บทที่4

วิธีการออกตัวควบคุมคาสเคดแบบหลายตัวแปร

วิธีการออกตัวควบคุมคาสเคดแบบหลายตัวแปร (Design Multivariable cascade control) แบ่งออกเป็น 2 แบบด้วยกัน คือ (2x1) และ (2x2)

4.1 การออกแบบระบบ (2x1) (one-input two-output)

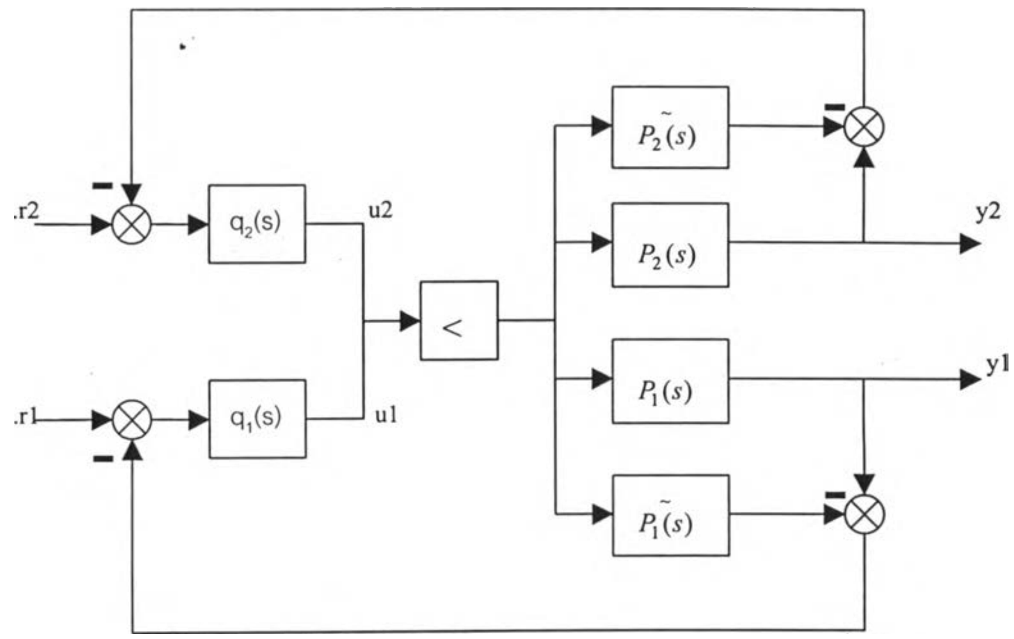
$$y_1 = p_1 u(s)$$

$$y_2 = p_2 u(s)$$

<u>ขั้นที่1</u> หาค่าตัวควบคุมสำหรับ y_1 คือ q_1จาก p_1 โดยวิธี IMC

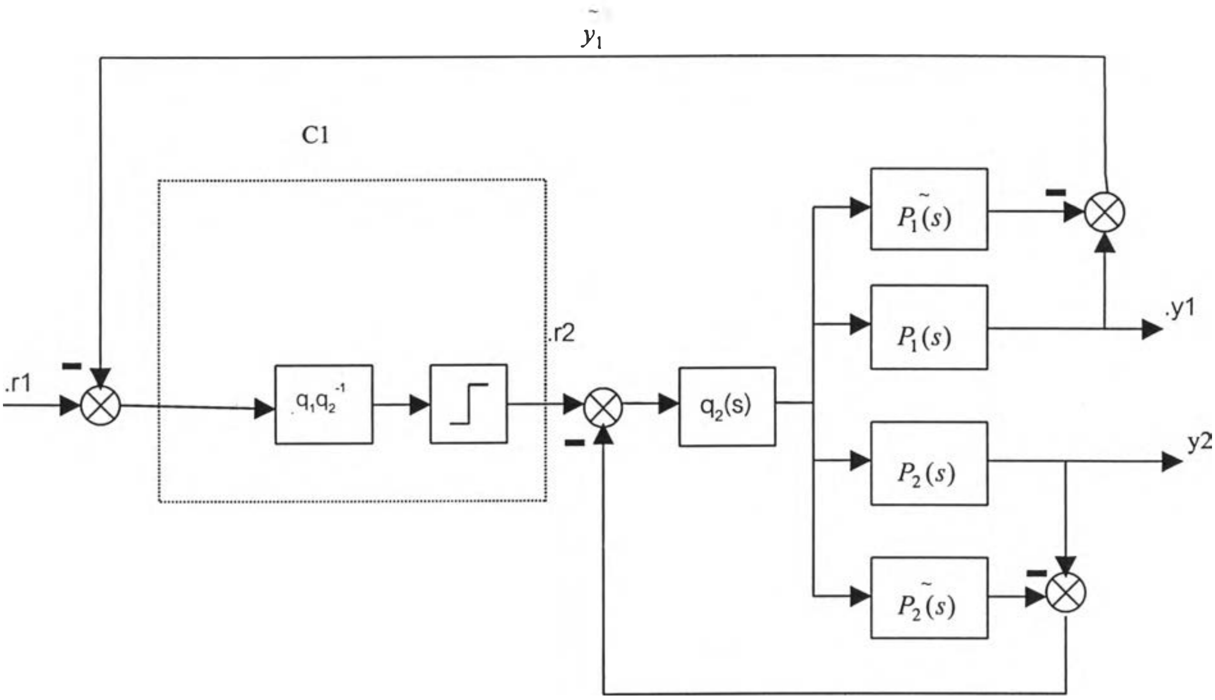
หาค่าตัวควบคุมสำหรับ y_2 คือ q_2จาก p_2 โดยวิธี IMC

<u>ขั้นที่2</u> จัดรูปแบบการควบคุมแบบโอเวอร์ไรด์ (override) ขึ้นก่อน



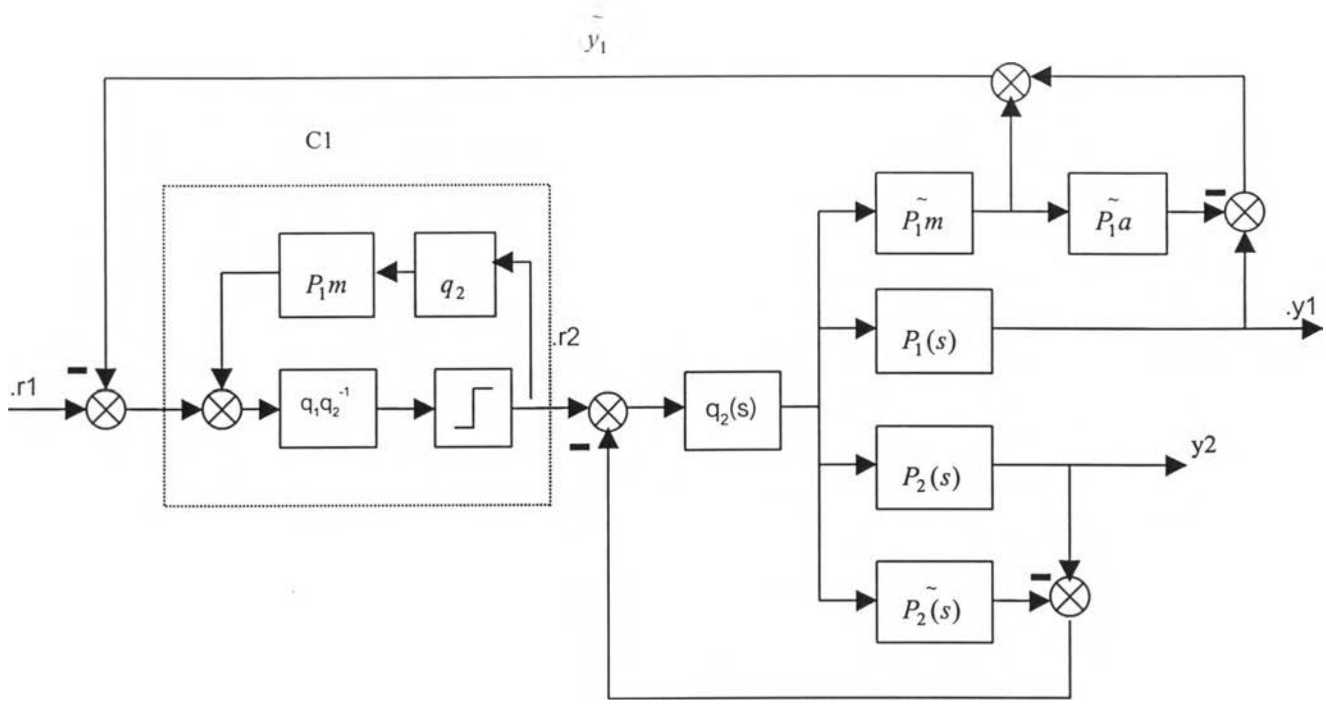
รูปที่ 4.1 การควบคุมแบบ IMC โอเวอร์ไรด์

ถ้าให้ y_1 เป็นตัวแปรที่ต้องการควบคุมและ y_2 เป็นตัวแปรไม่มีเงื่อนไข ดังนั้น r_1 มีค่าเซ็ทพอยท์ ส่วน $r_2 = 0$ เราจะสามารถจัดโครงสร้างเป็นคาสเคดได้โดยให้ y_1 เป็นลูพนอก ให้ y_2 เป็นลูพใน



รูปที่ 4.2 โครงสร้างของการควบคุมแบบคาสเคดที่ได้จากรูปที่ 4.1

ประสิทธิภาพของตัวควบคุมแบบคาสเคดที่ได้นี้จะเหมือนกับของตัวควบคุมโอเวอร์ไรด์เดิมทุกประการ แต่เราสามารถเพิ่มประสิทธิภาพของตัวควบคุมแบบคาสเคดนี้ได้อีกโดยการทำการชดเชยเดดไทม์หรือหรือตัวทำนายของสมิท



รูปที่ 4.3 โครงสร้างภายหลังการทำตัวทำนายสมิท (Smith predictor)

โดยการแบ่ง $p_1\tilde{}(s)$ ออกเป็น $p_1\tilde{}m$ คือ ส่วนไม่มีไทมดีเลย์และ $p_1\tilde{}a$ คือ ไทม์ดีเลย์ ส่วนตัวควบคุมเดิมเพิ่มลูพปิดของ q_2กับ $p_1\tilde{}m$ เข้าไป ดังรูปที่ 4.3

<u>ตัวอย่างที่ 1</u>

$$y_1 = \frac{K_1 e^{-T_1 s}}{\tau_1 s + 1} u(s) \qquad 2.7 \le K_1 \ge 3.3$$

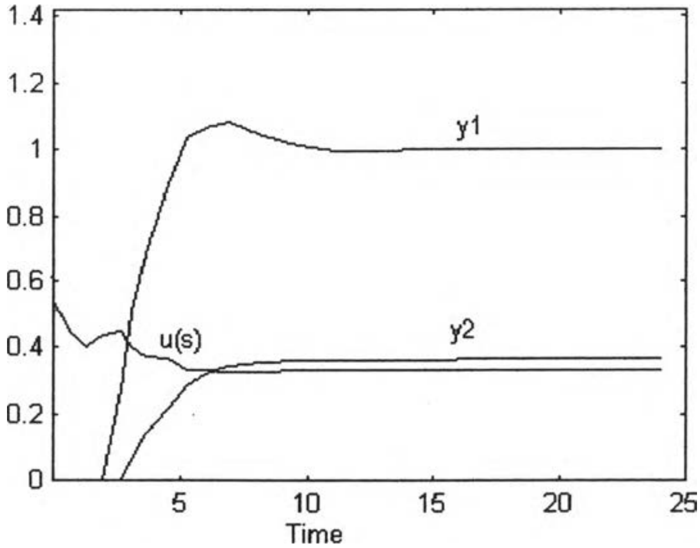
$$1.8 \le \tau_1, T_1 \ge 2.2$$

$$y_2 = \frac{K_2 e^{-T_2 s}}{\tau_2 s + 1} u(s) \qquad 2.7 \le \tau_2, T_2 \ge 3.3$$

ทำการออกแบบระบบควบคุมแบบโอเวอร์ไรด์ตามรูปที่ 4.2 หาตัวควบคุม q_1และ q_2 ของ y_1 และ y_2 แบบ IMC จะได้ q_1= (2s+1)/3(ε_1s+1) และ q_2= (3s+1)/(ε_2s+1) ตามรูปโดย การใช้โปรแกรม

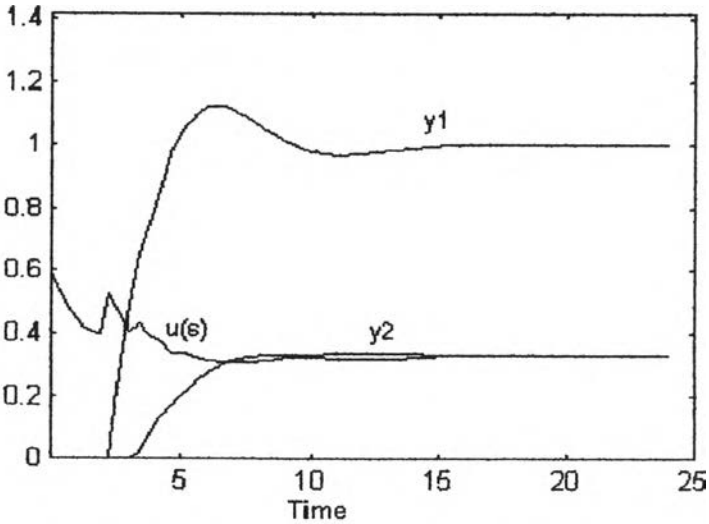
I 18814691

IMCTUNE หาค่าฟิวเตอร์ไทม์คอนสเต็นท์ (filter time-constant) จะได้ค่า $\varepsilon_1 = 1.24$, $\varepsilon_2 = 1.88$ ทำการเลียนแบบกระบวนการโดยให้เซ็ทพอยท์ y1=1,y2=0 และได้ผลการทดลองดังรูป 4.4



รูปที่ 4.4 กราฟแสดงผลการควบคุมแบบโอเวอร์ไรด์

ทำการออกแบบระบบควบคุมแบบคาสเคดแบบใหม่ตามรูปที่ 3.21 หาตัวควบคุม q_1และ q_2 ของ y_1 และ y_2 แบบ IMC จะได้ q_1= (2s+1)/3(ε_1s+1) และ q_2= (3s+1)/(ε_2s+1) โดยการใช้ โปรแกรม IMCTUNE หาค่า ฟิวเตอร์ไทม์คอนสเต็นท์ จะได้ค่า $\varepsilon_1 = 1.12$, $\varepsilon_2 = 1.88$ ได้ผลการทดลองดังรูป 4.5



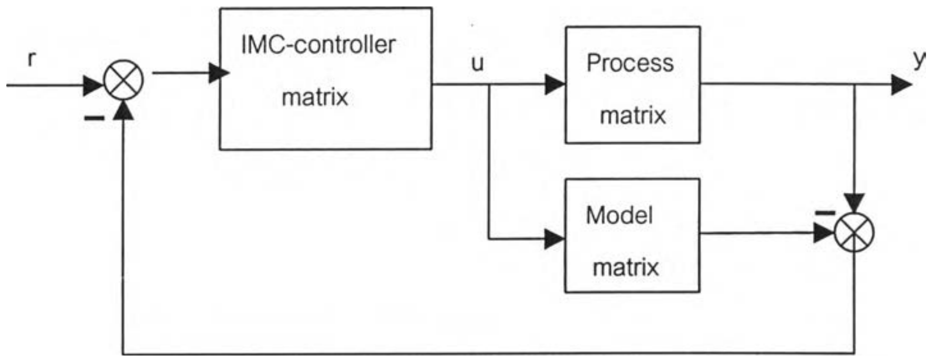
รูปที่ 4.5 กราฟแสดงผลการควบคุมแบบคาสเคดหลายตัวแปร

4.2 ระบบ (2x2) (two-input two-output)

$$y_1 = p_{11} u_1(s) + p_{12} u_2(s)$$

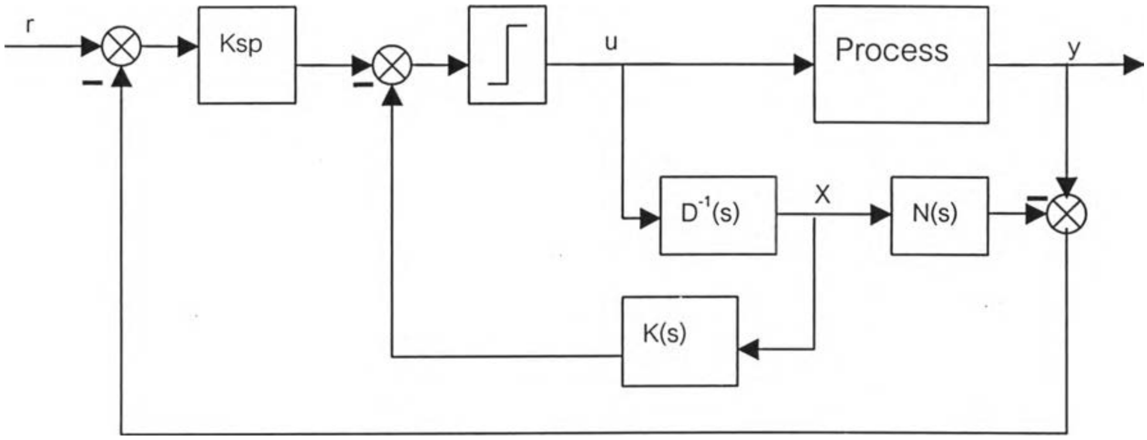
$$y_2 = p_{21} u_1(s) + p_{22} u_2(s)$$

<u>ขั้นที่1</u> เขียนบล็อกไดอะแกรมของIMCในรูปของเมทริกซ์



รูปที่ 4.6 แสดงบลอกไดอะแกรมของ IMC ในรูปเมทริกซ์

<u>ขั้นที่2</u> จัดโครงสร้างระบบคาสเคดโดยใช้ตัวแปรภายในตัวแปรภาวะเป็นลูพในตามวิธีของ Brosilow และ Joseph Boyce (1996)



รูปที่ 4.7 แสดงแบบจำลอง state feedback ของตัวควบคุมแบบหลายตัวแปรแบบ IMC

y คือ ตัวแปร ออก

r คือ set point

d คือ disturbances

u คือ IMC control effort

u_S คือ limit valveของ u

K_{SP} คือ แมทริกซ์ ของ constants

$K(s,e^{-S}), D(s), N(s,e^{-S})$ คือ แมทริกซ์ของโพลิโนเมียลของ sและ e^{-S}

แมทริกซ์ $K(s,e^{-s})$ คำนวณได้จาก $K(s,e^{-s}) = KspF^{-1}(s)\widetilde{P}_a^{-1}(s)N(s,e^{-s}) - D(s)$

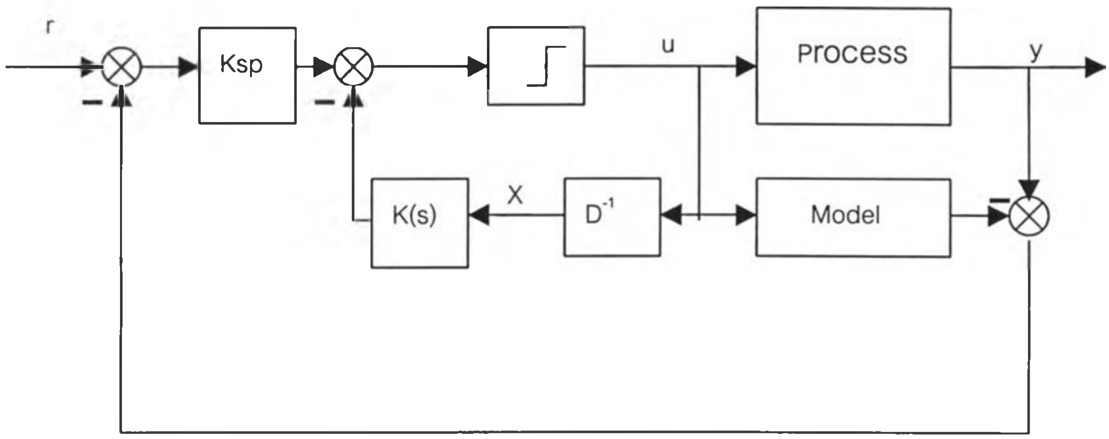
$\widetilde{P}_a^{-1}$ คือ อินเวอร์ส noninverted part

F^{-1} คือ อินเวอร์ดของ IMC controller filter

$$u(s) = Ksp(r(s) - d) - K(s, e^{-s})\,x(s)$$

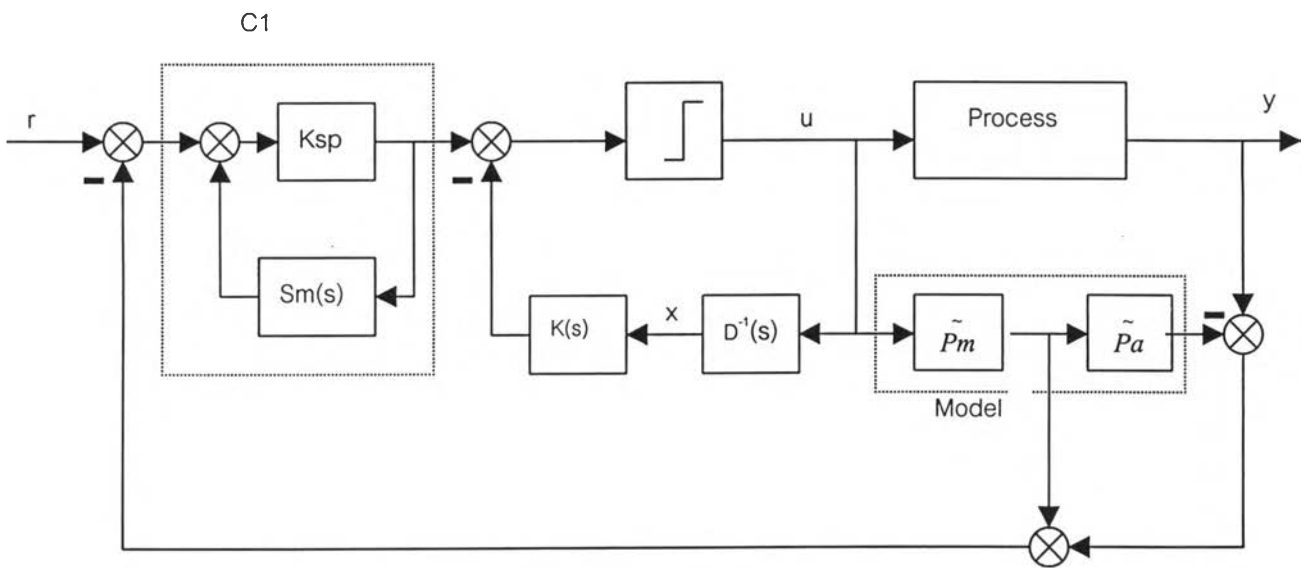
จากโครงสร้างดังกล่าวจะเห็นได้ว่าเราสามารถจะเพิ่มประสิทธิภาพของตัวควบคุมเช่นเดียวกับระบบ (2x1) โดยการใช้ตัวทำนายของสมิท (Smith predictor) เพื่อกำจัดเดดไทม์ของระบบซึ่งจะทำให้ตัวควบคุมมีค่าฟิวเตอร์ไทม์คอนสเต็นท์ลดลง ซึ่งสามารถทำดังขั้นที่ 3

<u>ขั้นที่3</u> จัดรูปใหม่เพื่อสะดวกในการทำตัวทำนายสมิท (Smith predictor)



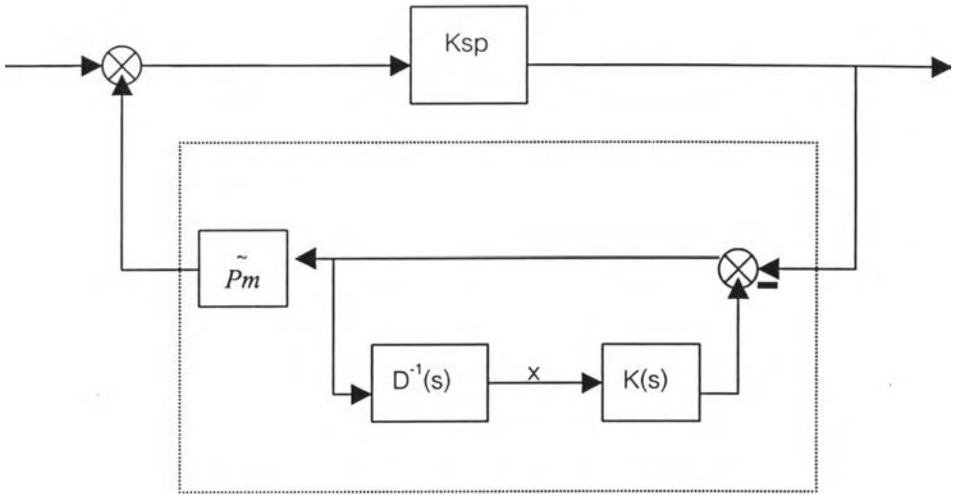
รูปที่ 4.8 แบบจำลอง state feedback ของตัวควบคุมที่ได้จากรูปที่ 4.7

<u>ขั้นที่4</u> ทำการออกแบบตัวทำนายสมิท (Smith predictor_)



รูปที่ 4.9 แบบจำลอง state feedback ของตัวควบคุมใส่ตัวทำนายสมิทแล้ว

โดยการแบ่งโดยการแบ่งแบบจำลองออกเป็น $\tilde{pm}$ คือ ส่วนไม่มีไทมดีเลย์และ $\tilde{pa}$ คือ ไทมดีเลย์ ทำการดัดแปลงตัวควบคุมเดิมเพิ่มลูพปิดของสมิทแมทริกซ์ (Smith matrix: Sm(s)) เข้าไปดังรูปที่ 4.10



Smith matric :Sm(s)

รูปที่ 4.10 ส่วนประกอบของสมิทแมทริกซ์ (Smith matrix)

แมทริกซ์ $S_m(s,e^{-s})$ คำนวณได้จาก

$$Sm(s) = \tilde{pm}\,[I+D^{-1}(s)K(s)] \tag{4.1}$$

ถ้าระบบ (2×2) มีค่าต่างๆ ดังนี้

$$P(s) = \begin{pmatrix} \dfrac{k_{11}}{\tau_{11}s+1}e^{-\Theta_{11}s} & \dfrac{k_{12}}{\tau_{12}s+1}e^{-\Theta_{12}s} \\ \dfrac{k_{21}}{\tau_{21}s+1}e^{-\Theta_{21}s} & \dfrac{k_{22}}{\tau_{22}s+1}e^{-\Theta_{22}s} \end{pmatrix} \tag{4.2}$$

$$D(s) = \begin{pmatrix} (\tau_{11}s+1)(\tau_{22}s+1 & 0 \\ 0 & (\tau_{12}s+1)(\tau_{22}s+1) \end{pmatrix} \tag{4.3}$$

$$N(s) = \begin{pmatrix} k_{11}(\tau_{21}s+1)e^{-\Theta_{11}s} & k_{12}(\tau_{12}s+1)e^{-\Theta_{12}s} \\ k_{21}(\tau_{11}s+1)e^{-\Theta_{21}s} & k_{22}(\tau_{22}s+1)e^{-\Theta_{22}s} \end{pmatrix} \tag{4.4}$$

$$P_a = \begin{pmatrix} e^{-\Theta_{11}s} & 0 \\ 0 & e^{-\Theta_{22}s} \end{pmatrix} \tag{4.5}$$

$$Pm = \begin{pmatrix} \dfrac{k_{11}}{\tau_{11}s+1} & \dfrac{k_{12}}{\tau_{12}s+1}e^{-(\Theta_{12}-\Theta_{11})s} \\ \dfrac{k_{21}}{\tau_{21}s+1}e^{-(\Theta_{21}-\Theta_{22})s} & \dfrac{k_{22}}{\tau_{22}s+1} \end{pmatrix} \tag{4.6}$$

$$Ksp = \begin{pmatrix} \dfrac{\tau_{11}}{k_{11}\varepsilon_{11}} & 0 \\ 0 & \dfrac{\tau_{22}}{k_{22}\varepsilon_{22}} \end{pmatrix} \tag{4.7}$$

$$K(s) = \begin{pmatrix} (Ksp_{11}k_{11}(\tau_{21}+\varepsilon_{11})-(\tau_{11}+\tau_{21}))s+(ksp_{11}k_{11}-1) & Ksp_{11}k_{12}(\tau_{22}\varepsilon_{11}s^2+(\tau_{22}+\varepsilon_{11})s+1)e^{-(\Theta_{12}-\Theta_{11})s} \\ Ksp_{22}k_{21}(\tau_{11}\varepsilon_{22}s^2+(\tau_{11}+\varepsilon_{22})s+1)e^{-(\Theta_{21}-\Theta_{22})} & Ksp_{22}k_{22}(\tau_{12}+\varepsilon_{22}-(\tau_{12}+\tau_{22}))s+(Ksp_{22}k_{22}-1) \end{pmatrix}$$

(4.8)

$$Sm(s) = \begin{pmatrix} \dfrac{k_{11}\varepsilon_{11}/\tau_{11}}{\varepsilon_{11}s+1} & 0 \\ 0 & \dfrac{k_{22}\varepsilon_{22}/\tau_{22}}{\varepsilon_{22}s+1} \end{pmatrix} \tag{4.9}$$

ตัวอย่าง 2

$$y_1 = \frac{4.05e^{-27s}}{50s+1}u_1(s) - \frac{1.77e^{-28s}}{60s+1}u_2(s)$$
$$y_2 = \frac{5.39e^{-18s}}{50s+1}u_1(s) - \frac{5.72e^{-14s}}{60s+1}u_2(s)$$

จากการใช้โปรแกรม IMCTUNE ได้ ε_{11}= 4.0, ε_{22}= 8.0 และสามารถคำนวณค่าต่างๆได้ดังนี้

$$P_a = \begin{pmatrix} e^{-27s} & 0 \\ 0 & e^{-14s} \end{pmatrix} \tag{4.10}$$

$$Pm = \begin{pmatrix} \frac{4.05}{50s+1} & \frac{-1.77}{60s+1}e^{-1s} \\ \frac{5.39}{50s+1}e^{-4s} & \frac{-5.72}{60s+1} \end{pmatrix} \tag{4.11}$$

$$D(s) = \begin{pmatrix} (50s+1)(50s+1 & 0 \\ 0 & (60s+1)(60s+1) \end{pmatrix} \tag{4.12}$$

$$N(s) = \begin{pmatrix} 4.05(50s+1)e^{-27s} & -1.77(60s+1)e^{-28s} \\ 5.39(50s+1)e^{-18s} & -5.72(60s+1)e^{-14s} \end{pmatrix} \tag{4.13}$$

$$Ksp = \begin{pmatrix} 3.086 & 0 \\ 0 & 1.311 \end{pmatrix} \tag{4.14}$$

$$K(s)=\begin{pmatrix} 575s-11.5 & (1311.1s^2+349.63s+5.46)e^{-s} \\ (2826.9s^2+409.9s+7.07)e^{-4s} & 390s+6.5 \end{pmatrix} \quad (4.15)$$

$$Sm(s)=\begin{pmatrix} \frac{0.32}{4s+1} & 0 \\ 0 & \frac{0.76}{8s+1} \end{pmatrix} \quad (4.16)$$

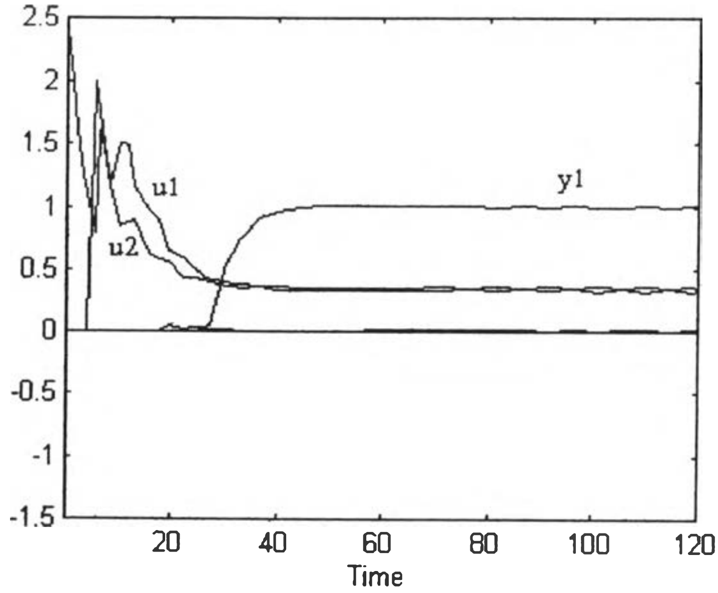
ผลที่ได้จากการเลียนแบบกระบวนการแยกออกเป็นกรณีต่างๆ ดังนี้

กรณีที่1 เซ็ทพอยท์ y1 = 1 ,y2 = 0

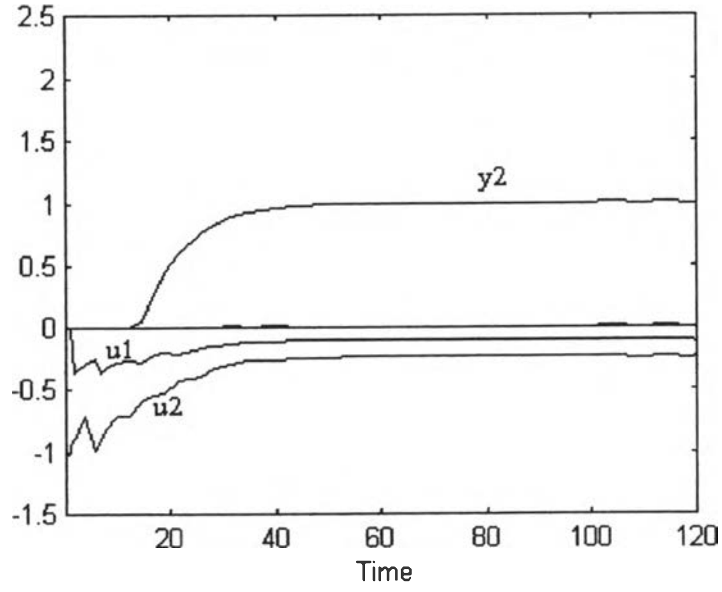
กรณีที่2 เซ็ทพอยท์ y1 = 0 , y2 = 1

กรณีที่3 เซ็ทพอยท์ y1 = 1 ,y2 = 2

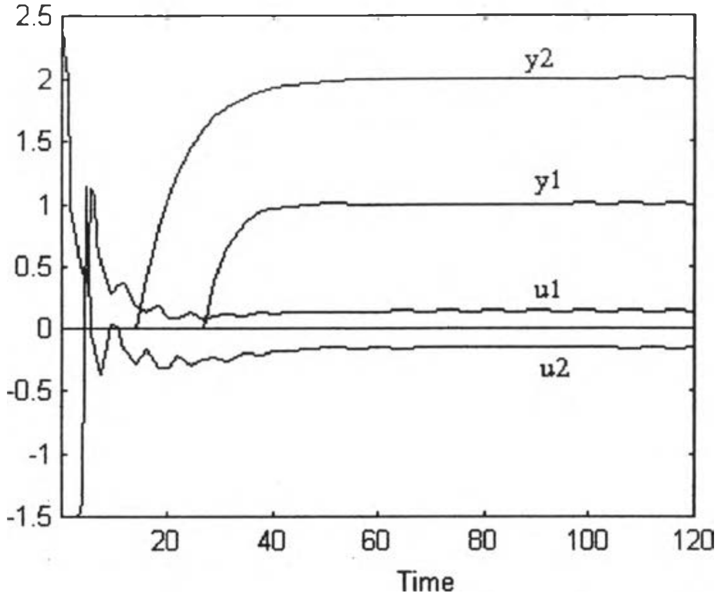
กรณีที่4 เซ็ทพอยท์ y1 = 1 , y2 = 1



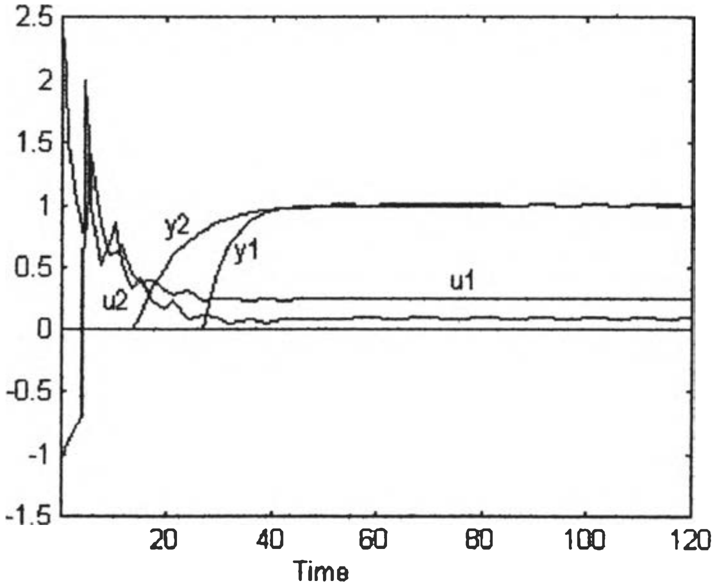
รูปที่ 4.11 ผลที่ได้จากการให้เซ็ทพอยท์ y1 = 1 ,y2 = 0



รูปที่ 4.12 ผลที่ได้จากการให้เซ็ทพอยท์ y1 = 0 , y2 = 1



รูปที่ 4.13 ผลที่ได้จากการให้เซ็ทพอยท์ y1 = 1 , y2=1



รูปที่ 4.14 ผลที่ได้จากการให้เซ็ทพอยท์ y1 = 1 , y2=1

4.3 วิธีการจูนตัวควบคุม (Controller Tuning)

ในการหาค่า τ_f ของตัวควบคุมแบบ IMC หรือที่เรียกว่า การ Tuning นั้นมีอยู่หลายวิธีด้วยกัน ได้แก่ IMCTUNE solfware, Ciancone correlation, วิธีการลองผิดลองถูก (Tiral and error)

4.3.1 วิธีการลองผิดลองถูก (Trial and error)

วิธีการนี้ก็คือทำการเดาค่า τ_f ขึ้นมาค่าหนึ่งนำค่าที่ได้ไปทดลองเลียนแบบกระบวนการสังเกตผลที่เกิดขึ้นถ้าผลที่ได้ยังใช้ไม่ได้จะทำการเดาค่า τ_f ใหม่โดยการเพิ่มหรือลดจากค่า τ_f เดิมขึ้นอยู่ผลที่ได้การจากเลียนแบบกระบวนการทำเช่นนี้ไปเรื่อยๆ จนกว่าจะได้ค่าที่ถูกต้องหรือพอใจดังนั้นถ้าต้องการค่าละเอียดมาก จำนวนครั้งของการเดาก็จะมากและยังขึ้นกับค่าเริ่มต้นหรือค่าที่เดาขึ้นมาครั้งแรก ถ้าค่าเริ่มต้นมีค่าใกล้เคียงค่าที่ถูกต้องจะทำให้จำนวนครั้งของการเดาลดลง นอกจากนี้ยังขึ้นกับวิธีการเดาค่าที่ดีอีกด้วย ดังนั้นจึงต้องมีวิธีเลือกค่าเริ่มต้นที่เหมาะสมและวิธีในการเดาค่าที่ดี

- การหาค่าเริ่มต้นซึ่งอาจจะได้มาจากวิธีการอื่นก็ได้เช่น Ciancone correlation หรือที่ง่ายๆ ก็ คือ ใช้ค่า τp เป็นค่าเริมต้นจะทำให้การเดาค่าเหลือ 2 เส้นทาง คือ มากกว่ากับน้อยกว่า τp ซึ่งจะทำให้ช่วงการเดาแคบลง
- วิธีการเดาค่าจะเริ่มจากการเดาจากการแบ่งซอยแบบหยาบๆ ก่อนเพื่อจะหาช่วง ของค่าจริงก่อนอาจจะทำซ้ำโดยการแบ่งซอยช่วงของค่าจริงให้ย่อยลงไปอีกทำการเดาค่าเพื่อหาช่วงของค่าจริงที่แคบลงจากนั้นจะทำการแบ่งอย่างระเอียดเพื่อเดาค่าอย่างละเอียดอีกทีหนึ่ง

4.3.2 วิธี Ciancone correlation

วิธีการนี้จะหาค่า τ_f แบบคร่าวๆจากทรานเฟอร์ฟังชันก์ของ process:Gp (s) เฉพาะระบบ SISO เท่านั้น ซึ่งอยู่ในรูปของสมการ (correlation)

$$y = -0.5X + 0.55 \tag{4.17}$$

กำหนดให้

$$X = \frac{\Theta}{\tau_p + \Theta}$$
$$y = \frac{\tau_f}{\tau_p + \Theta}$$

ในการหาค่า τ_f ด้วยวิธีนี้จะแบ่งออกเป็นขั้นๆดังนี้

<u>ขั้นที่1</u> หาค่า $X = \Theta/(\tau p + \Theta)$

<u>ขั้นที่2</u> หาค่า y จากสมการ $y = -0.5X + 0.55$

<u>ขั้นที่3</u> หาค่า τ_f จากความสัมพันธ์ $y = \tau_f/(\tau p + \Theta)$

<u>ตัวอย่างเช่น</u> $Gp(s) = \dfrac{5e^{-3s}}{10s+1}$

$X = 3/(10+3) = 0.23$

$y = -0.5x0.23+0.55 = 0.43$

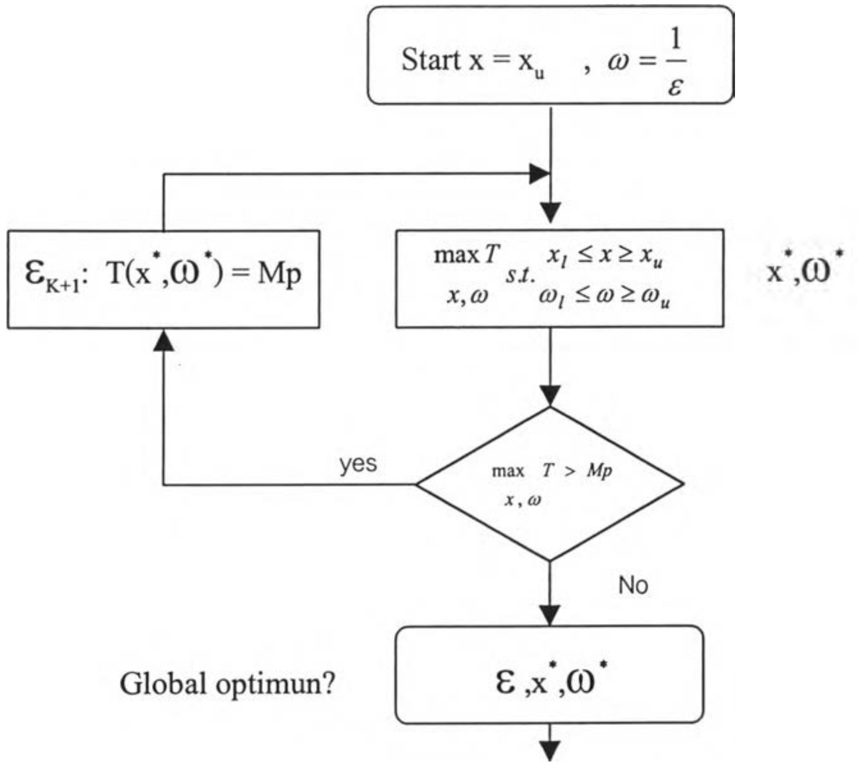
$\tau_f = 0.43x(10+3) = 5.65$

จากตัวอย่างค่า τ_f ที่หาได้คือ 5.56 ยังไม่ใช่ค่าที่ดีที่จะนำไปใช้แต่เป็นค่าที่ดีที่นำไปใช้เป็นค่าเริ่มต้นให้กับวิธี Tiral and error เพื่อหาค่าที่ถูกต้องต่อไป

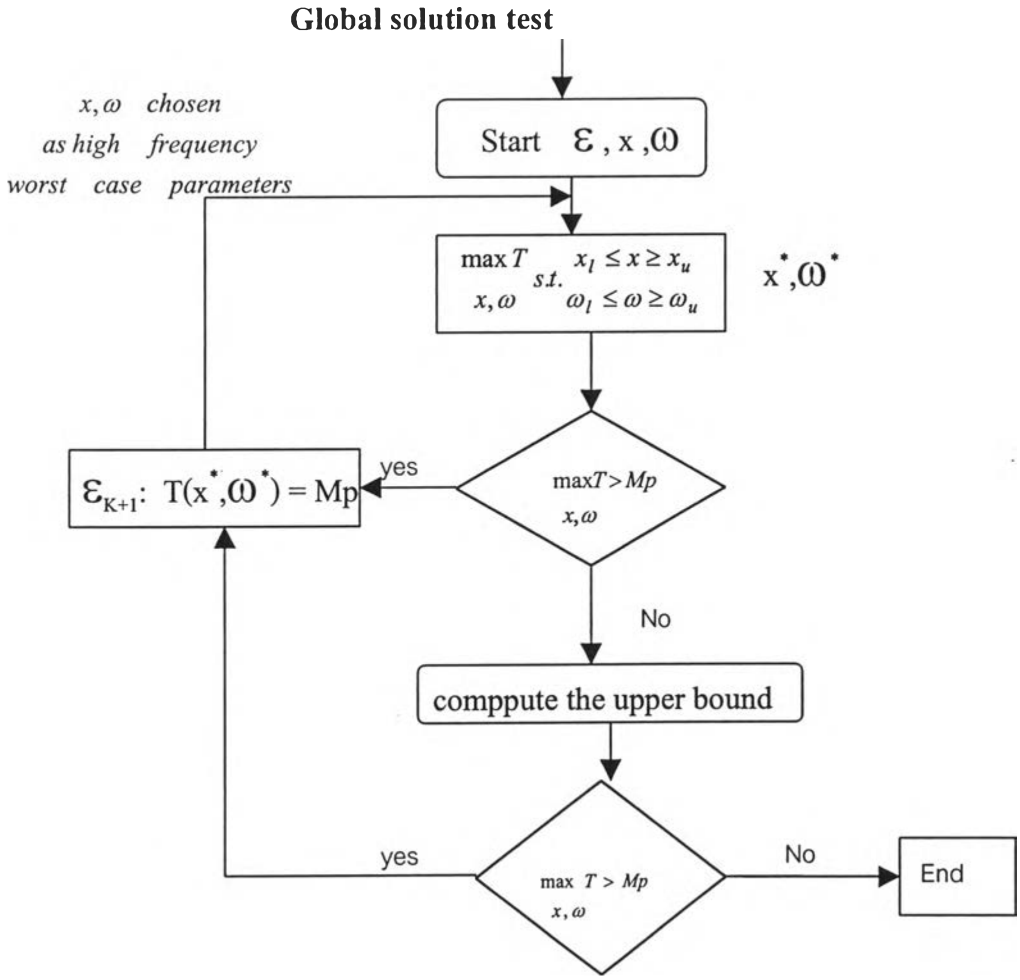
4.3.3 โปรแกรม IMCTUNE

IMCTUNE เป็นวิธีการหาค่า τ_f โดยการวิเคราะห์เชิงความถี่ (frequency response) ที่เรียกว่า Maximum Peak (Mp) Performance Criterion ของ sensitivity transferfunction โดยจะทำการเลือกค่า τ_f ที่ทำให้ $Mp \leq 1.05$ ตรอดทุกช่วงของความถี่ (ω)ในทางปฏิบัติวิธีการนี้ยุ่งยากและซ้ำซ้อนมากต้องใช้เครื่องคอมพิวเตอร์เข้าช่วยจึงต้องเขียนขึ้นเป็นโปรแกรมซึ่งมีอัลกอริทึม ดังนี้

Tuning algorithm



รูปที่ 4.15 การทำงานของโปรแกรมคอมพิวเตอร์ในการหาค่า τ_f



รูปที่ 4.16 การทำงานของโปรแกรมคอมพิวเตอร์ในการทดสอบค่า τ_r ที่จากขั้นตอนแรก

ค่าที่ได้จากวิธีนี้จะเป็นค่าที่ถูกต้องและใช้ได้เลย โดยไม่ต้องใช้วิธีการอื่นอีกและจะรับประกันผลในแง่ของ ประสิทธิภาพ (Performance) ความทนทาน (Robustness) และในแง่ของความเหมาะสม (Optimization) พร้อมๆ กัน