บทที่ 2

การควบคุมแบบจูนปรับตัวเอง

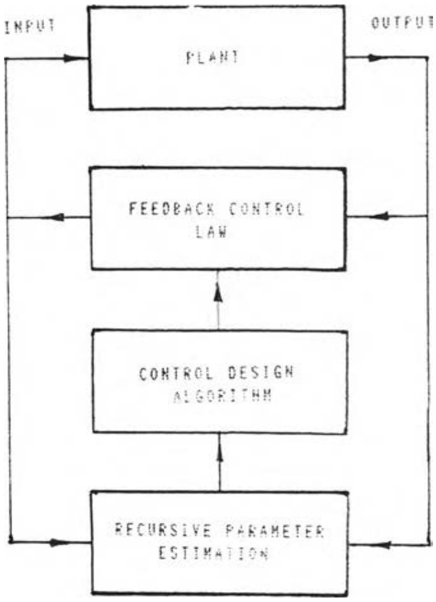
การควบคุมแบบจูนปรับตัวเองเป็นการควบคุมแบบจูนอัตโนมัติ สามารถประยุกต์ใช้ได้กับการควบคุมชนิดต่าง ๆ ได้โดยง่ายและใช้ได้กับระบบหลายอินพุทหลายเอาต์พุต (Multi-input multi-output) ในบทนี้จะกล่าวถึงลักษณะของการควบคุมแบบจูนปรับตัวเองในแบบ Classical control และ Modern control และขั้นตอนการออกแบบที่เป็นพื้นฐานชนิดต่าง ๆ ของการควบคุมแบบจูนปรับตัวเอง

ลักษณะของการควบคุมแบบจูนปรับตัวเอง

ในการควบคุมระบบที่มีความไม่แน่นอน (Uncertainty) เนื่องจากมีการเปลี่ยนแปลงค่าพารามิเตอร์อย่างช้า ๆ ตามเวลาหรือมีสิ่งรบกวนเกิดขึ้น ให้มีสมรรถนะ (Performance) และข้อกำหนด (Specification) เป็นไปตามต้องการนั้น วิธีการควบคุมแบบจูนปรับตัวเองเป็นวิธีหนึ่งที่ใช้ได้ดีสำหรับระบบดังกล่าวและสามารถใช้กับระบบที่มีลักษณะดังต่อไปนี้

1. ระบบที่มีพลวัต (Dynamic) ยุ่งยาก เช่น ระบบที่มีเฟสตามหลังนานมาก ๆ เป็นต้น
2. ระบบมีภาวะไม่เชิงเส้น เช่น อัตราขยายของระบบเปลี่ยนแปลงตามสัญญานเข้าใน pH neutralisation process [1] เป็นต้น
3. ระบบมีพารามิเตอร์เปลี่ยนแปลงตามเวลา
4. มี Interaction ระหว่างลูปในระบบ
5. ระบบที่มีสิ่งรบกวนจากสภาวะแวดล้อม (Environmental disturbance) เข้ามาเกี่ยวข้อง เช่น Noise เป็นต้น

จุดประสงค์ของในการควบคุมแบบจูนปรับตัวเองก็เพื่อต้องการควบคุมผลตอบของระบบที่มีลักษณะข้างต้นให้เป็นไปตามข้อกำหนดที่วางไว้ โดยการปรับค่าของพารามิเตอร์ของตัวควบคุมให้เหมาะสมกับการเปลี่ยนแปลงที่เกิดขึ้นอย่างอัตโนมัติตามกฎเกณฑ์ของการควบคุม (Criteria) ในการควบคุมจะต้องพิจารณาเสถียรภาพ สมรรถนะและการลู่เข้าเป็นหลัก โดยทั่วไปแล้วการควบคุมแบบจูนปรับตัวเองสามารถแบ่งออกเป็น 3 ส่วนใหญ่ ๆ ได้แก่



รูปที่ 2.1 โครงสร้างของการควบคุมแบบจูนปรับตัวเอง

1. การบอกเอกลักษณ์ (Identification) หรือ การประมาณค่าพารามิเตอร์ (Recursive parameter estimation)
2. การสังเคราะห์ตัวควบคุม (Control synthesis) หรือ อัลกอริทึมของการออกแบบการควบคุม (Control-design algorithm)
3. กฎของการควบคุม (Feedback control law)

การควบคุมแบบจูนปรับตัวเองสามารถแบ่งได้เป็น 2 แบบ คือ แบบ Explicit กับแบบ Implicit โดยที่แบบ Explicit จะประมาณ (Estimate) ค่าของพารามิเตอร์ของระบบแล้วแทนใน Control-design algorithm เพื่อหาค่าสัมประสิทธิ์ที่ใช้ในกฎการควบคุม ส่วนแบบ Implicit จะข้ามช่วง Control-design algorithm โดยจัดสมการเสียใหม่แล้วทำการประมาณ (Estimation) ค่าสัมประสิทธิ์ของตัวควบคุมโดยตรงทำให้การคำนวณรวดเร็วขึ้น แต่ว่าความแม่นยำ (Accuracy) และความมั่นคง (Robust) ของแบบ Explicit ดีกว่าของแบบ Implicit ซึ่งเปรียบเทียบได้กับวิธีตรง (Direct) และ วิธีอ้อม (Indirect) ในการควบคุมแบบปรับตัวเอง (Adaptive control)

ในการควบคุมแบบจูนปรับตัวเองจำเป็นต้องอาศัยค่าจากการบอกเอกลักษณ์หรือการประมาณค่าพารามิเตอร์ (Parameter estimation) เป็นหลัก ดังนั้นการเลือกใช้อัลกอริทึมใน

การประมาณ (Estimation algorithm) จะต้องพิจารณารูปแบบของระบบ (System model) และขั้นตอนการออกแบบ (Design procedure) ถ้าใช้อัลกอริทึมที่ขาดความมั่นคง จะทำให้การควบคุมแบบจูนปรับตัวเองล้มเหลวได้ การประมาณค่าพารามิเตอร์ที่ใช้ในอัลกอริทึมของการควบคุมแบบจูนปรับตัวเองเป็นการประมาณแบบเชิงเส้น (Linear estimation) ดังนั้นการนำการควบคุมแบบจูนปรับตัวเองไม่สามารถใช้กับกระบวนการที่มีความไม่เป็นเชิงเส้นยุ่งยากมาก ๆ ได้ โดยทั่วไปการประมาณพารามิเตอร์จะนิยมใช้อยู่ 3 วิธี คือ Least square estimation (LS) [4] , Maximum likelihood estimation (ML) [9] และ Extended least square estimation (ELS) [4] การเลือกใช้จะต้องให้เหมาะสมกับรูปแบบของกระบวนการและความต้องการของผู้ใช้

การควบคุมแบบจูนปรับตัวเองได้รับการพัฒนามานานในต่างประเทศเริ่มจาก Astrom [2] ได้เสนอตัวควบคุมแบบจูนปรับตัวเองอย่างง่าย (Basic self-tuning controller) โดยมีกฎเกณฑ์ที่จะการลดความแปรปรวนของผลตอบ (Minimized output variance) และ Clarke [3,4,10] ใช้ยุทธวิธีควบคุมแบบ Suboptimal ลด Cost function ของความแปรปรวนของผลตอบออกช่วย (Auxiliary output variance) ให้มีค่าน้อยที่สุด ส่วน Gawthrop [5] ได้ใช้เงื่อนไขของระบบเพื่อใช้การควบคุมแบบจูนปรับตัวเองชนิด PID (Self-tuning PID controller) แต่ระบบบางอย่างไม่สามารถใช้กับการควบคุมแบบจูนปรับตัวเองแบบเล็งเลิศ (Optimal self-tuning control) ได้ ตัวอย่างเช่น ระบบที่มี Delay time เปลี่ยนแปลง หรือ ระบบที่เป็น Non-minimum phase system เป็นต้น จึงได้มีการนำวิธีกำหนดโพล (Pole placement method) เข้ามาประยุกต์กับปัญหาเหล่านี้ เช่น Wellstead [11,12] เสนอ Pole assignment self-tuning regulator ในขณะที่ Astrom [13] ใช้การกำหนดโพลและซีโร (Pole zero placement) กับปัญหา· Servo และ Allidina [14] ได้ประยุกต์ยุทธวิธีแบบ Suboptimal กับ การกำหนดโพลเข้าด้วยกัน เป็นต้น และสำหรับระบบที่มีตัวแปรหลายตัว Prager กับ Wellstead [15] ได้เสนอ Multivariable pole placement self-tuning regulator ด้วย นอกจากนี้การควบคุมแบบจูนปรับตัวเองสามารถประยุกต์ใช้ในทฤษฎีควบคุมแบบ Modern control เช่น Warwick [7] ได้พิสูจน์คุณสมบัติของ State space self-tuning control และ Tsay [16] ประยุกต์การกำหนดโพลเข้ากับ State space self tuner และ Shieh [17] สามารถแก้ปัญหาสมการ Riccati ในการป้อนกลับแบบสถานะ (State feedback) เป็นต้น

<u>รูปแบบและปัญหาของการควบคุมแบบจูนปรับตัวเอง</u>

โดยทั่วไปรูปแบบของระบบสามารถแทนได้ในรูปของ Discrete time ส่วนใหญ่รูปแบบของระบบ [3,4] แทนได้เป็น

$$A(z^{-1})\ y(t) = z^{-k}\ B(z^{-1})\ u(t) + x(t) \qquad \text{\_\_\_\_\_(2.1)}$$

โดยที่

$u(t), y(t)$	-	อินพุตและเอาต์พุตของระบบที่เวลาใด ๆ
$A(z^{-1}), B(z^{-1})$	-	โพลีโนเมียลที่สอดคล้องกับอินพุตและเอาต์พุตของระบบ
$x(t)$	-	สิ่งรบกวนในระบบ

การแทนรูปแบบของระบบนี้ $x(t)$ เป็นสิ่งรบกวนที่เกิดขึ้นภายในระบบจึงสามารถพิจารณาได้หลายรูปแบบ ตัวอย่างเช่น

1. เป็นค่าคงตัว (Constant)
2. เป็นสิ่งรบกวนเนื่องจากโหลด (Load disturbance)
3. สิ่งรบกวนที่สามารถวัดได้ (Measurement disturbance)
4. Non zero mean disturbance
5. สิ่งรบกวนแบบ Stationary stochastic procees

การแทนรูปแบบของระบบให้ใกล้เคียงความเป็นจริงนั้น จะต้องเลือกเงื่อนไขของ $x(t)$ ให้เหมาะสมกับลักษณะของกระบวนการมากที่สุด ตัวอย่างเช่น

$$x(t) = d + C(z^{-1}) / A(z^{-1})\ \xi(t)$$

โดยที่

d - ค่าคงที่

$\xi(t)$ - Noise sequence ของตัวแปรอิสระที่มีมัชฌิม (Mean) เท่ากับ 0 และความแปรปรวน (Covariance) มีค่าเท่ากับ σ_v^2

แต่ปกติมักจะแทนรูปแบบของระบบ [4] ด้วย

$$A(z^{-1})\ y(t) = z^{-k}\ B(z^{-1})\ u(t) + C(z^{-1})\ \xi(t) \qquad \text{\_\_\_\_\_(2.2)}$$

ปัญหาการแทนรูปแบบของระบบมีความสำคัญมากเนื่องจากในอัลกอริทึมของการควบคุมแบบจูนปรับตัวเองต้องอาศัยพารามิเตอร์ของระบบในการควบคุม ในกรณีที่แทนรูปแบบของการบอกเอกลักษณ์ไม่เหมาะสมก็จะเป็นสาเหตุหนึ่งที่ทำให้การควบคุมแบบจูนปรับตัวเองล้มเหลว นอกจากปัญหาของการกำหนดรูปแบบซึ่งเป็นปัญหาเกี่ยวกับเสถียรภาพแล้ว ปัญหาที่สำคัญอีกประการได้แก่

ปัญหาของสมรรถนะของการควบคุม ปัญหาที่เกิดขึ้นเสมอคือ ปัญหา Offset ซึ่งสาเหตุของการเกิดขึ้นกับลักษณะของการควบคุมและสิ่งรบกวนที่เกิดในระบบ ตัวอย่างเช่น สิ่งรบกวนที่เป็น Non zero mean ก็เป็นสาเหตุสำคัญที่ทำให้เกิดปัญหา Offset

การแก้ไขปัญหาของการเกิด Offset มีด้วยกันหลายวิธี เช่น การใส่อินทิเกรเตอร์ (Integrator) ในลูป การชดเชยค่า Offset โดยการเพิ่มค่าคงที่ หรือ ใส่กิริยาอินทิกรัล (Integral action) ให้กับตัวควบคุม เป็นต้น จากบทความของ Clarke [10] สามารถกำจัดค่า λ-offset และ Prediction offset โดยใช้ K-incremental prediction ในขณะที่ Gawthrop [12] ได้ใช้เงื่อนไขของระบบกับใช้กิริยาอินทิกรัลในลักษณะของตัวควบคุมแบบ PI หรือ PID เข้ามาเกี่ยวข้อง เพื่อกำจัด Offset ตัวอย่างเช่น ตัวควบคุมแบบจูนปรับตัวเองชนิด PI ของ Proudfoot [1] เป็นต้น และเมื่อเร็ว ๆ นี้ Gawthrop [5] ได้เสนอบทความเกี่ยวกับตัวควบคุมแบบจูนปรับตัวเองชนิด PID โดยใช้ Zero gain prediction ในการกำจัดสิ่งรบกวนแบบ Non zero mean disturbance อีกด้วย

ชนิดของการประมาณมีทั้งแบบ Unbias และ Bias การเลือกใช้ที่ถูกต้องควรพิจารณาลักษณะรูปแบบของระบบเป็นหลัก ส่วนใหญ่จะนิยมใช้ Recursive least square (RLS) [4] เนื่องจากมีความมั่นคงและสามารถดัดแปลงได้โดยง่ายตัวอย่างเช่น บทความของ Fuchs [18] หรือ การใช้ Forgetting factor [19] เพื่อกำหนดการลู่เข้าของอัลกอริทึม เป็นต้น RLS เป็นการประมาณแบบ Unbias จึงใช้ได้กับ Zero mean data เท่านั้น ส่วน Non zero mean data จะนิยมใช้ Recursive maximum likelihood (RML) [2,9] หรือ Recursive extended least square algorithm (ELS) ซึ่งดัดแปลงจาก RLS โดยสามารถประมาณค่าของพารามิเตอร์ของสิ่งรบกวนได้

ชนิดของการควบคุมแบบจูนปรับตัวเอง

จากบทความ [3,4,6,13,14,15] การควบคุมแบบจูนปรับตัวเองสามารถประยุกต์ใช้ได้กับวิธีการควบคุมได้หลายชนิด ซึ่งสามารถใช้ได้ทั้งกับระบบหนึ่งอินพุตหนึ่งเอาต์พุตและระบบหลายอินพุตหลายเอาต์พุต ซึ่งมีวิธีควบคุมแบบพื้นฐานที่นิยมใช้ได้แก่ การควบคุมแบบเล็งเลิศและการควบคุมแบบกำหนดโพล เป็นต้น การควบคุมสามารถแบ่งออกเป็น 2 แบบ คือ การควบคุมแบบ Classical control และ การควบคุมแบบ Modern control

1. การควบคุมแบบ Classical control

เป็นการควบคุมโดยอาศัยการแทนรูปแบบของระบบด้วยสมการแบบโพลีโนเมียลซึ่งอยู่ในรูปของ Discrete time การประยุกต์วิธีการควบคุมเข้ากับการจูนแบบปรับตัวเองสามารถใช้ได้ดังนี้คือ

1.1 ระบบหนึ่งอินพุตหนึ่งเอาต์พุต

ระบบ Single loop โดยทั่วไปจะมีลักษณะเป็นระบบหนึ่งอินพุตหนึ่งเอาต์พุตซึ่งพบเห็นบ่อย การประยุกต์การควบคุมแบบจูนปรับตัวเองสามารถควบคุมระบบดังกล่าวได้โดยง่าย เนื่องจากลักษณะของระบบไม่ซับซ้อน วิธีการควบคุมแบบนิยมใช้ได้แก่

1.1.1 การควบคุมแบบเล็งเลิศ (Optimal control)

การควบคุมแบบเล็งเลิศสำหรับระบบที่มีหนึ่งอินพุตหนึ่งเอาต์พุตปกติจะมีลักษณะการออกแบบคล้าย ๆ กันคือ ตั้งกฏเกณฑ์เพื่อพยายามลดค่า Cost function ของกลุ่มตัวแปร ซึ่งผู้ใช้สามารถจะดัดแปลงกฏเกณฑ์ให้ความเหมาะสมตามต้องการ จากบทความ [3,4,20,21] วิธีการควบคุมแบบเล็งเลิศนี้จะประยุกต์รูปแบบคาดการณ์ (Predictive model) ในการออกแบบซึ่งมีการควบคุมอยู่ 2 แบบ คือ การควบคุมแบบ Minimum variance control จากบทความของ Astrom และ การควบคุมแบบ Generalized minimum variance control จากบทความของ Clarke แต่เนื่องจากวิธีของ Clarke มีข้อได้เปรียบกว่าจึงเป็นที่นิยมใช้มากกว่า [4,8,10]

การควบคุมแบบ Generalized minimum variance control สามารถกำหนดโพลให้กับสมการระบบวงปิด (Closed loop equation) ที่มีโครงสร้างเป็นแบบ Model reference control เสนอโดย Clarke [4] อาศัยการพิจารณาผลตอบช่วย (Auxiliary output) แทนที่จะใช้ผลตอบจริงของระบบ โดยทำให้ผลตอบช่วยมีค่าน้อยที่สุดกำหนดให้

$$\Phi(t) = P(z^{-1})\, y(t) \qquad \text{(2.3)}$$

โดยที่

$\Phi(t)$ - Auxiliary output

$P(z^{-1})$ - User specified transfer function

และ $P(z^{-1}) = P_n(z^{-1})/P_d(z^{-1})$

จาก [4] กำหนดให้

$$C(z^{-1})P(z^{-1}) = E(z^{-1})A(z^{-1}) + z^{-k} F(z^{-1})/P_d(z^{-1}) \qquad (2.4)$$

จาก (2.2)-(2.4) จะได้

$$\Phi(t+k) = F(z^{-1})\, y''(t) + G(z^{-1})\, u'(t) + E(z^{-1})\, \xi(t) \qquad (2.5)$$

โดยที่

$$y''(t) = y(t)/C(z^{-1})P_d(z^{-1})$$

$$u'(t) = u(t)/C(z^{-1})$$

$$G(z^{-1}) = E(z^{-1})\, B(z^{-1})$$

เมื่อลดค่าสมการ (2.5) ให้น้อยที่สุด จะได้

$$\Phi^*(t+k) = F(z^{-1})\, y''(t) + G(z^{-1})\, u'(t) \qquad (2.6)$$

ความคลาดเคลื่อนของการคาดการณ์ (Prediction error) จะเท่ากับ

$$\Phi^\sim(t+k) = \Phi(t+k) - \Phi^*(t+k)$$

$$= E(z^{-1})\, \xi(t+k)$$

และที่สภาวะคงตัว (Steady state) จะได้ว่า

$$\Phi^*(t+k) = w(t)$$

โดยที่ $w(t)$ - ค่าที่ตั้งไว้ (Set point)

ดังนั้นจากสมการ (2.5)

$$u(t) = 1/G(z^{-1})\, [w(t) - F(z^{-1})y(t)/T(z^{-1})P_d(z^{-1})] \qquad (2.7)$$

และได้สมการวงปิดเป็น

$$y(t) = \frac{1}{P(z^{-1})}\, w(t-k) + \frac{E(z^{-1})}{P(z^{-1})}\, \xi(t)$$

$$u(t) = \frac{A(z^{-1})}{P(z^{-1})B(z^{-1})}\, w(t) - \frac{F(z^{-1})}{P_N(z^{-1})B(z^{-1})}\, \xi(t)$$

แต่เนื่องจากสมการ Characteristic ของการควบคุมแบบจูนปรับตัวเองนี้จะตอบสนองไวต่อความคลาดเคลื่อนของรูปแบบ (Error modelling) จึงทำให้ทั้งผลตอบของระบบและตัวควบคุมมีการเปลี่ยนแปลงอย่างรวดเร็ว ดังนั้นจึงต้องแก้ไขด้วยการลดการจูนรูปแบบอ้างอิง (Detuned model reference) โดยการ weight ผลตอบของตัวควบคุมด้วย $Q(z^{-1})$ ซึ่งเป็น User specified transfer function เช่นเดียวกับ $P(z^{-1})$ โดยจะ

ให้ Cost function เป็น

$$\psi(t+k) = \Phi^*(t+k) + Q(z^{-1})\ u(t) - w(t) \qquad \_\_\_\_(2.8)$$

ที่สภาวะคงตัวจะได้ $\psi(t+k) = 0$ ดังนั้น

$$\Phi^*(t+k) + Q(z^{-1})\ u(t) = w(t)$$

หรือ $u(t) = 1/Q(z^{-1})\ [w(t) - \Phi^*(t+k)]$

แทนค่าในสมการ (2.6) จะได้

$$u(t) = 1/(Q + G/T)\ [w(t) - F\ y(t)/T\ P_d] \qquad \_\_\_\_(2.9)$$

และได้สมการวงปิดเป็น

$$y(t) = B/(PB+QA)\ w(t-k) + (EB+QT)/(PB+QA)\ \xi(t)/T \qquad \_\_\_\_(2.10)$$

$$u(t) = A/(PB+QA)\ w(t) - (F/P_d)/(PB+QA)\ \xi(t)/T \qquad \_\_\_\_(2.11)$$

การเลือกค่าของ $Q(z^{-1})$ จะต้องเลือกให้เหมาะสม เนื่องจาก $Q(z^{-1})$ เป็นสาเหตุหนึ่งที่ทำให้เกิดปัญหา Offset [10]

ขั้นตอนการออกแบบการควบคุมแบบปรับตัวเองชนิด Generalized minimum variance control สามารถทำได้ดังนี้

1. เลือก User specified transfer function $P(z^{-1})$ และ $Q(z^{-1})$ โดยให้ $P(z^{-1}) = 1/M(z^{-1})$ เมื่อ $M(z^{-1})$ เป็น Desired closed loop models และ $Q(z^{-1}) = \lambda Q'(z^{-1})$ เมื่อ $1/Q'(z^{-1}) = L(z^{-1})$ ซึ่งปกติจะมีเทอมของอินทิกรัลอยู่ด้วย

2. หาค่าของพารามิเตอร์ $F(z^{-1}),G(z^{-1})$ โดยการประมาณแบบ RLS หรือ ELS ในสมการ (2.5)

3. แทนค่าพารามิเตอร์ที่ได้ในสมการ (2.8) เพื่อหาผลตอบของตัวควบคุม u(t)

นอกจากนี้การออกแบบการควบคุมอาจจะดัดแปลงให้เหมาะกับลักษณะของระบบและการควบคุม เช่น วิธีการออกแบบของ Clarke และ Gawthrop [3] จะให้

$$\psi(t) = \Phi(t) + Q\ u(t-k) - w(t-k) \qquad \_\_\_\_(2.12)$$

ซึ่งสามารถจัดรูปแบบคาดการณ์ (Prediction model) ได้เป็น

$$\psi^*(t+k|t) = \Phi^*(t+k|t) + Q\ u(t) - w(t) \qquad \_\_\_\_(2.13)$$

$$C\psi^*(t+k|t) = C\Phi^*(t+k|t) + CQ\ u(t) - Cw(t) \qquad \_\_\_\_(2.14)$$

ที่สภาวะคงตัว $C\psi^{*}(t+k|t) = 0$

ดังนั้น $C\Phi^{*}(t+k|t) + CQ\,u(t) - Cw(t) = 0$

จากสมการ (2.5) แทนค่าจะได้

$$F'y^{f}(t) + G'u(t) + H'w(t) = 0 \qquad (2.15)$$

โดยที่ $y^{f} = y(t)/P_{d}(z^{-1})$

การหาค่าพารามิเตอร์ของตัวควบคุมก็จะประมาณสมการ (2.15) แทนสมการ (2.6) เพื่อหา F',G',H' ซึ่งสามารถนำไปแทนค่าหาผลตอบของตัวควบคุมได้

ส่วนการควบคุมแบบ Minimum variance control สามารถดัดแปลงได้เช่นกันโดยจากสมการ (2.3),(2.8) ซึ่งเป็น cost function ของการควบคุมแบบ Generalized minimum variance control สามารถเขียนได้เป็น

$$\psi(t+k) = P(z^{-1})y(t+k) + Q(z^{-1})u(t) - w(t)$$

การแทนค่า $P(z^{-1})$ และ $Q(z^{-1})$ สามารถเลือกได้อย่างอิสระ ดังนั้นในกรณีที่ต้องการควบคุมแบบ Minimum variance control ก็จะกำหนดให้ $P(z^{-1}) = 1$, $Q(z^{-1}) = 0$ และ $w(t) = 0$ ทำให้ cost function กลายเป็น

$$\psi(t) = y(t)$$

ซึ่งเมื่อใช้ในการควบคุมจะเป็นการลดค่าของผลตอบของกระบวนการให้น้อยที่สุดสามารถใช้ได้ทำนองเดียวกับวิธีของ Astrom [2]

1.1.2 <u>ตัวควบคุมชนิดกำหนดโพล (Pole placement Method)</u>

การควบคุมแบบจูนปรับตัวเองชนิดกำหนดโพลถึงแม้จะไม่ได้ค่าเล็งเลิศก็ตามแต่ว่าสามารถใช้ได้กับระบบที่ควบคุมแบบเล็งเลิศไม่ได้ และสามารถรักษาคุณสมบัติของการปรับตัวเอง (Self tuning property) เช่นเดียวกับการควบคุมแบบจูนปรับตัวเองเชิงเล็งเลิศ และมีความมั่นคงกว่า ข้อดีของการควบคุมแบบจูนปรับตัวเองชนิดกำหนดโพล คือ

1. ใช้กับ Non minimum phase system ได้ง่ายในขณะที่การควบคุมแบบจูนปรับตัวเองเชิงเล็งเลิศทำไม่ได้
2. ใช้กับระบบที่มีกาลล่าช้าเปลี่ยนแปลงตามเวลาหรือไม่ทราบค่า
3. สามารถกำหนดข้อจำกัดให้กับผลตอบของระบบ

การควบคุมแบบจูนปรับตัวเองชนิดกำหนดโพลมีขั้นตอนเหมือนการควบคุมแบบปรับตัวเองโดยทั่วไป แตกต่างที่การสังเคราะห์สัญญาณควบคุมโดยอาศัยกำหนดโพลของระบบวงปิดที่อยู่ในรูปของโพลีโนเมียล ซึ่งการเลือกส่วนใหญ่จะเลือกโพลที่สำคัญเพียงหนึ่งคู่เท่านั้น ส่วนที่เหลือให้อยู่ใกล้ ๆ กับ Origin เพื่อให้สะดวกต่อการกำหนดผลตอบสนองปกติจะเลือกค่า Damping ratio หรือ Rise time แทนการกำหนดโพลโดยตรงเมื่อได้ Desired closed loop poles polynomial แล้ว จะเปรียบเทียบกับ Closed loop poles polynomial ที่ได้จากการสังเคราะห์จากรูปแบบของระบบ ก็จะสามารถหาพารามิเตอร์สำหรับใช้กับกฎการควบคุมซึ่งต่างจากการควบคุมแบบจูนปรับตัวเองเชิงเล็งเลิศที่ใช้การกำจัดซีโร [3,4] เพียงอย่างเดียวจึงทำให้ผลตอบของระบบและตัวควบคุมมีการเปลี่ยนแปลงอย่างรวดเร็ว

สำหรับตัวอย่างการเลือกโพลในทางปฏิบัติ [2,13] ที่นิยมใช้กันจะกำหนดค่า ζ และ ความถี่ของผลตอบดังสมการต่อไปนี้

$$T(z^{-1}) = 1 - 2e^{-\zeta wh}\cos wh\sqrt{1-\zeta^2}\, z^{-1} + e^{-2\zeta wh} z^{-2}$$

โดยที่ ζ - damping ratio

w - ความถี่ที่ต้องการกำหนด

h - คาบของการสุ่ม

$T(z^{-1})$ - Desired closed loop poles polynomial

การใช้อัลกอริทึมแบบกำหนดโพลในการควบคุมแบบจูนปรับตัวเองเริ่มจากปัญหา Regulator ก่อนโดย Edmunds จากนั้นก็มีบทความที่เกี่ยวข้องมากขึ้น ตัวอย่างเช่น Self-tuning pole assignment regulators ของ Wellstead หรือ Stochastic pole assignment strategy ของ Wouters เป็นต้น นอกจากนี้วิธีการกำหนดโพลสามารถใช้ได้กับปัญหา Tracking ด้วย ตัวอย่างเช่น บทความของ Astrom กับ Wittenmark [13] หรือ บทความของ Wellstead กับ Zanker [11,12] เป็นต้น และยังสามารถใช้ร่วมกับการควบคุมเชิงเล็งเลิศได้อีกด้วย โดยจากบทความของ Allidina [14] ที่เสนอ Generalised self-tuning controller with pole assignment เพื่อลดความสูญเสีย (Losses) เนื่องจากการใช้วิธีกำหนดโพล

การควบคุมแบบจูนปรับตัวเองชนิดกำหนดโพลสามารถทำได้ง่ายจากสมการ Discrete time ของระบบ [17] ดังนี้

$$A(z^{-1})\, y(t) \;=\; z^{-k}\, B(z^{-1})\, u(t) \;+\; C(z^{-1})\, e(t) \qquad \text{\_\_\_\_(2.16)}$$

โดยที่

$$A(z^{-1}) = 1 + a_1 z^{-1} + \dots\dots\dots + a_n z^{-n}$$

$$B(z^{-1}) = b_o + b_1 z^{-1} + \dots\dots\dots + b_n z^{-n}$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + \dots\dots\dots + c_n z^{-n}$$

และ $\{ e(t) \;:\; t = 0,1 \dots\dots \}$ เป็น Zero mean white noise gaussian sequence

สำหรับปัญหา Regulator ถ้าเลือกกฎการควบคุมเป็น

$$F(z^{-1})\, u(t) \;=\; G(z^{-1})\, y(t) \qquad \text{\_\_\_\_(2.17)}$$

โดยที่

$$F(z^{-1}) = 1 + f_1 z^{-1} + \dots\dots\dots + f_n z^{-n}$$

$$G(z^{-1}) = g_o + g_1 z^{-1} + \dots\dots\dots + g_n z^{-n}$$

จากสมการ (2.16),(2.17) กำหนดให้

$$n_b + k - 1 = n_f \quad \text{และ} \quad n_g = n_a - 1$$

จะได้สมการวงปิดเป็น

$$(A(z^{-1})F(z^{-1}) - z^{-k}B(z^{-1})G(z^{-1}))y(t) = C(z^{-1})F(z^{-1})e(t) \qquad \text{\_\_\_\_(2.18)}$$

จากสมการ (2.18) ถ้าเลือก Desired closed loop poles polynomial $T(z^{-1})$ เป็น

$$T(z^{-1}) = 1 + t_1 z^{-1} + \dots\dots\dots + t_n z^{-1} \qquad \text{\_\_\_\_(2.19)}$$

แทนสมการ (2.19) จะได้

$$A(z^{-1})\, F(z^{-1}) - z^{-k}\, B(z^{-1})\, G(z^{-1}) \;=\; T(z^{-1}) \qquad \text{\_\_\_\_(2.20)}$$

สมการ (2.16) ทำการประมาณค่าพารามิเตอร์ โดยใช้ RLS จะได้

$$A^*(z^{-1})\, y(t) \;=\; B^*(z^{-1})\, u(t) \;+\; \epsilon(t) \qquad \text{\_\_\_\_(2.21)}$$

โดยที่

$$A^*(z^{-1}) = 1 + a_1^* z^{-1} + \dots\dots\dots + a_n^* z^{-n}$$

$$B^*(z^{-1}) = b_o^* z^{-1} + \dots\dots\dots + b_{n+k-1}^* z^{-n-k}$$

แทนพารามิเตอร์ที่ได้จากการประมาณ $a_1^*, \dots\dots, b_o^*, b_1^*, \dots\dots$ ใน สมการ (2.20)

$$A^*(z^{-1})\, F(z^{-1}) - B^*(z^{-1})\, G(z^{-1}) \;=\; T(z^{-1}) \qquad \text{\_\_\_\_(2.22)}$$

แก้สมการ (2.22) หาค่า $F(z^{-1})$, $G(z^{-1})$ ก็สามารถหา $u(t)$ ได้จะเห็นว่า

$$T(z^{-1})\, y(t) \;=\; F(z^{-1})\, \epsilon(t)$$

ดังนั้นเมื่อ $t \longrightarrow \infty$ และ $C(z^{-1}) = 1$

และ $A^{*}(z^{-1}) \longrightarrow A(z^{-1})$, $B^{*}(z^{-1}) \longrightarrow B(z^{-1})$

จะได้ว่า $\epsilon(t) \longrightarrow e(t)$

นั่นคือ $T(z^{-1})\, y(t) = F(z^{-1})\, e(t)$ ____(2.23)

ขั้นตอนการควบคุมแบบจูนปรับตัวเองชนิดกำหนดโพลก็คือ

1. ทำการประมาณค่าพารามิเตอร์หา $A^{*}(z^{-1}), B^{*}(z^{-1})$ จากสมการ(2.21)
2. หา $F(z^{-1}), G(z^{-1})$ จากสมการ (2.22)
3. หา $u(t)$ จากสมการ (2.17)

โดยจากทฤษฎีแล้ว เมื่อ $A^{*}(z^{-1}), B^{*}(z^{-1})$ สามารถลู่เข้าหา $A(z^{-1}), B(z^{-1})$ ได้แล้วระบบก็จะมีรูปลักษณ์ของระบบวงปิด (Closed loop configuration) ตามต้องการ

1.2 <u>ระบบหลายอินพุตหลายเอาต์พุต</u>

การควบคุมระบบหลายอินพุตหลายเอาต์พุตด้วยการควบคุมแบบจูนปรับตัวเองสามารถทำได้เช่นเดียวกับระบบระบบหนึ่งอินพุตหนึ่งเอาต์พุตแต่จะมีความยุ่งยากมากกว่า เนื่องจากคุณสมบัติและขีดจำกัดบางอย่างของการควบคุมแบบจูนปรับตัวเอง ตัวอย่างเช่น การออกแบบควบคุมชนิดเล็งเลิศกับระบบหลายอินพุตหลายเอาต์พุตจะประสบปัญหาเช่นเดียวกับระบบหนึ่งอินพุตหนึ่งเอาต์พุต คือ ใช้กับ Non-minimum phase system ไม่ได้และมีขั้นตอนการคำนวณยุ่งยาก เช่น ขั้นตอน ' Cut and try ' ของ Clarke เป็นต้น นอกจากนี้ยังไวต่อการเปลี่ยนแปลงและมีข้อกำหนดว่ากาลล่าช้าในแต่ละลูปต้องเท่ากัน จึงไม่เหมาะที่จะใช้กับระบบหลายอินพุตหลายเอาต์พุต วิธีที่ใช้ได้ดีกับระบบหลายอินพุตหลายเอาต์พุตเห็นจะเป็นการควบคุมแบบจูนปรับตัวเองชนิดกำหนดโพลแบบหลายตัวแปร เพราะนอกจากการจะใช้วิธีกำหนดโพลกับระบบหนึ่งอินพุตหนึ่งเอาต์พุต (single input single output) แล้วยังสามารถใช้กับระบบหลายอินพุตหลายเอาต์พุต (Multivariable system) ได้ การใช้วิธีกำหนดโพลสามารถแก้ไขปัญหาที่เกิดขึ้นกับการออกแบบควบคุมชนิดเล็งเลิศได้ รวมทั้งสามารถสมมติ Noise disturbance ให้เป็น Independent white noise vector process ได้

ลักษณะขั้นตอนของการควบคุมแบบจูนปรับตัวเองชนิดหลายตัวแปรโดยกำหนดโพลจะคล้ายกับการควบคุมแบบจูนปรับตัวเองตัวแปรเดียวโดยอาศัยการสมมูลของสมการของระบบปิดแต่จะพิจารณารูปแบบของ Transfer function ในรูปของเมตริกซ์ ตัวอย่างเช่น จากบทความของ Prager และ Wellstead [15] เป็นต้น คุณสมบัติที่เห็นได้ชัดเมื่อใช้การกำหนดโพลในระบบหลายตัวแปร

1. ใช้กับ Non minimum phase system ได้โดยง่าย
2. อัตราการสุ่มในแต่ละลูปไม่จำเป็นต้องเท่ากัน
3. ผลตอบของตัวควบคุมไม่ไวต่อการเปลี่ยนแปลง

การใช้การควบคุมแบบจูนปรับตัวเองชนิดกำหนดโพลกับระบบหลายตัวแปร นอกจากสามารถกำหนดรูปลักษณ์ของระบบวงปิด (Closed loop configuration) ได้ตามต้องการแล้ว ความมั่นคงก็ยังมีมากกว่าการควบคุมแบบจูนปรับตัวเองเชิงเล็งเลิศ และสามารถประยุกต์ใช้กับระบบต่าง ๆ ได้มากกว่าด้วย

2. การควบคุมแบบ Modern control

ในการควบคุมแบบจูนปรับตัวเองถึงแม้จะสามารถประยุกต์ใช้กับระบบหลายตัวแปรได้ก็ตาม แต่เมื่อมีปัญหา Noise disturbance source ที่มากกว่าหนึ่ง การแทน Transfer function ด้วยสมการพลวัต (Dynamic equation) จะเหมาะสมกว่าและสามารถใช้กับระบบหลายตัวแปรได้ นอกจากนี้สามารถที่จะใช้ทฤษฎีการควบคุมเข้ามาประยุกต์ได้โดยง่าย ตัวอย่างเช่น การทำ State estimation หรือ Observer เพื่อทำการควบคุมแบบ State feedback หรือ ใช้การกำหนดโพล เป็นต้น

การควบคุมแบบจูนปรับตัวเองในแบบ Modern control approach จะมีคุณสมบัติเหมือนทำใน Classical control approach ซึ่งพิสูจน์ไว้ในบทความของ Warwick [7] ตัวอย่างของ State space self tuner ได้แก่บทความของ Tsay [16] ที่ทำ Pole assignment self tuning feedback control โดยใช้ System identification ร่วมกับ State estimation หรือบทความของ Hesketh [6] และ บทความของ Shieh [17] ที่ทำ Suboptimal state space self tuner กับปัญหา Linear stochastic multivariable system โดยใช้สมการ Riccati และใช้ Matrix sign function หาคำตอบของสมการ Riccati ในการทำ State feedback

2.1 การควบคุมแบบกำหนดโพล

จากสมการสถานะของระบบหลายอินพุตหลายเอาต์พุต (MIMO) [17]

$$x(k+1) = Ax(k) + Bu(k) + w(k) \qquad (2.24)$$

$$y(k) = C^T x(k) + v(k) \qquad (2.25)$$

โดยที่

$x(k)$ - n x 1 state vector

$u(k), y(k)$ - อินพุตและเอาต์พุตของระบบ

A - n x n system matrix

B, C - n x 1 vectors

$w(k)$ - n x 1 white vector process ที่มีมัชฌิม (Mean) เท่ากับ 0 และความแปรปรวน (Covariance) มีค่าเท่ากับ $Q\delta(k_1-k_2)$

$v(k)$ - White process ที่มีมัชฌิมเท่ากับ 0 และความแปรปรวนมีค่าเท่ากับ σ_v^2

จากสมการ (2.24),(2.25) ทำ state estimation จะได้

$$x^*(k) = Ax^*(k-1) + Bu(k-1) + K(k)e(k-1) \qquad (2.26)$$

$$Y(k) = C^T x^*(k) + e(k) \qquad (2.27)$$

โดยที่

$x^*(k)$ - Estimation of $x(k)$

$e(k)$ - White noise process ที่มีมัชฌิมเท่ากับ 0

$K(k)$ - Kalman gain .

ถ้าระบบในสมการ (2.26),(2.27) Observable จะได้

$$x_o^* = A_o x_o^*(k-1) + B_o u(k-1) + K_o(k)e(k-1) \qquad (2.28)$$

$$y(k) = C_o^T x_o^*(k) + e(k) \qquad (2.29)$$

โดยที่

$$x^*(k) = Hx_o^*(k)$$

$$H^{-1} = H_2^{-1} H_1$$

$$H_1 = [C, A^T C, \ldots\ldots, (A^{n-1})^T C]$$

$$H_2^{-1} = [C_o, A_o^T C_o, \ldots, (A_o^{n-1})^T C_o]^{-T}$$

$$C_o = [1,0, \ldots ,0]^T$$

$$A_o = H^{-1}AH$$

$$= \begin{vmatrix} -a_1 & 1 & 0 & \cdots & \\ -a_2 & 0 & 1 & 0 & \cdots \\ \cdots & & & 1 & \cdots \\ -a_{n-1} & \cdots & & & 1 \\ -a_n & \cdots & & & 0 \end{vmatrix}$$

$$B_o = H^{-1}B = [b_1, b_2, \ldots , b_n]^T$$

$$K_o(k) = H_{-1}K(k) = [k_1(k), k_2(k), \ldots, k_n(k)]^T$$

ทำนองเดียวกันใน Warwick [7] จะได้

$$x^*(k) = (I - A_o z^{-1})^{-1} [B_o z^{-1} u(k) + K_o(k) z^{-1} e(k) \qquad (2.30)$$

$$y(k) = C_o^T (I - A_o z^{-1}) B_o z^{-1} u(k) + (1 + C_o^T (I - A_o z^{-1})^{-1} K_o(k) z^{-1}) e(k) \qquad (2.31)$$

เมื่อเทียบสมการ (2.31) กับ

$$y(t) = B(z^{-1})/A(z^{-1}) u(t) + C(z^{-1})/A(z^{-1}) e(t) \qquad (2.32)$$

โดยที่

$$A(z^{-1}) = 1 + a_1 z^{-1} + \ldots + a_n z^{-1}$$

$$B(z^{-1}) = 1 + b_1 z^{-1} + \ldots + b_n z^{-1}$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + \ldots + c_n z^{-1}$$

และ $c_i = a_i + k_i(k) \quad ; \quad i = 1,2,3,\ldots,n$

จะได้ $A(z^{-1})y(k) = B(z^{-1})u(k) + C(z^{-1})e(k) \qquad (2.33)$

จากสมการ (2.33) เมื่อทำการหาเอกลักษณ์ของ $A(z^{-1}), B(z^{-1}), C(z^{-1})$ แทนในสมการ (2.28) จะได้

$$x_o^*(k+1) = A_o^*(k) x_o^*(k-1) + B_o^*(k) u(k-1) + K_o^*(k) e^*(k-1) \qquad (2.34)$$

ถ้าพิจารณาสมการ (2.28) เป็น Controllable และระบบไม่มี noise

$$x_c(k+1) = A_c x_c(k) + B_c u(k) \qquad (2.35)$$

$$y_c(k) = C_c^T x_c(k) \qquad (2.36)$$

โดยที่

$$A_c = \begin{vmatrix} -a_1 & -a_2 & \cdots\cdots & -a_n \\ 1 & 0 & \cdots\cdots & . \\ 0 & 1 & \cdots\cdots & . \\ \cdots & \cdots & \cdots\cdots & . \\ \cdots & \cdots & \cdots\cdots 1 & 0 \end{vmatrix}$$

$$B_c = [b_1, b_2, \ldots\ldots\ldots, b_n]$$

$$C_c = [1, 0, \ldots\ldots\ldots\ldots, 0]^T$$

กำหนดให้ Feedback control law คือ

$$u(k) = r(k) - F_c^T x_c(k) \qquad \text{\_\_\_\_\_}(2.37)$$

$$F_c = [f_{c1}, f_{c2}, \ldots\ldots\ldots]$$

ดังนั้นสมการวงปิดจะเป็น

$$x_c(k+1) = A_c x_c(k) - B_c F_c^T x_c\ (k) + B_o r(k)$$

$$= (A_c - B_o F_c^T) x_c(k) + B_c r(k) \qquad \text{\_\_\_\_\_}(2.38)$$

ทำนองเดียวกับสมการ (2.28)-(2.33) โดยแทน $x_o^*(k) = Tx_c(k)$

โดยที่

$$T = T_1 T_2^{-1}$$

$$T_1 = [B_o, A_o B_o, \ldots\ldots\ldots, A_o^{n-1} B_o]$$

$$T_2^{-1} = [B_c, A_c B_c, \ldots\ldots\ldots, A_c^{n-1} B_c]^{-1}$$

จะได้

$$x_o^*(k) = (I-(A_o-B_oF_c^TT^{-1})z^{-1})^{-1}(B_oz^{-1}r(k)+K_o(k)z^{-1}e(k)) \qquad \text{\_\_\_\_\_}(2.39)$$

$$y(k) = C_o^T(I-(A_o-B_oF_c^TT^{-1})z^{-1})^{-1}B_oz^{-1}r(k)+(1+C_o^T(I-(A_o-B_o F_c^TT^{-1})^{-1}K_o(k)z^{-1}\ e(k) \qquad \text{\_\_\_\_\_}(2.40)$$

ทำนองเดียวกับสมการ (2.33)

$$y(k) = B(z^{-1})/A_c(z^{-1})r(k) + D_c(z^{-1})/A_c(z^{-1})\ e(k) \qquad \text{\_\_\_\_\_}(2.41)$$

โดยที่

$$A_c(z^{-1}) = 1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \ldots + \alpha_n z^{-n}$$

$$D_c(z^{-1}) = 1 + \delta_1 z^{-1} + \delta_2 z^{-2} + \ldots + \delta_n z^{-n}$$

และ $\delta_i = d_i + f_{di}$; $i = 1,2,3,...$

จากสมการ (2.38) Closed loop characteristic polynomial คือ

$$G_c(z) = DET\ [zI - A_c + B_cF_c^T]$$

$$= z_n + \Sigma\ (a_i + f_{ci})z^{n-1} \qquad \text{\_\_\_\_\_(2.42)}$$

และ Desired closed loop polynomial

$$G_c(z) = \Pi\ (z-p_i)$$

$$= z_n + \Sigma\ \alpha_i z^{n-1} \qquad \text{\_\_\_\_\_(2.43)}$$

จากสมการ (2.42),(2.43)

$$\alpha_i = a_i + f_{ci}$$

หรือ

$$f_{ci} = \alpha_i - a_i \qquad \text{\_\_\_\_\_(2.44)}$$

โดยที่

$$F_d^T = [f_{d1}, f_{d2}, ...] = F_c^T T^{-1} S$$

$$S = S_1 S_2^{-1}$$

$$S_1 = [B_o, DB_o, ..., D^{n-1}B_o]$$

$$S_2^{-1} = [B_c, D^TB_c, ..., (D^T)^{n-1}B_c]^{-1}$$

$$D = \begin{vmatrix} -d_1 & 1 & 0 & \cdots \\ -d_2 & 0 & 1 & \cdots \\ \cdots & & & \\ -d_{n-1} & \cdots & & 1 \\ -d_n & \cdots & & 0 \end{vmatrix}$$

ที่สภาวะคงตัวผลตอบจะเท่ากับค่าที่ตั้งไว้

$$E[y(k)] = B(1)/A(1)\ r_c(k) \qquad \text{\_\_\_\_\_(2.45)}$$

และจากสมการ (2.37) จะได้

$$u(k) = K_r r(k) - F_c^T T^{-1} x^*(k) \qquad \text{\_\_\_\_\_(2.46)}$$

โดยที่

$$K_r = A_c(1)/B_c(1) = \frac{(1 + \Sigma\ \alpha_i)}{\Sigma\ b_i} \quad ; \quad B_c(1) \neq 0$$

สมการ (2.46) จะเป็นกฎการควบคุมที่ใช้สำหรับ State space self-tuning algorithm โดยทำ State feedback ด้วยวิธีกำหนดโพลในบทความของ Tsay [16]

สรุป

ตัวควบคุมแบบจูนปรับตัวเองสามารถใช้ควบคุมระบบที่ซับซ้อนได้แก่ ระบบที่มีภาวะไม่เชิงเส้น หรือ ระบบที่ไม่ทราบค่าพารามิเตอร์แน่นอน ทำให้การใช้งานค่อนข้างกว้าง แต่ทั้งนี้ทั้งนั้นการเลือกใช้อัลกอริทึมต้องคำนึงถึงเสถียรภาพ การลู่เข้าและสมรรถนะของระบบด้วย ตัวอย่างเช่น การใช้ตัวควบคุมแบบจูนปรับตัวเองชนิด PI ในขบวน pH neutralisation process [1] การควบคุมอุณหภูมิโดยใช้ K-incremental predictor [10] เป็นต้น จากบทความ [4] จะเห็นว่าการควบคุมแบบจูนปรับตัวเองมีอยู่หลายวิธี สามารถเลือกลักษณะการควบคุมได้ทั้งแบบ Classical control และแบบ Modern control ซึ่งมีวิธีที่เป็นพื้นฐานและน่าสนใจได้แก่ Generalized minimum variance control และ Pole placement method สามารถประยุกต์ใช้กับไมโครคอมพิวเตอร์ในการควบคุมได้ง่าย ตัวอย่างเช่น Simplied self-tuning controller based on single chip microcomputer [8] และ การใช้ z-80 ใน State space self tuner [6] เป็นต้น