ปีการศึกษา 2531

การควบคุมแบบปรับตัว

โดย

นาย ขวัญรัฐ ส่วนพงษ์

นาย ชาญชัย สุทธิธนากุล

อาจารย์ที่ปรึกษา

อาจารย์ จงกล งามวิวิทย์

เรื่อง การควบคุมแบบปรับตัวเอง

นาย ขวัญรัฐ ส่วนพงษ์
นาย ชาญชัย สุทธิธนากุล
อ. จงกล งามวิวิทย์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2531

บทคัดย่อ

วิทยานิพนธ์นี้ นำเสนอเรื่องการควบคุมแบบปรับตัว ซึ่งตัวควบคุมที่ใช้นี้ สามารถปรับค่าควบคุมตามการเปลี่ยนแปลงของพารามิเตอร์ของระบบ เราสามารถตรวจวัดการเปลี่ยนแปลงของพารามิเตอร์ของระบบด้วย ตัวประมาณค่าแบบลีสสแควร์(Least-Squares estimation) แล้วจึงหาค่าควบคุมด้วยทฤษฎีทางพีชคณิต(Diophantine equation) และค่าควบคุมที่หาได้นี้จะเป็นค่าที่นำสมรรถนะของระบบกลับคืนสู่ค่าที่ต้องการ

Title SELF-TUNNING CONTROL

Kwanradh Suanpong

Chanchai Suttithanakul

Jongkol Ngamwiwit Advisor

Abstract

This thesis present the self-tuning control. The controler ,we used here,will adapt the control value according to the changing process. We can detect any deviation in parameter by Least-Squares estimation.Then we calculate the control value by the Algebra theorem(Diophantine equation).This control value will bring the process performance back to the desired value.

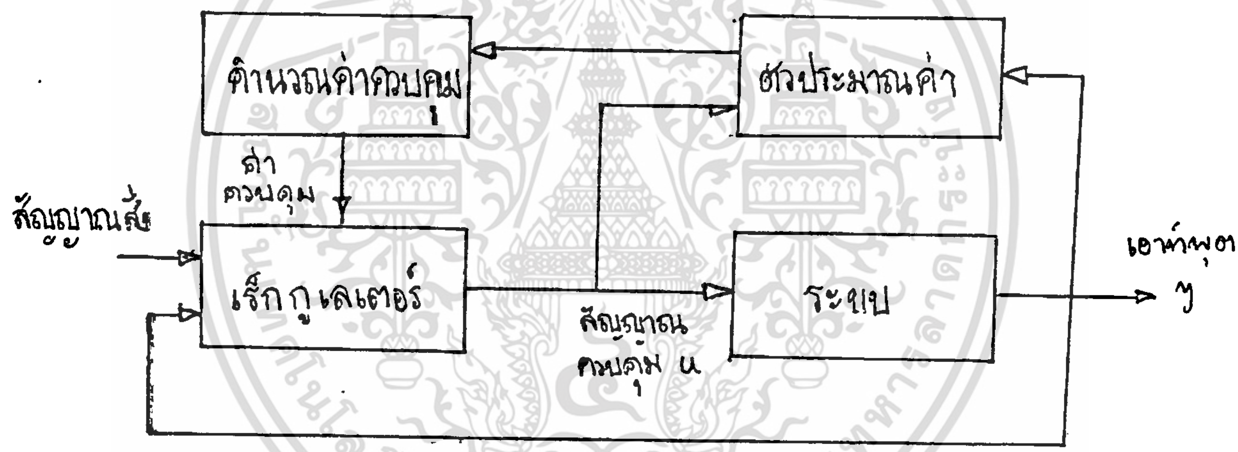
สารบัญ

บทที่ 1

บทนำ การควบคุมแบบปรับตัว

ถึงแม้ว่าตัวควบคุมแบบ PID จะมีสมรรถนะในการควบคุมสูงอยู่แล้วแต่มักจะต้องเสียเวลาและค่าใช้จ่ายในการปรับสำหรับตอนเริ่มทำงานถ้าระบบมีพลวัตช้า และมักจะมีสมรรถนะด้อยลงเนื่องจากพารามิเตอร์เปลี่ยนแปลง ดังนั้นจึงได้มีการพัฒนาการควบคุมแบบปรับตัวขึ้นมา

วิธีในการปรับนั้นมีหลายวิธี แต่วิธีที่ง่ายที่สุด คือ การควบคุมแบบปรับตัวเอง (Self-tuning Control) ซึ่งจะเอา อินพุต และ เอาท์พุต ของระบบไปตรวจสอบหาค่าของระบบที่เปลี่ยนแปลงไป แล้วนำค่านั้นมาคำนวณค่าควบคุมที่ยังคงรักษาระบบให้มีสมรรถนะดังเดิม โดยมีโครงสร้างดังรูป



รูป 1 การควบคุมแบบปรับตัว

การทำงานตามโครงสร้างนี้จะเรียกว่า อัลกอริทึ่มแบบชัดแจ้ง (Explicit Algorithm) และถ้าลดขั้นตอนการทำงานให้น้อยลงกว่านี้โดยยุบตัวประมาณค่าพารามิเตอร์ ไปรวมกับอัลกอริทึ่มการหาค่าตัวควบคุม จะเรียกวิธีนี้เรียกว่า อัลกอริทึ่มแบบแฝง (Implicit Algorithm)

วิทยานิพนธ์นี้ได้ยกเรื่องโมเดลของระบบมาไว้เป็นหัวข้อแรก เพราะโมเด็ลเป็นเรื่องสำคัญที่สุดในทฤษฎีการปรับตัว (Adaptive Control Theory) ตามมาด้วยการศึกษาระบบทางเวลาไม่ต่อเนื่อง และเสถียรภาพ แล้วจึงเริ่มกล่าวถึงการควบคุมปิดลูปอย่างจริงจัง โดยกล่าวถึง แนวทางการควบคุมที่จะทำให้เสถียรภาพของการ

ควบคุมปิดลูปดีที่สุด แล้วจึงนำทฤษฎีทางพีชคณิตมาใช้หาค่าควบคุม

หลังจากที่ได้กล่าวถึงการควบคุมที่ถูกสมมุติว่าทราบค่าของพารามิเตอร์ของระบบแล้ว จึงมาพูดถึงเรื่องของตัวประมาณค่า โดยตัวประมาณค่าจะเป็นแบบเร็คคอซีฟ ลีสสแควร์ (Recusive-Least Squares)

หลังจากนั้นก็นำแต่ละส่วนมาประกอบกันตามโครงสร้างในหน้าที่แล้ว และก็ศึกษาถึงผลของการควบคุมและการปรับตัว

คณะผู้จัดทำ หวังว่าวิทยานิพนธ์นี้จะช่วยให้ผู้ที่มาศึกษาต่อไป ใช้เวลาในการศึกษาทฤษฎีพื้นฐานน้อยลง และสามารถประยุกต์การควบคุมแบบปรับตัวไปใช้ในงานด้านอื่นๆได้

บทที่ 2 ทฤษฎี

2.1.1 โมเด็ลของระบบ

เมื่อเราต้องการที่จะศึกษาระบบในทางปฏิบัติว่ามีคุณสมบัติอย่างไร ก่อนอื่นจะต้องหาโมเด็ลของระบบนั้นขึ้นมาเสียก่อน แล้วจึงทำการควบคุมระบบนั้น ในขั้นแรกเราอาจแบ่งโมเด็ลของระบบตามแบบทฤษฎีการควบคุม ได้ 2 แบบ ดังนี้

1) โมเด็ลปริภูมิสภาวะ (State-Space Model) หรือ โมเด็ลภายใน (internal model) เป็นโมเด็ลที่แสดงความสัมพันธ์ของแต่ละตัวแปรในระบบในรูปของเมตริกซ์

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

2) โมเด็ลอินพุตเอาท์พุต (Input - Output Model) หรือ โมเด็ลภายนอก (External Model) เป็นโมเด็ลที่แสดงความสัมพันธ์ของแต่ละตัวแปรในระบบในรูปของฟังก์ชั่นถ่ายโอน (Transfer Function)

$$\frac{Y(t)}{U(t)} = \frac{B(s)}{A(s)} = \frac{b_o + b_1 s + b_2 s^2 + \ldots + b_n s^m}{a_o + a_1 s + a_2 s^2 + \ldots + a_n s^n}$$

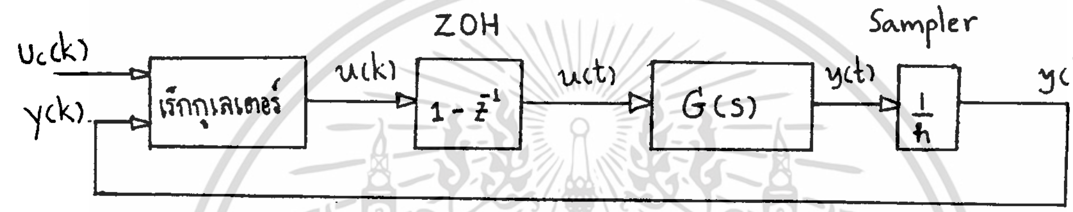
นอกจากจะแบ่งโมเด็ลตามชนิดของทฤษฎีการควบคุมแล้ว ยังแบ่งโมเด็ลตามการใช้งานได้อีก 2 แบบ คือ

1) โมเด็ลแบบเร็กกูเลเตอร์ (Regulator) ใช้แทนระบบที่มีสัญญาณรบกวนมาก และ จุดมุ่งหมายของการควบคุมก็คือ รักษาให้เอาท์พุตคงที่ตลอดเวลา ดังนั้นโมเด็ลแบบนี้ต้องมีเทอมของสัญญานรบกวน (Noise) และ ภาระ(Load)รวมอยู่ด้วย

2) โมเด็ลแบบเซอร์โว (Servo) ใช้แทนระบบที่ต้องการควบคุมสัญญาณเอาท์พุตให้เปลี่ยนแปลงตามอินพุต โดยถือว่าสัญญาณอินพุตเป็นสัญญาณที่ต้องการ และไม่มีสัญญาณรบกวนเข้ามาเป็นสัญญาณที่ต้องการ ดังนั้นจึงจะไม่คิดสัญญาณรบกวน

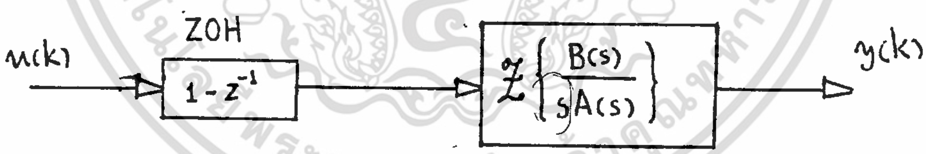
2.1.2 โมเด็ลของระบบในการควบคุมแบบเวลาไม่ต่อเนื่อง (Discrete Time Control)

เนื่องจากการควบคุมแบบปรับตัวเองต้องใช้คอมพิวเตอร์ในการควบคุม ดังนั้น จึงต้องแปลงระบบจากแบบเวลาต่อเนื่องให้เป็นแบบเวลาไม่ต่อเนื่อง และมีโครงสร้างการควบคุม ดังรูป



รูป 2.1 การควบคุมระบบเวลาต่อเนื่อง ด้วยการควบคุมแบบเวลาไม่ต่อเนื่อง

จะเห็นว่ายังเหลือส่วนที่เป็นระนาบ s อยู่ที่เดียวคือ G(s) ดังนั้นต้องทำการแปลงชุดนี้ แล้วดูว่าจากรูปมาตรฐาน B(s)/A(s) เมื่อนำมารวมกับ ZOH แล้วแปลงให้อยู่ในระนาบ z จะมีรูปมาตรฐานอย่างไร จึงพิจารณาเฉพาะ



รูป 2.2 การแปลงระบบเวลาต่อเนื่อง มาเป็นระบบมาตรฐานแบบเวลาไม่ต่อเนื่อง

กระจาย $\frac{B(s)}{sA(s)}$ ออกโดยใช้ทฤษฎีแยกส่วน(Partial fraction)

เมื่อสมมุติให้ระบบเป็นแบบที่มีโพลแบบต่างกัน

จะได้ $\frac{G(0)}{s} + \frac{R_i}{(s+P_i)}$; P_i เป็นโพลที่แตกต่างกัน i ตัว

$$R_i = \left[\frac{B(s)}{\frac{d}{ds}(sA(s))}\right]_{s=-P_i} = \left[\frac{B(s)(s+P_i)}{(sA(s))}\right]_{s=-P_i}$$

$$G^*(z^{-1}) = \frac{B^*(Z^{-1})}{A^*(z^{-1})} = G(O) + \frac{Ri(1-z^{-1})}{(1-qiz^{-1})}$$

จากเทอม $1-qiz^{-1}$ จะเห็นว่า เมื่อคูณกัน n ตัว จัทำให้สปส.ของ A^* เป็น 1

$$G^*(z^{-1}) = \frac{b0+b1z^{-1}+...+bnz^{-n}}{1+a1z^{-1}+...+anz^{-n}} \qquad 2.1.2.1$$

นี่คือรูปมาตรฐานของโมเดลของระบบแบบเวลาไม่ต่อเนื่อง

ข้อสังเกต z^{-1} เรียกว่า backward shift operator มักเอาไว้ใช้ในการพิจารณาเสถียรภาพและใช้ในการแสดงความสัมพันธ์ของสมการผลต่าง ส่วน shift operator มักใช้ในการหากำลัง(deg) ในเนื้อหาต่อไปจะใช้ z^{-1} แทน z หมด ซึ่งเปลี่ยนได้โดย นำ z ไปหารทั้งเศษและส่วน

2.1.3 เสถียรภาพของระบบเวลาไม่ต่อเนื่อง

สิ่งที่สำคัญที่สุดในการควบคุมเวลาแบบไม่ต่อเนื่องก็คือ ตำแหน่งของโพลและซีโร่ของ $G^*(Z^{-1})$ ในระนาบ Z โดยสำหรับแต่ละโพลในระนาบ S จะมีการถ่ายเท (map) ไปเป็นโพลในระนาบ Z เพียงตัวเดียว (Uniqueness) ด้วย $Z = e^{sh}$ หรือโพลที่เสถียร (ในระนาบ S ทางครึ่งซ้าย) จะถ่ายเทลงใน (map into) โพลในวงกลมหนึ่งหน่วย ดังนั้นจะสรุปได้ว่า ระบบทางเวลาต่อเนื่องที่มีโพลเสถียร เมื่อแปลงมาเป็นระบบในเวลาไม่ต่อเนื่องแล้วยังคงมีโพลที่เสถียรอยู่ แต่ซีโร่ไม่เป็นเช่นนั้น

ถ้าในระบบไม่มีดิเลย์เศษ (Fractional Delay) และถ้ากำลังของ B(s) เท่ากับกำลังของ $B^*(Z^{-1})$ แล้วซีโร่ของ B^* จะอยู่ที่

$$zi = 1+hPi+o(h^2) \quad (4)$$

; โดย Pi เป็นรากที่ i ของ B(s) และ $o(h^2)$

$$= 1+hPi$$

เป็นฟังก์ชันประมาณที่มีค่าเข้าใกล้ 0 เมื่อ $h \to 0$ และฟังก์ชันนี้ขึ้นกับ h^2

กฎนี้แสดงว่าระบบที่มีเฟสต่ำสุด (minimum phase) หรือที่มี $pi < 0$ และกำลังของ B^* เท่ากับกำลังของ B จะมีซีโร่ในระบบเวลาไม่ต่อเนื่องอยู่ในเขตเสถียร

อย่างไรก็ตามในระบบที่มีการแปลง Z จะทำให้กำลังของ $B^* = n-1$ ดังนั้นกฎนี้จึงใช้ได้กับระบบที่มีโพลมากกว่าซีโร่อยู่หนึ่งตัวในระบบ S ซึ่งมีระบบในทางปฏิบัติน้อยมากที่มีเฟสล้าหลังเพียง 90 องศา และโพลในระบบ S ที่เกินมานี้ จะทำให้เกิดซีโร่ในระนาบ Z โดยที่ตำแหน่งของซีโร่ตัวนี้จะทำให้เกิดปัญหาในอัลกอริทึมการปรับตัว เพราะอาจจะอยู่นอกเขตเสถียรก็ได้

ถ้า G(s) มีโพลมากกว่าซีโร่ p ตัว $G^*(z^{-1})$ จะลู่เข้าสู่ $G^*(z^{-1})$ ที่หาได้จาก $G(s) = s^{-r}$ โดย

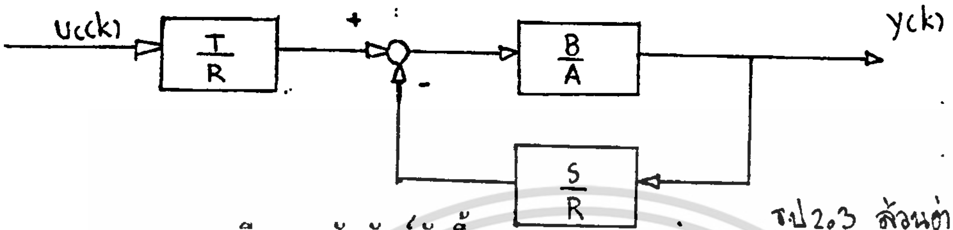
$$\text{เมื่อ } h \to 0 \quad G^*(z^{-1}) \to bm(1-z^{-1}) \frac{D^p}{p!}\left[\frac{1}{(1-z^{-1})}\right] \quad (4) \quad \text{โดย } D= -hz.d/dz$$

เช่นมีโพลมากกว่าซีโร่ 1,2 และ 3 ตัวตามลำดับ จะได้เศษของ $G^*(Z^{-1})$ เป็น $B^* = 1$; $B^* = 1-z^{-1}$; $B^* = 1+4z^{-1}+z^{-2}$ ซึ่งจะเห็นว่าจะได้ซีโร่ที่ไม่เสถียรที่ -1 สำหรับโพลมากกว่าซีโร่ 2 ตัว และที่ -3.73 สำหรับโพลมากกว่าซีโร่ 3 ตัว และเมื่อ z มากขึ้น ขนาดและจำนวนของซีโร่จะเพิ่มมากขึ้นด้วย ดังนั้นจะพบว่าซีโร่ที่ไม่เสถียรอาจจะเกิดขึ้นในระบบเดิมที่เคยเสถียรได้ถ้าอัตราการสุ่มตัวอย่างสูงพอ

2.1.4 การควบคุมปิดลูปทางระบบเวลาไม่ต่อเนื่องด้วยการกำหนดโพล

เมื่อต้องการควบคุมระบบอย่างสมบรูณ์ตัวควบคุมจะต้องมีทั้งการป้อนกลับและการป้อนตรง(Feedback And Feedforward) ดังนั้นจึงกำหนดให้มีการป้อนกลับด้วยฟังก์ชั่นถ่ายโอน $S(Z^{-1})/R(Z^{-1})$ และป้อนตรงด้วยฟังก์ชั่นถ่ายโอน $T(Z^{-1})/R(Z^{-1})$

ตัวควบคุมนี้เราอาจจะเรียกว่าเร็กกูเลเตอร์ (Regulator) (ไม่เกี่ยวกับโมเด็ลแบบเร็กกูเลเตอร์) มีโครงสร้างดังรูป



รูป2.3 ส่วนต่างๆของเร็กกูเลเตอร์

และมีความสัมพันธ์ดังนี้

$$R(z^{-1})U(k) = T(z^{-1})Uc(k) + S(z^{-1})Y(k) \qquad 2.1.4.1$$

โดย U เป็นสัญญาณควบคุม (Control Signal) Uc เป็นสัญญาณคำสั่ง (Command Signal) หรือสัญญาณอ้างอิงในระบบแบบเร็กกูเลเตอร์

ระบบเปิดลูปมีความสัมพันธ์ดังนี้

$$G(z^{-1}) = \frac{Y(k)}{U(k)} = \frac{B(z^{-1})}{A(z^{-1})} \qquad 2.1.4.2$$

ดังนั้นเมื่อทำการควบคุมปิดลูปจะได้

$$Gm(z^{-1}) = \frac{Y(k)}{Uc(k)} = \frac{Bm(z^{-1})}{Am(z^{-1})} = \frac{TB}{AR+BS} \qquad 2.1.4.3$$

G เป็นทรานสเฟอร์ฟังก์ชั่นของระบบที่ต้องการควบคุม Gm เป็นทรานสเฟอร์ฟังก์ชั่นที่ให้สมรรถะตามต้องการ จากสมการที่ 2.1.4.3 จะเห็นว่าโดยทั่วไปแล้ว กำลังของ Am มักจะน้อยกว่า กำลังของ AR + BS ดังนั้นแสดงว่าจะต้องมีองค์ประกอบบางตัวที่หักล้างกันได้ และจากทฤษฎีทางปริภูมิสเตท (ในภาคผนวก) จะเห็นว่าเร็กกูเลเตอร์ตามสมการที่ 2.1.4.1 มีความสัมพันธ์กับการรวมกันของ ตัวสังเกตสเตท (Observer) และตัวป้อนกลับสเตท (State Feedback) แต่จะไม่มีความสัมพันธ์กับ Uc ดังนั้นในทรานสเฟอร์ฟังก์ชั่นของระบบปิดลูป จะไม่มีโพลิโนเมียลของตัวสังเกตสเตท A_o เพราะ A_o อยู่ในเร็กกูเลเตอร์แต่เมื่อมาเกี่ยวข้องกับ Uc ก็จะหักล้างกันไป สมการจึงเปลี่ยนเป็นทรานส

เฟอร์ฟังก์ชั่นของระบบปิดลูปที่สมบูรณ์ ดังนี้

$$\frac{TB}{AR+BS} = \frac{BmAo}{AmAo} \qquad 2.1.4.4$$

2.1.5 เสถียรภาพของระบบปิดลูป

เมื่อพิจารณาผลของ Uc ต่อ Y แล้ว มาพิจารณาผลของ Uc ต่อ U บ้าง จากสมการที่ 1 และ 2 จะได้

$$\frac{U(k)}{Uc(k)} = \frac{A(z^{-1})}{B(z^{-1})} \cdot \frac{Bm(z^{-1})}{Am(z^{-1})} \qquad 2.1.5.1$$

จะเห็นว่าในกรณีที่ BAm อยู่นอกเขตเสถียรในระนาบ Z จะทำให้ U เกิดการกระเพื่อม หรือไม่เสถียรได้ และเนื่องจาก Am เป็นโพลของระบบปิดลูปที่เราต้องการ ดังนั้นมันจะต้องเสถียรอย่างแน่นอน จึงเหลือ B ซึ่งเป็นซีโร่ของระบบเปิดลูปที่กลายเป็นตัวสำคัญในการกำหนดลักษณะของสัญญาณควบคุม ซึ่งถ้า B เสถียร U ก็จะเสถียรด้วย แต่ถ้า B มีองค์ประกอบที่ไม่เสถียรจะต้องหาทางกำจัดไม่ให้ตัวที่ไม่เสถียรมีผล โดยแยก B เป็นองค์ประกอบ

$$B = B^{+} \cdot B^{-}$$ โดย B^{+} เป็นองค์ประกอบที่เสถียร

และ B^{-} เป็นองค์ประกอบที่ไม่เสถียร

การกำจัดเทอมที่ไม่เสถียรนั้นเกิดขึ้นจากการหักล้างระหว่าง B กับ Bm โดยการหักล้างเกิดขึ้นได้ 3 วิธีดังนี้

1. หักล้างซิโร่ทุกตัวของ B (หักล้างทั้งตัวที่เสถียร และตัวที่ไม่เสถียร) จากสมการที่ 2.1.4.5 R ต้องแบ่งเป็น

$$R = R_1 \cdot B$$

และจะทำให้เทอมของ B ในเศษและส่วนทางด้านซ้ายของสมการหักล้างกัน

$$\frac{TB}{B(AR_1 + S)} = \frac{Bm}{Am} \longrightarrow \frac{T}{AR_1 + S} = \frac{Bm}{Am}$$

จึงเห็นว่า Bm จะไม่มีเทอมของ B อยู่ ดังนั้นเมื่อเขียนอยู่ในรูปของ U กับ Uc

$$\frac{U(k)}{Uc(k)} = \frac{A.Bm}{B.Am}$$

ซึ่งจะเห็นว่า Bm ไม่หักล้าง B ทำให้ B ที่มีเทอมที่ไม่เสถียรปนอยู่ มีผลทำให้ U ไม่เสถียรตาม

2. หักล้างเฉพาะซิโร่ที่ไม่เสถียร B^{-}

R ต้องแบ่งเป็น $R_2 \cdot B^-$

และ $$\frac{T\,B^+\,B^-}{B^-(AR_2+B^+S)} = \frac{Bm}{Am} \longrightarrow \frac{TB^+}{AR_2+S} = \frac{Bm}{Am}$$

ดังนั้น Bm ต้องมีเทอมของ B^+ ด้วย

$$Bm = B^+\,Bm_2$$

และ $$\frac{A}{B^+B^-}\cdot\frac{B^+\,Bm_2}{Am} \longrightarrow \frac{A}{B^-}\cdot\frac{Bm_2}{Am}$$

จะเห็นว่าถ้าหักล้างซีโร่ที่ไม่เสถียร ซีโร่ที่ไม่เสถียรนั้นยังคงมีผลต่อ U อยู่

3. หักล้างเฉพาะซีโร่ที่เสถียร

R ต้องแบ่งเป็น $R = R_3 \cdot B^+$

และ $$\frac{T\,B^+\,B^-}{B^+(AR_3+B^-S)} = \frac{Bm}{Am} \longrightarrow \frac{TB^-}{AR_3+B^-S}$$

ดังนั้น Bm ต้องมีเทอมของ B^- อยู่ด้วย โดย $Bm = B^-\,Bm_3$

ทำให้ได้ความสัมพันธ์

$$\frac{A}{B^-B^+}\cdot\frac{B^-Bm_3}{Am} \qquad \frac{A}{B^+}\cdot\frac{Bm_3}{Am}$$

จะเห็นว่าตัว Bm มีส่วนของ B ที่ไม่เสถียร จะทำให้เกิดอาการแกว่งใน U หายไป แต่จะเกิดผลของซีโร่ที่ไม่เสถียรในระบบปิดลูป ซึ่งจะมีผลน้อยกว่ากรณีที่ B^- เป็นโพล

อาจสรุปได้ว่า เพื่อรักษาเสถียรภาพของสัญญาณควบคุม U จะต้องหักล้างซีโร่ที่เสถียรของระบบเปิดลูปด้วยการทำให้ซีโร่ของระบบปิดลูปมีเทอมที่ไม่เสถียร ซึ่งจะเห็นอีกว่า การแกว่งของ U ขึ้นกับการกำหนดซีโร่ของระบบปิดลูปว่าเป็นอย่างไร ดังนั้นจะเห็นว่าเราได้กำหนดขอบเขตของค่าของเร็กกูเลเตอร์ให้แคบลงแล้ว และต่อไปจะได้กล่าวถึงการหาค่าของเร็กกูเลเตอร์อย่างละเอียดมากยิ่งขึ้นใน 2.1.6

2.1.6 การหาค่าของเร็กกูเลเตอร์

กลับมาดูสมการ

$$\frac{TB}{AR+BS} = \frac{BmAo}{AmAo}$$

จะเห็นว่าทางด้านส่วน เป็นสมการ 2 ตัวแปร ซึ่ง R และ S มีได้หลายค่า แต่เมื่อต้องการเพียงค่าเดียว และต้องเป็นค่าที่เป็นไปได้ในความเป็นจริงในฐานะที่ S/R เป็นส่วนของการป้อนกลับ ดังนั้นในการหา R และ S จะต้องหาค่า R และ S ใดใด แล้วค่อยหา R และ S ตามข้อกำหนดของระบบที่ใช้ และตามหลักการป้อนกลับ จะเริ่มหา R และ S จากสมการ

$$AR + BS = Am \cdot Ao$$

ซึ่งเป็นสมการ 2 ตัวแปร มีชื่อเรียกว่า สมการไดโอแฟนไทน์ (Diophantine Equation)

สมการไดโอแฟนไทน์มีรูปแบบมาตรฐานดังนี้

$$aX + bY = c \qquad 2.1.6.1$$

โดยที่ a,b,c เป็นโพลิโนเมียลใดใด

สมการที่ 2.1.6.1 จะหาคำตอบได้ก็ต่อเมื่อ ห.ร.ม.ของ a และ b หาร c ได้ และเมื่อ ห.ร.ม.ของ a และ b เป็น g จะได้

$$a = ga_o , \quad b = gb_o , \quad c = gc_o$$

และ $g(a0\ X + b0\ Y) = gc_o$

ในการหาคำตอบของ 2.1.6.1 จะต้องใช้คู่โพลิโนเมียล p,q และ r,s ที่

$$ap + bq = g \qquad 2.1.6.2$$

$$ar + bs = 0 \qquad 2.1.6.3$$

เราสามารถหา g,p,q,r,s ทั้งหมดได้ในขั้นตอนเดียว ด้วยอัลกอริทึมใน ผ.3

จากสมการ 2.1.6.2 และ $c = gc_o$

จะได้ $$a(pc_o) + b(qc_o) = c$$

ดังนั้นจะได้ $X_o = pc_o$ และ $Y_o = qc_o$ ที่เป็นคำตอบเฉพาะของสมการ 2.1.6.1 จากสมการ 2.1.6.3 และ $a = ga_o$ และ $b = gb_o$ จะได้

$$ga_o\, r + gb_o\, s = 0$$

$$a_o\, r + b_o\, s = 0$$

ดังนั้น $a_o = s$ และ $b_o = -r$

ถ้าให้ X_o , Y_o เป็นคำตอบเฉพาะของสมการ 2.1.6.1 จะได้คำตอบใดใดของสมการเป็น

$$X = X_o - b_o\, t \qquad 2.1.6.4$$

$$Y = Y_o + a_o\, t \qquad 2.1.6.5$$

จัดรูปใหม่ได้

$$X = pc_o + rt \qquad 2.1.6.6$$

$$Y = qc_o + st \qquad 2.1.6.7$$

โดยที่ t เป็นโพลิโนเมียลใดๆ

มีหลายวิธีที่จะแก้สมการ 2.1.6.1 วิธีที่ 1 เป็นวิธีเทียบสัมประสิทธิ์ โดยเอาสมการ 2.1.6.1 เป็นเงื่อนไขกำลังของโพลิโนเมียล ตาม ผ.5เพื่อให้ทราบว่ากำลังของโพลิโนเมียลแต่ละตัวเป็นอย่างไร เมื่อทราบว่าโพลิโนเมียลใดมีกำลังเท่าใดแล้วก็แทนลงไป แล้วเทียบสัมประสิทธิ์แก้สมการออกมาโดยค่าที่ได้จะเป็น R ,S เพียงค่าเดียวที่ตรงตามค่ากำหนดของการเป็นตัวป้อนกลับ ซึ่งเหมาะสมที่จะใช้สำหรับโพลิโนเมียลที่มีกำลังต่ำๆ เพราะเมื่อใช้กับโพลิโนเมียลกำลังสูงๆ จะทำให้แก้สมการหา R , S ได้ยาก ซึ่งอาจต้องใช้กฎของแครเมอร์ (Kramer's rule)เข้าช่วย

อีกวิธีหนึ่งเป็นวิธีตามทฤษฎีของสมการไดโอแฟนไทน์ซึ่งจะเป็นวิธีที่ใช้ในวิทยานิพนธ์นี้ เพราะเป็นการศึกษาและทดลองตามทฤษฎีทางพีชคณิตโดยตรง และยังสามารถขยายไปใช้กับเร็กกูเลเตอร์ที่มีกำลังสูงๆ และ ระบบที่มีความละเอียดสูงๆได้อย่างง่ายดาย

รายละเอียดของการคำนวณจะได้กล่าวถึงเมื่อได้ทราบรูปแบบของโมเดลอย่างแน่นอนแล้ว และต่อไปเราจะพูดถึงตัวประมาณค่าซึ่งเป็นส่วนสำคัญที่ทำให้การควบคุมแบบปรับตัวสามารถปรับตัวได้

2.2 **การประมาณค่าแบบรีเคอร์ซีฟลีสแควร์** (Recursive Parameter Estimation)

การเลือกไคร์โตเรียล (Criterion) ของรีเคอร์ซีฟลีสแควร์คือการจักโมเด็ลโดยให้กำลังสองของความผิดพลาดระหว่างโมเด็ลของเอาท์พุตและจากค่าที่เราสังเกตมามีค่าน้อยที่สุด

พิจารณาโมเด็ลที่เป็นพารามิเตอร์เชิงเส้นดังนี้

$\phi(t) = \theta_1 x_1(t)+\theta_2 x_2(t)+\ldots+\theta_n x_n(t)$

หรือให้ $\phi(t) = x'(t)\theta+e(t)$ 2.2.1

ซึ่ง θ คือ ค่าเวกเตอร์จำนวน n ค่า ของพารามิเตอร์ที่ไม่ทราบค่า

$x(t)$ คือ เวกเตอร์ของข้อมูลที่ทราบค่า

$e(t)$ คือ ความผิดพลาดทางด้านสถิติไม่ขึ้นกับค่า $x_i(t)$

$\phi(t)$ คือ ค่าเอาท์พุตของการสังเกต

สมมุติว่าเรามีการสังเกตเข้ามา N ค่า โดยที่ $N > n$ จากสมการที่ 2.2.1

จะได้ว่า $\phi_N = X_N \theta + e_N$ 2.2.2

โดยที่ ϕ_N , e_N มีขนาดจำนวน N เวกเตอร์

และ X_N คือ เมตริกซ์ขนาด N*n มิติ

การประมาณค่าของ θ คือ การทำให้ค่าของฟังก์ชั่น (Lost Function) $L = \sum_1^N e^2(t)$ มีค่าน้อยที่สุด โดย $e(t)$ เป็นโมเด็ลของความผิดพลาด $\phi(t)-x'(t)\hat{\theta}$ ซึ่งจะแปรค่าตาม θ

เราจะสามารถแสดงสมการลีสแควร์ได้ดังนี้

$$\hat{\theta} = (X_N' X_N)^{-1} X_N' . \phi_N \qquad 2.2.3$$

ดังนั้นจากสมการที่ 2.2.2 จะได้

$$\hat{\theta} = (X_N' X_N)^{-1} X_N'(X_N\theta+e_N)$$

$$= \hat{\theta}+(X_N' X_N)^{-1} X_N' e_N \qquad 2.2.4$$

แต่ค่าของ $x_i(t)$ และ $e(t)$ จะไม่ขึ้นต่อกัน ดังนั้นค่าคาดหวังของ $\hat{\theta}$ จะขึ้นอยู่กับค่าความคาดหวังของ e_N ถ้าเราสมมติให้ค่าของสัญญาณรบกวนมีค่าเฉลี่ยเป็น 0 (ถ้าไม่เป็นเช่นนั้น $x_{n+1}(t)$ สามารถกำหนดให้มีค่าเท่ากับ 1 ได้ และค่าเฉลี่ยของ e จะถูกประมาณค่าได้) $E\{\hat{\theta}\}$ ก็คือ θ เป็นค่าที่แท้จริงของเวกเตอร์พารามิเตอร์ ในกรณีเช่นนี้เราจึงเรียกว่าเป็นการประมาณแบบไม่มีไบแอส

ยิ่งกว่านั้นในกรณีที่ $e(t)$ คือ สัญญาณรบกวนขาว (White Noise) ซึ่งการประมาณแบบลีสแควร์สามารถแสดงได้ว่าค่าของความแปรปวนมีค่าน้อยที่สุด เมื่อเปรียบเทียบกับการประมาณค่าแบบไม่มีไบแอสชนิดอื่น . และความแม่นยำจะขึ้นกับโควาเรียลเมตริกซ์ (Covariance Metrixs) ด้วย

$$E\{(\hat{\theta}-\theta).(\hat{\theta}-\theta)'\} = \rho^2(X_N'X_N)^{-1} \qquad 2.2.5$$

ρ^2 คือ ความแปรปวนของ $e(t)$

จากสมการที่แสดงมานี้จะเหมาะกับขบวนการแบบแบทช์ ซึ่งจำนวนของข้อมูลและการคำนวณจะเพิ่มขึ้นเป็น N แสดงว่าตามทฤษฎีนี้จะไม่เหมาะสมสำหรับการนำมาใช้ในการควบคุมแบบปรับตัวเอง เมื่อการคำนวณไม่ควรจะเพิ่มตามเวลา

ตามคุณสมบัติที่ต้องการนี้เราใช้สมการในรูปแบบของรีเคอร์ซีฟที่เราสามารถจำกัดการคำนวณในรูปแบบของแต่ละการสังเกตของข้อมูลครั้งล่าสุด จะไม่มีมิติของ $X'X$ ที่จะเพิ่มขึ้นถึง N

$$S(t) \triangleq (X_t'X_t) \qquad 2.2.6$$

เมื่อ X_t คือ เมตริกซ์ของข้อมูลที่เราทราบค่าซึ่งจะขึ้นกับเวลา t จากสมการที่ 2.2.2 เราจะเขียนในรูปแบบได้ดังนี้

$$\begin{bmatrix} \phi_{t-1} \\ \phi(t) \end{bmatrix} = \begin{bmatrix} X_{t-1} \\ x(t) \end{bmatrix} \theta + \begin{bmatrix} e_{t-1} \\ e(t) \end{bmatrix} \qquad 2.2.7$$

$\hat{\theta}(t)$ เป็นการประมาณของ n เวกเตอร์ใช้สำหรับข้อมูลทั้งหมดที่ขึ้นกับเวลา t จาก 2.2.3 จะเขียนเป็น

$$\hat{\theta}(t) = S(t)^{-1}[X'_{t-1}\phi_{t-1} + x(t)\phi(t)]$$

$$= S(t)^{-1}[S(t-1)\hat{\theta}(t-1) + x(t)\phi(t)] \qquad 2.2.8$$

แต่ $S(t) = S(t-1) + x(t)\,x'(t)$

ดังนั้น $\hat{\theta}(t) = S(t)^{-1}[S(t)\hat{\theta}(t-1) - x(t)\,x'(t)\hat{\theta}(t-1) + x(t)\phi(t)]$

$$= \hat{\theta}(t-1) + S(t)^{-1}x(t)[\phi(t) - x'(t)\hat{\theta}(t-1)] \qquad 2.2.9$$

สมมุติให้ $S(t)^{-1}\ x(t) = K(t)$

รูปแบบของริเคอร์ซีฟ θ คือ

prediction error

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[\phi(t) - x'(t)\hat{\theta}(t-1)] \qquad 2.2.10$$

K(t) คือ n เวกเตอร์ (หรือเรียกว่า คาร์มานเกน)

การประมาณค่าพารามิเตอร์จะต้องใช้ข้อมูลที่เป็นค่าปัจจุบันสำหรับแต่ละพารามิเตอร์ที่อยู่ในรูปของการป้อนกลับ

[new estimate] = [old estimate] + gain × (prediction error of old model) 2.2.11

เราสมมุติให้ส่วนกลับของ S(t) คือ P(t)

$$P(t) = (S(t-1) + xx')^{-1}$$

$$= \frac{P(t-1) - P(t-1)\,xx'\,P(t-1)}{(1 + x'P(t-1)\,x)} \qquad 2.2.12$$

จากทฤษฎีของ RLS ในสมการที่ 2.2.10 จะได้

$$K(t) = \frac{P(t-1)\ x(t)}{1 + x'(t)\ P(t-1)\ x(t)} \qquad 2.2.13$$

$$P(t) = (I-K(t)\ x'(t))P(t-1) \qquad 2.2.14$$

จากสมการที่ 2.2.13 และ 2.2.14 นี้จะเห็นได้ว่าไม่จำเป็นต้องมีการอินเวอรส์เมตริกซ์ โดยที่ S^{-1} ที่ปรากฎอยู่เพียงหลังจากการได้มีการสังเกต n ค่า แต่เรามักนิยมให้ค่าเริ่มต้นของอัลกอริทึมนี้เป็น P(O) เป็นรูปแบบของเมตริกซ์ทะแยงมุม I ซึ่งเราสามารถเลือกค่าเริ่มต้นเท่าใดก็ได้ แต่ถ้า P(O) มีค่ามาก สมมุติตามวิทยานิพนธ์เราใช้ 10^5 จะมีผลทำให้การเปลี่ยนแปลงค่าเริ่มต้นของ $\hat{\theta}(t)$ เปลี่ยนแปลงเข้าสู่การประมาณค่าพารามิเตอร์ที่แท้จริงได้อย่างรวดเร็วกว่า ถ้า P(O) น้อยๆ จะทำให้เกิดการเปลี่ยนแปลงของ $\hat{\theta}(t)$ เป็นไปอย่างช้าๆ แต่การที่กำหนดค่าเริ่มต้นที่มากๆ เราจำเป็นที่จะต้องมีตัวจำกัดการควบคุม เพื่อป้องกันการเปลี่ยนแปลงในตอนเริ่มต้นที่มีค่ามากๆ

จาก $S(t) = \Sigma\ x(i)x'(i)$

ดังนั้นขนาดของ S จะมีค่าเริ่มต้นที่น้อยๆ แต่จะมีแนวโน้มเข้าสู่ค่าอนันต์ดังนั้นขนาดของ K จะมีแนวโน้มเข้าสู่ 0 และ $\hat{\theta}(t)$ จะเป็นค่าเวกเตอร์คงที่ ซึ่งถ้าเป็นจริงดังนี้ ค่าพารามิเตอร์ที่แท้จริงจากการประมาณจะคงที่ แต่ในทางปฏิบัติเราต้องการอัลกอริทึมที่ติดตามการเปลี่ยนแปลงค่าพารามิเตอร์อย่างช้าๆ (คือ ตัวควบคุมแบบปรับตัวเองยังคงมีการปรับตัวอยู่)

เรามีอยู่หลายทาง จากสมการที่ 2.2.10 ในการที่จะจัดการให้มีการปรับตัวอยู่ตลอดเวลาอย่างช้าๆ แต่ที่นิยมใช้คือ การให้ความสำคัญของข้อมูลเก่าลดลง และให้ความสำคัญกับข้อมูลปัจจุบันมากกว่าโดยใช้ค่าแฟกตอร์การลืม (Forgeting Factor) โดยที่ $0 < \beta <= 1,\ I = \sum_{1}^{t} \beta^{t-i}.e^2(i)$

การจะทำให้มีค่าต่ำที่สุดตามทฤษฎี RLS จาก 2.2.13 และ 2.2.14 คือ

$$K(t) = \frac{P(t-1)\ x(t)}{\beta + x'(t)P(t-1)x(t)}$$

$$P(t) = (I - K(t)x'(t))P(t)/\beta$$

จากกรณีนี้ขนาดของ P และ ขนาดของ K จะไม่มีทางเป็น 0 ได้เพื่อที่ $\theta(t)$ สามารถเปลี่ยนค่าสำหรับเวลา t ที่มากๆได้

ถ้าค่าฟังก์ชั่นการลืมมีค่ายิ่งน้อย เช่น 0.95 จะทำให้มีผลของการเปลี่ยนแปลงอย่างรวดเร็ว แต่ถ้ามีค่าเป็น 0.999 จะเปลี่ยนแปลงไปอย่างช้าๆ

การประมาณค่าที่จะใช้ในการทดลองต่อไปนี้ จะมีสัญญลักษณ์ที่แตกต่างจากทฤษฎีที่กล่าวมาข้างต้นบ้าง ซึ่งทฤษฎีข้างต้นนี้ ได้อ้างอิงจากวารสารอ้างอิง (4) ส่วนสัญญลักษณ์ที่นำมาใช้ในการทดลองนั้น ได้อ้างอิงจากวารสารอ้างอิง (2)

บทที่ 3 การคำนวณ

3.1 โมเด็ลเปิดลูป

เราถือว่าระบบที่เราต้องการควบคุมเป็นระบบอันดับสอง และมีทรานสเฟอร์ฟังก์ชันดังนี้

$$\frac{Kp.wo^2}{s^2+2xiwo.s+wo^2}$$

เพราะมีคุณสมบัติครอบคลุมระบบเกือบทุกชนิด

เมื่อแปลงมาเป็นโมเด็ลทางเวลาไม่ต่อเนื่อง จะต้องรวมกับ ZOH แล้วจึงทำการแปลง Z จะได้โมเด็ลเป็น

$$\frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}$$

โดย

S_1=1-alpha*(beta+(xi*wo*gamma/omegad))

S_2=sqr(alpha)+alpha*((xi*wo*gamma/omegad)-beta)

b_1=Kp.S_1

b_2=Kp.S_2

a_1=-2*alpha*beta

a_2=sqr(alpha)

จะเห็นว่าในการคำนวณหาโมเด็ลทางระบบทางเวลาไม่ต่อเนื่อง เราสามารถกำหนดคุณสมบัติได้จาก xi และ wo

3.2 โมเด็ลปิดลูป

เรากำหนดให้เป็นโมเด็ลทางเวลาไม่ต่อเนื่องอยู่แล้ว

$$\frac{d_1z^{-1}+d_2z^{-2}}{1+c_1z^{-1}+c_2z^{-2}}$$

โดย

c_1=-2*exp(-xis*wos*h)*cos(wos*h*sqrt(1-sqr(xis)))

c_2=exp(-2*xis*woh*h)

เรากำหนดให้ซีโร่ของระบบปิดลูปอยู่ใกล้ๆ ออริจิน(origin) และมีอัตราการการขยายไฟตรงเป็น 1

$d_1=0.9(1+c_1+c_2)$

$d_2=0.1(1+c_1+c_2)$

เช่นกัน เราสามารถกำหนดคุณสมบัติของระบบได้ด้วย xis และ wos

3.2 โมเด็ลของตัว ประมาณค่า

พิจารณาสมการความสัมพันธ์ของระบบปิดลูป

$$\frac{y(k)}{u(k)} = \frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}$$

เขียนในรูปสมการผลต่างจะได้

$$y(k) = -a_1y(k-1)-a_2y(k-2)+b_1u(k-1)+b_2u(k-2)$$

ดังนั้น โมเด็ลการประมาณค่า จะต้องอยู่ในรูปเดียวกับโมเด็ลของระบบจริง ซึ่งมีความสัมพันธ์

$$y(k) = \hat{\theta}^T(k).\phi(k+1)$$

โดย $\phi(k+1) = [-y(k)\ \ -y(k-1)\ \ u(k)\ \ u(k-1)]$

$\hat{\theta}(k) = [\ \hat{a}_1\quad \hat{a}_2\quad \hat{b}_1\quad \hat{b}_2\ ]$

ทำการประมาณค่าพารามิเตอร์ ตามขั้นตอนต่อไปนี้

1)กำหนดค่าเริ่มต้น $k=2$ $P(2)=10^5I$ $\theta(2)=[\ 0\ 0\ 0\ 0\ ]$

$y(2)=y(1)=u(2)=u(1) = 1$

เพราะโมเด็ลของระบบของเราเป็นโมเด็ลเวลาไม่ต่อเนื่องอันดับสอง ซึ่งต้องการค่าเริ่มต้น 2 ค่า

2) $\phi(3) = [-y(2)\ \ -y(1)\ \ u(2)\ \ u(1)]$

3) $K(k) = P(k)\phi^T(k+1)[\ lamda+\phi(k+1)P(k)\phi^T(k+1)\ ]^{-1}$

$K(2) = P(2)\phi^T(3)[\ lamda+\phi(3)P(2)\phi^T(3)\ ]^{-1}$

4) นำข้อมูลเก่า $\phi(3)$ ป้อนไปใน $G(z^{-1})$ จะได้ $y(3)$ เป็นเอาท์พุต

5) $P(k+1) = [\ I - K(k)\phi(k+1)\ ]P(k)/lamda$

$P(3) = [\ I - K(2)\phi(3)\ ]P(2)/lamda$

6) $\hat{\theta}(k+1) = \hat{\theta}(k) + K(k)[\ y(k+1)-\phi(k+1)\theta(k)\ ]$

$\hat{\theta}(3) = \hat{\theta}(2) + K(2)[\ y(3)-\phi(k+1)\theta(k)\ ]$

เมื่อได้ ϕ ใหม่ ก็จะนำไปคำนวณหาค่าเร็กกูเลเตอร์ที่ยังคงรักษาสมรรถนะของระบบปิดลูปให้คงเดิม

3.4 โมเด็ลของเร็กกูเลเตอร์

เราสามารถคำนวณหาค่าของเร็กกูเลเตอร์ได้จาก AR+BS = ? ซึ่งทางขวาของสมการมีโอกาสเกิดขึ้นได้ 2 กรณี คือ

1) กรณีหักล้าง B

$$\frac{TB}{B(AR_1+S)} = \frac{BmAo}{AmAo} \quad \begin{array}{l} \rightarrow\ Bm = 1 \quad T = Ao \\ \rightarrow\ AR+BS = BAmAo \end{array}$$

2) กรณีไม่หักล้าง B

$$\frac{TB}{AR+BS} = \frac{BmAo}{AmAo} \quad \begin{array}{l} \rightarrow\ Bm = B \quad T = Ao \\ \rightarrow\ AR+BS = AmAo \end{array}$$

จะพบว่าไม่มีปัญหาในการหาค่า T เราจึงหันมาพิจารณา ส่วนของสมการ ว่าจะหาค่า R,S จากวิธีเทียบสปส.หรือ วิธีไดโอแฟนไทน์ และมีผลแตกต่างกันอย่างไร

3.4.1 วิธีเทียบสปส. วิธีนี้กำลังของ R,T,S จะถูกกำหนดตายตัวตาม ผ.5

3.4.1.1 กรณีหักล้าง B จาก ผ.5 จะได้

$$degR = degB + degAo + degAm - degA$$

$$degS = degA - 1$$

$$degAo >= 2degA - degAm - degB - 1$$

$$degAo >= 4 - 2 - 1 - 1 = 0$$

$* \ Ao = 1$

$$degS = 2 - 1 = 1$$

$* \ S = s_0 z + s_1$

$$degR = 1 + 0 + 2 - 2 = 1$$

* $R = r_o z + r_1$

แทนค่าใน AR+BS ได้

$$(z^2+\hat{a}_1 z+\hat{a}_2)(r_o z+r_1)+(\hat{b}_1 z+\hat{b}_2)(s_o z+s_1) = (\hat{b}_1+\hat{b}_2)(z^2+c_1 z+c_2).1$$

เทียบสปส. ได้

$$r_o = \hat{b}_1$$

$$r_1 = \frac{\hat{b}_1\hat{b}_2c_2-\hat{b}_1c_2+\hat{b}_2c_1-\hat{a}_2\hat{b}_1-\hat{b}_2c_1-\hat{b}_2a_1-\hat{b}_2/\hat{b}_1}{\hat{a}_1+\hat{b}_1\hat{a}_2-\hat{b}_2/\hat{b}_1}$$

$$s_o = c_1+\hat{a}_1+(\hat{b}_2-r_1)/\hat{b}_1$$

$$s_1 = \hat{b}_2c_2-\hat{a}_2r_1$$

3.4.1.2 ไม่หักล้าง B

มีสมการ AR+BS = AmAo และจาก ผ.5 จะได้

degR = degAo + degAm - degA

degS = degA - 1

degAo >= 2degA - degAm - 1

degAo >= 4 - 2 - 1 = 1

Ao = z

degS = 2 - 1 = 1

$S = s_o z + s_1$

degR = 1 + 2 - 2 = 1

$R = r_o z + r_1$

ถ้าสังเกตจะพบว่า degAo จะเป็น 1 เพื่อทำให้กำลังทางด้านขวาเท่ากับกำลังทางด้านซ้าย แทนค่าลงใน AR+BS จะได้

$$(z^2+\hat{a}_1 z+\hat{a}_2)(r_o z+r_1)+(\hat{b}_1 z+\hat{b}_2)(s_o z+s_1) = (z^2+c_1 z+c_2).z$$

เทียบสปส.จะได้

$$r_o = 1$$

$$r_1 = -\hat{b}_2 . s_1/\hat{a}_2$$

$$s_o = (c_1-\hat{a}_1-r_1)/\hat{b}_1$$

$$s_1 = \frac{c_2-\hat{a}_2-\hat{b}_2(c_1-\hat{a}_1)/\hat{b}_1}{\hat{b}_1-(\hat{a}_1-\hat{b}_2/\hat{b}_1)\hat{b}_2/\hat{a}_2}$$

3.4.2 วิธีไดโอแฟนไทน์

คำตอบของสมการอยู่ในรูป

$$X = pc_o+rt$$

$$Y = qc_o+st$$

เมื่อเราพูดถึง deg ใน 3.4.1 ทำให้เราหันมาสนใจ degR,degS ที่อาจเป็นไปได้ โดยเราจะยก กรณีหักล้าง B ขึ้นมาเป็นตัวอย่าง AR+BS = BAmAo พบว่า R และ S พบว่าเกิดขึ้นได้ 3 แบบ

1) degR = degS นั่นคือ

A	R	+	B	S	=	B	Am	Ao
2	1		1	1		1	1	0

กรณีเช่นนี้เกิดขึ้นได้ โดยกำลังที่ 2 ของ AR และ BS รวมกันก่อนแล้วจึงไปเท่ากับ BAmAo ส่วนกำลังที่ 3 ของ AR จะไปเท่ากับ BAmAo โดยตรง วิธีนี้พบในกรณีเทียบสปส.

2) degR>degS นั่นคือ

A	R	+	B	S	=	B	Am	Ao
2	1		1	2		1	2	0

กรณีนี้ทุกๆกำลังของ AR จะไปรวมกับ BS ก่อนแล้วจึงได้ผลเท่ากับ BAmAo ซึ่งจะพบในการใช้ ไดโอแฟนไทน์ ซึ่งเป็นค่าที่ไม่ต้องการ

3) degR>degS

จาก $X = X_o - b_o t$

$$Y = Y_o + a_o t$$

จะเห็นว่ากำลังของ X และ Y เกิดขึ้นได้หลายกรณี ขึ้นกับว่า $b_o t$ หักล้างกับ X_o ได้มากแค่ไหน และ $a_o t$ หักล้างกับ Y_o ได้มากแค่ไหน สำหรับ สมการไดโอแฟนไทน์ AR+BS = BAmAo = C R,S จะอยู่ในรูปของ

$$R = R_o - B_o t$$
$$S = S_o + A_o t$$

ถ้าเราต้องการลดกำลังของ S เราจะต้องกำหนดค่าของ t ให้ $A_o t$ หักล้างกับ S_o มากที่สุด โดยเขียน S_o ใหม่ให้อยู่ในรูปที่หักล้างกับ $A_o t$ ได้ นั่นคือ

$S_o = A_o u + v$ โดย u เป็นผลหารที่ได้จากการนำ A_o ไปหาร Y_o และ v เป็นเศษที่เหลือ และกำลังของเศษนี้ จะต้องน้อยกว่ากำลังของตัวหาร

เมื่อแทนค่ากลับไปจะได้

$$S_o = A_o u + v - A_o t$$

ถ้าต้องการให้ S มีกำลังน้อยสุด t จะต้องเท่ากับ u

ได้ $S = v$

$$R = R_o - B_o u$$

ในทางปฏิบัติ A และ B จะไม่มีตัวประกอบร่วม ซึ่งจะทำให้ หรม.เป็น 1 ดังนั้น

$$A_o = A,\ B_o = B,\ C_o = C$$

จาก degv<degAo จะได้ degv<degA degv < 2 degS < 2

และจาก AR+BS = C เมื่อ degA = 2 degB = 1 degC = 3 จะเห็นว่า $\deg S_o$ จะต้องเป็น 1 และเมื่อรวมกับ $-B_o u$ หรือ $-Bu$ ซึ่งมี degu = 1 และ degB = 1

ดังนั้น degBu = 2 ทำให้ degS = 2 นั่นคือ เมื่อใช้วิธีนี้จะทำให้ degR>degS ตามต้องการ

กำหนดพารามิเตอร์ของระบบปิดลูปให้เท่ากับของระบบเปิดลูปก่อน โดย

Kp = 1

xi = 0.65 xis = 0.65

wo = 5 sec. wos = 5 sec.

และทำการทดสอบความสามารถในการปรับตัว โดยเพิ่ม xi ของระบบเปิดลูปอีก 0.09 เป็น 0.74 ที่รอบการทำงานที่ 40

เมื่อทำการทดลองกับการควบคุมแบบปรับตัวที่ใช้วิธีเทียบสปส.หาค่าเร็กกูเลเตอร์ ค่า R,T,S,Ao จะถูกกำหนดไว้ก่อน โดย degR = 1 degS = 1 degT = 0 และ

กรณีหักล้าง B จะได้ degAo = 0, Ao = 1

กรณีไม่หักล้าง B จะได้ degAo = 1, Ao = q

แต่เมื่อเรากำหนด xi = 0.65 และ wo = 5 ดังนั้นซีโร่ของระบบเปิดลูป จะอยู่ในวงกลมหนึ่งหน่วย และเป็นกรณีไม่หักล้าง B

เมื่อเริ่มทำงาน ทั้ง u และ y จะมีค่าสูงมาก เพราะตัวประมาณค่า ยังให้ค่าที่ไม่ถูกต้อง จนมาถึงรอบที่ 20 ค่า u และ y จึงลู่เข้าหา u_c

เมื่อถึงรอบการทำงานที่ 40 กำหนดให้เพิ่ม xi อีก 0.09 เป็น 0.74 แต่การเปลี่ยนแปลงนี้ ตัวประมาณค่าไม่สามารถตรวจวัดได้ แม้ว่าพารามิเตอร์จะเปลี่ยนไปแค่ไหนก็ตาม

ก่อนอื่นเราต้องมาทำความเข้าใจในความสัมพันธ์ระหว่าง ตัวประมาณค่า กับโมเด็ลของระบบ ถ้าโมเด็ลเป็นแบบเร็กกูเลเตอร์ ตัวประมาณค่า จะต้องถูกกระตุ้นให้ทำงานด้วยสัญญาณรบกวน และถ้าโมเด็ลเป็นแบบเซอร์โว จะต้องกระตุ้นตัวประมาณค่าด้วย การเปลี่ยนแปลงของสัญญาณอ้างอิง ดังนั้น ถ้าเรากำหนดให้ u_c มีค่าเดียวตลอด ตัวประมาณค่าจะไม่สามารถวัดการเปลี่ยนแปลงของพารามิเตอร์ได้

ดังนั้น จึงกำหนดให้ มีการกระตุ้นตัวประมาณค่าด้วย u_c ที่มากกว่า 1 ณ ขณะที่เปลี่ยนแปลง xi โดยเปลี่ยนเป็น 2,5,10 แต่ก็ยังไม่เพียงพอที่จะกระตุ้นตัวประมาณค่าให้ค่า $\hat{\theta} = \theta$ ได้ จึงเพิ่ม u_c เป็น 100 ดังรูป และ $\hat{\theta}$ สามารถลู่เข้าสู่ θ ได้

y สามารถติดตาม u ได้อย่างรวดเร็ว เพราะ มีซีโร่ 1 ตัวเท่ากับโพล

เมื่อพิจารณาให้ตัวควบคุมนี้แก้ปัญหาเร็กกูเลเตอร์ โดยป้อนสัญญาณรบกวนที่มีค่าเฉลี่ย 1 เข้าไปที่ u พบว่าเพียงพอที่จะกระตุ้นตัวประมาณค่าเทียบเท่ากับ $u_c = 100$

และมีการเปลี่ยนแปลงอย่างรวดเร็ว เพราะมีซีโร่เท่ากับโพล เป็นตัวป้อนกลับ

ต่อไปทำการทดลองด้วยวิธีไดโอแฟนไทน์ ที่มี degR = 2 degS = 1 และ $A_o = 1$ นั่นคือ มีตัวป้อนกลับเป็นอินทิเกรต พบว่าตอนเริ่มทำงาน จะไม่มีการแกว่งของสัญญาณ และเมื่อเปลี่ยน x_i เพียงอย่างเดียว ตัวประมาณค่าก็ไม่สามารถวัดพารามิเตอร์ที่เปลี่ยนแปลงได้ จึงต้องเพิ่ม u_c เป็น 100 เช่นกัน

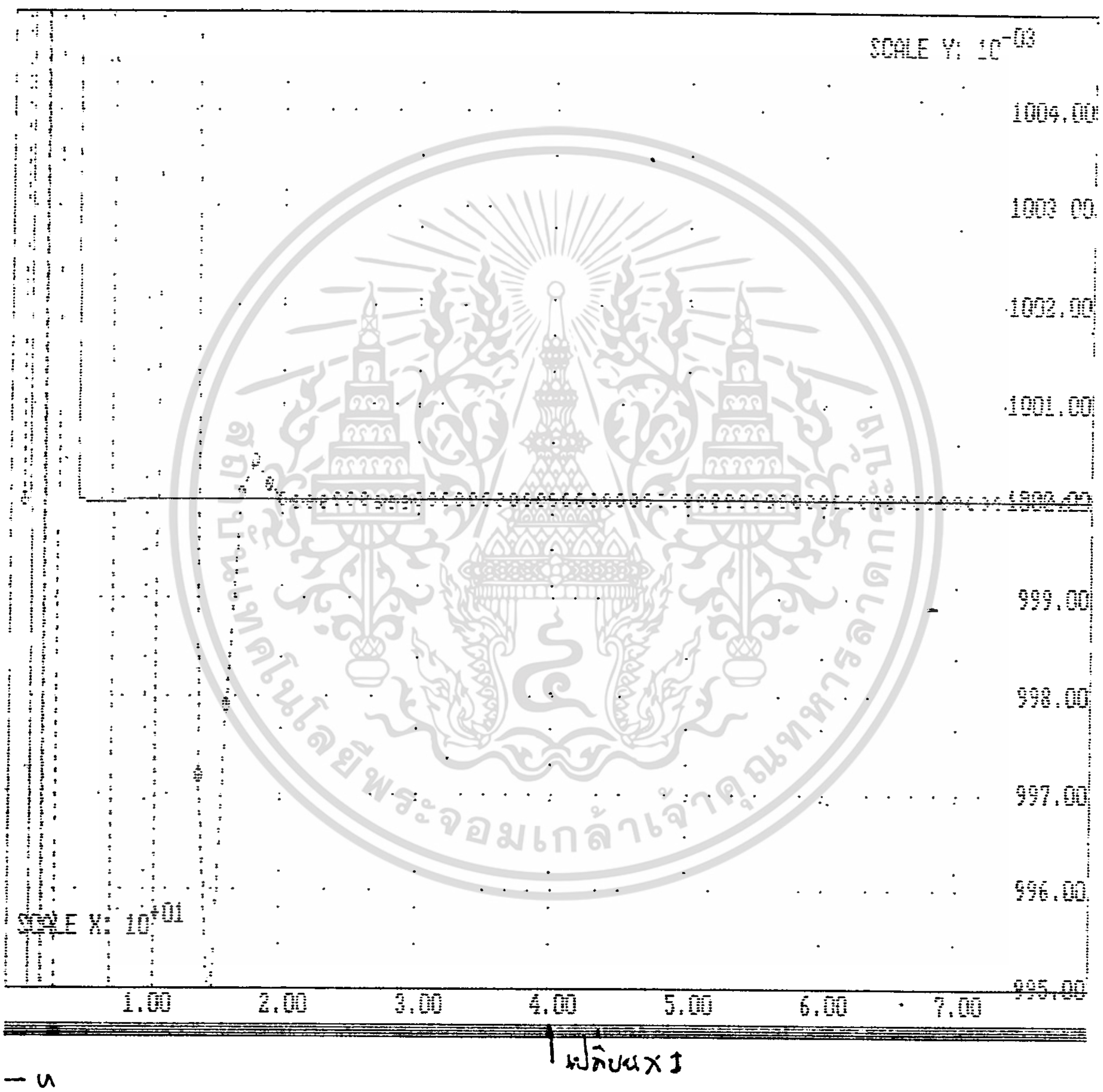
เมื่อป้อนสัญญาณรบกวนเข้าไปที่ u การเปลี่ยนแปลงของสัญญาณจะไม่เปลี่ยนตามสัญญาณรบกวนทันที เพราะมีการป้อนกลับเป็นอินทิเกรเตอร์

```
                                           Kp.wo~2
OPEN LOOP PROCESS     G(S)=  -----------------
                            s~2+2wi.xi.s+wo~2
PROCESS PARAMETER
                Kp =1
        xi(0<xi<1) =0.65
        wo(sec)    =5

                                              d1q~1+d2q~2
CLOSED LOOP SYSTEM    Hm(q~1) = ------------------
                                      1 + c1q~1 + c2q~2
SYSTEM PARAMETER
      xis(0<xis<1) =0.65
      wos(sec)     =5
Sampling time    h =0.265
Observer pole   A0 = z1q~1 + z2q~2
  For cancel B        A0 = 1
  For not cancel B    A0 = q
forgeting factor LAMDA=0.9
```

รูป 4.1 แสดง การป้อนค่า สำหรับวิธีเทียบ สัมประสิทธิ์

```
loop1=28
a1=-4.51771925E-01theta[1]=-4.5177191701E-01
a2= 1.78619042E-01theta[2]= 1.7861903785E-01
b1= 4.68584316E-01theta[3]= 4.6858431510E-01
b2= 2.58262801E-01theta[4]= 2.5826280587E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.7E-21 0=      01
s0=-3.4175834232E-09
s1= 4.1862778197E-09
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-6.0528814225E-09

d1=-2.4E-26d2=-1.8E+31
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+00U= 1.0000000053E+00Y= 9.9999992194E-01
PHI[1]=-9.9999992194E-01
PHI[2]=-9.9999987389E-01
PHI[3]= 1.0000000053E+00
PHI[4]= 1.0000000053E+00


loop1=100
a1=-4.71512144E-01theta[1]=-4.5177191702E-01
a2= 1.40717633E-01theta[2]= 1.7861903792E-01
b1= 4.43226326E-01theta[3]= 4.6858431507E-01
b2= 2.25979163E-01theta[4]= 2.5826280584E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.7E-21 0=      01
s0=-3.5316193971E-09
s1= 4.1424016793E-09
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-5.9894415121E-09

d1=-2.4E-26d2=-1.8E+31
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+00U= 1.0000000054E+00Y= 1.0000000054E+00
PHI[1]=-1.0000000054E+00
PHI[2]=-1.0000000054E+00
PHI[3]= 1.0000000054E+00
PHI[4]= 1.0000000054E+00
Press any key to return to Turbo Pascal
```

รูป 4.2 ผลการทดลอง ก่อนเปลี่ยน XI กับหลังเปลี่ยน
เมื่อไม่มีการกระตุ้น ตัวประมาณค่า (HC คงที่)



— u
-o- y

รูป 4.3 แสดง อินพุต และเอาท์พุต ของระบบ และผลของการเปลี่ยน x1 เมื่อไม่มีการกระตุ้น ตัวประมาณค่า

```
loop1=32
a1=-4.51771925E-01theta[1]=-4.5177191701E-01
a2= 1.78619042E-01theta[2]= 1.7861903785E-01
b1= 4.68584316E-01theta[3]= 4.6858431509E-01
b2= 2.58262801E-01theta[4]= 2.5826280587E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.7E-21 0=      01
s0=-3.4290450273E-09
s1= 4.1816197979E-09
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-6.0461464525E-09

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+00U= 1.0000000053E+00Y= 1.0000000066E+00
PHI[1]=-1.0000000116E+00
PHI[2]=-1.0000000138E+00
PHI[3]= 1.0000000053E+00
PHI[4]= 1.0000000053E+00


loop1=100
a1=-4.71512144E-01theta[1]=-4.4813186446E-01
a2= 1.40717633E-01theta[2]= 1.7435199906E-01
b1= 4.43226326E-01theta[3]= 4.6652704775E-01
b2= 2.25979163E-01theta[4]= 2.5968477571E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.8E-15 0=      01
s0= 2.4324402530E-03
s1= 3.2058339777E-03
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-4.7748593763E-03

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 2.0000000000E+00U= 1.9982746419E+00Y= 1.9982733399E+00
PHI[1]=-1.9982751278E+00
PHI[2]=-1.9982756733E+00
PHI[3]= 1.9982705095E+00
PHI[4]= 1.9982766136E+00
Press any key to return to Turbo Pascal
```

รูป 4.9 ผลของการเปลี่ยน Uc เป็น 2

```
loop1=29
a1=-4.51771925E-01theta[1]=-4.5177191701E-01
a2= 1.78619042E-01theta[2]= 1.7861903785E-01
b1= 4.68584316E-01theta[3]= 4.6858431510E-01
b2= 2.58262801E-01theta[4]= 2.5826280587E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.7E-21 0=      01
s0=-3.4205087119E-09
s1= 4.1853297892E-09
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-6.0515106781E-09

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+00U= 1.0000000053E+00Y= 9.9999999110E-01
PHI[1]=-9.9999992194E-01
PHI[2]=-9.9999987389E-01
PHI[3]= 1.0000000053E+00
PHI[4]= 1.0000000053E+00


loop1=100
a1=-4.71512144E-01theta[1]=-4.3028896303E-01
a2= 1.40717633E-01theta[2]= 1.4690571088E-01
b1= 4.43226326E-01theta[3]= 4.5277441278E-01
b2= 2.25979163E-01theta[4]= 2.6383722800E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 7.1E-15 0=      01
s0= 2.9771639834E-02
s1= 1.9467437517E-02
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-3.4962798382E-02

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 5.0000000000E+00U= 4.9296228295E+00Y= 4.9295978727E+00
PHI[1]=-4.9296048638E+00
PHI[2]=-4.9295992440E+00
PHI[3]= 4.9295772795E+00
PHI[4]= 4.9296245301E+00
Press any key to return to Turbo Pascal
```

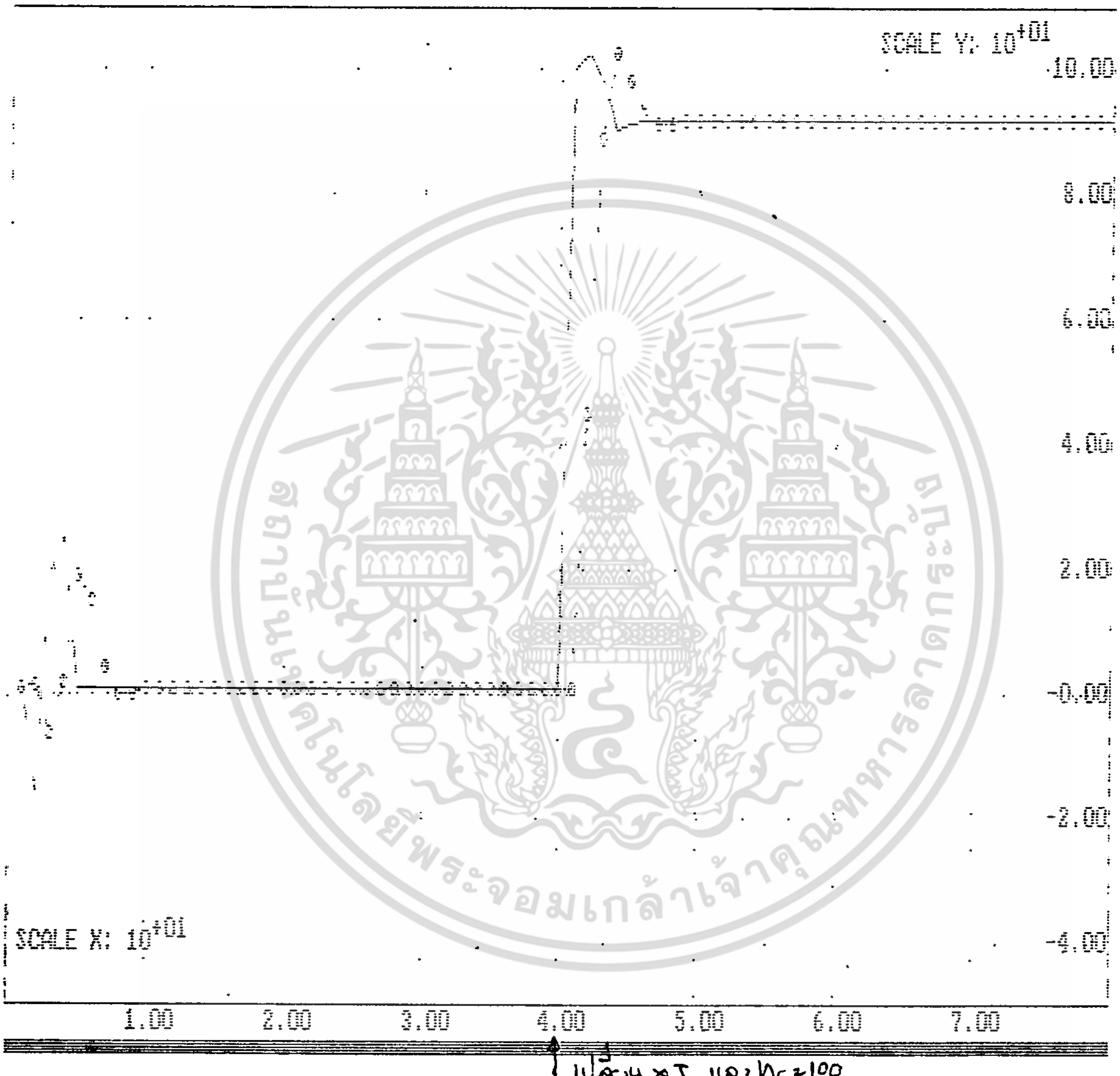
รูป 4.5 ผลของการเปลี่ยน uc เป็น 5

```
loop1=38
a1=-4.51771925E-01theta[1]=-4.5177191702E-01
a2= 1.78619042E-01theta[2]= 1.7861903787E-01
b1= 4.68584316E-01theta[3]= 4.6858431509E-01
b2= 2.58262801E-01theta[4]= 2.5826280586E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.7E-21 0=       01
s0=-3.4490429468E-09
s1= 4.1732517848E-09
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-6.0340472570E-09

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+00U= 1.0000000053E+00Y= 1.0000000053E+00
PHI[1]=-1.0000000053E+00
PHI[2]=-1.0000000053E+00
PHI[3]= 1.0000000053E+00
PHI[4]= 1.0000000053E+00


loop1=100
a1=-4.71512144E-01theta[1]=-4.4117523731E-01
a2= 1.40717633E-01theta[2]= 1.3621572576E-01
b1= 4.43226326E-01theta[3]= 4.4513516957E-01
b2= 2.25979163E-01theta[4]= 2.4990399709E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 7.1E-15 0=       01
s0= 6.3427500719E-02
s1= 2.1165425990E-02
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-3.8830498648E-02

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+01U= 9.5624014042E+00Y= 9.5623553917E+00
PHI[1]=-9.5623713468E+00
PHI[2]=-9.5623615674E+00
PHI[3]= 9.5623127091E+00
PHI[4]= 9.5624096625E+00
Press any key to return to Turbo Pascal
```

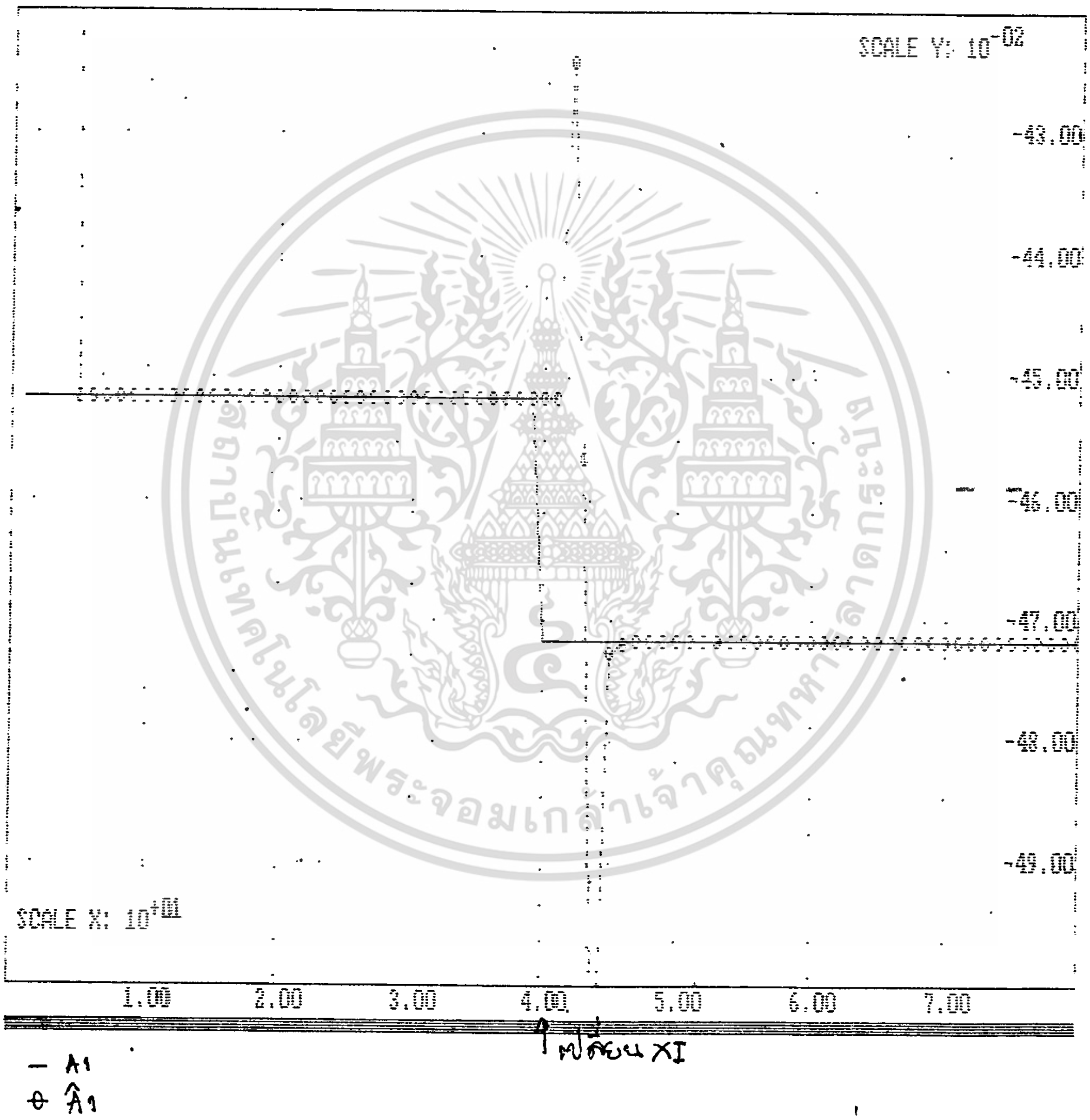
รูป 4.6 ผลต่อการเปลี่ยน Uc เป็น 10

```
loop1=38
a1=-4.51771925E-01theta[1]=-4.5177191702E-01
a2= 1.78619042E-01theta[2]= 1.7861903787E-01
b1= 4.68584316E-01theta[3]= 4.6858431509E-01
b2= 2.58262801E-01theta[4]= 2.5826280586E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.7E-21 0=      01
s0=-3.4490429468E-09
s1= 4.1732517848E-09
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-6.0340472570E-09

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+00U= 1.0000000053E+00Y= 1.0000000053E+00
PHI[1]=-1.0000000053E+00
PHI[2]=-1.0000000053E+00
PHI[3]= 1.0000000053E+00
PHI[4]= 1.0000000053E+00


loop1=100
a1=-4.71512144E-01theta[1]=-4.7160643587E-01
a2= 1.40717633E-01theta[2]= 1.4079157386E-01
b1= 4.43226326E-01theta[3]= 4.4346302223E-01
b2= 2.25979163E-01theta[4]= 2.2572209401E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 3.6E-15 0=      01
s0= 9.4467087514E-02
s1= 1.3758518046E-02
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-2.2058148927E-02

d1=-2.0E+19d2=-0.0E+00
c1=-4.5E-01c2= 1.8E-01
Uc= 1.0000000000E+02U= 9.2066929114E+01Y= 9.2065597396E+01
PHI[1]=-9.2065604020E+01
PHI[2]=-9.2065194026E+01
PHI[3]= 9.2065034156E+01
PHI[4]= 9.2066437110E+01
Press any key to return to Turbo Pascal
```

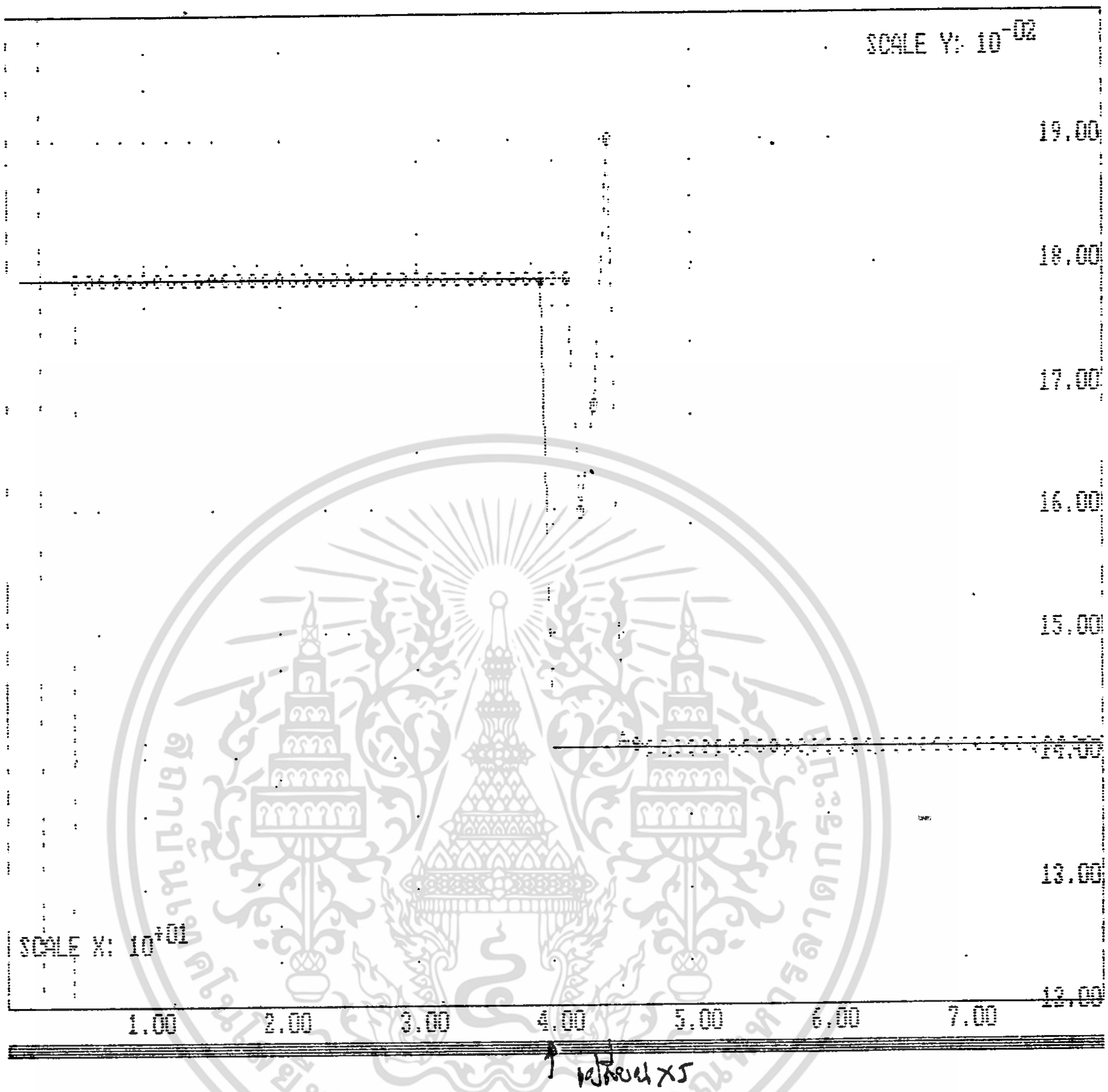
รูป4.7 ผลของการเปลี่ยน Uc เป็น 100



รูป 4.8 ผลของ Kc=100 และผลของการเปลี่ยน XI เป็น 0.74

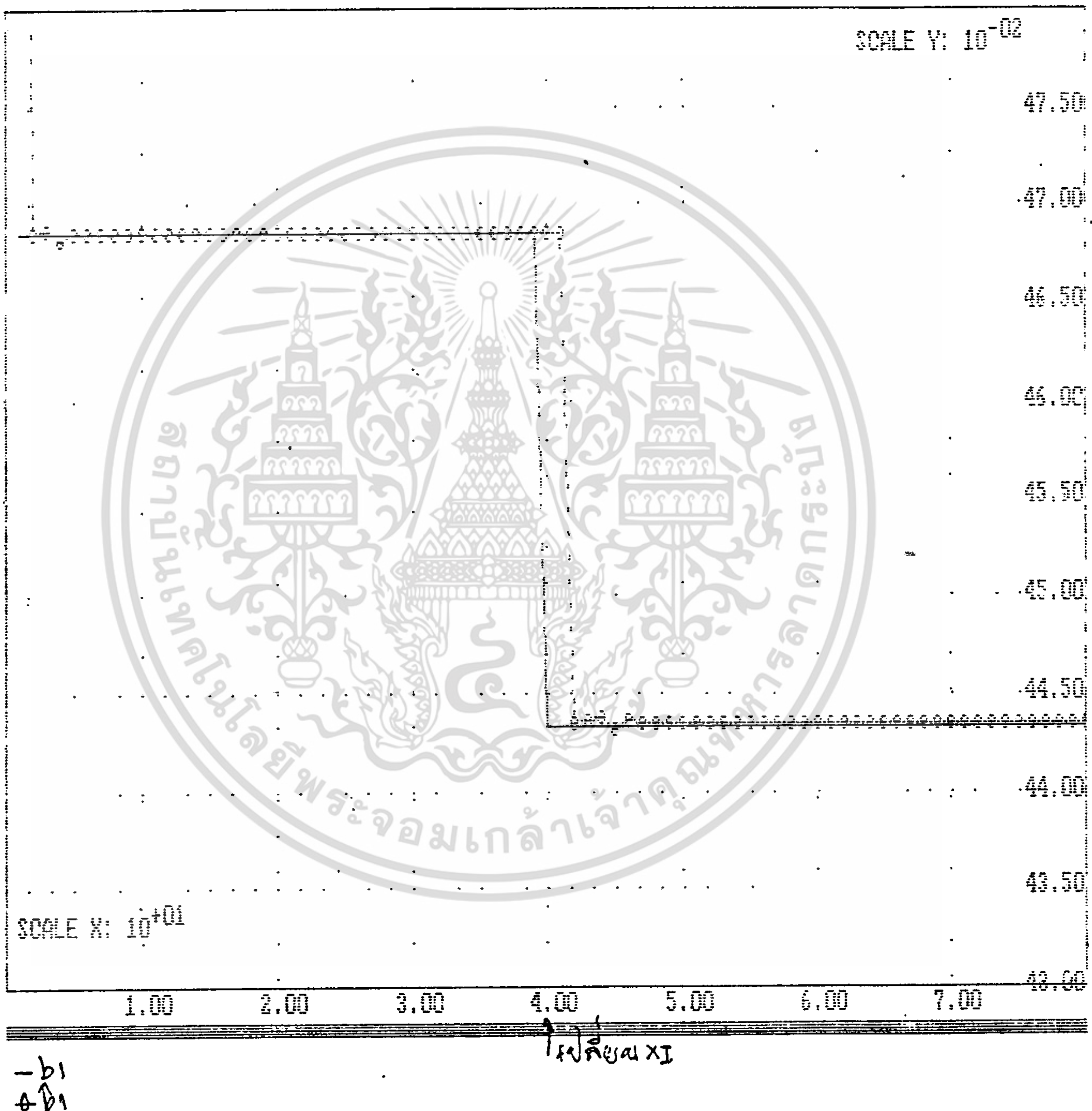


รูป 4.9 แสดงการเปลี่ยนแปลงค่า A1 เนื่องจากการเปลี่ยน XI

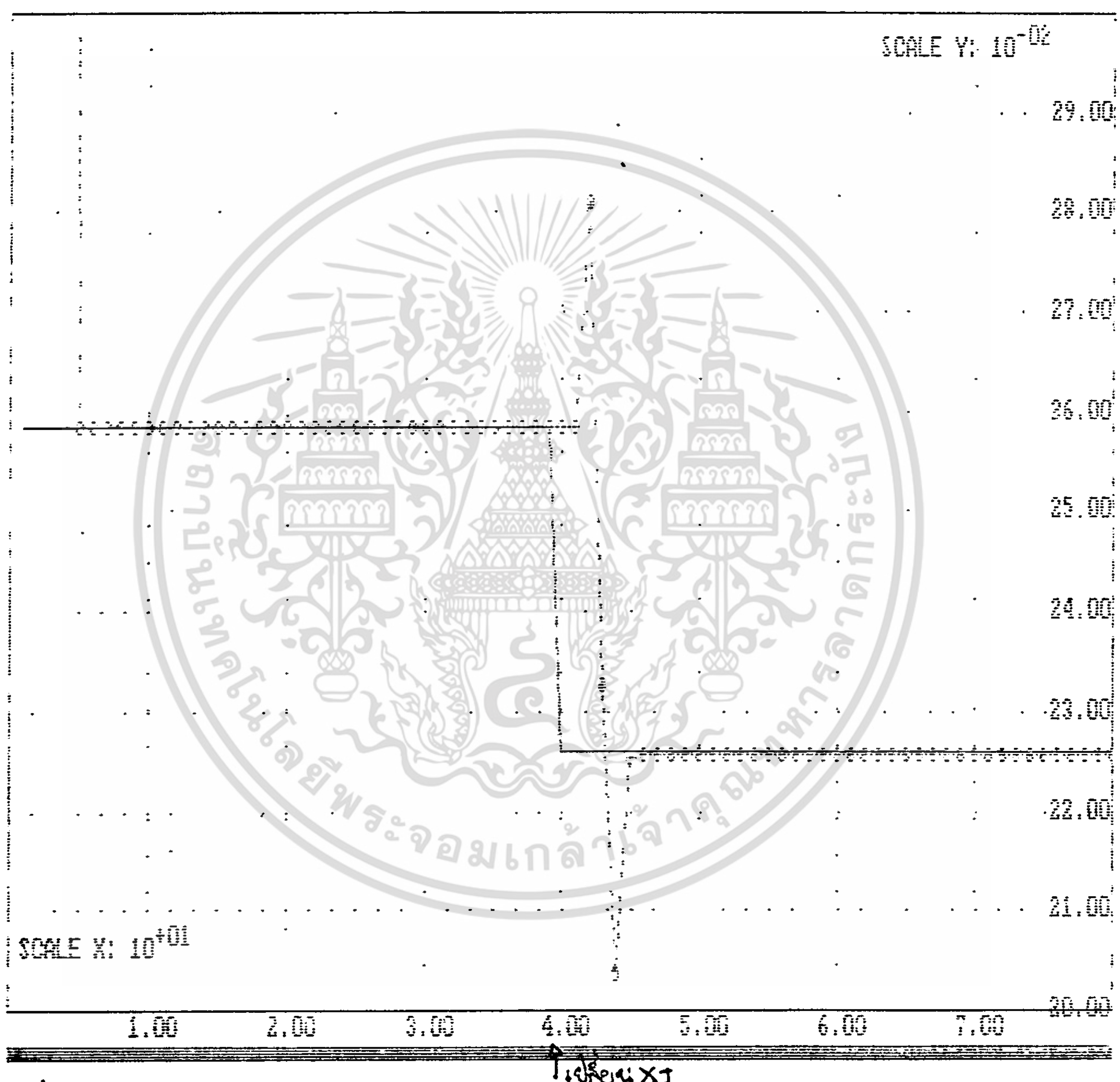


− A2
θ Â2

รูป 4.10 แสดง การเปลี่ยนแปลง ของ A2 เมื่อจาก X5



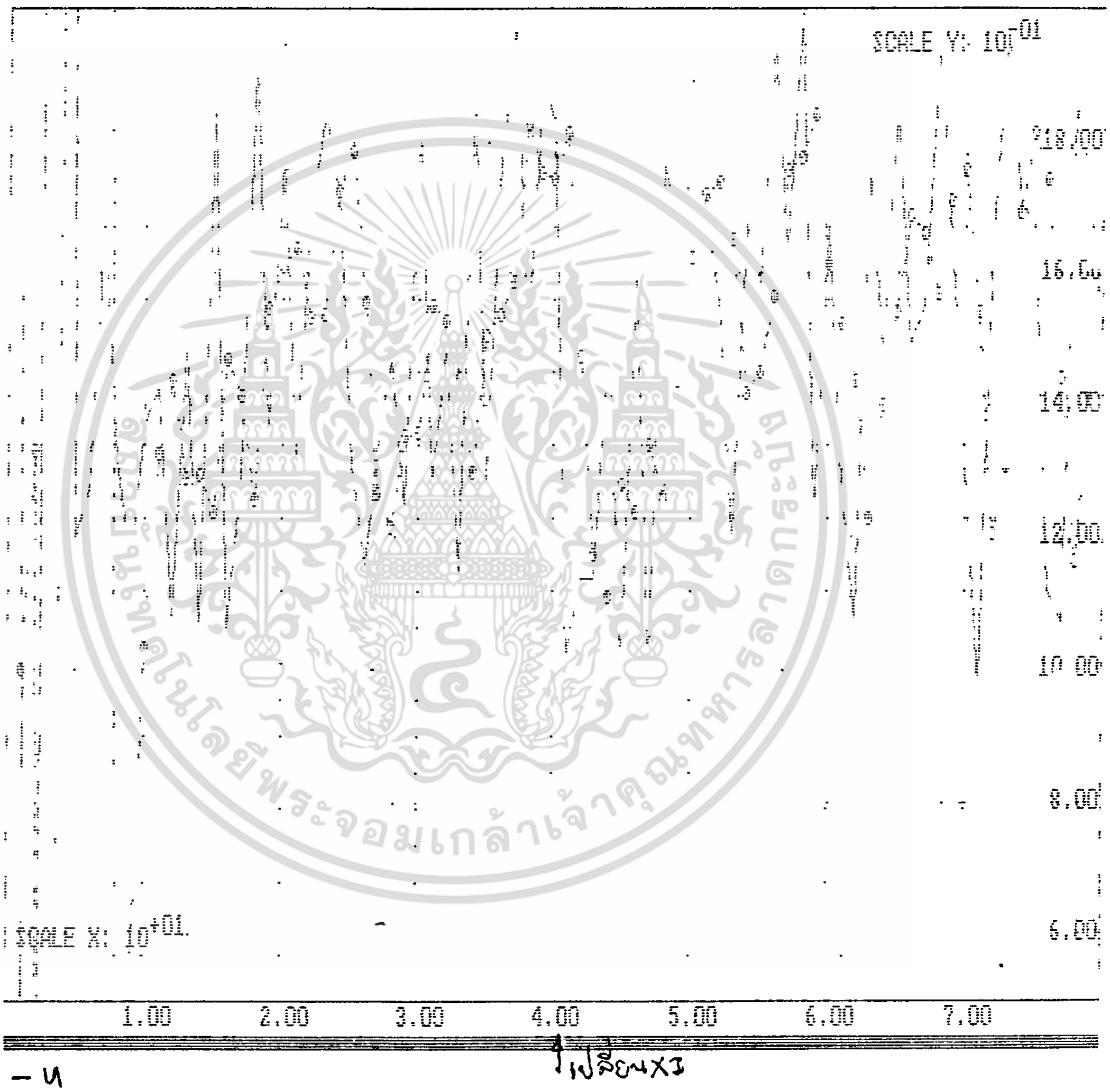
รูป 4.11 . แสดงผลของการเปลี่ยน b1 เนื่องจากการเปลี่ยน XI



— b_2

θ $\hat{b}_2$

รูป 4.12 แสดง ผลการเปลี่ยน b_2 เนื่องจาก การเปลี่ยน XJ

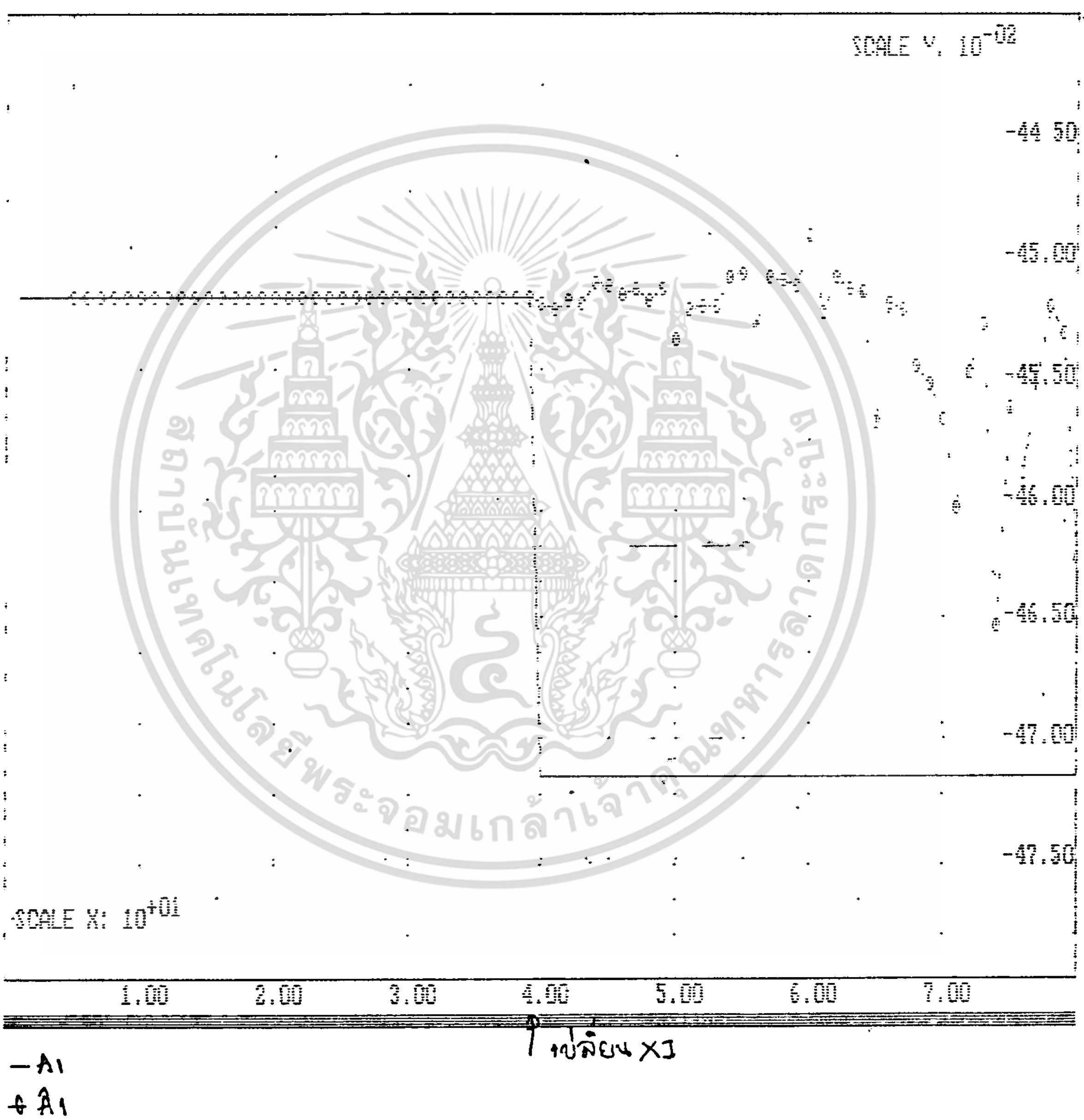


− น
+ y

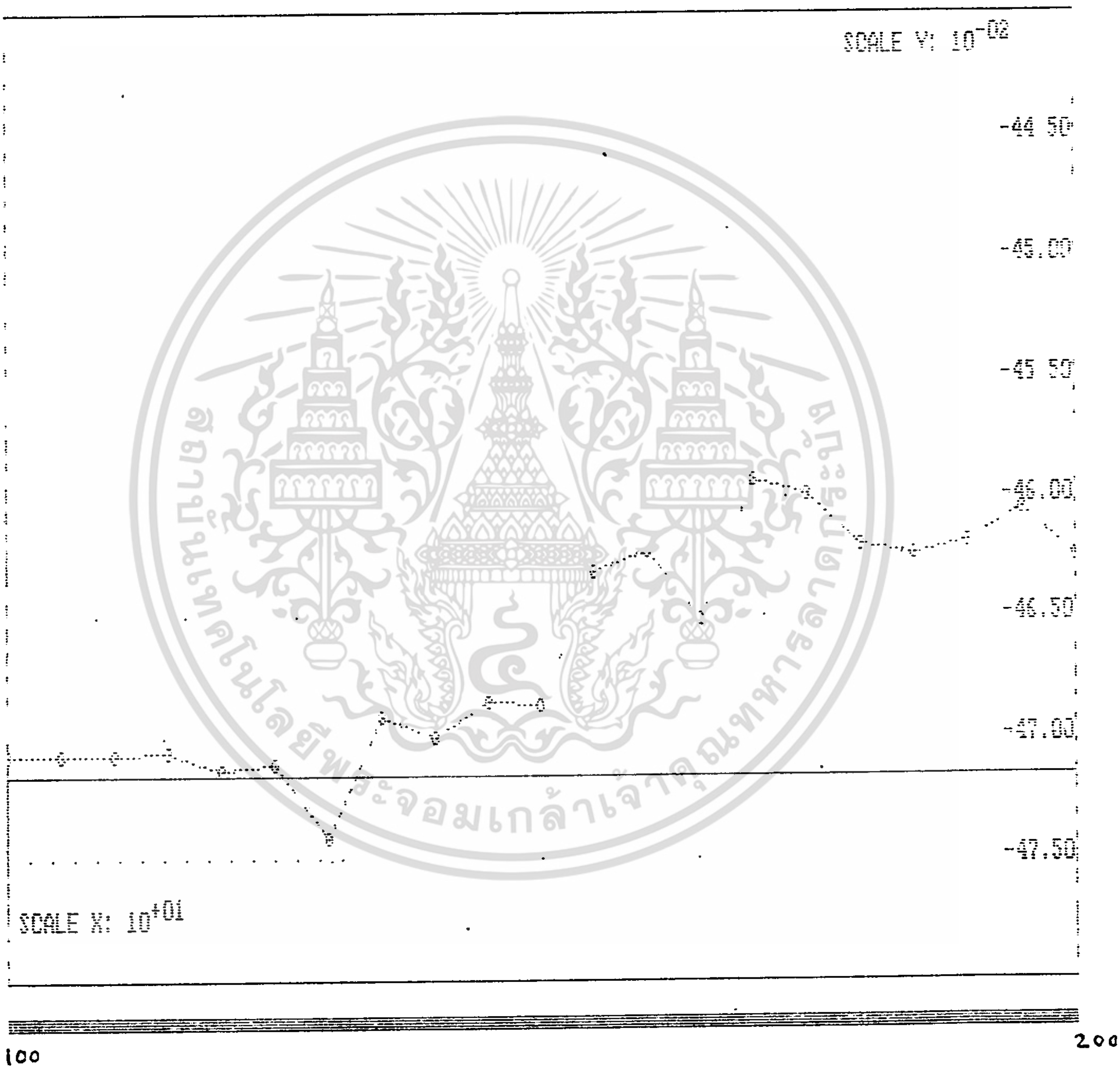
รูป 4.13 แสดงผลการควบคุมแบบเรกกูเลเตอร์ โดยการป้อนสัญญาณรบกวน ค่าเฉลี่ย 1 ที่ [illegible]

```
loop1=36
a1=-4.51771925E-01theta[1]=-4.5177191696E-01
a2= 1.78619042E-01theta[2]= 1.7861903787E-01
b1= 4.68584316E-01theta[3]= 4.6858431516E-01
b2= 2.58262801E-01theta[4]= 2.5826280578E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 1.7E-21 0=      01
s0=-3.5184799694E-09
s1= 4.1866028353E-09
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-6.0533513551E-09

d1= 1.2E-15d2= 2.0E+19
c1=-4.5E-01c2= 1.8E-01
Uc= 1.9764598897E+00U= 1.7727532751E+00Y= 1.5351459678E+00
PHI[1]=-1.5351459678E+00
PHI[2]=-1.5045768084E+00
PHI[3]= 1.7727532751E+00
PHI[4]= 1.2872942142E+00


loop1=100
a1=-4.71512144E-01theta[1]=-4.7071882742E-01
a2= 1.40717633E-01theta[2]= 1.4391250316E-01
b1= 4.43226326E-01theta[3]= 4.4534097373E-01
b2= 2.25979163E-01theta[4]= 2.2801409365E-01
NOT CANCEL B degA0=0 A0 =1 T=Bm
am3= 4.7E-01 x3= 4.7E-01
am2=-4.5E-01c1=-4.5E-01
am1= 1.8E-01c2= 1.8E-01
am0= 3.6E-15 0=      01
s0= 8.6974030733E-02
s1= 1.2488180274E-02
t0= 0.0000000000E+00
t1= 1.0000000000E+00
r0= 1.0000000000E+00
r1=-1.9786196780E-02

d1= 1.2E-15d2= 2.0E+19
c1=-4.5E-01c2= 1.8E-01
Uc= 1.7396899601E+00U= 1.5135017644E+00Y= 1.1552947152E+00
PHI[1]=-1.3215447674E+00
PHI[2]=-1.4043022212E+00
PHI[3]= 1.0472434373E+00
PHI[4]= 1.1753939096E+00
Press any key to return to Turbo Pascal
```

รูป 4.14 แสดงข้อมูลที่ได้จากการจำลองสถานการณ์ค่าเฉลี่ยเข้าไปสิ้น



รูป 4.15 แสดง ผลลู่เข้าของ Â1 เมื่อสัญญาณกระตุ้นตัวประมาณค่าด้วยสัญญาณรบกวนที่มีค่าเฉลี่ย 1


SCALE Y: 10^-02
-44.50
-45.00
-45.50
-46.00
-46.50
-47.00
-47.50
SCALE X: 10^+01
100
200

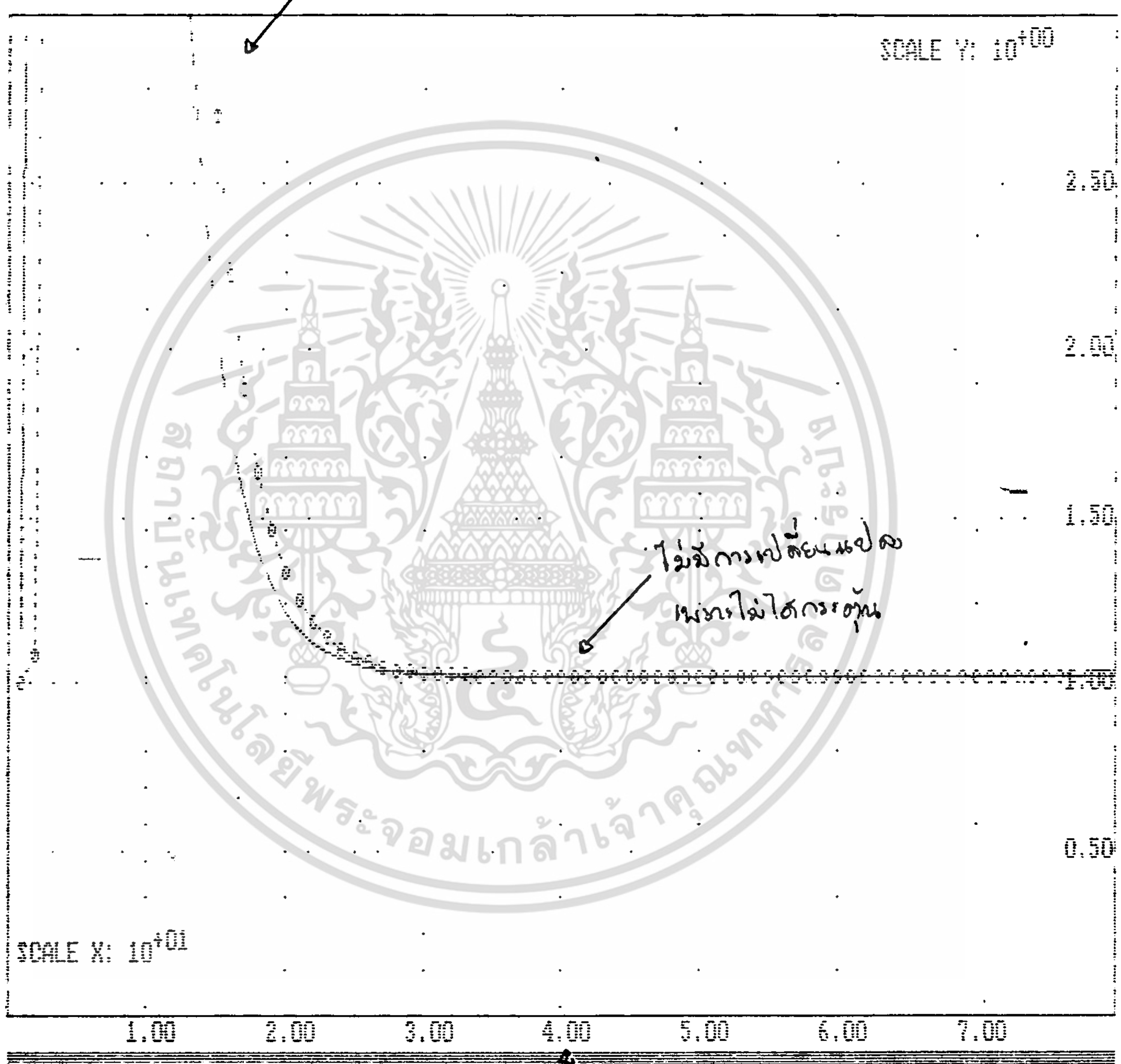
```
                                                Kp.wo^2
OPEN LOOP PROCESS        G(S)=  ----------------
                                s^2+2wi.xi.s+wo^2
PROCESS PARAMETER
                    Kp =1
       xi(0<xi<1)  =0.65
       wo(sec)      =5

                                                 d1q^1+d2q^2
CLOSED LOOP SYSTEM       Hm(q^1) = ------------------
                                                 1 + c1q^1 + c2q^2
SYSTEM PARAMETER
      xis(0<xis<1) =0.65
      wos(sec)       =5
Sampling time       h =0.265
Observer pole     A0 = z1q^1 + z2q^2
                    z1 =0
                    z2 =1
forgeting factor LAMDA=0.9
```

รูป 4.16 แสดงการป้อนค่า สำหรับวิธี ไดโอแฟนไทน์

```
a1=-4.5177192467E-01theta[1]=-4.5177191278E-01
a2= 1.7861904166E-01theta[2]= 1.7861903727E-01
b1= 4.6858431563E-01theta[3]= 4.6858431519E-01
b2= 2.5826280136E-01theta[4]= 2.5826280956E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]=-3.0E-02
R[0]=-5.2E-01T[0]= 1.0E+00S[0]= 2.0E-02


loop1=30
d1= 4.7E-01d2= 2.6E-01
c1=-4.5E-01c2= 1.8E-01


Uc[4]= 1.0000000000E+00 U[4]= 1.0260211891E+00 Y[4]= 1.0301953055E+00


a1=-4.7151214436E-01theta[1]=-4.5177189127E-01
a2= 1.4071763291E-01theta[2]= 1.7861902052E-01
b1= 4.4322632562E-01theta[3]= 4.6858432401E-01
b2= 2.2597916292E-01theta[4]= 2.5826281968E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]=-3.0E-02
R[0]=-5.2E-01T[0]= 1.0E+00S[0]= 2.0E-02


loop1=100
d1= 4.7E-01d2= 2.6E-01
c1=-4.5E-01c2= 1.8E-01


Uc[4]= 1.0000000000E+00 U[4]= 1.0192631668E+00 Y[4]= 1.0192631668E+00

Press any key to return to Turbo Pascal
```

รูป 4.17 เมื่อไม่มีการกระตุ้น ตัวประมาณค่า

ราบเรียบ เพราะมีอินทิเกรเตอร์



เปลี่ยน XI

$\bar{\theta}$ η

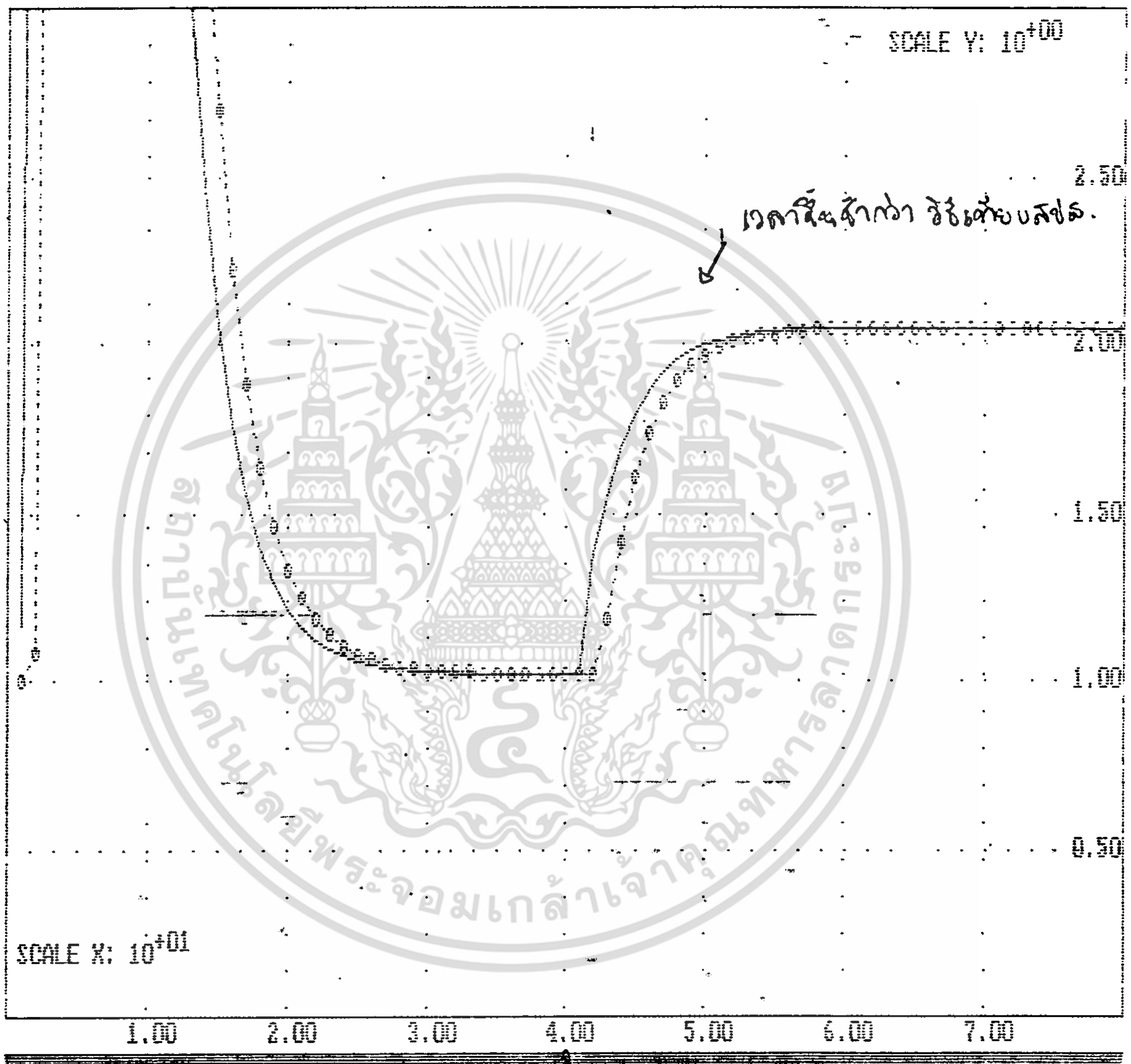
รูป 4.18 สัญญาณที่ได้ และ ω เปลี่ยนแปลง เมื่อไม่มีการกระตุ้น
ตัวประมาณค่า

```
a1=-4.5177192467E-01theta[1]=-4.5177191278E-01
a2= 1.7861904166E-01theta[2]= 1.7861903727E-01
b1= 4.6858431563E-01theta[3]= 4.6858431519E-01
b2= 2.5826280136E-01theta[4]= 2.5826280956E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]=-3.0E-02
R[0]=-5.2E-01T[0]= 1.0E+00S[0]= 2.0E-02

loop1=30
d1= 4.7E-01d2= 2.6E-01
c1=-4.5E-01c2= 1.8E-01


Uc[4]= 1.0000000000E+00 U[4]= 1.0260211891E+00 Y[4]= 1.0301953055E+00
```

ไม่เปลี่ยนตาม θ จริง

```
a1=-4.7151214436E-01theta[1]=-4.5162859994E-01
a2= 1.4071763291E-01theta[2]= 1.7797689164E-01
b1= 4.4322632562E-01theta[3]= 4.6811723102E-01
b2= 2.2597916292E-01theta[4]= 2.5821599595E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]=-2.0E-02
R[0]=-5.3E-01T[0]= 1.0E+00S[0]= 1.0E-02

loop1=100
d1= 4.7E-01d2= 2.6E-01
c1=-4.5E-01c2= 1.8E-01


Uc[4]= 2.0000000000E+00 U[4]= 2.0481282458E+00 Y[4]= 2.0481282434E+00
Press any key to return to Turbo Pascal
```

รูป 4.17 แสดงผล เมื่อ Uc=2



เปลี่ยน XS และ nc=2

— u
+ y

รูป 4.20 สัญญาณ และการเปลี่ยนแปลง เมื่อ nc=2

```
a1=-4.5177192467E-01theta[1]=-4.5177191278E-01
a2= 1.7861904166E-01theta[2]= 1.7861903727E-01
b1= 4.6858431563E-01theta[3]= 4.6858431519E-01
b2= 2.5826280136E-01theta[4]= 2.5826280956E-01

------------------------NOTCANCELB------------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]=-3.0E-02
R[0]=-5.2E-01T[0]= 1.0E+00S[0]= 2.0E-02

loop1=30
d1= 4.7E-01d2= 2.6E-01
c1=-4.5E-01c2= 1.8E-01

Uc[4]= 1.0000000000E+00 U[4]= 1.0260211891E+00 Y[4]= 1.0301953055E+00

a1=-4.7151214436E-01theta[1]=-4.5377662292E-01
a2= 1.4071763291E-01theta[2]= 1.7151772382E-01
b1= 4.4322632562E-01theta[3]= 4.6331000922E-01
b2= 2.2597916292E-01theta[4]= 2.5442283706E-01

------------------------NOTCANCELB------------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.8E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]= 2.0E-02
R[0]=-5.0E-01T[0]= 1.0E+00S[0]= 1.0E-02

loop1=100
d1= 4.6E-01d2= 2.5E-01
c1=-4.5E-01c2= 1.8E-01

Uc[4]= 5.0000000000E+00 U[4]= 4.6937087622E+00 Y[4]= 4.6937091266E+00
Press any key to return to Turbo Pascal
```

รูป 4.21 แสดงผลเมื่อ Uc = 5

```
a1=-4.5177192467E-01theta[1]=-4.5177191278E-01
a2= 1.7861904166E-01theta[2]= 1.7861903727E-01
b1= 4.6858431563E-01theta[3]= 4.6858431519E-01
b2= 2.5826280136E-01theta[4]= 2.5826280956E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]=-3.0E-02
R[0]=-5.2E-01T[0]= 1.0E+00S[0]= 2.0E-02

loop1=30
d1= 4.7E-01d2= 2.6E-01
c1=-4.5E-01c2= 1.8E-01

Uc[4]= 1.0000000000E+00 U[4]= 1.0260211891E+00 Y[4]= 1.0301953055E+00

a1=-4.7151214436E-01theta[1]=-4.7011119927E-01
a2= 1.4071763291E-01theta[2]= 1.6421324533E-01
b1= 4.4322632562E-01theta[3]= 4.5723408291E-01
b2= 2.2597916292E-01theta[4]= 2.3689082957E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 3.0E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.3E+00T[1]= 0.0E+00S[1]=-7.0E-02
R[0]=-5.0E-01T[0]= 1.0E+00S[0]= 1.1E-01

loop1=100
d1= 4.6E-01d2= 2.4E-01
c1=-4.5E-01c2= 1.8E-01

Uc[4]= 1.0000000000E+01 U[4]= 8.3444259049E+00 Y[4]= 8.3444893751E+00
Press any key to return to Turbo Pascal
```

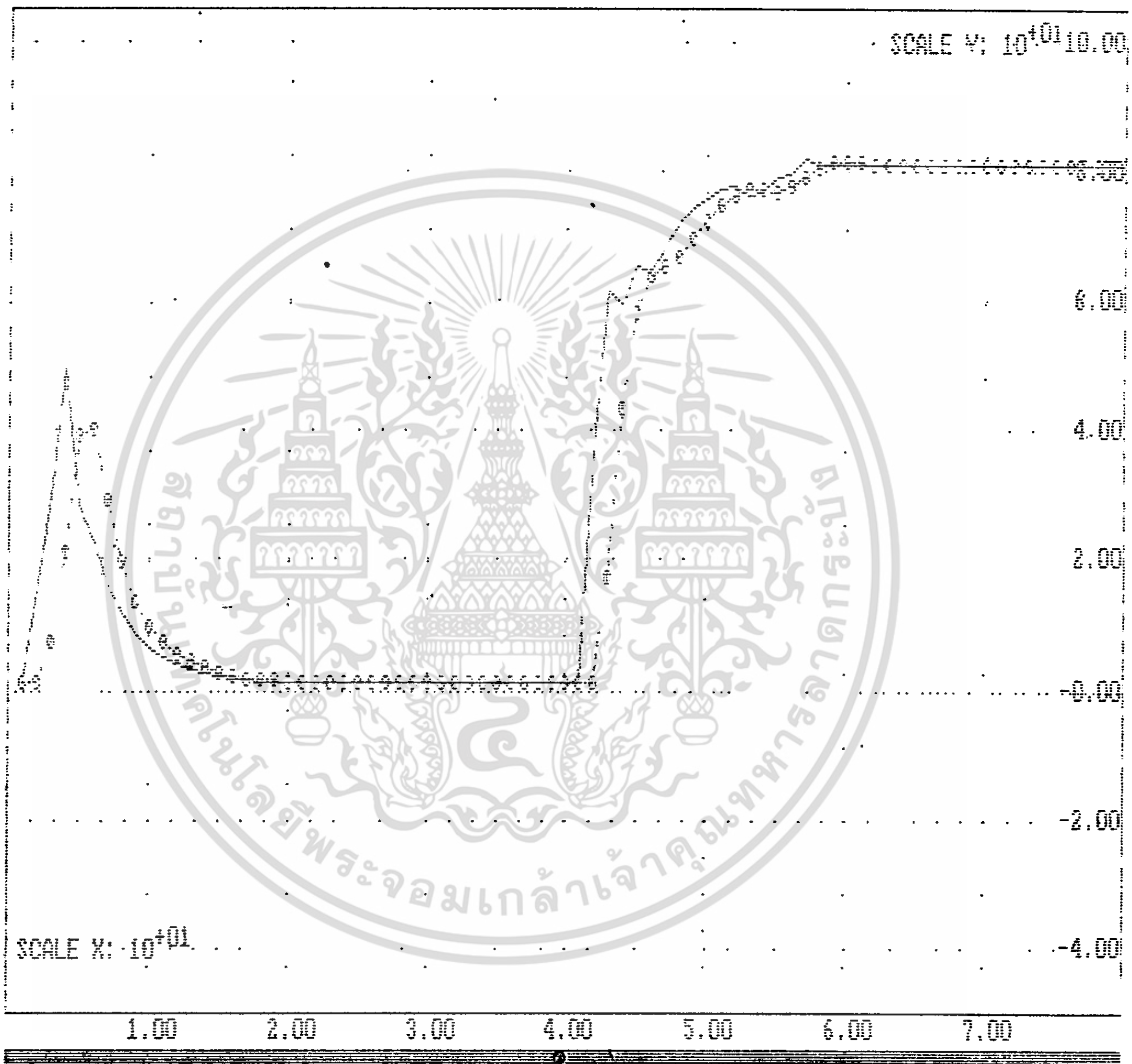
รูป 4.22 แสดงผลเมื่อ Uc = 10

```
=-4.5177192467E-01theta[1]=-4.5177191278E-01
= 1.7861904166E-01theta[2]= 1.7861903727E-01
= 4.6858431563E-01theta[3]= 4.6858431519E-01
= 2.5826280136E-01theta[4]= 2.5826280956E-01

---------------------NOTCANCELB-----------------
7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
1]=-1.2E+00T[1]= 0.0E+00S[1]=-3.0E-02
0]=-5.2E-01T[0]= 1.0E+00S[0]= 2.0E-02

op1=30
= 4.7E-01d2= 2.6E-01
=-4.5E-01c2= 1.8E-01

[4]= 1.0000000000E+00 U[4]= 1.0260211891E+00 Y[4]= 1.0301953055E+00
```

ว ค่าใกล้ θ

```
=-4.7151214436E-01theta[1]=-4.6728015835E-01
= 1.4071763291E-01theta[2]= 1.3888720285E-01
= 4.4322632562E-01theta[3]= 4.4562886620E-01
= 2.2597916292E-01theta[4]= 2.2597818646E-01

----------------------NOTCANCELB----------------
7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
2]= 3.2E+00T[2]= 0.0E+00S[2]= 0.0E+00
1]=-1.4E+00T[1]= 0.0E+00S[1]=-1.8E-01
0]=-5.2E-01T[0]= 1.0E+00S[0]= 1.8E-01

op1=100
= 4.5E-01d2= 2.3E-01
=-4.5E-01c2= 1.8E-01

[4]= 1.0000000000E+02 U[4]= 8.0793880641E+01 Y[4]= 8.0793880645E+01
ess any key to return to Turbo Pascal
```

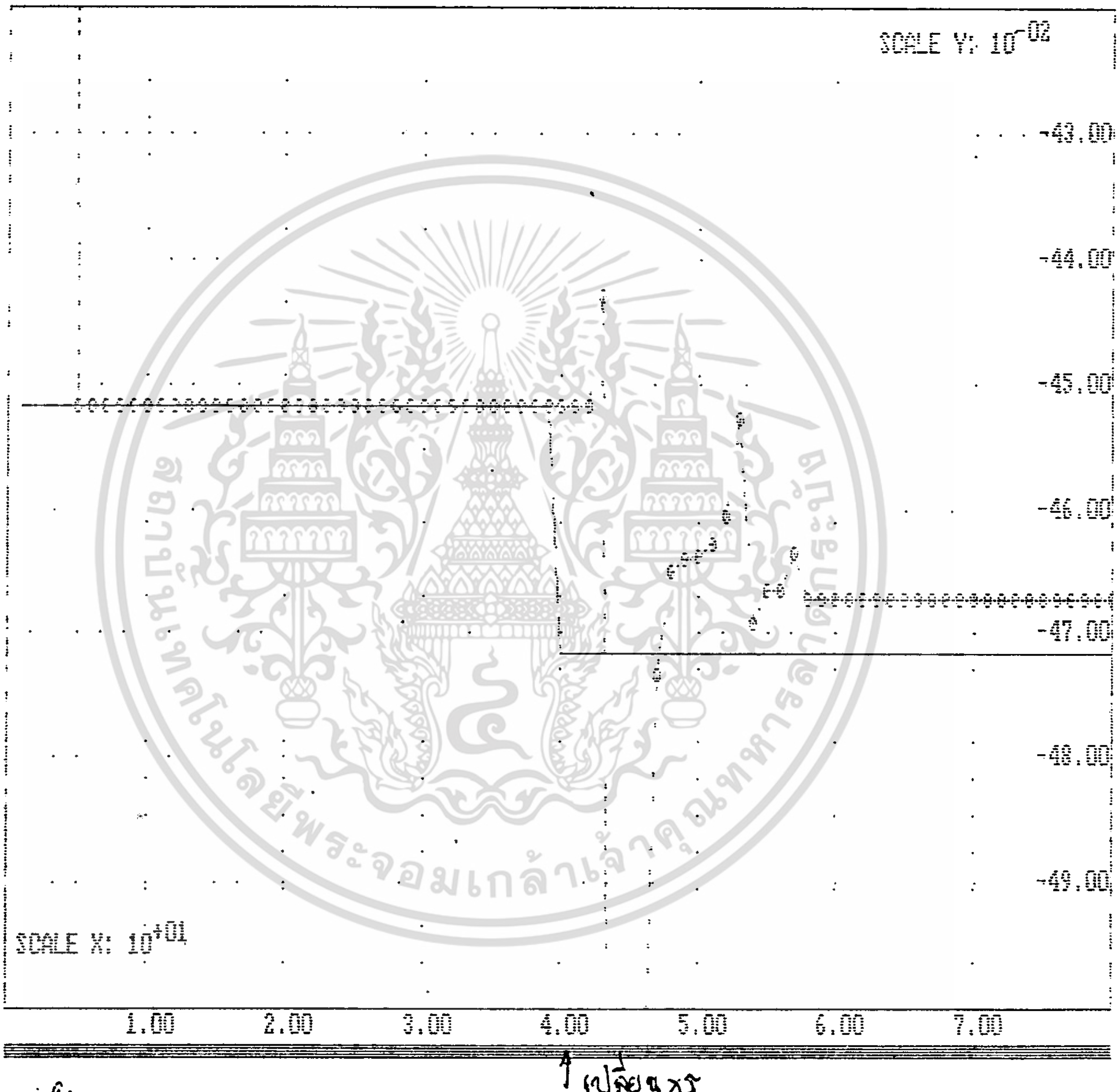
รูป 4.23 แสดง Uc = 100



เปลี่ยน xJ และ Nc=100

— u
+ y

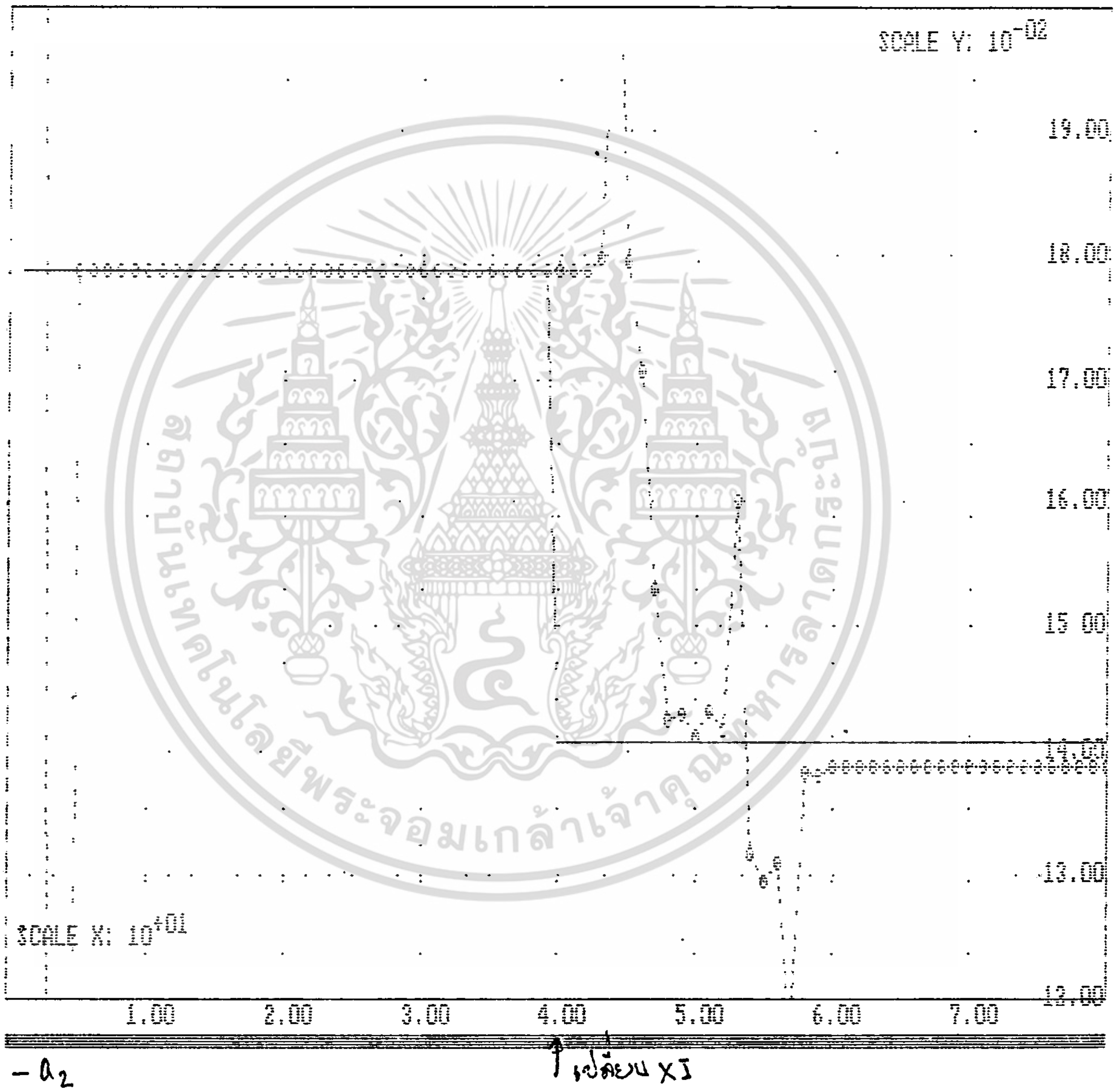
รูป 4.24 สัญญาณ และการเปลี่ยนแปลง เมื่อ MC=100



— a_1

⊕ $\hat{a}_1$

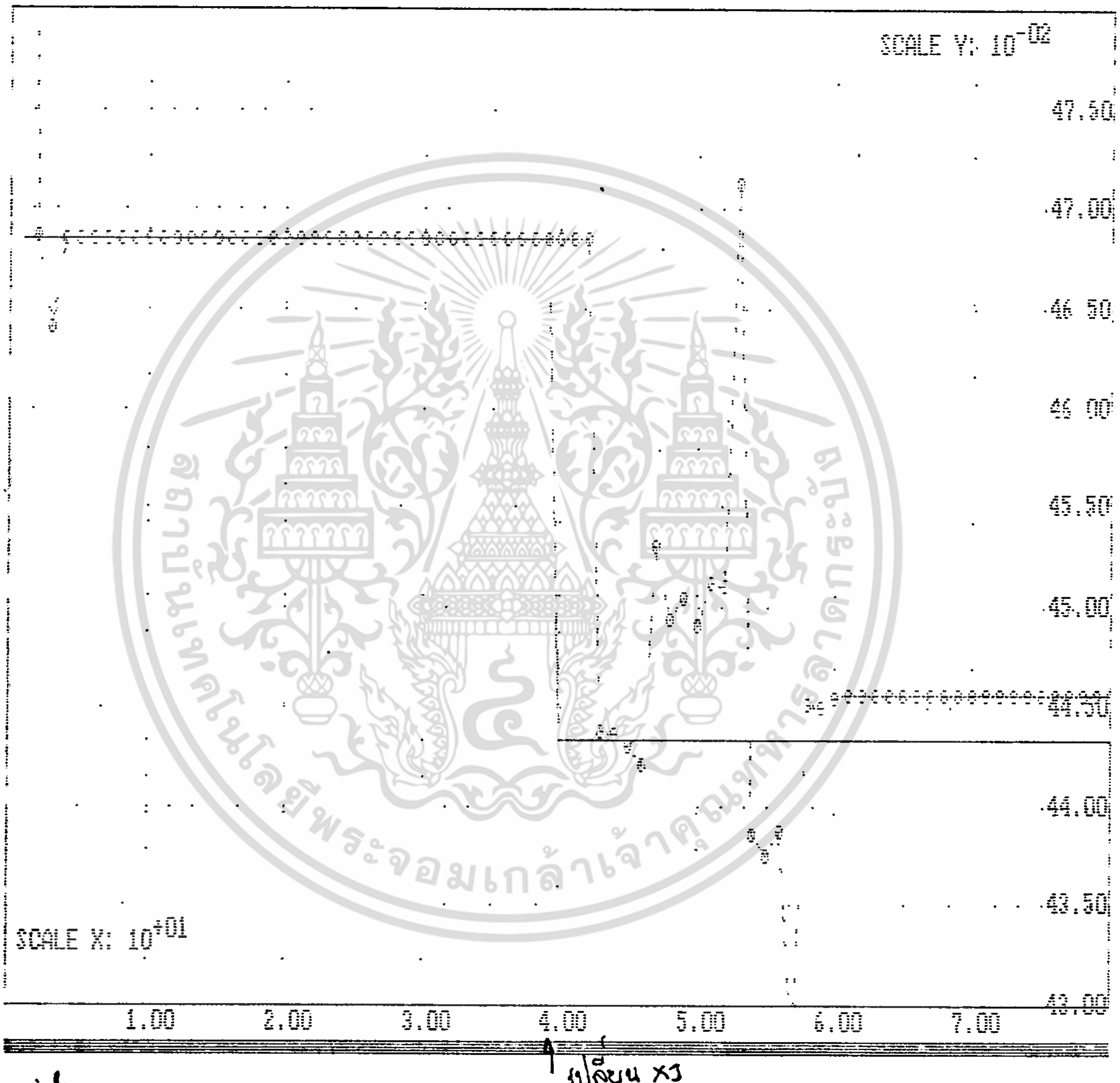
รูป 4.25 แสดงการลู่เข้าของ $\hat{a}_1$ เมื่อ kc=100



— a_2

+ $\hat{a}_2$

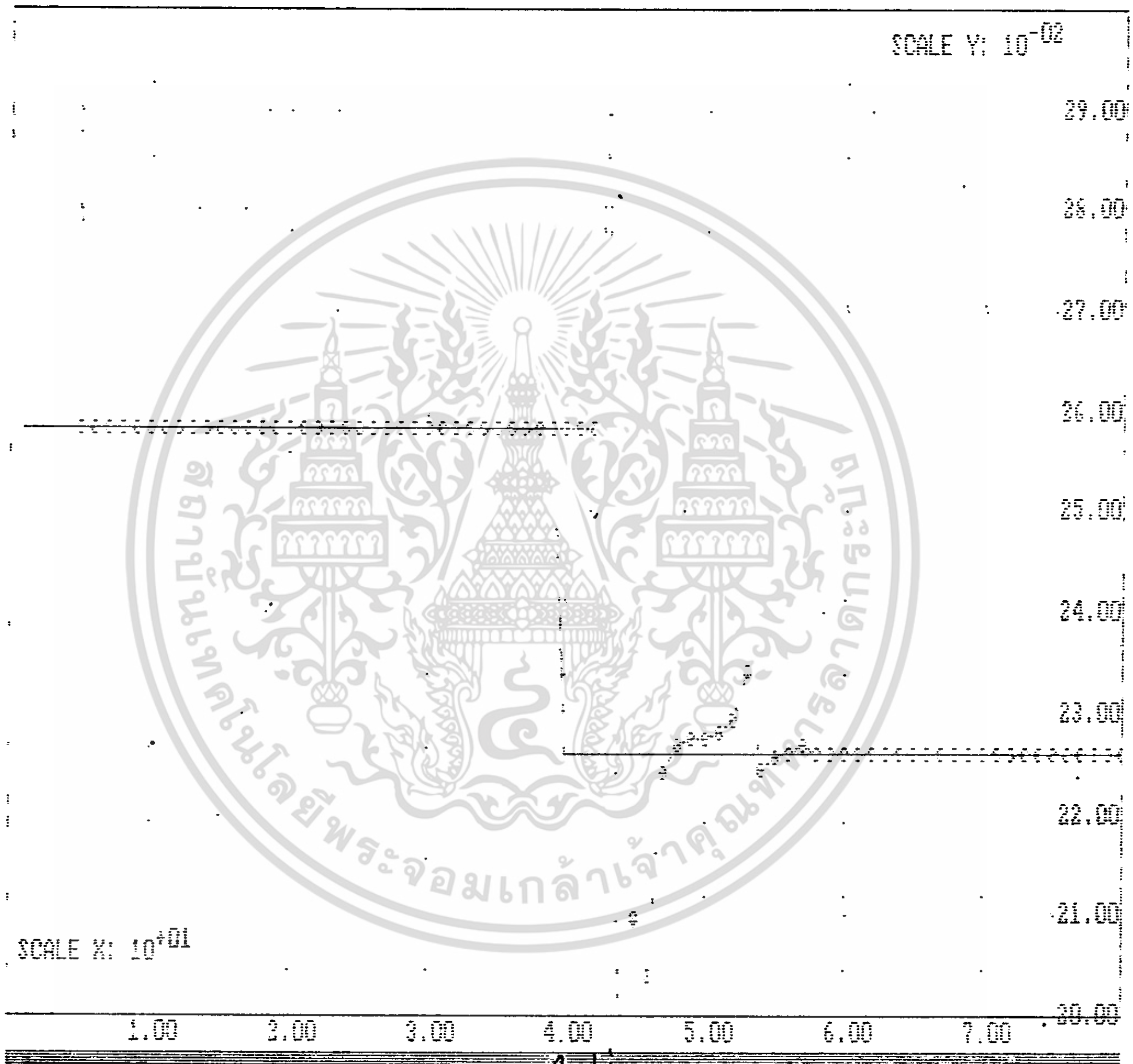
รูป 4.26 แสดงการลู่เข้าของ $\hat{a}_2$ เมื่อ nc = 100



— b_1

ϕ $\hat{b}_1$

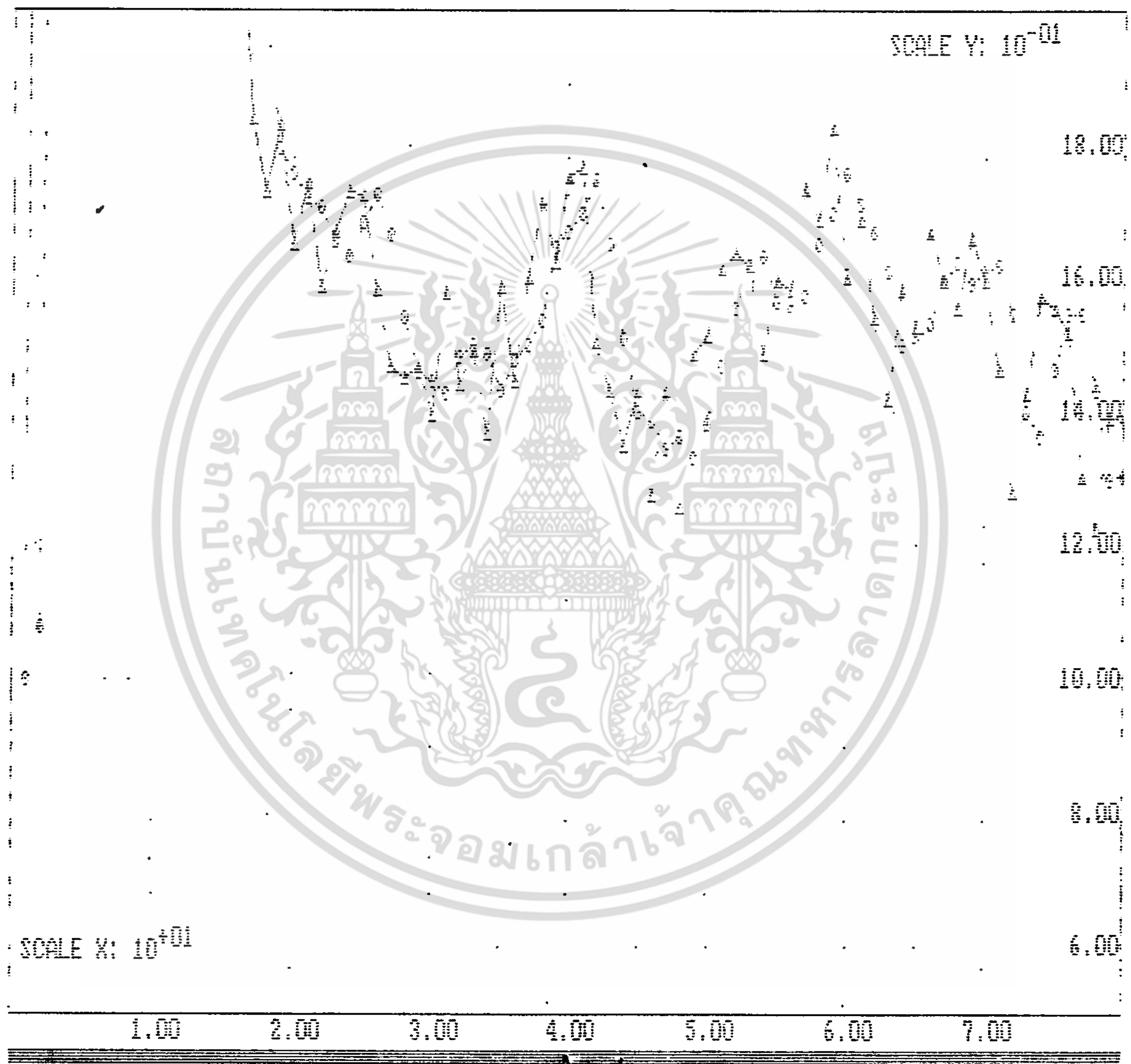
รูป 4.27 แสดงการลู่เข้าของ $\hat{b}_1$ เมื่อ $n_c = 100$



— b_2

θ $\hat{b}_2$

รูป 4.28 แสดงการลู่เข้าของ $\hat{b}_2$ เมื่อ $n_c = 100$



A M

θ y

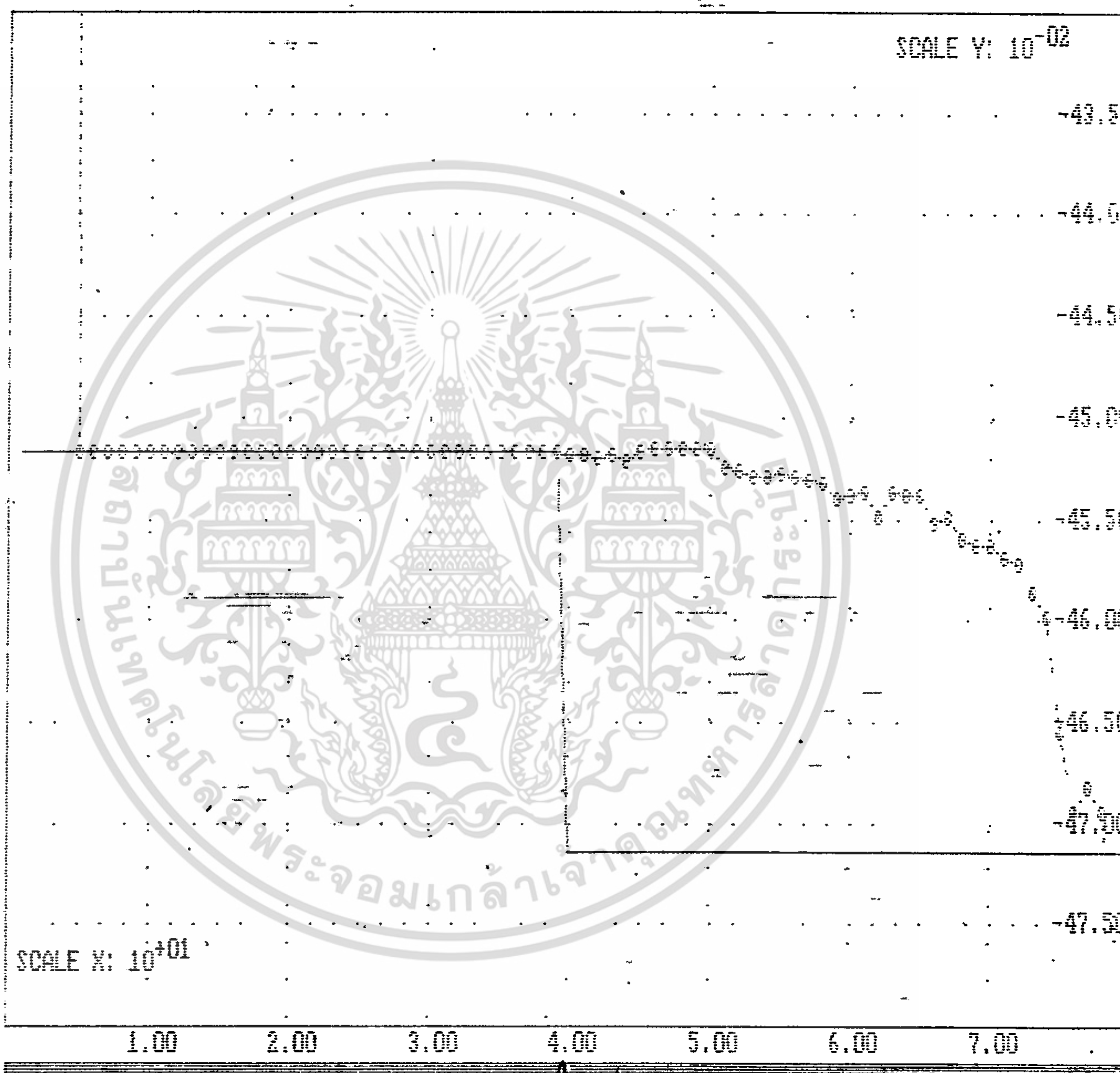
รูป 4.29 แสดง ผลของการซ้อม สัญญาณรบกวน ค่าเฉลี่ย 1 ห้าด้าน

```
a1=-4.5177192467E-01theta[1]=-4.5177191319E-01
a2= 1.7861904166E-01theta[2]= 1.7861903741E-01
b1= 4.6858431563E-01theta[3]= 4.6858431520E-01
b2= 2.5826280136E-01theta[4]= 2.5826280926E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 2.7E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.2E+00T[1]= 0.0E+00S[1]=-3.0E-02
R[0]=-5.2E-01T[0]= 1.0E+00S[0]= 2.0E-02


loop1=30
d1= 4.7E-01d2= 2.6E-01
c1=-4.5E-01c2= 1.8E-01


Uc[4]= 1.2790294834E+00 U[4]= 1.3935803619E+00 Y[4]= 1.4462029389E+00
```

เริ่มสุ่มหา โดยไม่ต้อง
ป้อน NC ถึง 100

```
a1=-4.7151214436E-01theta[1]=-5.0357092065E-01
a2= 1.4071763291E-01theta[2]= 1.7998462424E-01
b1= 4.4322632562E-01theta[3]= 4.6305213209E-01
b2= 2.2597916292E-01theta[4]= 2.1445397153E-01

------------------------NOTCANCELB-----------------
R[7]= 0.0E+00T[7]= 0.0E+00S[7]= 0.0E+00
R[6]= 0.0E+00T[6]= 0.0E+00S[6]= 0.0E+00
R[5]= 0.0E+00T[5]= 0.0E+00S[5]= 0.0E+00
R[4]= 0.0E+00T[4]= 0.0E+00S[4]= 0.0E+00
R[3]= 0.0E+00T[3]= 0.0E+00S[3]= 0.0E+00
R[2]= 3.2E+00T[2]= 0.0E+00S[2]= 0.0E+00
R[1]=-1.4E+00T[1]= 0.0E+00S[1]= 1.3E-01
R[0]=-3.6E-01T[0]= 1.0E+00S[0]= 2.0E-02

loop1=100
d1= 4.6E-01d2= 2.1E-01
c1=-4.5E-01c2= 1.8E-01


Uc[4]= 1.7396899601E+00 U[4]= 8.4925381952E-01 Y[4]= 9.6686275472E-01
Press any key to return to Turbo Pascal
```

รูป 4.30 แสดงข้อมูล เมื่อ ป้อนสัญญาณรบกวนค่า เฉลี่ย ๆ เข้าด้วย



เปลี่ยน x_3

— a_1

⊕ $\hat{a}_1$

รูป 4.31 แสดงผลลู่เข้าของ $\hat{a}_1$ โดยไม่ต้องใช้ผลเฉลี่ย 100

บทที่ 5 สรุป

อาจสรุปได้ว่า ในปัญหาเซอร์โวนั้น จะมีปัญหาที่ การเปลี่ยนแปลงของสัญญาณอ้างอิงเพียงค่าน้อยๆดังที่พบกันทั่วไปนั้น ไม่สามารถกระตุ้นให้ตัวประมาณค่า วัดค่าพารามิเตอร์ได้ถูกต้อง จึงทำให้การควบคุมแบบปรับตัวนี้ ไม่สามารถนำไปใช้แก้ปัญหาเซอร์โวได้จริง

วิธีคำนวณทั้งสองวิธีนี้ ให้ผลใกล้เคียงกัน เพียงแต่ วิธีไดโอแฟนไทน์ใช้เวลาคำนวณช้า และเนื่องจากวิธีไดโอแฟนไทน์นี้ กำหนด R,T,S ด้วยโพลิโนเมียลใดใด t จึงทำให้ยากต่อการเลือก t ให้ได้ R,T,S เหมาะสมกับปัญหาที่กำหนดมา

เนื่องจากการศึกษาการควบคุมแบบปรับตัวนี้ เป็นการจำลองการทำงานโดยไม่มีการติดต่อกับอุปกรณ์ภายนอกจริง ดังนั้นจึงไม่เห็นผลของเวลาในการคำนวณต่อเสถียรภาพของการควบคุม

ในการศึกษาต่อไปนั้น ควรที่จะศึกษาและพัฒนาวิธีเทียบสปส. เพราะง่ายต่อการศึกษา ใช้เวลาในการคำนวณสั้น กำหนดค่าของเร็กกูเลเตอร์ได้ทันที จึงทำให้มีแนวโน้มที่จะนำไปควบคุมอุปกรณ์ภายนอกได้

```
PROGRAM HEAD;
USES CRT;
type
     POL7= array[0..7] of real;  POLPOL=array[1..4,1..4] of real;
     POL4= array[1..4] of real; POL9  =array[0..9] of real;
var
    AA:FILE OF REAL;
    QQQ:REAL;
    POLA,POLB,POLC,POLU,POLV,POLZ,POLG,POLC0,POLL,POLM,LAMDN:POL7;
    DEGA,DEGB,DEGC,DEGU,DEGV,DEGW,DEGZ,DEGX,DEGY,DEGG,DEGC0,DEGL,
          DEGM:INTEGER;
    I,J,N,W,F,E,K,L,MINGK,G,Q,MAXLw,MAXMZ:INTEGER;
    LAM,leafA,leafB,leafC,LEAFG,LEFC0,LEAFA0,LEAFU,LEAFW,LEAFV,
         LEAFZ,LEAFX:REAL;
    LEAFY,LEAFM,LEAFL,SUMGR,stop,ramda:REAL;
    Kp,WO,XI,WOS,XIS,Z1,Z2,Ynew,h:REAL;
    THETA,Ka,PHI,Y,Uc,U:pol4;
    POLAR,POLBS,POLARBS,POLBmc,POLAmc:pol9;
    p:polpol;
    b1,b2,a1,a2,c1,c2,d1,d2,r0,r1,s0,s1,t0,t1,nu,de,am3,am2,am1,
                                              am0,x3,x2,x1,x0:real;
    POLW,POLK,POLAm,POLA0,POLXX,POLBm,POLR,POLT,POLS:POL7;
    DEGA0,DEGAm,DEGXX,loop1,loop2,MINBK,DEGBm,COU,COY,COUc,
           DEGR,DEGS,DEGT,DEGAR,DEGBS,DEGBmc,DEGAmc:INTEGER;
    LEAFAm,LEAFBm,LEAFXX,SUMBR,LEAFBmc,LEAFAmc,LEAFAR,LEAFBS,
            LEAFT, LEAFS,LEAFR:REAL;
{---------------------------------------------------------------}
PROCEDURE FIND( POLX:POL7;VAR DEGX:Integer;VAR leafX:REAL);
{Input is pol. Output are degree and leading coeficient of pol,
  leaving the value of pol unchange}
var
   W:INTEGER;
begin
```

```
      W:=7;
      WHILE (POLX[W] = 0) AND (W>0) DO
           W:=W-1;
       degx:=w;
      leafX:=POLX[W];

end;
{----------------------------------------------------------------}
PROCEDURE LAMDA(VAR POLX:POL7;POLY:POL7;LAM:REAL;VAR M:INTEGER);
{Input is polx,the greater degree pol,and poly,the lower degree pol.
 This procedure will operate 'POLX = POLX-LAM*POLY'in matrix form.
 output is pol LAMDN ,leaving POLX,POLYand LAM unchange.}
var
   E:INTEGER;LAMDN:POL7;
begin
     FIND(POLX,DEGX,LEAFX);
     FIND(POLY,DEGY,LEAFY);
     IF M=0 THEN FOR E:=7 DOWNTO 0 DO LAMDN[E]:=POLY[E];
     IF M>0 THEN
            begin
                 FOR E:= 7 downto M DO   LAMDN[E]:=POLY[E-M];
                 FOR E:= M-1 downto 0 DO LAMDN[E]:=0;
            end;
     FOR E:=7 downto 0 DO
            POLX[E] :=ROUND(100*(POLX[E]-(lam*LAMDN[E])))/100;
end;
PROCEDURE INDATA(VAR KP,WO,XI,WOS,XIS,H,Z1,Z2:REAL);
BEGIN
     CLRSCR;
     WRITELN('                                    Kp.wo^2');
     WRITELN('OPEN LOOP PROCESS    G(S)=  ----------------');
     WRITELN('                               s^2+2wi.xi.s+wo^2');
```

```
    WRITELN('PROCESS PARAMETER');
    WRITELN('                  Kp =');
    WRITELN('         xi(0<xi<1) =');
    WRITELN('         wo(sec)    =');
    WRITELN;
    WRITELN('                                        d1q^1+d2q^2');
    WRITELN('CLOSED LOOP SYSTEM   Hm(q^1) = ----------------');
    WRITELN('                                      1 + c1q^1 + c2q^2');
    WRITELN('SYSTEM PARAMETER');
    WRITELN('       xis(0<xis<1) =');
    WRITELN('       wos(sec)     =');
    WRITELN('Sampling time     h =');
    WRITELN('Observer pole    A0 = z1q^1 + z2q^2');
    WRITELN('                 z1 =');
    WRITELN('                 z2 =');
    GOTOXY(21,5);READLN(KP);
    GOTOXY(21,6);READLN(XI);
    GOTOXY(21,7);READLN(WO);
    GOTOXY(21,13);READLN(XIS);
    GOTOXY(21,14);READLN(WOS);
    GOTOXY(21,15);READLN(H);
    GOTOXY(21,17);READLN(z1);
    GOTOXY(21,18);READLN(z2);
 END;
PROCEDURE OPENLOOP(WO,XI,KP,H:REAL;VAR B1,B2,A1,A2:REAL);
VAR
   OMEGAD,ALPHA,BETA,GAMMA,S1,S2:REAL;
BEGIN
    OMEGAD:=WO*(SQRT(1-SQR(XI)));
    ALPHA:=EXP(-XI*WO*H);
    BETA:=COS(OMEGAD*H);
    GAMMA:=SIN(OMEGAD*H);
```

```
      S1:=1-ALPHA*(BETA+(XI*WO*GAMMA/OMEGAD));
      S2:=SQR(ALPHA)+ALPHA*((XI*WO*GAMMA/OMEGAD)-BETA);
      A1:=-2*ALPHA*BETA;
      A2:=SQR(ALPHA);
      B1:=KP*S1;
      B2:=KP*S2;
 END;
 PROCEDURE CLOSEDLOOP(WOS,XIS,H:REAL;VAR D1,D2,C1,C2:REAL);
 BEGIN
      C1:=-2*EXP(-XIS*WOS*H)*COS(WOS*H*SQRT(1-SQR(XI)));
      C2:=EXP(-2*XIS*WOS*H);
      D1:=0.9*(1+C1+C2);
      D2:=0.1*(1+C1+C2);
 END;
 PROCEDURE TEETA(VAR THETA:POL4;Ka,phi:pol4;Ynew:real);
 VAR
   i:integer;PHITHETA:REAL;
 begin
   PHITHETA:=0;
   FOR i:=1 TO 4 DO
   begin
     PHITHETA:=PHITHETA + PHI[i]*THETA[i];
   end;
   FOR i:=1 TO 4 DO
     THETA[i]:=THETA[i]+Ka[i]*(Ynew-PHITHETA);
 end;
 PROCEDURE  PP(VAR P:POLPOL;Ka,PHI:POL4;RAMDA:REAL);
  VAR
     KPHI,IKPHI,PEE:POLPOL;i,j,l,m:integer;
 begin
   FOR i:=1 TO 4 DO
   FOR j:=1 TO 4 DO
```

```
   KPHI[i,j]:=Ka[i]*PHI[j];
  FOR i:=1 TO 4 DO
  FOR j:=1 TO 4 DO
   IF i=j THEN IKPHI[i,j]:=1-KPHI[i,j]
          ELSE IKPHI[i,j]:= -KPHI[i,j];
  FOR i:=1 TO 4 DO
  FOR l:=1 TO 4 DO
  begin
   PEE[i,l]:=0;
   FOR m:=1 TO 4 DO
   PEE[i,l]:=PEE[i,l]+IKPHI[i,m]*P[l,m];
  end;
  FOR i:=1 TO 4 DO
  FOR j:=1 TO 4 DO
   P[i,j]:= PEE[i,j]/RAMDA;
end;
PROCEDURE GAIN(VAR Ka,PHI:POL4;P:POLPOL;RAMDA:REAL);
 VAR
    PPHI:POL4;i,j:integer;PHIPPHI:REAL;
begin
   FOR J:=1 TO 4 DO
   begin
    PPHI[j]:=0;
    FOR i:=1 TO 4 DO
     PPHI[j]:=PPHI[j]+P[j,i]*PHI[i];
   end;
   PHIPPHI:=0;
   FOR j:=1 TO 4 DO
   PHIPPHI:=PHIPPHI+PHI[j]*PPHI[j];
   FOR j:=1 TO 4 DO
   Ka[j]:=PPHI[j]/(RAMDA+PHIPPHI);
end;
```

```
PROCEDURE INITIAL(VAR PHI,THETA,Y,Uc,U:POL4;VAR P:POLPOL;
                                                  VAR RAMDA:REAL);
VAR
   i,j:INTEGER;
begin
      Y[4]:=0;Uc[4]:=0;U[4]:=0;
      FOR I:=3 DOWNTO 1 DO
       begin
             Y[I]:=1;
            Uc[I]:=1;
             U[I]:=1;
       end;
      PHI[1]:=-Y[3];PHI[2]:=-Y[2];PHI[3]:=U[3];PHI[4]:=U[2];
      FOR I:=1 TO 4 DO
      FOR J:=1 TO 4 DO
       IF I=J THEN P[I,J]:=10E+10 ELSE P[I,J]:=0;
      gotoxy(1,19);
      WRITE('forgeting factor LAMDA=');READLN(RAMDA);
      FOR I:=1 TO 4 DO THETA[I]:=1;
end;
PROCEDURE DIOPHANTINE (POLA,POLB,POLC:POL7;VAR POLL,POLM,POLW,
                                                        POLZ:POL7);
LABEL OUT;
VAR
   POLU,POLV,POLG,POLC0,POLX,LAMDN:POL7;
   I,J,K,N,G,Q,E,F,DEGA,DEGB,DEGC,DEGC0,DEGU,DEGV,DEGW,DEGZ,DEGA0,
                  DEGAm,DEGG,DEGL,DEGM,MIN,MAXLW,MAXMZ,DEGX:INTEGER;
   SUM,LAM,LEAFA,LEAFB,LEAFG,LEAFC,LEFC0,LEAFU,LEAFV,LEAFW,LEAFZ,
                 LEAFL,LEAFM,LEAFA0,LEAFAm,LEAFX:REAL;
begin
      FOR J:=0 TO 7 DO
      begin
```

```
      polw[j] := 0;
      polv[j] := 0;
      polu[j] := 0;
      polz[j] := 0;
      poll[j] :=0;
      polm[j]:=0;
      lamdn[j]:=0;
end;
polu[0] := 1;
polz[0] := 1;
REPEAT
    FIND(POLA,DEGA,leafA);
    FIND(POLB,DEGB,leafB);
    FIND(POLU,DEGU,leafU);
    FIND(POLW,DEGW,leafW);
    FIND(POLV,DEGV,leafV);
    FIND(POLZ,DEGZ,leafZ);
    IF ((DEGA>DEGB) AND (LEAFB<>0)) OR
      ((DEGA=DEGB)AND(LEAFB>=LEAFA)AND(LEAFB<>0)) THEN
      begin
          LAM := leafA/leafB;
          N := DEGA-DEGB;
          LAMDA(POLA,POLB,LAM,N);
          LAMDA(POLU,POLW,LAM,N);
          LAMDA(POLV,POLZ,LAM,N);
     end
     ELSE
        IF ((DEGB>DEGA) AND (LEAFA<>0)) OR
          ((DEGA=DEGB)AND(LEAFB<LEAFA)AND(LEAFA<>0)) THEN
           begin
                LAM := leafB/leafA;
                N := DEGB-DEGA;
```

```
                    LAMDA(POLB,POLA,LAM,N);
                    LAMDA(POLW,POLU,LAM,N);
                    LAMDA(POLZ,POLV,LAM,N);
                end;
          FIND(POLA,DEGA,LEAFA);
          FIND(POLB,DEGB,LEAFB);
   UNTIL ((DEGA=0) AND (LEAFA=0)) OR ((DEGB=0) AND (LEAFB=0));
         IF (DEGB=0) AND (leafB=0) THEN
         begin
           FOR I:=7 DOWNTO 0 DO
            begin
               POLG[I]:=POLA[I];
            end;
         end;
        IF (DEGA=0) AND (leafA=0) THEN
        begin
           FOR I:= 7 DOWNTO 0 DO
            begin
             POLG[I]:=POLB[I];
           end;
        END;
{MAIN DEVIDING}
      FIND(POLC,DEGC,LEAFC);
      FIND(POLG,DEGG,LEAFG);
      FOR K:=0 TO 7 DO POLC0[K]:=0;
      FOR K:=0 TO 7 DO POLX[K]:=POLC[K];
      REPEAT
          begin
               IF (DEGX=0) AND (LEAFX =0) THEN GOTO OUT
               ELSE
                   FIND(POLX,DEGX,LEAFX);
                   LAM:=LEAFX/LEAFG;
```

```
                N:=DEGX-DEGG;
                LAMDA(POLX,POLG,LAM,N);
                POLCØ[N]:=LAM;
        end
     UNTIL N=Ø;
    OUT:
    FIND(POLU,DEGU,LEAFU);
    FIND(POLV,DEGV,LEAFV);
    FIND(POLCØ,DEGCØ,LEFCØ);
    FOR G:=Ø TO DEGU DO
    begin
      FOR Q:=Ø TO DEGCØ DO
      POLL[G+Q]:=POLL[G+Q]+POLU[G]*POLCØ[Q];
    end;
    FOR G:=Ø TO DEGV DO
    begin
      FOR Q:=Ø TO DEGCØ DO
      POLM[G+Q]:=POLM[G+Q]+POLV[G]*POLCØ[Q];
    end;
    END;
PROCEDURE CANCELB(POLA,POLAm,POLAØ,POLB:POL7;VAR POLBm,POLL,POLM,
                                      POLW, POLZ,POLR,POLT,POLS:POL7);
VAR
  I,J,K,L,DEGBAØ,DEGBmAØ,DEGTTZ,DEGTT,DEGTTW,MAX,MIN,DEGX,DEGL,
    DEGZ, DEGBAm,DEGAL,DEGB_M,DEGAW,DEGBZ,DEGNU,DEGDE:INTEGER;
  POLBAØ,POLBmAØ,POLTT,POLTTZ,POLTTW,POLX,POLBAm,POLAL,POLB_M,
         POLAW, POLWX,POLBZ,POLNU,POLDE:POL7;
  LEAFBAØ,LEAFBmAØ,LEAFTTZ,LEAFTT,LEAFTTW,SUM,LEAFX,LEAFL,LEABAm,
         LEAFAL,FEAFB_M,LEAFAW,LEAFBZ,LEAFNU,LEAFDE:REAL;
begin
    {----------------For CANCELB,C=B*Am*AØ-------------------}
```

```
FOR I:=0 TO 7 DO POLBA0[I]:=0;
FIND(POLB,DEGB,LEAFB);
FIND(POLA0,DEGA0,LEAFA0);
FOR I:=0 TO DEGB DO
FOR J:=0 TO DEGA0 DO
   POLBA0[I+J]:=POLBA0[I+J]+POLB[I]*POLA0[J];
FOR I:=0 TO 7 DO POLC[I]:=0;
FIND(POLBA0,DEGBA0,LEAFBA0);
FIND(POLAm,DEGAm,LEAFAm);
FOR I:=0 TO DEGBA0 DO
FOR J:=0 TO DEGAm DO
   POLC[I+J]:=POLC[I+J]+POLBA0[I]*POLAm[J];
DIOPHANTINE(POLA,POLB,POLC,POLL,POLM,POLW,POLZ);
    {      writeln('-----------------CANCELB-------------');
           FOR I:=7 DOWNTO 0 DO
           begin
             WRITE  ('POLL[',I,']=',POLL[I]:7);
             WRITELN('POLW[',I,']=',POLW[I]:7);
           end;
           FOR I:=7 DOWNTO 0 DO
           begin
             WRITE  ('POLM[',I,']=',POLM[I]:7);
             WRITELN('POLZ[',I,']=',POLZ[I]:7);
          end;readln;}
FOR I:=0 TO 7 DO POLX[I]:=0;
REPEAT
      FIND(POLM,DEGM,LEAFM);
      FIND(POLZ,DEGZ,LEAFZ);
      LAM:=LEAFM/LEAFZ;
      N:=DEGM-DEGZ;
      LAMDA(POLM,POLZ,LAM,N);
      POLX[N]:=POLX[N]+LAM;
```

```
            FIND(POLM,DEGM,LEAFM);
            FIND(POLZ,DEGZ,LEAFZ);
      UNTIL DEGM<DEGZ;
      FOR I:=0 TO 7 DO POLS[I]:=POLM[I];
      FOR I:=0 TO 7 DO POLWX[I]:=0;
      FIND(POLW,DEGW,LEAFW);
      FIND(POLX,DEGX,LEAFX);
      FOR I:=0 TO DEGW DO
      FOR J:=0 TO DEGX DO
          POLWX[I+J]:=POLWX[I+J]+POLW[I]*POLX[J];
      FOR I:=0 TO 7 DO POLR[I]:=POLL[I]+POLWX[I];
{---------------------------T=Bm*A0------------------------------}
       FOR I:=0 TO 5 DO POLT[I]:=0;
       FIND(POLBm,DEGBm,LEAFBm);
       FIND(POLA0,DEGA0,LEAFA0);
       FOR I:=0 TO DEGBm DO
       FOR J:=0 TO DEGA0 DO
       POLT[I+J]:=POLT[I+J]+POLBm[I]*POLA0[J];
      WRITELN('--------------------------CANCELB------------------');
      FOR I:=5 DOWNTO 0 DO
      WRITELN('R[',I,']=',POLR[I]:5,'T[',I,']=',POLT[I]:5,'S[',I,']=',
                           POLS[I]:5);
      READLN;
end;
PROCEDURE NOTCANCELB(POLA,POLAm,POLA0,POLB:POL7;VAR POLBm,POLL,
                         POLM,POLW,POLZ, POLR,POLT,POLS:POL7);
VAR
     i,j,k,DEGK,DEGBmA0,DEGBm,DEGA0,MIN,DEGX,DEGAL,DEGB_M,DEGAW,
              DEGBZ,DEGNU,DEGDE,DEGTT,DEGTTW,DEGTTZ,MAX:integer;
     LEAFK,LEAFBmA0,LEAFBm,LEAFA0,SUM,LEAFX,LEAFAL,LEAFB_M,LEAFAW,
            LEAFBZ,LEAFNU,LEAFDE,LEAFTT,LEAFTTW,LEAFTTZ:REAL;
     POLK,POLBmA0,POLX,POLAL,POLB_M,POLAW,POLBZ,POLNU,POLDE,POLTT,
```

```
            POLTTW,POLTTZ,POLWX:POL7;
begin
      FOR I:=0 TO 7 DO POLBm[I]:=POLB[I];
      FOR I:=0 TO 7 DO POLC[I]:=0;
      FIND(POLA0,DEGA0,LEAFA0);
      FIND(POLAm,DEGAm,LEAFAm);
      FOR I:=0 TO DEGA0 DO
      FOR J:=0 TO DEGAm DO
         POLC[I+J]:=POLC[I+J]+POLA0[I]*POLAm[J];
      DIOPHANTINE(POLA,POLB,POLC,POLL,POLM,POLW,POLZ);
{      FOR I:=7 DOWNTO 0 DO
      begin
          WRITE  ('POLL[',I,']=',POLL[I]:7);
          WRITELN('POLW[',I,']=',POLW[I]:7);
      end;
      FOR I:=7 DOWNTO 0 DO
      begin
          WRITE  ('POLM[',I,']=',POLM[I]:7);
          WRITELN('POLZ[',I,']=',POLZ[I]:7);
      end;readln;}
      FOR I:=0 TO 7 DO POLX[I]:=0;
      REPEAT
            FIND(POLM,DEGM,LEAFM);
            FIND(POLZ,DEGZ,LEAFZ);
            LAM:=LEAFM/LEAFZ;
            N:=DEGM-DEGZ;
            LAMDA(POLM,POLZ,LAM,N);
POLX[N]:=POLX[N]+LAM;
            FIND(POLM,DEGM,LEAFM);
            FIND(POLZ,DEGZ,LEAFZ);
     UNTIL DEGM<DEGZ;
     FOR I:=0 TO 7 DO POLS[I]:=POLM[I];
```

```
     FOR I:=Ø TO 7 DO POLWX[I]:=Ø;
     FIND(POLW,DEGW,LEAFW);
     FIND(POLX,DEGX,LEAFX);
     FOR I:=Ø TO DEGW DO
     FOR J:=Ø TO DEGX DO
         POLWX[I+J]:=POLWX[I+J]+POLW[I]*POLX[J];
     FOR I:=Ø TO 7 DO POLR[I]:=POLL[I]+POLWX[I];
      {---------------------T=AØ-------------------------------}
      FOR I:=Ø TO 7 DO POLT[I] :=POLAØ[I];
     WRITELN('-----------------------NØTCANCELB-----------------');
     FOR I:=7 DOWNTO Ø DO
      WRITELN('R[',I,']=',POLR[I]:5,'T[',I,']=',POLT[I]:7,'S[',I,']=',
                         POLS[I]:5);
     {READLN;}
 end;
 {------------MAIN----------------}
 BEGIN
    ASSIGN(AA,'REALDATA');
    REWRITE(AA);
    {WRITELN(' the type of open and closed loop pole in POLE-ZERO form');
    writeln('OPEN-LOOP POLE    1 + a1 q^1 +a2 q^2');
    write  ('a1=');readln(a1);
    write  ('a2=');readln(a2);
    writeln('OPEN-LOOP ZERO          b1 q^1 + b2 q^2');
    write  ('b1=');readln(b1);
    write  ('b2=');readln(b2);
    writeln('CLOSED-LOOP POLE 1 + c1 q^1 + c2 q^2');
    write  ('c1=');readln(c1);
    write  ('c2=');readln(c2);
    writeln('CLOSED-LOOP ZERO        d1q^1 + d1q^2');
    write  ('d1=');readln(d1);
    write  ('d2=');readln(d2);
```

```
WRITE('SAMPLING TIME      h =');READLN(h);
WRITELN('OBSERVER POL              AØ=[Z1q+Z2]');
WRITE('Z1=');READLN(Z1);
WRITE('Z2=');READLN(Z2);}
INDATA(KP,WO,XI,WOS,XIS,H,Z1,Z2);
OPENLOOP(WO,XI,KP,H,B1,B2,A1,A2);
CLOSEDLOOP(WOS,XIS,H,D1,D2,C1,C2);
INITIAL(PHI,THETA,Y,Uc,U,P,RAMDA);
clrscr;
for i:=1 to 30 do
    begin
         gotoxy(1,1);
         writeln('preestimation ');
         u[4]:=u[4]+i;
         GAIN(KA,PHI,P,RAMDA);
         Y[4]:=A1*PHI[1]+A2*PHI[2]+B1*PHI[3]+B2*PHI[4];
         Ynew:=Y[4];
         TEETA(THETA,KA,PHI,Ynew);
         PP(P,KA,PHI,RAMDA);
         U[1]:=U[2];U[2]:=U[3];U[3]:=U[4];
         Y[1] :=Y[2]; Y[2] :=Y[3]; Y[3] :=Y[4];
         PHI[1]:=-Y[3];PHI[2]:=-Y[2];PHI[3]:=U[3];PHI[4]:=U[2];
         writeln('a1=',a1:7,'theta[1]=',theta[1]:7);
         writeln('a2=',a2:7,'theta[2]=',theta[2]:7);
         writeln('b1=',b1:7,'theta[3]=',theta[3]:7);
         writeln('b2=',b2:7,'theta[4]=',theta[4]:7);
     end;
for i:=1 to 3 do
    begin
         u[i]:=1;
         y[i]:=1;
    end;
```

```
FOR LOOP2:=0 TO 2 DO
 BEGIN
     Uc[4]:=1+loop2;
  FOR loop1 :=1 TO 100 DO
    begin
     if loop1 = 30 then
        begin
             xis:=xis+0.02;
             CLOSEDLOOP(WOS,XIS,H,D1,D2,C1,C2);
      end;
     FOR I:=0 TO 7 DO
     begin
          POLAm[I]:=0;POLA[I]:=0;POLA0[I]:=0;
          POLBm[I]:=0;POLB[I]:=0;
     end;
     POLAm[2]:=1;                   POLA[2]:=1;
     POLAm[1]:=c1;POLBm[1]:=d1;POLA[1]:=a1;POLB[1]:=b1;
     POLAm[0]:=c2;POLBm[0]:=d2;POLA[0]:=a2;POLB[0]:=b2;
     POLA0[1]:=z1;
     POLA0[0]:=z2;
     GAIN(KA,PHI,P,RAMDA);
     Y[4]:=A1*PHI[1]+A2*PHI[2]+B1*PHI[3]+B2*PHI[4];
     Ynew:=Y[4];
     TEETA(THETA,KA,PHI,Ynew);
     PP(P,KA,PHI,RAMDA);
     writeln('a1=',a1:7,'theta[1]=',theta[1]:7);
     writeln('a2=',a2:7,'theta[2]=',theta[2]:7);
     writeln('b1=',b1:7,'theta[3]=',theta[3]:7);
     writeln('b2=',b2:7,'theta[4]=',theta[4]:7);
     IF ABS(THETA[4]/THETA[3])>1
       THEN
```

```
    CANCELB(POLA,POLAm,POLA0,POLB,POLBm,POLL,POLM,POLW,POLZ,
                                                      POLR,POLT,POLS)
  ELSE
      NOTCANCELB(POLA,POLAm,POLA0,POLB,POLBm,POLL,POLM,POLW,
                            POLZ,POLR,POLT,POLS);
 U[4]:=-POLR[1]*U[3]/POLR[2]-POLR[0]*U[2]/POLR[2]
      +POLT[2]*Uc[4]/POLR[2]+POLT[1]*Uc[3]/POLR[2]
      +POLT[0]*Uc[2]/POLR[2]
      -POLS[2]*Y[4]/POLR[2]-POLS[1]*Y[3]/POLR[2]
      -POLS[0]*Y[2]/POLR[2];
 U[1]:=U[2];U[2]:=U[3];U[3]:=U[4];
 Uc[1]:=Uc[2];Uc[2]:=Uc[3];Uc[3]:=Uc[4];
 Y[1] :=Y[2]; Y[2] :=Y[3]; Y[3] :=Y[4];
 PHI[1]:=-Y[3];PHI[2]:=-Y[2];PHI[3]:=U[3];PHI[4]:=U[2];
 GOTOXY(1,1);
 WRITELN('loop2=',loop2);
 WRITELN('loop1=',loop1);
 WRITELN('d1=',polbm[1]:5,'d2=',polbm[0]:5);
 WRITELN('c1=',c1:5,'c2=',c2:5);
 gotoxy(1,30);
 WRITE('Uc[4]=',Uc[4]:7);
 WRITE(' U[4]=',U[4]:7);
 QQQ:=LOOP1;
 WRITELN(' Y[4]=',Y[4]:7);{readln;}
 WRITE(AA,QQQ,U[4],Y[4]);
{ COUc:=ROUND(Uc[4]);
 COU:=ROUND(U[4]);
 COY:=ROUND(Y[4]);
 GOTOXY(loop1,COUc);WRITE('-');
 GOTOXY(loop1,COU);WRITE('.');
 GOTOXY(loop1,COY);WRITE('*');}
end; end;END.
```

ผ.2 สมการไดโอแฟนไทน์

$ax + by = c$ โดยที่ a,b,c เป็นโพลิโนเมียลของ R[d] (1)

ทฤษฎีที่ 1. โพลิโนเมียล a,b ที่อยู่ในเซ็ทของ R[d] จะมีหรม. g และ ครน. l สำหรับคู่โพลิโนเมียล p,q และ r,s ที่มีอินเวอร์สใน R[d] โดย

$$ap + bq = g \quad (2)$$

$$ar + bs = \emptyset \quad (3)$$

$$l = ar = -bs$$

ทฤษฎีที่ 2. สมการ (1) มีคำตอบก็ต่อเมื่อ หรม.ของ a และ b หาร c ได้

พิสูจน์ ให้ $x\emptyset$ และ y_o เป็นคำตอบของ (1) และถ้า หรม.ของ a และ b เป็น g

$a = ga_o$ และ $b = gb_o$

ดังนั้น $g(a_ox_o + b_oy_o) = c$

ถ้ากำหนดให้ หรม.ของ a และ b (c) หาร b ได้ $c = gc_o$

จาก ทฤษฎีที่ 1 $ap + bq = g$

$a(pc_o) + b(qc_o) = c$

ดังนั้น คำตอบของสมการคือ

$x_o = pc_o$ และ $y_o = qc_o$

จาก $ar + bs = \emptyset$ จะได้ $a_or + b_os = \emptyset$

ดังนั้น $a_o = s$ และ $b_o = -r$

ทฤษฎีที่ 3. ให้ x_o ,y_o เป็นคำตอบเฉพาะของ (1) จะได้คำตอบทั่วไปของ (1) เป็น

$x = x_o - b_ot$

$y = y_o + a_ot$ โดย t เป็นโพลิโนเมียลใดใดใน R[d]

พิสูจน์ สมมติ $ax_o + by_o = c$ ดังนั้น

$a(x - x_o) = -b(y - y_o)$

ถ้า a และ b มีอินเวอร์สใน R[d] สมการนี้สามารถย้ายข้างได้

$(x - x_o)/(y - y_o) = -b/a$

เมื่อนำ t คูณทั้ง b และ a แล้วคิดส่วนเท่ากับส้วนและเศษเท่ากับเศษจะได้

$x - x_o = -bt$

$y - y_o = at$

จัดรูปได้

$x = x_o - bt$ และ $y = y_o + at$

จากทฤษฎีที่ 2 ได้

$x = pc_o + rt$ และ $y = qc_o + st$

ผ.3 อัลกอริทึ่มการคำนวณ

หา g,p,q,r,s จาก

$$ap+bq = g$$

$$ar+bs = \emptyset$$

จาก ap+bq = g จะเห็นว่าถ้าตัวใดตัวหนึ่งเป็นศูนย์ อีกตัวจะเท่ากับ g ดังนั้นขั้นตอนในการคำนวณ จะต้องทำ reduce echelon จนโพลิโนเมียลตัวใดตัวหนึ่งเป็นศูนย์

จัดรูปสำหรับการคำนวณ

$$F = [\,a \quad b\,]$$

$$V = \begin{bmatrix} p & r \\ q & s \end{bmatrix}$$

$$FV \begin{bmatrix} g & \emptyset \end{bmatrix}$$

อัลกอริทึ่ม

1. ให้ V = I (unit matrix)
2. ถ้า a หรือ b ตัวใดตัวหนึ่งเป็นศูนย์ ให้ข้ามไปที่ 7 ถ้าไม่ ให้เลือกโพลิโนเมียลตัวที่มีกำลังน้อยกว่ามา
3. เอา สปส. โพลิโนเมียล ตัวกำลังมากตั้ง หารด้วย สปส. โพลิโนเมียล ตัวกำลังน้อยได้ λ
4. เอากำลังของโพลิโนเมียลตัวกำลังมากตั้ง ลบด้วยกำลังของโพลิโนเมียลตัวกำลังน้อย ได้ n
5. เอาโพลิโนเมียลตัวกำลังมากตั้ง ลบด้วย $\lambda\, d^{n}$ * โพลิโนเมียลตัวกำลังน้อย และทำเช่นเดียวกับคอลัมน์ใน V ที่สัมพันธ์กัน เช่นถ้าเอา a - b จะต้องเอา p - r และ q - s
6. กลับไปทำ 2
7. ระหว่าง a และ b ตัวที่ไม่เป็นศูนย์ จะเป็น g และได้ V ตามต้องการ
8. หยุด

ผ.4 ตัวสังเกตสเตท และ ตัวป้อนกลับสเตท(Observer and State Feedback)

วัตถุประสงค์

1) เพื่อให้ทราบว่า เร็กกูเลเตอร์มีที่มาจากทั้ง ทฤษฎีทางปริภูมิสเตทและทฤษฎีทางอินพุต-เอาท์พุต

2) เพื่อให้ทราบว่า ถ้าใช้เร็กกูเลเตอร์ควบคุมปิดลูปเพื่อแก้ปัญหาเซอร์โว จะต้องกำหนดให้มีโพลของตัวสังเกตสเตท Ao ในเร็กกูเลเตอร์

สมการพลวัตของระบบในปริภูมิสเตท

$$x(k+1) = \Phi x(k)+\Gamma u(k) \qquad ผ.4.1$$

$$y(k) = Cx(k) \qquad ผ.4.2$$

เมื่อเพิ่ม L ซึ่งเป็นตัวป้อนกลับสเตท

$$u(k) = -Lx(k) \qquad ผ.4.3$$

เรียกสมการ ผ.4.3 ว่า กฎการควบคุม(Control law) ซึ่งทำให้สมการนี้มีการควบคุม-ปิดลูป และมีสมการคุณลักษณะ

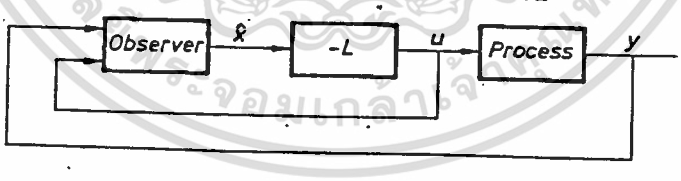
$$\left|zI-[\Phi-\Gamma L]\right| = 0 \qquad ผ.4.4$$

จากการกำหนด z ซึ่งเป็นค่าไอเก็นของระบบปิดลูป จะหา L ได้

แต่ถ้าไม่ทราบสเตทภายใน ก็จะทำการควบคุมปิดลูปไม่ได้ ดังนั้นจึงต้องสร้างตัวสังเกตสเตทขึ้นมา โดยตัวสังเกตสเตทจะต้องทำให้เอาท์พุตที่ได้จากตัวสังเกตสเตทและเอาท์พุตที่ได้จากระบบจริง ใกล้เคียงกัน นำหลักการนี้มาเป็นสมการของตัวสังเกตสเตท

$$\hat{x}(k+1/k) = \Phi\hat{x}(k/k-1)+\Gamma u(k)+K[y(k)-C\hat{x}((k/k-1)] \qquad ผ.4.5$$

โดยที่ K เป็นกำลังขยายของตัวสังเกตสเตท



รูป ผ.4.1

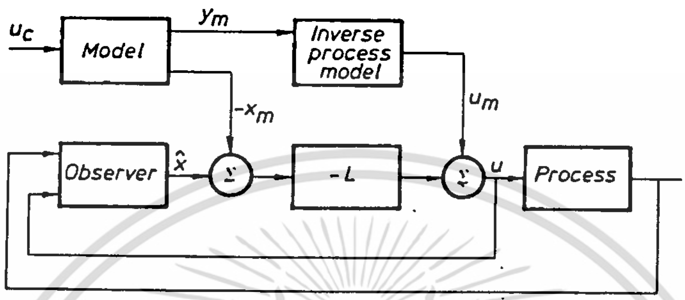
จัดให้อยู่ในรูปของผลต่างของการสังเกตสเตท $\tilde{x}=x-\hat{x}$ จาก ผ.4.1 และผ.4.4 จะได้

$$\tilde{x}(k+1/k) = \Phi x(k/k-1)+K[y(k)-C\hat{x}(k/k-1)]$$

$$= [\Phi-KC]\tilde{x}(k/k-1) \qquad ผ.4.6$$

ดังนั้น จะได้สมการคุณลักษณะ $\left|zI-[\Phi-KC]\right| = 0$ ซึ่งทำให้หากำลังขยายของตัวสังเกตสเตทได้ในแบบเดียวกัน

ต่อไปเมื่อพิจารณาการแก้ปัญหาเซอร์โวของระบบ จะต้องกำหนดสมรรถนะของระบบในรูปของโมเด็ล และควบคุมให้ระบบมีสมรรถนะตามโมเด็ลด้วยการกำหนดโพล-ซีโร่(Pole-Zero Placement) ซึ่งจะได้โครงสร้างของการควบคุมแบบเซอร์โวทางปริภูมิสเตท



รูป ผ.4.2

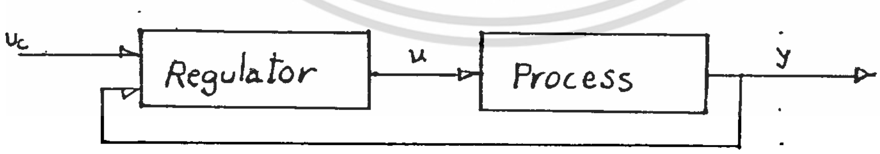
จากรูปได้สมการพลวัติ

$$x_m(k+1) = \Phi x_m(k)+Tu_m(k)$$

$$y_m(k) = C_m x_m(k) \qquad ผ.4.7$$

โมเด็ลจะเป็นตัวกำหนดโพลและซีโร่ของระบบปิดลูป นั่นคือโมเด็ลจะไปบังคับให้เอาท์พุตของระบบเป็นตามโมเด็ล ส่วนตัวป้อนกลับจะลดส่วนเบี่ยงเบนต่างๆของเอาท์พุตที่เบี่ยงเบนไปจากโมเด็ล

เปรียบเทียบรูป ผ.4.2 กับรูปโครงสร้างของเร็กกูเลเตอร์ในทฤษฏีทางอินพุต-เอาท์พุต โดยให้ส่วนที่อยู่ภายในเส้นประเป็นเร็กกูเลเตอร์



รูป ผ.4.3

จะพบว่า ถ้าใช้เร็กกูเลเตอร์ทางอินพุต-เอาท์พุตโมเด็ลไปควบคุมระบบแบบเซอร์โว จะต้องมีส่วนของตัวสังเกตสเตท หรือโพลของตัวสังเกตสเตท Ao อยู่ในเร็กกูเลเตอร์ด้วย

จากรูป ผ.4.2

$$x(k+1) = \Phi x(k)+Tu(k)$$

$$u(k) = L[x_m(k)-\hat{x}(k)]+u_m(k)$$

$$\hat{x}(k+1/k) = \Phi\hat{x}(k/k-1)+Tu(k)+K[y(k)-C\hat{x}(k/k-1)] \qquad ผ.4.8$$

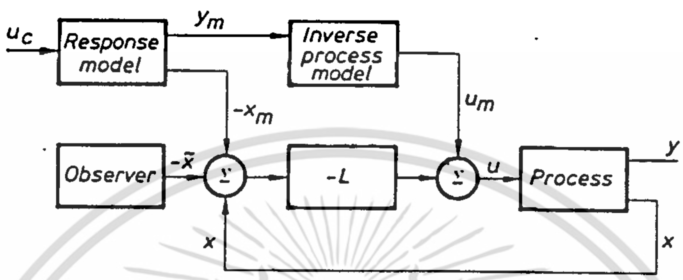
ถ้ากำหนด $\tilde{x}=x-\hat{x}$ และ $\tilde{u}_c(k)=u_m(k)+Lx_m(k)$ จะได้

$$x(k+1) = [\Phi-TL]x(k)+TL\tilde{x}(k/k-1)+T\tilde{u}_c(k)$$

$$\tilde{x}(k+1/k) = [\Phi-KC]\tilde{x}(k/k-1)$$

$$u(k) = -L\hat{x}(k)+\tilde{u}_c(k) \qquad \text{ผ.4.9}$$

และเมื่อเขียนโครงสร้างตามสมการ ผ.4.9 จะได้



รูป ผ.4.4

เมื่อเปรียบเทียบกับเร็กกูเลเตอร์ในทฤษฎีทางอินพุต-เอาท์พุต(รูป ผ.4.3)จะพบว่า โพลของตัวสังเกตสเตทที่อยู่ในเร็กกูเลเตอร์นั้น จะไม่มีผลต่อทรานสเฟอร์ฟังก์ชันระหว่าง y และ u_c

ผ.5 เงื่อนไขกำลังของเร็กกูเลเตอร์

1.ในกรณีที่หักล้าง B^+

$$AR+BS = B^+AmAo$$

โดยทั่วไปแล้ว $\deg S <= \deg R$ (เพราะตัวป้อนกลับจะต้องเป็นอินทิเกรเตอร์ไม่ก็เป็นค่าคงที่) และ $\deg B < \deg A$

จะได้ $\deg AR = \deg (AR+BS) = \deg B^+AmAo$

ดังนั้น $\deg R = \deg B^+ + \deg Ao + \deg Am - \deg A$ ****

จาก $T = BmAo/B^-$

จะได้ $\deg T = \deg Bm + \deg Ao - \deg B^-$ ****

จากทฤษฎีของสมการไดโอแฟนไทน์ สมการ $AX+BY = C$ จะมีคำตอบเมื่อหรม.ของ A และ B หาร C ได้ และถ้า Xo,Yo เป็นคำตอบเฉพาะจะได้คำตอบใดใด

$$X = Xo + QB$$

$$Y = Yo - QA$$

โดยที่ X และ Y ต้องจริงตาม

$$\deg X < \deg B \text{ และ } \deg Y < \deg A$$

ดังนั้นจาก $AR+BS = B^+AmAo$ จะได้ $\deg R < \deg B$ และ $\deg S < \deg A$

ถ้าเลือก $\deg S = \deg A - 1$ เมื่อนำมารวมกับ $\deg R >= \deg S$ จะได้

$$\deg B^+ + \deg Ao + \deg Am - \deg A >= \deg A - 1$$

$$\deg Ao >= 2\deg A - \deg Am - \deg B^+ -1$$ ****

2)กรณีที่ไม่หักล้าง B^+

$$AR + BS = AoAm$$

จาก $\deg S <= \deg R$ และ $\deg B <= \deg A$ จะได้

$$\deg R = \deg (AR + BS) =\deg AoAm$$

ดังนั้น $\deg R = \deg Ao + \deg Am - \deg A$ ****

จาก $T = BmAo/B$ ดังนั้น

$$\deg T = \deg Bm + \deg Ao - \deg B$$ ****

จาก สมการ $AX + BY = C$ จะต้องมี

$$\deg X < \deg B \text{ และ } \deg Y < \deg A$$

ดังนั้นจาก $AR + BS = AoAm$ จะได้ $\deg R < \deg B$ และ $\deg S < \deg A$

ถ้าเลือก $\deg S = \deg A - 1$ เมื่อรวมกับ $\deg R >= \deg S$ จะได้

$$\deg Ao + \deg Am + \deg A >= \deg A - 1$$

$$\deg Ao >= 2\deg A - \deg Am - 1$$ ****

กิตติกรรมประกาศ

ผู้จัดทำทั้งสองคนขอขอบพระคุณ อาจารย์ จงกล งามวิวิทย์ สำหรับการดูแลเอาใจใส่และกำลังใจในการทำงานเป็นอย่างมาก นอกจากนี้ยังขอขอบคุณ ท่านอาจารย์และท่านผู้มีพระคุณท่านอื่นๆ สำหรับข้อคิดเห็นและคำแนะนำ

หนังสืออ้างอิง

1.Prof.K.J.Astrom And B.Wittenmark,Ph.D. :"Self-tuning Controllers Based On Pole-zero Placement"
IEE Proceedings,Vol.127,Pt ,No.3,MAY 1980

2.Prof.K.J.Astrom And B.Wittenmark,Ph.D:"Computer Controlled Systems Theroy And Design"
Prentice-Hall International, 1984

3.Kucera :"Dicrete Linear Control.Prague"
Academia,1979

4.C.J.Harris:"Self-tuning And Adaptive Control"
IEE Series 15 ,1981