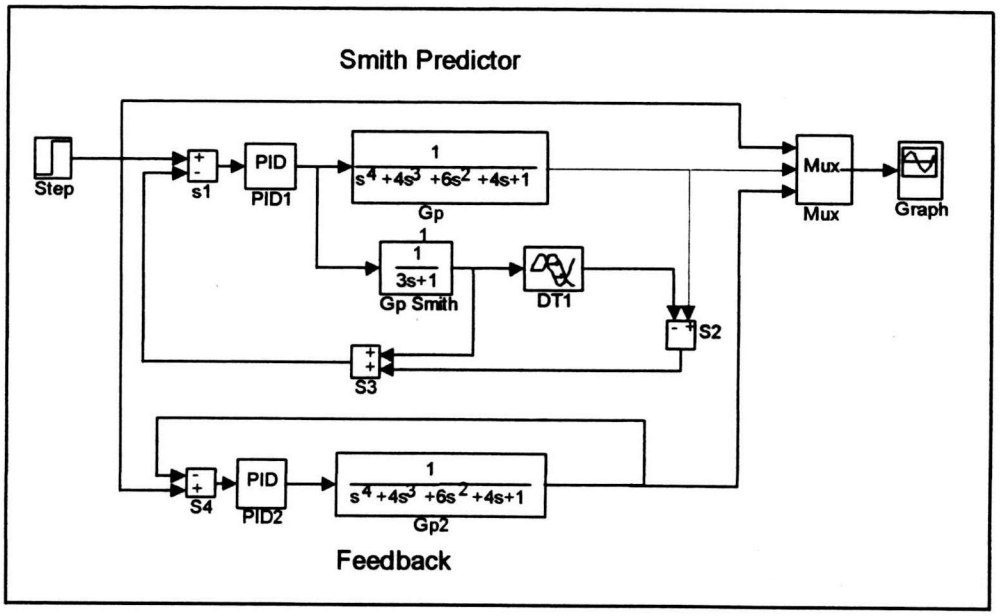
$$G_p(S) = \left(\frac{1}{S+1}\right)^4 \tag{3.3}$$

และเมื่อเปลี่ยนสมการที่ 3.3 เป็นแบบจำลองอันดับหนึ่งที่มีเดดไทม์ จะได้ดังนี้

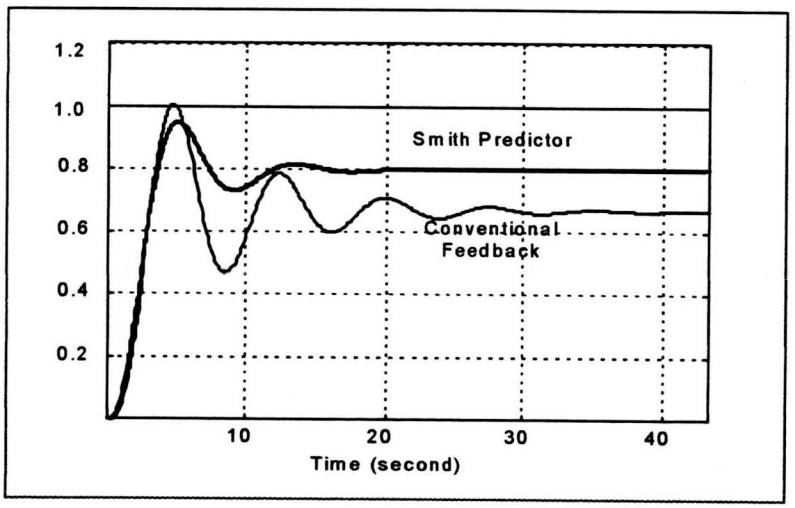
$$G_p(S) = \frac{1}{(3S+1)} e^{-1.5S} \tag{3.4}$$

ทำการสร้างแบบจำลองการควบคุมแบบป้อนกลับและการจำลองที่มีการชดเชยค่าเดดไทม์โดยวิธีการทำนายค่าแบบสมิธตามรูปที่ 3.9 โดยที่แบบจำลองแบบป้อนกลับตามสมการที่ 3.3 เมื่อทำการจูนโดยวิธี Z-N สามารถหาค่า $K_{cu} = 4.0$ และค่า $P_u = 2\pi$ ดังนั้นจึงกำหนดค่าตัวควบคุมให้ $K_c = 2.0$ เมื่อกำหนดให้เกิดการเปลี่ยนแปลงแบบเสต็พขึ้นในกระบวนการ จะพบว่ามีการแกว่งซึ่งค่อยๆ ลดลง โดยให้ค่าออฟเซทประมาณ 0.333 ตามรูปที่ 3.9 ส่วนการควบคุมที่มีการชดเชยค่าเดดไทม์ โดยสมการที่ 3.4 เป็น G_p จะพบว่า โดยวิธีการที่จูนที่เหมือนกัน จะกำหนดค่าตัวควบคุมที่ $K_c = 4.0$ ซึ่งค่า

นี้เป็นค่า K_{cu} ของวิธีการแรก และเมื่อมีการเปลี่ยนแปลงแบบเสต็พ จะให้ค่าออฟเซทประมาณ 0.2 ตามรูปที่ 3.10



รูปที่ 3.9 แสดงแบบจำลองการควบคุม



รูปที่ 3.10 แสดงผลการควบคุมของตัวอย่างที่ 3.1

ตัวอย่างที่ 3.2 จะเป็นการเปรียบเทียบผลการควบคุมที่มีเดดไทม์โดยวิธีการต่างๆ กัน 3 วิธี คือ การทำนายค่าเชิงวิเคราะห์ (Analytical Predictor - AP) การทำนายค่าเชิงวิเคราะห์แบบทั่วไป

(General Analytical Predictor - GAP) และการควบคุมภายใน (Internal Model Control - IMC) โดยมีทรานส์เฟอร์ฟังก์ชัน ดังนี้

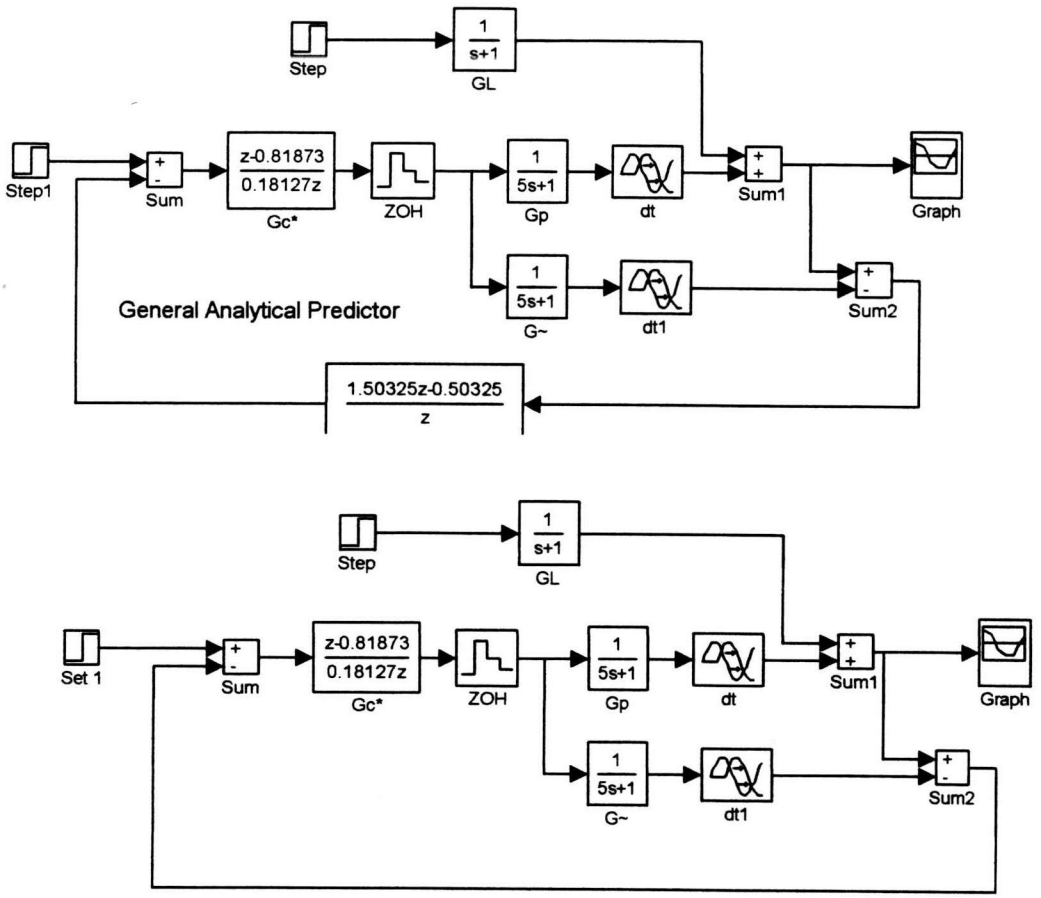
$$G_p(S) = \frac{e^{-2S}}{5S+1} \quad (3.5)$$

$$G_L(S) = \frac{1}{S+1} \quad (3.6)$$

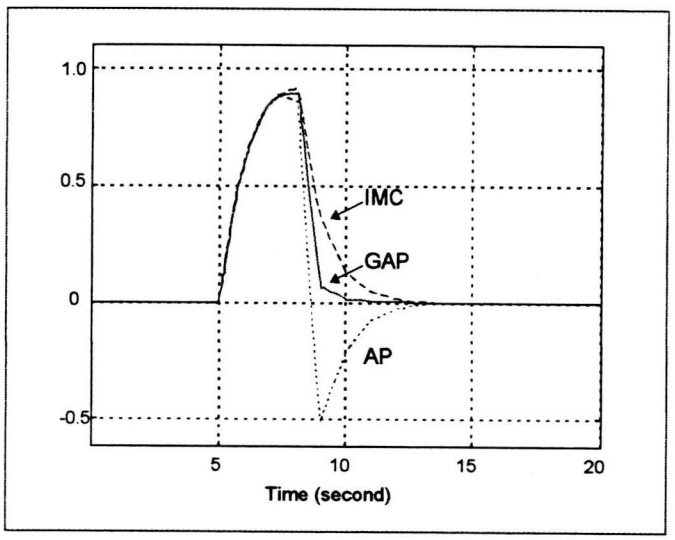
โดยกำหนดให้มีการเปลี่ยนแปลงของโหลดเกิดขึ้นที่เวลา 5 วินาที และกำหนดว่าแบบจำลองที่ใช้ในการควบคุมไม่มีความผิดพลาด สร้างแบบจำลองทั้งหมดที่ต้องใช้ในการควบคุมตามรูปที่ 3.11 และผลการควบคุมจะเป็นไปตามรูปที่ 3.12

ซึ่งจากตัวอย่างที่ 3.1 และตัวอย่างที่ 3.2 ได้ผลการควบคุมที่สอดคล้องกับเอกสารอ้างอิงตามตัวอย่างที่ 18.4 ของหนังสือ Process Control Analysis and Control โดย Coughaunowr, D.R. 1991 หน้าที่ 270 และตัวอย่างที่ 26.7 ของหนังสือ Process Dynamics and Control โดย DALE E. SEBORG, 1989 หน้าที่ 641

ส่วนรายละเอียดอื่นๆ ที่เกี่ยวข้องกับการใช้งานของโปรแกรมทั้งสองนี้สามารถศึกษาได้จากคู่มือการใช้งานของแมทแล็บ



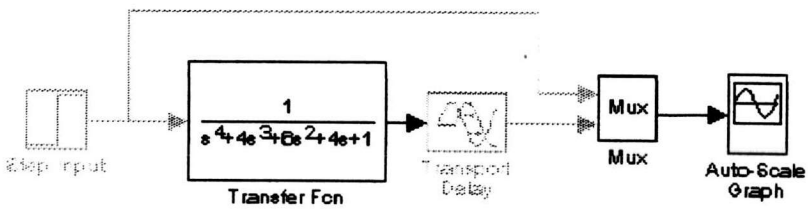
รูปที่ 3.11 แสดงแบบจำลองการควบคุมด้วยวิธีการต่างๆ



รูปที่ 3.12 แสดงผลการควบคุมของตัวอย่างที่ 3.2

3.5 การจัดบล็อกไดอะแกรมที่มีอันดับที่มากกว่าหนึ่งโดยวิธีการเปลี่ยนแปลงแบบสเต็พ (Fitting Higher Order Model using Step Change)

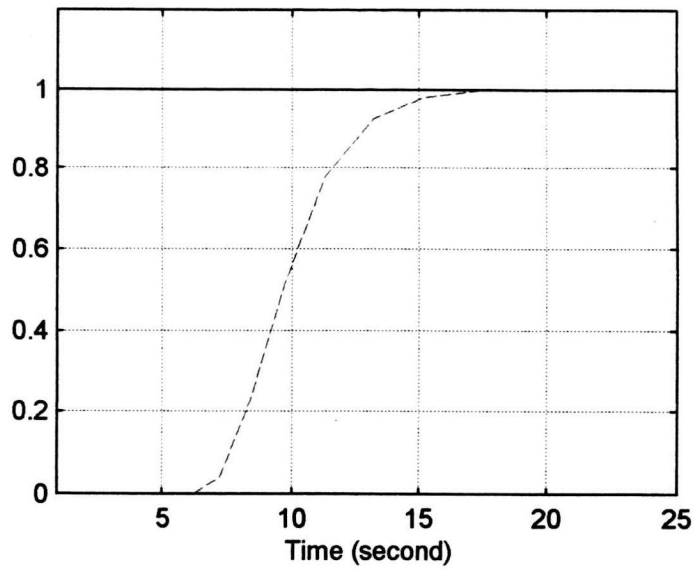
จากการที่กระบวนการผลิตของโรงงานอุตสาหกรรมทั่วไปมักจะมีอันดับที่มากกว่าหนึ่งขึ้นไปเสมอ อันจะมีผลทำให้การปรับแต่งตัวควบคุมกระบวนการทำได้ลำบากมากโดยเฉพาะการทำการชดเชยเดดไทม์ เพราะเราจะไม่ทราบเลยว่ากระบวนการจะมีอันดับเป็นเท่าไร ดังนั้นในการแก้ปัญหาที่เกิดขึ้นนี้จะทำได้โดยการสร้างแบบจำลองของกระบวนการดังกล่าว ให้มีอันดับเท่ากับหนึ่งเสีย ก่อน โดยการนำทรานส์เฟอร์ฟังก์ชันของกระบวนการ (Process Transfers Functions) มาทำการ เปลี่ยนแปลงแบบสเต็พ (Step change) แล้วนำการตอบสนอง (Response) ที่ได้มาหาค่า τ และค่า θ (เดดไทม์)จากบล็อกไดอะแกรมของรูป 3.13 จะพบว่าทรานส์เฟอร์ฟังก์ชันของกระบวนการมีอันดับ เท่ากับ 4 ดังนั้นจะของนำทรานส์เฟอร์ฟังก์ชันของกระบวนการนี้มาทำการเปลี่ยนแปลงแบบสเต็พ เพื่อทำการเปลี่ยนกระบวนการดังกล่าวให้มีอันดับเท่ากับหนึ่ง ดังรูป 3.16 จะเป็นลักษณะของบล็อก ไดอะแกรมที่จะทำการเปลี่ยนแปลงแบบสเต็พ



Deadtime = Transport delay = 6 Sec

รูปที่ 3.13 แสดงลักษณะของบล็อกไดอะแกรมที่จะทำการเปลี่ยนแปลงแบบสเต็พ

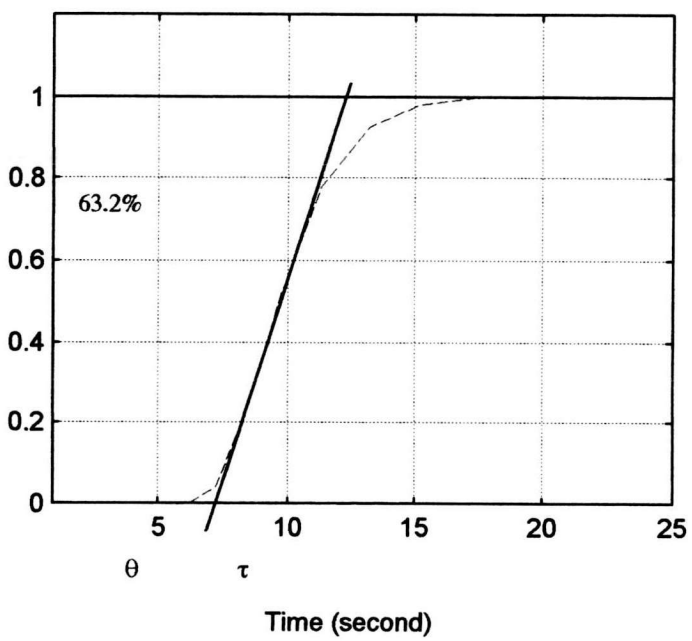
การตอบสนองของการทำการเปลี่ยนแปลงแบบสเต็พของบล็อกไดอะแกรมดังรูปที่ 3.13 จะมีลักษณะดังรูปที่ 3.14



รูปที่ 3.14 แสดงการเปลี่ยนแปลงแบบสเต็พของกระบวนการที่อันดับมากกว่าหนึ่ง

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

$$G_m(s) = \frac{e^{-\theta s}}{\tau s+1} \qquad (3.7)$$

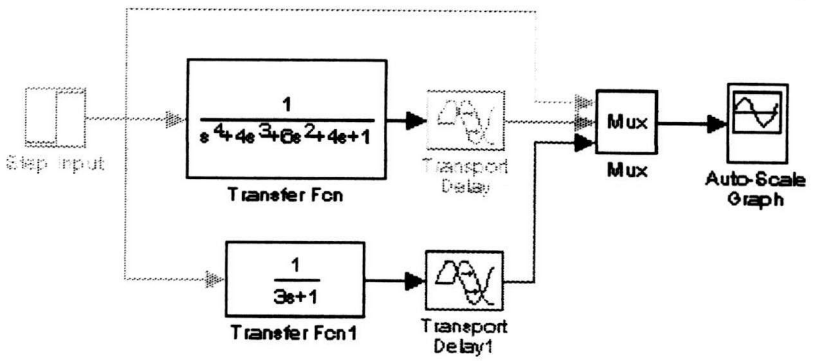


รูปที่ 3.15 แสดงการหาค่า θ และค่า τ ของ G_p ใหม่

จากวิธีการดังกล่าวข้างต้นทำให้สามารถหาค่า θ ได้เท่ากับ 7.3 และหาค่า τ ได้เท่ากับ 3 ดังนั้นจะได้ G_p ใหม่คือ

$$G_p(s) = \frac{e^{-7.3s}}{3s+1} \quad (3.8)$$

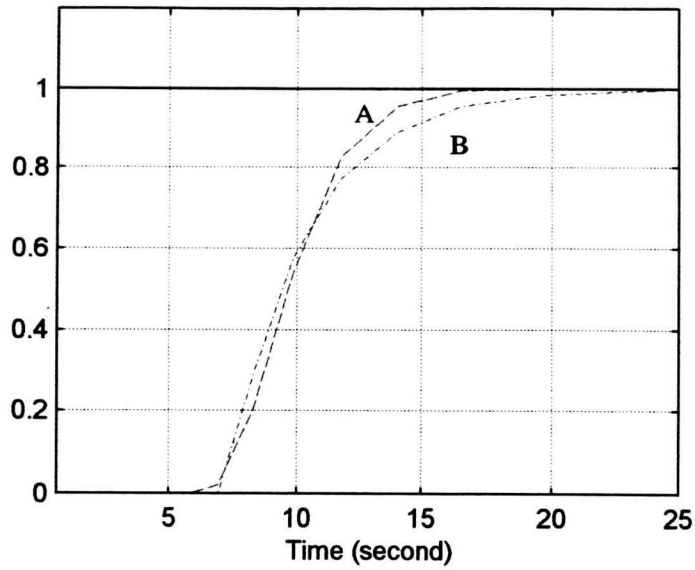
G_p ที่ได้จากสมการ 3.8 จะเป็น G_m ใหม่เราจะสามารถทดสอบ G_p ใหม่นี้ได้โดยการนำมาทำการเปลี่ยนแปลงแบบสเต็พ เปรียบเทียบกับ G_p เก่าดังรูป 3.16 จากนั้นนำการตอบสนอง ที่ได้มาวาดกราฟลงในแกนเดียวกันแล้วสังเกตถึงการเปลี่ยนแปลงดังรูปที่ 3.17 ถ้า G_p ใหม่ที่ได้ถูกต้องการตอบสนองก็ควรจะมีลักษณะคล้ายคลึงกัน



Deadtime = Transport Delay = 6

New Deadtime = Transport Delay1 = 7.3

รูปที่ 3.16 แสดงลักษณะของบล็อกไดอะแกรมที่จะทำการเปลี่ยนแปลงแบบสเต็พ



รูปที่ 3.17 แสดงการตอบสนองที่มีการเปลี่ยนแปลงแบบสเต็พของกระบวนการที่ทรานส์เฟอร์ฟังก์ชันมีอันดับมากกว่าหนึ่งและกระบวนการที่ทรานส์เฟอร์ฟังก์ชันได้รับการเปลี่ยนเป็นอันดับหนึ่ง โดยวิธีการแบบสเต็พ

$$A = G_p \tag{3.9}$$

$$B = G_m' \tag{3.10}$$

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

จากการที่สามารถหาแบบจำลองของกระบวนการผลิตได้ในรูปของทรานส์เฟอร์ฟังก์ชันที่มีอันดับเท่ากับหนึ่ง จะมีประโยชน์ในการทำการเลียนแบบ (Simulation) ของตัวชดเชยเดดไทม์ ซึ่งจะต้องสร้างแบบจำลองของกระบวนการผลิตให้ใกล้เคียงกับกระบวนการผลิตจริงมากที่สุด อีกทั้งจะต้องทำการเลียนแบบ (Simulation) ในรูปแบบของการกำหนดแบบจำลองผิดพลาด (Modeling Error)