

อัลกอริทึมสำหรับระบบควบคุมแบบปรับตัวเอง ที่ใช้อัตราการซีกตัวอย่างสัญญาณที่สูง

ผู้วิจัย: ดร. สุขสันต์ นุ่นงาม

บทคัดย่อ

บทความนี้เสนออัลกอริทึมสำหรับการปรับตัวเองของระบบควบคุมเชิงดิสครีตที่ใช้อัตราการซีกตัวอย่างสัญญาณที่รวดเร็ว อัลกอริทึมนี้ได้พัฒนาขึ้นในลักษณะที่สามารถจะนำเอาข้อมูลเบื้องต้นเกี่ยวกับกระบวนการจริง (Actual plant) มาใช้ประโยชน์ได้ ข้อมูลนี้อยู่ในรูปของความรู้เกี่ยวกับกระบวนการที่ระบุ (nominal plant) นั่นคือข้อมูลดังกล่าวจะถูกนำไปใช้ในการออกแบบตัวควบคุมที่ระบุ (nominal robust controller) เพื่อควบคุมกระบวนการจริง จากนั้นอัลกอริทึมของการปรับตัวเองจะทำการประมาณค่าแบบกำลังสองน้อยสุดเพื่อไปปรับตัวควบคุม Q ที่เพิ่มเข้าไปเพื่อเสริมสมรรถนะของ ตัวควบคุมที่ระบุในกรณีที่กระบวนการจริงมีการเปลี่ยนแปลงหรือแตกต่างไปจากกระบวนการที่ระบุ ในอัลกอริทึมที่เสนอนี้ ตัวควบคุม Q จะถูกแปลงไปอยู่ในโคออดิเนตใหม่ เพื่อให้พารามิเตอร์ของตัวควบคุม Q สามารถจะแทนอย่างประมาณในลักษณะที่เป็นอนุกรมโดยใช้จำนวนพารามิเตอร์ที่น้อยกว่าการใช้พารามิเตอร์ในโคออดิเนตเดิม วิธีการนี้จะทำให้อัลกอริทึมของการปรับพารามิเตอร์มีประสิทธิภาพมากยิ่งขึ้น

Abstract

This paper presents a discrete time adaptive control algorithm for rapidly sampled continuous time plants. The algorithm is formulated in a general framework to incorporate any a priori plant information: in the form of knowledge of a nominal plant. The nominal model of the plant is used to design a fixed robust nominal controller for the actual plant. The adaptive algorithm uses standard least squares method to tune an augmented controller Q to enhance the performance of this fixed robust nominal controller in the face of perturbation of the actual plant from the nominal one. A novel co-ordinate basis transformation for the augmented controller Q is introduced so that the parameters of Q can be approximated by a power series with a small number of terms. This gives rise to a more efficient identification scheme.

1. บทนำ

ในปัจจุบันนี้เราอาจแบ่งการพัฒนาทฤษฎีระบบควบคุมออกเป็น 2 แนวทาง คือ ทฤษฎีการควบคุมแบบคงทน (Robust control theory) และทฤษฎีการควบคุมแบบปรับตัวเอง (Adaptive control theory) [1],[2] การพัฒนาทฤษฎีในแนวทางแรกนั้นมีจุดประสงค์เพื่อให้ได้มาซึ่งวิธีการออกแบบตัวควบคุมเพื่อให้ระบบมีความคงทน (Robustness) ต่อการเปลี่ยนแปลงของกระบวนการ กล่าวคือ การออกแบบตัวควบคุมจะใช้ข้อมูลจากกระบวนการที่ระบุ (nominal plant) ซึ่งเป็นแบบจำลองทางคณิตศาสตร์ที่ใช้แทนกระบวนการจริง (Actual plant) การออกแบบนี้ต้องการที่จะให้ตัวควบคุมสามารถที่จะควบคุมกระบวนการจริงได้อย่างมีประสิทธิภาพ ถึงแม้ว่ากระบวนการจริงนั้นจะมีความแตกต่างไปจากกระบวนการที่ระบุที่ใช้ในการออกแบบ ส่วนการพัฒนาทฤษฎีในแนวทางหลังนั้นมีจุดประสงค์ในการสร้างอัลกอริทึม (algorithm) ของการเรียนรู้กระบวนการหรือเรียนรู้ระบบเพื่อใช้ในการปรับกลยุทธ์ในการควบคุม (control strategy) อย่างต่อเนื่อง (on-line) ตามการเปลี่ยนแปลงของกระบวนการเพื่อให้สัมฤทธิ์ผลอย่างใดอย่างหนึ่งของการควบคุม

อย่างไรก็ตามได้มีการเสนอแนวความคิดในการรวมเอาแนวทางทั้งสองดังกล่าวนี้มาใช้ในการออกแบบระบบควบคุมเพื่อเสริมประสิทธิภาพของกันและกัน ในแนวความคิดนี้เราจะสมมุติว่าเรารู้ข้อมูลของกระบวนการมากพอเพียงที่จะใช้ในการออกแบบตัวควบคุมแบบคงทน เพื่อทำให้ระบบมีเสถียรภาพก่อนในเบื้องต้น แต่ข้อมูลที่ใช้ในการออกแบบนี้จะมีความแตกต่างไปบ้างจากกระบวนการจริง ดังนั้นเมื่อนำตัวควบคุมนี้ไปควบคุมกระบวนการจริง ก็จะทำให้สมรรถนะของระบบไม่ดีเท่าที่ควรจะเป็น ด้วยเหตุนี้เราจึงเพิ่มส่วนที่ใช้ในการปรับตัวเองเพื่อเสริมสมรรถนะของระบบ แนวความคิดนี้นับว่ามีความสอดคล้องกับการปฏิบัติ กล่าวคือในการออกแบบระบบควบคุมนั้นส่วนใหญ่มักจะรู้ข้อมูลเกี่ยวกับกระบวนการอย่างคร่าวๆ ก่อน แล้วจึงออกแบบตัวควบคุมแบบคงทน เพียงเพื่อให้ระบบมีเสถียรภาพ แต่สมรรถนะอาจจะยังไม่ดีเท่าที่ควร จากนั้นจึงเสริมสมรรถนะด้วยการใช้ส่วนที่ปรับตัวเองเพื่อทำการเรียนรู้ระบบ หลังจากข้อมูลของระบบมากขึ้นเราก็สามารถออกแบบระบบควบคุมให้มีสมรรถนะดีขึ้น ตัวอย่างของการพัฒนาระบบควบคุมตามแนวความคิดนี้ได้เสนอไว้ใน [1] และ [3] และบทความนี้ได้ยึดแนวทางดังกล่าวสำหรับการออกแบบ โดยจะพัฒนาต่อจากวิธีการออกแบบที่ได้เสนอไว้ใน [3] เพื่อมุ่งเน้นในการแก้ปัญหาการควบคุมที่ใช้อัตราการซีกตัวอย่างสัญญาณที่สูง (High sampling rate)

การเลือกอัตราการซีกตัวอย่างสัญญาณ (sampling rate) นับว่าเป็นปัจจัยสำคัญที่ต้องพิจารณาในการออกแบบระบบควบคุมดิจิทัล การใช้อัตราการซีกตัวอย่างสัญญาณที่สูงนั้นเป็นสิ่งที่พึงปรารถนา เพราะจะทำให้สมรรถนะของระบบในด้านการทำจัดผลของสัญญาณรบกวน (disturbance rejection) ดีขึ้น ได้มีการแนะนำแนวทางในการเลือกอัตราการซีกตัวอย่างสัญญาณไว้ใน [5] โดยเสนอให้ใช้อัตราเป็น 20 เท่าของแถบความถี่ (bandwidth) ของกระบวนการ

ปัญหาที่สำคัญของระบบควบคุมดิจิทัลที่ใช้อัตราสุ่มตัวอย่างสัญญาณที่สูงนั้นก็คือ โพล (poles) ของระบบซึ่งประกอบด้วยโพลของกระบวนการและโพลของตัวควบคุมจะมารวมกันอยู่ใกล้ๆ จุด $z=1$ ในระนาบ z (z -Plane) นอกจากนี้ยังเป็นที่ยืนยันทั่วไปว่าอัลกอริทึมของระบบควบคุมแบบปรับตัวเอง จะมีประสิทธิภาพด้อยลงหากโพลของระบบมาอยู่ในบริเวณดังกล่าว

บทความนี้ได้เสนอวิธีการออกแบบระบบควบคุมเพื่อแก้ปัญหาดังกล่าว การแก้ปัญหานี้ใช้หลักการแปลงโคออดิเนต (co-ordinate transformation) กล่าวคือ ฟังก์ชันถ่ายโอน (Transfer function) ของระบบที่มีโพลอยู่บริเวณใกล้ๆ จุด $z=1$ ของโคออดิเนตเดิม จะถูกแปลงมาเป็นฟังก์ชันถ่ายโอน ซึ่งมีโพลอยู่บริเวณใกล้ๆ จุดกำเนิด (origin) ของโคออดิเนตใหม่ และให้อัลกอริทึมของการเรียนรู้ทำงานในโคออดิเนตใหม่ หลักการนี้จะทำให้ประสิทธิภาพการทำงานของอัลกอริทึมดีขึ้น และส่งผลให้เสถียรภาพของระบบในช่วงการปรับตัวดีขึ้นด้วย นอกจากนี้หากเราแทนฟังก์ชันถ่ายโอนของระบบอย่างประมาณในลักษณะอนุกรม การแปลงโคออดิเนตดังกล่าวนี้จะทำให้พารามิเตอร์ของอนุกรมที่ใช้ในอัลกอริทึมมีจำนวนเทอมที่น้อยลงกว่าจำนวนเทอมของพารามิเตอร์ของอนุกรมในโคออดิเนตเดิม ซึ่งเป็นผลให้ประสิทธิภาพของอัลกอริทึมเพิ่มขึ้นไปอีก

2. ทฤษฎีตัวควบคุม (controller theory)

ต่อไปนี้จะได้กล่าวถึงทฤษฎีที่เป็นพื้นฐานในการออกแบบตัวควบคุม รายละเอียดและการพิสูจน์ทฤษฎีได้แสดงไว้ใน [3] และ [5] จึงจะไม่ขอกล่าวซ้ำในที่นี้ แต่จะสรุปเอาเฉพาะส่วนที่จำเป็นต้องนำมาใช้ในงานวิจัยนี้เท่านั้น และจะพยายามหลีกเลี่ยงคณิตศาสตร์ที่ซับซ้อน แต่จะสรุปผลที่ได้จากทฤษฎีและการตีความถึงประโยชน์ที่จะนำเอาไปใช้

ในบทความนี้สัญลักษณ์ R_p คือ class of rational proper transfer functions

และ RH^∞ คือ class of all asymptotically stable rational proper transfer functions

กระบวนการที่ระบุเชิงเวลาต่อเนื่อง, $G_o(s)$ (Nominal continuous time plant)

สมมติให้ $G_o(s)$ เป็นกระบวนการที่ระบุเชิงเวลาต่อเนื่อง ซึ่งเขียนในรูปของตัวแปรสแตต (state variable) ดังนี้

$$\dot{x}(t) = \bar{A}x(t) + \bar{B}u(t), \quad y(t) = \bar{C}x(t) + \bar{D}u(t) \quad (2.1)$$

โดยที่ $G_o(s)$ มีคุณสมบัติคือสามารถทำให้เสถียรได้ (stabilizable) และสามารถสังเกตได้ (observable)

กระบวนการที่ระบุเชิงเวลาดิสครีต, $G_o(z)$ (Nominal discrete time plant)

ให้ $G_o(z)$ เป็นกระบวนการที่ระบุเชิงเวลาดีสครีตสำหรับ $G_o(s)$ ซึ่งได้จากการแปลงสมการ (2.1) ด้วยตัวคงค่าอันดับศูนย์ (zero order hold) และใช้ช่วงชักตัวอย่างสัญญาณ (sampling interval) เป็น T และเขียนเป็นสมการสแตตเชิงเวลาดีสครีตดังนี้

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx(k) + Du(k) \quad (2.2)$$

ซึ่งอาจเขียนให้อยู่ในรูปเมตริกซ์ของฟังก์ชันถ่ายโอน (Transfer function matrix) ตามนิยามดังนี้

$$G_o(z) = C(zI - A)^{-1}B + D = \begin{bmatrix} A & B \\ C & D \end{bmatrix}_T \in R_p \quad (2.3)$$

$[]_T$ คือเมตริกซ์ของฟังก์ชันถ่ายโอนใดๆ ที่มีนิยามตามสมการ (2.3) โดยเขียนในรูปของการแบ่งเป็นบล็อก (block partitioning)

เราจะพิจารณา $G_o(z)$ ที่มีคุณสมบัติที่สามารถแยกเป็นแฟกเตอร์ชนิดโคพราิม (coprime factorizations) ได้ดังนี้

$$G_o = N_o M_o^{-1} = \tilde{M}_o^{-1} \tilde{N}_o \quad ; \quad N_o, M_o, \tilde{N}_o, \tilde{M}_o \in RH^\infty \quad (2.4)$$

ในที่นี้เราเขียน $G_o(z)$ เป็น G_o เพื่อความสะดวก

ตัวควบคุมที่ระบุ, K_o

สมมติให้ K_o เป็นตัวควบคุมที่ระบุ ที่ออกแบบสำหรับควบคุมกระบวนการที่ระบุ, G_o แล้วทำให้ระบบมีเสถียรภาพ ดังที่แสดงในรูปที่ 2.1 โดยที่ K_o มีคุณสมบัติที่สามารถแยกเป็นแฟกเตอร์ชนิดโคพราิมได้เช่นกันดังนี้

$$K_o = U_o V_o^{-1} = \tilde{V}_o^{-1} \tilde{U}_o \quad ; \quad U_o, V_o, \tilde{U}_o, \tilde{V}_o \in RH^\infty \quad (2.5)$$

และเป็นไปตามเงื่อนไขของสมการ Bezout คือ

$$\begin{bmatrix} \tilde{V}_o & -\tilde{U}_o \\ -\tilde{N}_o & \tilde{M}_o \end{bmatrix} \begin{bmatrix} M_o & U_o \\ N_o & V_o \end{bmatrix} = \begin{bmatrix} M_o & U_o \\ N_o & V_o \end{bmatrix} \begin{bmatrix} \tilde{V}_o & -\tilde{U}_o \\ -\tilde{N}_o & \tilde{M}_o \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (2.6)$$

ตัวควบคุม $K(Q)$

สมมติให้ $K(Q)$ เป็นตัวควบคุมใดๆ ที่สามารถเขียนให้อยู่ในรูปของฟังก์ชันถ่ายโอน $Q \in RH^\infty$ โดยใช้สมการ (2.4), (2.5) และ (2.6) ดังนี้

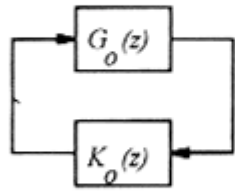
$$\begin{aligned} K(Q) &= UV^{-1}, \quad U = U_o + M_o Q, \quad V = V_o + N_o Q \\ &= \tilde{V}_o^{-1} \tilde{U}, \quad \tilde{U} = \tilde{U}_o + Q \tilde{M}_o, \quad \tilde{V} = \tilde{V}_o + Q \tilde{N}_o \end{aligned} \quad (2.7)$$

ซึ่งเขียนใหม่โดยใช้สมการ (2.6) ได้ดังนี้

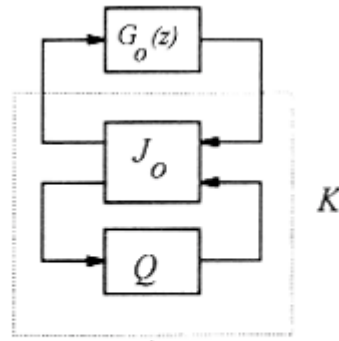
$$K(Q) = K_o + \tilde{V}_o^{-1} Q (I + V_o^{-1} N_o Q)^{-1} V_o^{-1} \quad (2.8)$$

และสามารถจัดรูปใหม่ดังในรูปที่ 2.2 โดยที่

$$J_0 = \begin{bmatrix} K_0 & -V_0^{-1} \\ V_0^{-1} & -V_0^{-1}N_0 \end{bmatrix} \in R_p \quad (2.9)$$



การควบคุมระบบด้วย K_0



การควบคุมระบบด้วย $K(Q)$

สรุปทฤษฎี

ถ้า G_o , K_o , Q และ $K(Q)$ เป็นไปตามเงื่อนไขดังในสมการ (2.4) ถึง (2.7) ดังกล่าวแล้ว และ K_o สามารถทำให้ระบบ G_o มีเสถียรภาพ เราสามารถพิสูจน์ได้ว่า [3]

- 1) ตัวควบคุม $K(Q)$ สามารถทำให้ระบบมีเสถียรภาพได้ ถ้า Q มีเสถียรภาพ และ
- 2) ในกรณีที่ Q เป็นโอเพอร์เรเตอร์ที่เปลี่ยนแปลงตามเวลา (time-varying operator) จะได้ว่าระบบจะเป็น BIBO (Bounded Input Bounded Output) ถ้า Q เป็น BIBO

ทฤษฎีนี้บอกเราว่าถ้าเราสามารถออกแบบตัวควบคุม K_o เพื่อให้ระบบ G_o มีเสถียรภาพได้แล้ว เราก็สามารถออกแบบตัวควบคุม $K(Q)$ ใดๆ ดังในรูปที่ 2.2 เพื่อให้ระบบ G_o มีเสถียรภาพได้เช่นกัน โดยมีเงื่อนไขว่า Q จะต้องมีเสถียรภาพ

จะสังเกตเห็นว่าถ้า $Q = 0$ ระบบในรูปที่ 2.2 จะกลายเป็นระบบในรูปที่ 2.1 ซึ่งมีเสถียรภาพอยู่แล้วตามเงื่อนไขของการออกแบบ K_o การจัดระบบควบคุมให้อยู่ในลักษณะรูปที่ 2.2 นั้น มีข้อดีที่เห็นชัดคือเราจะมีตัวแปรอิสระในการออกแบบตัวควบคุมมากขึ้น นั่นคือเรามีอิสระในการเลือก Q นั้นเอง

ถ้าระบบควบคุมที่มีกระบวนการ G_o ตามสมการ (2.2) และตัวควบคุม K_o เป็นชนิดการป้อนกลับตัวแปรสแตตจากออปเซอเวอร์ ที่มีอัตราขยายของการป้อนกลับ (feedback gain) เป็น F และมีอัตราขยายของออปเซอเวอร์ (observer gain) เป็น H ในที่นี้เราจะได้ [3]

$$K_0 = \left[\frac{A+BF+HC+HDF}{F} \mid \frac{-H}{0} \right]_T \quad (2.10)$$

และ

$$J_0 = \left[\frac{A+BF+HC+HDF}{F} \mid \frac{-H(B+HD)}{0 \quad I} \right. \\ \left. \frac{-(C+DF)}{I \quad -D} \right]_T \in R_p \quad (2.11)$$

การเลือก Q จะมีผลต่อสมรรถนะ (performance) ของระบบ ในที่นี้เราจะใช้ประโยชน์จาก Q ในแง่ที่ว่า เมื่อกระบวนการจริงเปลี่ยนไปจากกระบวนการที่ระบุ จะทำให้การควบคุมด้วย K_0 มีสมรรถนะด้อยลง ดังนั้นเราจะเลือก Q ในลักษณะที่ช่วยเสริม (enhance) สมรรถนะของระบบในกรณีที่กระบวนการจริงมีความแตกต่างไปจากกระบวนการที่ระบุ รายละเอียดจะได้กล่าวในตอนต่อไป

ดัชนีสมรรถนะ (Performance index)

ระบบควบคุมในทางปฏิบัตินั้นมักจะมีสัญญาณรบกวน (disturbances) จากภายนอกเข้ามากระทำต่อระบบ เช่น การแปรปรวนของกระแสอากาศที่กระทำต่อเครื่องบิน หรือกระแสคลื่นที่มาปะทะเรือเดินสมุทร ซึ่งเป็นผลให้วิถีการเดินทางของเครื่องบินหรือเรือเดินสมุทรเปลี่ยนแปลงไปเป็นต้น ระบบควบคุมที่ดีนั้นจะต้องสามารถกำจัดหรือลดผลกระทบต่อระบบอันเกิดจากสัญญาณรบกวนดังกล่าว ดังนั้นในที่นี้เราจะพิจารณาสมรรถนะของการควบคุมในด้านของความสามารถในการกำจัดผลของสัญญาณรบกวน (disturbance rejection)

สมมติให้ w_k เป็นสัญญาณรบกวนจากภายนอกที่กระทำต่อระบบ และ e_k เป็นผลตอบสนองอันเกิดจากสัญญาณรบกวน ซึ่ง e_k นี้อาจจะเป็นสัญญาณเอาต์พุตของระบบ, สัญญาณอินพุตของระบบ, ตัวแปรสแตตที่ต้องการควบคุม หรือการผสมของสัญญาณทั้งหมดนี้ก็ได้ เราจะสมมติให้ความสัมพันธ์ระหว่างสัญญาณเหล่านี้ในโดเมนความถี่เป็นดังนี้

$$\begin{bmatrix} e(z) \\ y(z) \end{bmatrix} = \begin{bmatrix} P_{11}(z) & P_{12}(z) \\ P_{21}(z) & P_{22}(z) \end{bmatrix} \begin{bmatrix} w(z) \\ u(z) \end{bmatrix} \quad (2.12)$$

โดยที่ $P_{11}(z), P_{12}(z), P_{21}(z)$ และ $P_{22}(z) = G(z)$ จะขึ้นอยู่กับกระบวนการและการเลือก $e(z)$

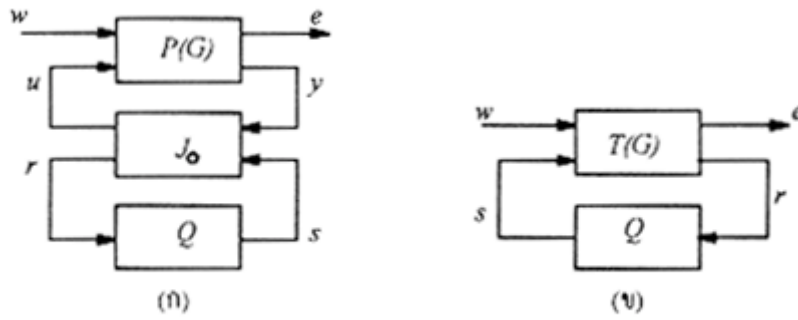
สิ่งที่เราสนใจในตอนนี้เป็น การหาความสัมพันธ์ระหว่าง $w(z)$ และ $e(z)$ ที่เขียนอยู่ในเทอมของ Q ซึ่งทำให้รู้ว่าการเปลี่ยนแปลงของ Q จะมีผลต่อ $e(z)$ อย่างไรบ้าง

ในกรณีที่ตัวควบคุม $K(Q)$ ทำการควบคุมกระบวนการจริง G เราสามารถเขียนความสัมพันธ์ของ $e(z)$ และ $w(z)$ ดังที่แสดงในรูปที่ 2.3 โดยที่

$$T(G) = \begin{bmatrix} P_{11} + P_{12}K_0(I - GK_0)^{-1}P_{21} & P_{12}(I - GK_0)^{-1}\tilde{V}_0^{-1} \\ V_0^{-1}(I - GK_0)^{-1}P_{21} & V_0^{-1}(I - GK_0)^{-1}G\tilde{V}_0^{-1} - V_0^{-1}N_0 \end{bmatrix} \quad (2.13)$$

และ
$$e(z) = P_{11}w(z) + (P_{12}U_0 + P_{21}M_0Q)r(z) = P_{11}w(z) + P_{12}U_0r(z) + P_{21}M_0s(z)$$

โดยที่
$$r(z) = \tilde{M}_0y(z) - \tilde{N}_0u(z) \quad \text{และ} \quad s(z) = Qr(z) \quad (2.14)$$



รูปที่ 2.3 การกำจัดผลของสัญญาณรบกวนด้วยตัวควบคุม Q

จากสมการ (2.14) เราจะเห็นว่า การปรับ Q จะมีผลต่อ $e(z)$ และเพื่อให้สัญญาณรบกวนมีผลต่อระบบน้อยที่สุด เราจะนิยามดรรชนีสมรรถนะ ดังนี้

$$I = \frac{1}{k} \sum_1^k e_k^l e_k \quad , \quad k \rightarrow \infty \quad (2.15)$$

และเราจะหาวิธีการปรับ Q เพื่อให้ค่าเฉลี่ย (expectation) ของ $e(z)$ (หรือ ค่า I) มีค่าน้อยที่สุด

จากกระบวนการตามสมการ (2.2) หากเราพิจารณาในกรณีที่มีสัญญาณรบกวน w_k ซึ่งเป็นชนิดสโตคาสติก (stochastic) มากจะทำต่อระบบดังนี้

$$x_{k+1} = Ax_k + Bu_k + \Gamma w_k \quad , \quad y_k = Cx_k \quad (2.16)$$

และถ้าเราพิจารณาดรรชนีสมรรถนะของการออกแบบด้วยวิธี LQG (Linear Quadratic Gaussian) คือ

$$I_{LQG} = \frac{1}{k} \sum_1^k (u_{i-1}^l R_c u_{i-1} + x_i^l Q_c x_i) \quad , \quad R_c > 0, Q_c \geq 0 \quad , \quad k \rightarrow \infty \quad (2.17)$$

ในกรณีนี้หากเราเลือก e_k ในสมการ (2.15) เป็นดังนี้

$$e_k = \begin{bmatrix} R_c^{1/2} u_{k-1} \\ Q_c^{1/2} x_k \end{bmatrix} \quad (2.18)$$

เราจะเห็นได้ชัดเจนว่า $I_{LQG} = I$ นั่นคือ การออกแบบตัวควบคุมด้วยวิธี LQG จะพ้องกับการทำให้ I ในสมการ (2.15) มีค่าน้อยที่สุด ถ้าเราเลือก e_k ตามสมการ (2.18) ดังนั้นเราจึงเลือกโครงสร้างของระบบควบคุมเป็นชนิดป้อนกลับตัวแปรสแตตจากออปเซอเวอ์เป็นพื้นฐานในการพัฒนาอัลกอริทึมของการปรับ Q โดยมี e_k ตามสมการ (2.18) และ F และ H ได้มาจากการออกแบบด้วยวิธี LQG ซึ่งเป็นวิธีการมาตรฐาน

ทั้งหมดที่ได้กล่าวมานี้เป็นทฤษฎีที่จะนำมาใช้ในการพัฒนาอัลกอริทึมของการปรับ Q เพื่อให้ดรรชนีสมรรถนะในสมการ (2.15) มีค่าน้อยที่สุด

3. การแปลงโคออดิเนตสำหรับ Q (co-ordinate transformation for Q)

การแปลงโคออดิเนตสำหรับ Q เป็นแนวความคิดที่บทความนี้นำเสนอ เพื่อเป็นแนวทางหนึ่งในการแก้ปัญหาของการออกแบบระบบควบคุมแบบปรับตัวเอง แต่ก่อนที่จะกล่าวถึงแนวทางดังกล่าวนี้ เราจะมาศึกษาถึงเงื่อนไขและปัญหาต่างๆ เป็นลำดับดังต่อไปนี้

เงื่อนไขที่ 1: การเลือกอัตราการชั่งตัวอย่างสัญญาณ (selection of sampling rate)

เพื่อให้ระบบสามารถกำจัดผลของสัญญาณรบกวนได้ดี เราจะพิจารณาเลือกอัตราการชั่งตัวอย่างสัญญาณที่สูง [5] เป็นประมาณ 20 เท่าของแถบความถี่ (bandwidth) ของกระบวนการ ด้วยการใช้อัตราขนาดนี้เราจะพบว่า ตำแหน่งโพลของระบบเชิงเวลาดิสครีตจะมารวมกันอยู่ภายในวงกลมที่มีรัศมี 0.1 หน่วยและมีจุดศูนย์กลางที่ $(1+j0)$ ในระนาบ z และหากเราเลือกการชั่งตัวอย่างสัญญาณเป็นอนันต์ โพลทุกๆ ตัวจะเข้ามาอยู่ที่จุด $z=1$

เงื่อนไขที่ 2: การออกแบบตัวควบคุม (Controller design)

ในทางทฤษฎีนั้นเราสามารถที่จะออกแบบตัวควบคุมโดยการวางโพลของระบบ (closed-loop poles) ให้อยู่ในตำแหน่งใดก็ได้ถ้ากระบวนการนั้นสามารถควบคุมได้ (controllable) และสามารถสังเกตได้ (observable) แต่ในทางปฏิบัตินั้นเราจำเป็นต้องพิจารณาข้อจำกัดอื่นๆ ของระบบด้วย เช่นขนาดของสัญญาณควบคุมและตัวขับ (actuator) เป็นต้น เพราะถ้าเราออกแบบให้โพลของระบบอยู่ห่างจากโพลของกระบวนการมาก การตอบสนองของระบบอาจจะเร็วเกินไป เป็นเหตุให้ต้องใช้ขนาดของสัญญาณควบคุมสูงเกินขีดจำกัด และเป็นเหตุให้ตัวขับอิมิตัว ซึ่งอาจจะทำให้ระบบไร้เสถียรภาพในที่สุด ดังนั้นเราจึงต้องเลือกตำแหน่งโพลของระบบให้ห่างจากโพลของกระบวนการพอสมควร ดังนั้นเงื่อนไขนี้คือ ตัวควบคุมจะถูกออกแบบให้สัญญาณควบคุมถูกจำกัดให้ทำงานในช่วงการทำงานปกติของตัวขับ (Actuator) ที่ใช้ในระบบจริง

จากเงื่อนไขทั้งสองนี้เราก็สรุปได้ว่าขณะนี้เรากำลังออกแบบระบบควบคุมที่มีโพลของตัวควบคุมและโพลของกระบวนการอยู่ในบริเวณเดียวกัน และอยู่ใกล้จุด $z=1$ บนระนาบ z

ปัญหาในการหาพารามิเตอร์ของ Q

ในอัลกอริทึมของการปรับ Q นั้นจะกระทำโดยการแทน Q ในรูปของพารามิเตอร์ของฟังก์ชันถ่ายโอน จากนั้นจะใช้การประมาณค่าแบบกำลังสองน้อยสุด (Recursive Least Squares) หรือ RLS ในการหาค่าพารามิเตอร์ดังกล่าวเพื่อให้ตรรกะนี้สมรรถนะสมการ (2.15) มีค่าน้อยที่สุด เป็นที่รู้กันทั่วไปว่าอัลกอริทึม RLS จะทำงานได้ดีและแม่นยำก็ต่อเมื่อระบบมีโพลอยู่ใกล้จุดกำเนิดของระนาบ z หรือระบบที่ใช้อัตราการชั่งตัวอย่างที่ต่ำมากๆ นั่นเอง แต่ถ้าระบบเป็นไปตามเงื่อนไขที่ 1 และ 2 นั่นคือมีโพลอยู่ใกล้จุด $z=1$ จะทำให้อัลกอริทึม RLS ทำงานอย่างไม่มีประสิทธิภาพ เช่นการลู่เข้าของพารามิเตอร์เป็นไปอย่าง

เชิงซ้ำ ความแม่นยำเชิงตัวเลขน้อยลง เป็นต้น นี่เป็นปัญหาสำหรับการหาพารามิเตอร์ของระบบที่ใช้อัตรา
การชักตัวอย่างสัญญาณที่สูง

การแก้ปัญหาด้วยวิธีการแปลงโคออดิเนตของ Q

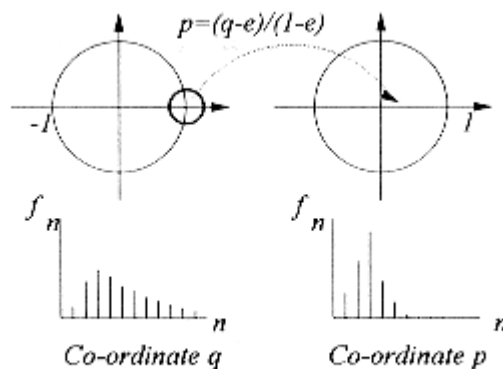
ถ้าให้ q เป็นสัญญาณลักษณะ แทนโอเปอเรเตอร์ชนิดเลื่อนไปข้างหน้า (forward shift operator)

และ q^{-1} เป็นสัญญาณลักษณะ แทนโอเปอเรเตอร์ชนิดเลื่อนไปข้างหลัง (backward shift operator)

เราจะให้นิยามของการแปลงโคออดิเนตจากโอเปอเรเตอร์ q มาเป็นโอเปอเรเตอร์ p ดังนี้

$$p = \frac{(q - e)}{(1 - e)}, \quad 0 < e < 1 \quad (3.1)$$

การแปลงในสมการ (3.1) คือการแมพ (mapping) พื้นที่ของวงกลมที่มีจุดศูนย์กลางที่ e และรัศมี $(1 - e)$
ไปเป็นวงกลมรัศมีหนึ่งหน่วย (unit circle) ที่มีจุดศูนย์กลางอยู่ที่จุดกำเนิดของโคออดิเนตชุดใหม่ ดังนั้น
ฟังก์ชันถ่ายโอนที่เขียนในเทอมของ q และมีโพลอยู่บริเวณใกล้ๆ $q = e$ เมื่อทำการแปลงโดยใช้สมการ
(3.1) แล้วจะกลายเป็นฟังก์ชันถ่ายโอนในเทอมของ p ที่มีโพลอยู่บริเวณใกล้ๆ $p = 0$ ดังที่ได้แสดงใน
รูปที่ 3.1



รูปที่ 3.1 การแปลงโคออดิเนตจาก q เป็น p ตามสมการ(3.1)

หากเราพิจารณาฟังก์ชันถ่ายโอนที่เขียนในเทอมของ q^{-1} ดังนี้

$$Q(q)^{-1} = \frac{b_0 + b_1 q^{-1} + \dots + b_n q^{-n}}{1 + a_1 q^{-1} + \dots + a_n q^{-n}} \quad (3.2)$$

หลังการแปลงโดยใช้สมการ (3.1) แล้วเราจะได้ฟังก์ชันถ่ายโอนในเทอมของ p^{-1} ดังนี้

$$Q(p)^{-1} = \frac{c_0 + c_1 p^{-1} + \dots + c_n p^{-n}}{1 + d_1 p^{-1} + \dots + d_n p^{-n}} \quad (3.3)$$

เราจะสังเกตเห็นได้ว่าการแปลงด้วยสมการ (3.1) จะไม่ทำให้อันดับ (order) ของระบบเปลี่ยนแปลงไป

การแทน Q ในลักษณะของอนุกรม

เป็นที่รู้กันทั่วไปว่าอัลกอริทึม RLS จะทำการประมาณค่าพารามิเตอร์ของแบบจำลองคณิตศาสตร์
ได้อย่างมีประสิทธิภาพดียิ่งขึ้น ถ้าแบบจำลองนั้นเป็นชนิด Moving Average หรือ MA ซึ่งมีรูปแบบเป็น
อนุกรมดังนี้

$$Q(q^{-1}) = e_{0+}e_1q^{-1} + e_2q^{-2} + \dots e_nq^{-n} \quad (3.4)$$

และจะเห็นได้ว่า Q จะเป็นฟังก์ชันของอินพุตเพียงอย่างเดียว

ถึงแม้ว่าฟังก์ชันถ่ายโอนของระบบในทางปฏิบัตินั้นจะอยู่ในรูปแบบตามสมการ (3.2) แต่บางครั้งเราก็อาจจะประมาณให้อยู่ในรูปของอนุกรมในสมการ (3.4) ได้โดยการหารยาวสมการ (3.2) ซึ่งเราจะได้ออกมาในรูปของอนุกรมอนันต์ดังนี้

$$Q(q^{-1}) = f_{0+}f_1q^{-1} + f_2q^{-2} + \dots f_nq^{-n} + f_{n+1}q^{-(n+1)} \quad (3.5)$$

ค่าสัมประสิทธิ์ f_n จะสัมพันธ์กับผลตอบสนองต่อสัญญาณอิมพัลส์ของฟังก์ชันถ่ายโอน Q และถ้า Q มีเสถียรภาพ เราจะพบว่าสัมประสิทธิ์ f_n จะมีค่าน้อยมากเมื่อ n มีค่ามากขึ้น และจะลู่เข้าสู่ศูนย์ เมื่อ n เข้าใกล้อนันต์ ดังนั้นหากเราตัดเทอม f_n ที่มีค่าน้อยๆ ออกไป เราก็สามารถจะประมาณ $Q(q^{-1})$ ในสมการ (3.2) ให้อยู่ในรูปของอนุกรมดังสมการ (3.4)

การที่เราจะพิจารณาเลือกว่าต้องใช้ f_n จำนวนกี่เทอมถึงจะพอเพียงนั้น ขึ้นอยู่กับตำแหน่งโพลของฟังก์ชันถ่ายโอน ถ้าโพลอยู่บริเวณใกล้กับจุด $q=1$, f_n จะลู่เข้าสู่ศูนย์ได้ช้า กรณีนี้จึงต้องใช้ f_n จำนวนมากเทอม แต่ถ้าโพลอยู่ใกล้กับจุด $q=0$, f_n จะลู่เข้าสู่ศูนย์ด้วยความเร็วและในกรณีนี้เราอาจจะใช้ f_n เพียงไม่กี่เทอมก็พอเพียงสำหรับการแทน Q อย่างประมาณ

ในระบบควบคุมที่เรากำลังพิจารณาอยู่นี้ การออกแบบระบบจะต้องเป็นไปตามเงื่อนไขที่ 1 และ 2 โพลของ $Q(q^{-1})$ จะมาอยู่ใกล้บริเวณจุด $q=1$ ดังนั้นหากเราประมาณ $Q(q^{-1})$ ด้วยอนุกรมแล้วเราจะต้องใช้จำนวนเทอมของ f_n มาก แต่ถ้าเราแปลง $Q(q^{-1})$ มาเป็น $Q(p^{-1})$ ด้วยสมการ (3.1) โดยเลือก e ให้เหมาะสมแล้ว เราจะได้โพลของ $Q(p^{-1})$ มาอยู่ใกล้กับจุด $p=0$ ในกรณีนี้การประมาณค่า $Q(p^{-1})$ ในรูปอนุกรมจะใช้สัมประสิทธิ์ f_n ในจำนวนเทอมที่น้อยกว่า ดังที่จะได้แสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 3.1 เราจะพิจารณาฟังก์ชันถ่ายโอนที่เขียนในโอเปอเรเตอร์ q ดังนี้

$$Q(q)^{-1} = \frac{q^{-1} - 0.8q^{-2}}{1 - 1.85q^{-1} + 0.853q^{-2}}$$

ในกรณีนี้ $Q(q)^{-1}$ มีโพลอยู่ที่ $q=0.9$ และ $q=0.95$ และเมื่อเขียนเป็นอนุกรมแล้วจะได้ f_n ดังที่แสดงในรูปที่ 3.1

หากเราแปลง $Q(q)^{-1}$ ให้อยู่ในรูปของโอเปอเรเตอร์ p โดยสมการ (3.1) และเลือกให้ $e=0.9$ เราจะได้

$$Q(p)^{-1} = \frac{10p^{-1} + 1p^{-2}}{1 - 0.5p^{-1}}$$

ในกรณีนี้ $Q(q)^{-1}$ จะมีโพลอยู่ที่ $p=0$ และ $p=0.5$ และมีอนุกรมของ f_n ดังที่แสดงในรูปที่ 3.1 จะสังเกตเห็นว่า f_n จะลดค่าลงอย่างรวดเร็ว ดังนั้นเราจึงสามารถใช้ f_n ด้วยจำนวนเทอมที่น้อยกว่าในกรณีแรก

โดยสรุปแล้วในกรณีที่ระบบใช้อัตราการซัดตัวอย่างสัญญาณที่สูง การแปลง $Q(q)^{-1}$ ให้มาอยู่ในโอเปอเรเตอร์ p นั้น จะทำให้การหาพารามิเตอร์ของ $Q(q)^{-1}$ ด้วยวิธี RLS มีความแม่นยำมากยิ่งขึ้น นอกจากนี้หากเราแทน $Q(q)^{-1}$ อย่างประมาณในรูปของอนุกรม เราก็สามารถใช้จำนวนเทอมของ พารามิเตอร์น้อยกว่า ซึ่งเป็นผลให้การหาพารามิเตอร์มีประสิทธิภาพมากยิ่งขึ้นไปอีก

4. อัลกอริทึมของการปรับ Q

ขั้นตอนการสร้างอัลกอริทึมของการปรับ Q จะได้แสดงเป็นลำดับดังต่อไปนี้

รูปของการป้อนกลับ (feedback loop)

เราจะเริ่มด้วยการออกแบบตัวควบคุม K_0 จากนั้นสร้าง J_0 ดังในสมการ (2.9) โดยการแยกแพกเตอร์ชนิดโคพราสมของ G_0 และ K_0 ในสมการ (2.4) และ (2.5) ถ้า K_0 เป็นตัวควบคุมชนิดที่ใช้การป้อนกลับตัวแปรสแตตจากออปเซอเวอร์จะทำให้การสร้าง J_0 นั้นง่ายขึ้น [3] และ J_0 ที่ได้นี้จะไม่เพิ่มไดนามิกส์ให้กับระบบ สัญญาณที่เกี่ยวข้องกับ J_0 ที่เพิ่มขึ้นมาคือ s_k และ r_k ดังนั้นจากสมการ (2.9) เราจะได้

$$\begin{bmatrix} u(z) \\ r(z) \end{bmatrix} = J_0 \begin{bmatrix} y(z) \\ s(z) \end{bmatrix} \quad (4.1)$$

ขออภัยสำหรับ G_0 และ K_0 ชุดหนึ่งอาจจะมี J_0 ได้หลายชุดทั้งนี้ขึ้นอยู่กับ การแยกแพกเตอร์ของ G_0 และ K_0 ซึ่งอาจแยกได้หลายแบบ

รูปของการปรับ Q

สมมติว่าเราได้ออกแบบระบบให้เป็นไปตามเงื่อนไขที่ 1 และ 2 ดังนั้นเราจะพิจารณาฟังก์ชันถ่ายโอน Q ที่เขียนเป็นพารามิเตอร์ในรูปของโอเปอเรเตอร์ p ในกรณีทั่วไปดังนี้

$$(1 + d_{1k}p^{-1} + d_{2k}p^{-2} + \dots + d_{nk}p^{-n})s_k = (e_{0k} + e_{1k}p^{-1} + e_{2k}p^{-2} + \dots + e_{mk}p^{-m})r_k \quad (4.2)$$

และเราจะนิยามให้

$$\hat{\theta}_k = [d_{1k} d_{2k} \dots d_{nk} \quad c_{0k} c_{1k} \dots c_{mk}]$$

ซึ่งเป็นพารามิเตอร์ของ $Q(p)^{-1}$ ณ เวลา k และในกรณีที่แทน $Q(p)^{-1}$ ในรูปของอนุกรม เราจะได้

$$s_k = (f_{0k} + f_{1k}p^{-1} + \dots + f_{mk}p^{-m})r_k \quad (4.3)$$

ในกรณีนี้เราจะนิยามพารามิเตอร์ของ $Q(p)^{-1}$ ณ เวลา k ดังนี้

$$\hat{\theta}_k = [f_{0k} f_{1k} f_{2k} \dots f_{mk}]$$

เราจะเห็นว่า $\hat{\theta}_k$ จะมีค่าแปรตามเวลา (time-varying)

การเขียน e_k ในเทอมของ $\hat{\theta}_k$

ก่อนที่จะกล่าวถึงทฤษฎี เราจะกำหนดสัญญาณ ζ และ ξ ซึ่งสร้างขึ้นดังนี้

$$\zeta_{k-i} = e_{k-i} - P_{12}M_0p^{-i}s_k, i = 0,1,2,3,\dots \quad (4.4)$$

และ
$$\xi_{k-i} = P_{12}M_0p^{-i}r_k, i = 0,1,2,3,\dots \quad (4.5)$$

ในที่นี้เราจะพิจารณา $P_{12}M_0$ ว่าเป็นตัวกรองสัญญาณ (filter) ของ $p^i s_k$ และ $p^i r_k$ โดยที่

$$s_k = Q(p^{-1})r_k$$

ทฤษฎีบทที่ 1: จากรูปที่ 2.3 และภายใต้เงื่อนไขของสมการ (2.12) และ (2.14) โดยที่ Q ถูกแทนด้วย $Q_k(p^{-1})$ ซึ่งแปรค่าตามเวลา เราจะได้ว่า

$$e_k = P_{11}w_k + [P_{12}U_0 + P_{12}M_0Q_k(p^{-1})]r_k \quad (4.6)$$

และถ้าให้ $Q_{k/\theta}$ คือ Q ที่เวลา k ที่เขียนในรูปของพารามิเตอร์ $\hat{\theta}_k$ เราจะได้

$$\begin{aligned} e_{k/\theta} &= P_{11}w_k + [P_{12}U_0 + P_{12}M_0Q_{k/\theta}(p^{-1})]r_k \\ &= \xi_k + P_{12}M_0Q_{k/\theta}(p^{-1})r_k \end{aligned} \quad (4.7)$$

นอกจากนี้ ถ้า $Q_{k/\theta}$ เป็นสเกลาร์ (scalar) สมการ (4.7) สามารถเขียนใหม่ได้ดังนี้

$$\text{โดยที่} \quad \xi_k = \varphi'_k \theta + e_{k/\theta} \quad (4.8)$$

$$\varphi'_k = [(e_{k-1} - \zeta_{k-1}) \dots (e_{k-n} - \zeta_{k-n}) \\ - \zeta_k - \zeta_{k-1} \dots - \zeta_{k-m}]$$

ทฤษฎีบทที่ 1 นี้จะคล้ายคลึงกับทฤษฎีที่ได้แสดงไว้ใน [3] ความแตกต่างอยู่ที่การสร้างสัญญาณ ζ_k และ ξ_k ในสมการ (4.4) และ (4.5) และการแปลงโอเปอเรเตอร์จาก q มาเป็น p ในฟังก์ชันถ่ายโอน Q ในสมการ (4.6) และ (4.7) ดังนั้นจึงไม่ขอแสดงรายละเอียดของการพิสูจน์ในที่นี้

ทฤษฎีบทที่ 1 นี้เป็นพื้นฐานสำคัญของการสร้างอัลกอริทึมในการปรับ Q เราจะเห็นจากสมการ (4.8) ว่า $e_{k/\theta}$ สามารถเขียนให้อยู่ในเทอมของ $\hat{\theta}$ ซึ่งเป็นพารามิเตอร์ของ Q และรูปแบบของสมการ (4.8) เอื้ออำนวยต่อการใช้วิธี RLS เพื่อประมาณค่า $\hat{\theta}$

การหา $\hat{\theta}$ ด้วยวิธี RLS

สุดท้ายนี้เราจะหาพารามิเตอร์ $\hat{\theta}$ โดยการใช้วิธี RLS ในลักษณะที่ทำให้บรรจบกันสมรรถนะในสมการ (2.15) มีค่าน้อยที่สุด นั่นคือ

$$\min_{\text{stabilizing } \theta} \frac{1}{k} \sum_1^k e_{k/\theta}^2, \quad k \rightarrow \infty$$

สมการ (4.8) อยู่ในรูปแบบมาตรฐานที่สามารถใช้วิธี RLS ได้โดยตรง และเนื่องจาก RLS เป็นวิธีการมาตรฐานซึ่งมีกล่าวไว้หลายแห่ง [6] จึงไม่ขอกล่าวรายละเอียดในที่นี้ แต่จะยกเอาสมการต่างๆ ที่เกี่ยวข้องในอัลกอริทึมของการหาพารามิเตอร์ Q ด้วยวิธี RLS ดังต่อไปนี้

$$\begin{aligned} \hat{\theta}_k &= \hat{\theta}_{k-1} + P_k \varphi_k \hat{e}_{k/k-1}, \quad \hat{e}_k = \zeta_k - \varphi'_k \hat{\theta}_{k-1} \\ \hat{P}_k &= \left(\sum_1^k \varphi_i \varphi_i' \right) \\ &= \hat{P}_{k-1} - \hat{P}_{k-1} \varphi_k [I + \varphi_k \hat{P}_{k-1} \varphi_k']^{-1} \varphi_k \hat{P}_{k-1} \end{aligned}$$

$$\text{โดยที่} \quad \varphi'_k = [(e_{k-1} - \zeta_{k-1}) \dots (e_{k-n} - \zeta_{k-n}) \\ - \zeta_k - \zeta_{k-1} \dots - \zeta_{k-m}]$$

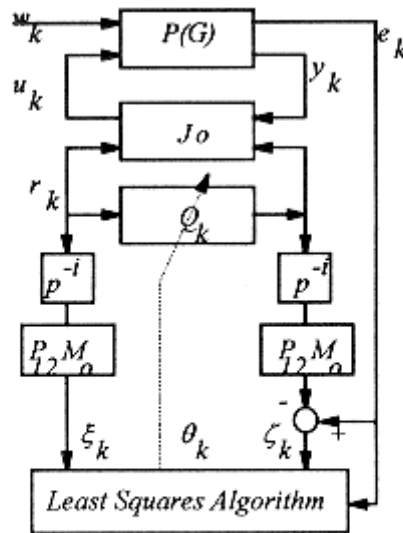
และถ้าเราแทน Q ด้วยอนุกรมที่มีพารามิเตอร์ m เทอม จะได้

$$\text{โดยที่} \quad \varphi'_k = [-\xi_k - \xi_{k-1} \dots - \xi_{k-m}]$$

$$\xi_i = P_{12} M_0 p^{-i} r_k$$

ในกรณีนี้เราจะเห็นว่า φ_k เป็นฟังก์ชันของ r_k เท่านั้น ผลดีที่ตามมาก็คือการทำงานของอัลกอริทึม RLS จะมีประสิทธิภาพดีขึ้น

จากขั้นตอนที่ได้กล่าวมาทั้งหมดนี้ เราสามารถเขียนโครงสร้างของระบบได้ดังในรูปที่ 4.1



รูปที่ 4.1 โครงสร้างของระบบที่มีการปรับ Q

การเลือก P_{12}

ในอัลกอริทึมของการปรับ Q นั้น เราจำเป็นต้องใช้ P_{12} แต่เนื่องจาก G เป็นกระบวนการจริงที่เราไม่รู้พารามิเตอร์ ดังนั้นเราจึงสามารถเลือก P_{12} ได้อย่างอิสระโดยที่ไม่ขึ้นกับ G ใดๆก็ตามการเลือก P_{12} ก็จะมีผลต่อสมรรถนะของระบบเช่นกัน วิธีการเลือก P_{12} ได้กล่าวไว้อย่างละเอียดใน [3] ในงานวิจัยนี้ เราเลือก $P_{12} = M_0^{-1}$ การเลือกเช่นนี้จะทำให้ตัวกรองสัญญาณ $P_{12} M_0$ กลายมาเป็นเพียงตัวขยายสัญญาณที่มีอัตราขยายเป็นหนึ่ง ซึ่งทำให้ง่ายในการสร้างอัลกอริทึม

5. ผลการทดลอง

ต่อไปนี้จะได้แสดงผลการทดสอบสมรรถนะของอัลกอริทึมที่ได้เสนอในบทความนี้ โดยจะเปรียบเทียบผลที่ได้จากอัลกอริทึมที่ได้เสนอไว้เดิมใน [3]

เพื่อให้การเปรียบเทียบมีผลสรุปที่เด่นชัด เราจึงสร้างเงื่อนไขของการทดสอบให้เป็นอุดมคติมากที่สุด นั่นคือใช้กระบวนการที่เป็นเชิงเส้น (linear plant) ซึ่งเขียนเป็นโปรแกรมอยู่ในคอมพิวเตอร์ และมีการ

สร้างสัญญาณรบกวนที่มากกระทำต่อระบบเป็นสัญญาณรบกวนขาว (white noise) ที่มีค่าเฉลี่ยเป็นศูนย์ (zero mean) และมีวาเรียนซ์ (variance) เป็นหนึ่ง

กระบวนการที่ระบุ ที่ใช้ในการทดสอบ มีฟังก์ชันถ่ายโอนดังนี้

$$G_0(s) = \frac{1}{s^2 - 0.2s + 100.01}$$

ซึ่งเป็นระบบที่ไม่มีเสถียรภาพ โดยมีโพลอยู่ที่ $0.1 \pm j10$ ในระนาบ S

เมื่อทำการแปลง $G_0(S)$ ให้เป็นกระบวนการที่ระบุเชิงดิคริต โดยใช้ช่วงเวลาชักตัวอย่างสัญญาณ (sampling interval) เป็น 0.01 วินาที จะได้กระบวนการในรูปของตัวแปรสแตตดิงในสมการ (2.2) โดยที่

$$A_0 = \begin{bmatrix} 0.9970 & -0.9994 \\ 0.0100 & 0.9950 \end{bmatrix}, B_0 = \begin{bmatrix} 0.01 \\ 0.00 \end{bmatrix}, C_0 = [0 \ 1]$$

ซึ่งมีโพลอยู่ที่ $0.996 \pm j0.0999$ ในระนาบ Z จะเห็นว่าโพลจะเข้ามาอยู่ใกล้จุด $z=1$

สมมติว่าสัญญาณรบกวน w_k ได้กระทำกับระบบดังที่แสดงในสมการ (2.16)

โดยที่ $\Gamma = [0.1 \ 0.1]$

เราได้ออกแบบ K_0 ในสมการ (2.10) โดยใช้วิธี LQG ซึ่งมีดรรชนีสมรรถนะ ดังในสมการ (2.17)

โดยที่

$R_c = I$ และ $Q = 1$ (ใช้ Control toolbox ใน Matlab ช่วยในการออกแบบ)

กระบวนการจริงจะแตกต่างไปจากกระบวนการที่ระบุ ในการทดสอบนี้เราจะพิจารณากระบวนการจริง 2 กระบวนการดังนี้

กระบวนการจริง 1

$$A_1 = \begin{bmatrix} 0.8 & -0.8 \\ 0.01 & 0.995 \end{bmatrix}, B_1 = \begin{bmatrix} 0.01 \\ 0.0001 \end{bmatrix}, C_1 = [0 \ 1]$$

มีโพลอยู่ที่ $z = 0.8587$ และ $z = 0.9363$ ซึ่งมีเสถียรภาพ

กระบวนการจริง 2

$$A_2 = \begin{bmatrix} 1.0 & -1.0022 \\ 0.0100 & 0.9950 \end{bmatrix}, B_2 = \begin{bmatrix} 0.01 \\ 0.0001 \end{bmatrix}, C_2 = [0 \ 1]$$

มีโพลอยู่ที่ $z = 0.9975 + j0.1001$ ซึ่งไม่มีเสถียรภาพ

ในการทดสอบทั้งสองอัลกอริทึมนี้ เราเลือก Q เป็นลักษณะอนุกรม โดยใช้พารามิเตอร์ 6 เทอม สำหรับอัลกอริทึม $Q(p^{-1})$ และใช้ 10 เทอมสำหรับอัลกอริทึม $Q(q^{-1})$ ที่เสนอใน [3] และใช้ $e=0.9$ สำหรับการแปลงโคออดิเนตในสมการ (3.1) การทดสอบสมรรถนะของแต่ละอัลกอริทึมได้กระทำในช่วงเวลา 200 วินาที หรือ 20,000 ช่วงการชักตัวอย่างสัญญาณและได้กระทำ 10 ครั้ง โดยแต่ละครั้งจะใช้สัญญาณรบกวนที่แตกต่างกันไป ค่าดรรชนีสมรรถนะในแต่ละครั้งที่ทดสอบได้คำนวณจากสมการ (2.15)

โดยพิจารณาในช่วง 50 วินาทีสุดท้าย นั่นคือพิจารณาหลังจากที่ระบบอยู่ในสภาวะอยู่ตัวแล้ว จากนั้นจึงนำเอาค่านี้ออกมาหารค่าเฉลี่ย ดังผลที่ได้แสดงในตารางที่ 1

ตารางที่ 1 : ค่าตรรกะนิสมรรถนะเฉลี่ย

	กระบวนการที่ระบุ	กระบวนการจริง 1	กระบวนการจริง 2
ตัวควบคุม K_0 เพียงอย่างเดียว	24.16	35.38	ไร้เสถียรภาพ
ตัวควบคุม K_0 พร้อมการปรับ $Q(q^{-1})$ ใช้พารามิเตอร์ 10 เทอม	24.17	14.52	4132
ตัวควบคุม K_0 พร้อมการปรับ $Q(p^{-1})$ ใช้พารามิเตอร์ 6 เทอม	24.16	10.24	2104
ตัวควบคุมที่เหมาะสมที่สุด (optimal control)	24.16	2.15	1214

บรรทัดสุดท้ายของตารางได้แสดงค่าตรรกะนิสมรรถนะเฉลี่ยของระบบควบคุมที่ออกแบบให้เหมาะสมที่สุดสำหรับแต่ละกระบวนการ ดังนั้นค่าที่แสดงไว้จึงเป็นค่าที่น้อยที่สุดของแต่ละกระบวนการ

จากตารางที่ 1 ถึงแม้ว่าตัวควบคุม K_0 จะสามารถควบคุมกระบวนการจริง 1 ได้ แต่สมรรถนะลดลง และสำหรับกระบวนการจริง 2 นั้น K_0 ไม่สามารถควบคุมได้

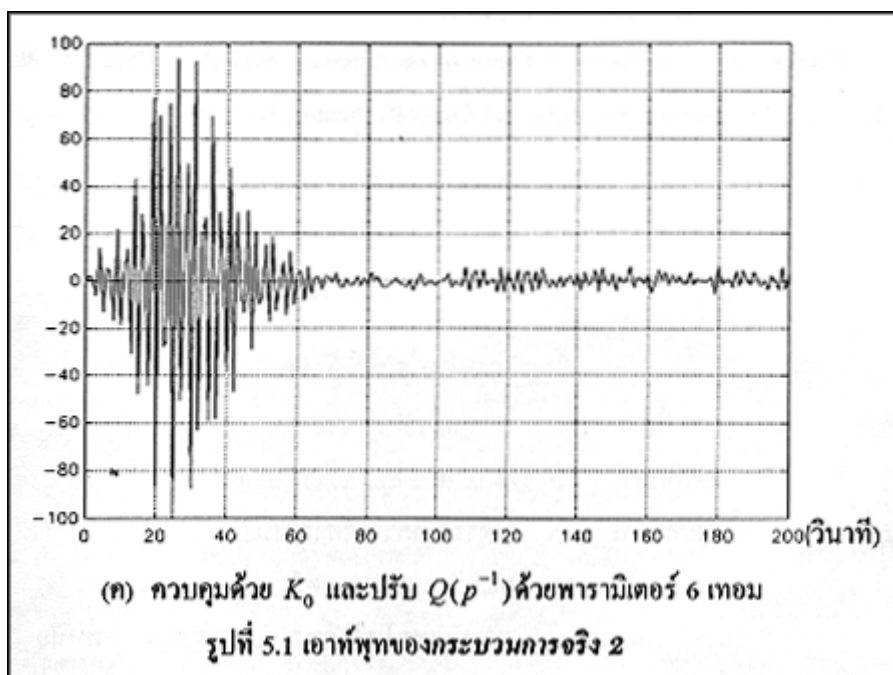
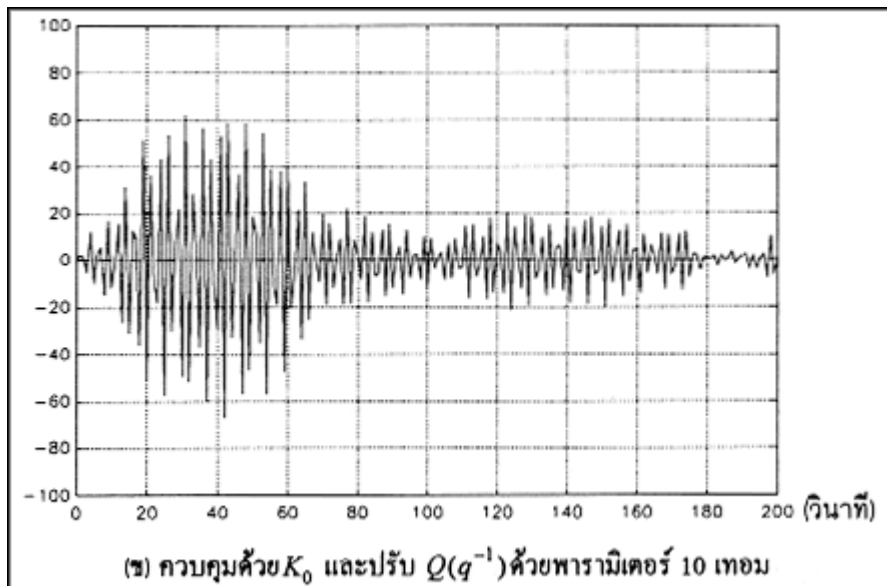
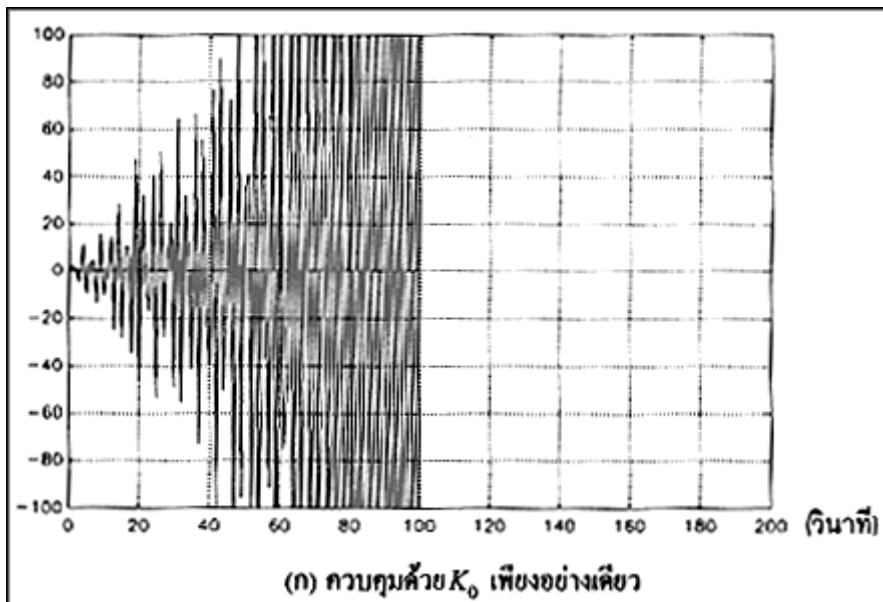
การปรับ Q โดยใช้โอเปอเรเตอร์ q ทำให้สมรรถนะของระบบเริ่มดีขึ้น จะเห็นได้จากค่าตรรกะนิสมรรถนะเฉลี่ยลดลงจาก 35.38 ไปเป็น 14.52 สำหรับกระบวนการจริง 1 นอกจากนี้ยังสามารถควบคุมกระบวนการจริง 2 ได้ โดยมีค่าตรรกะนิสมรรถนะเฉลี่ยเป็น 4132

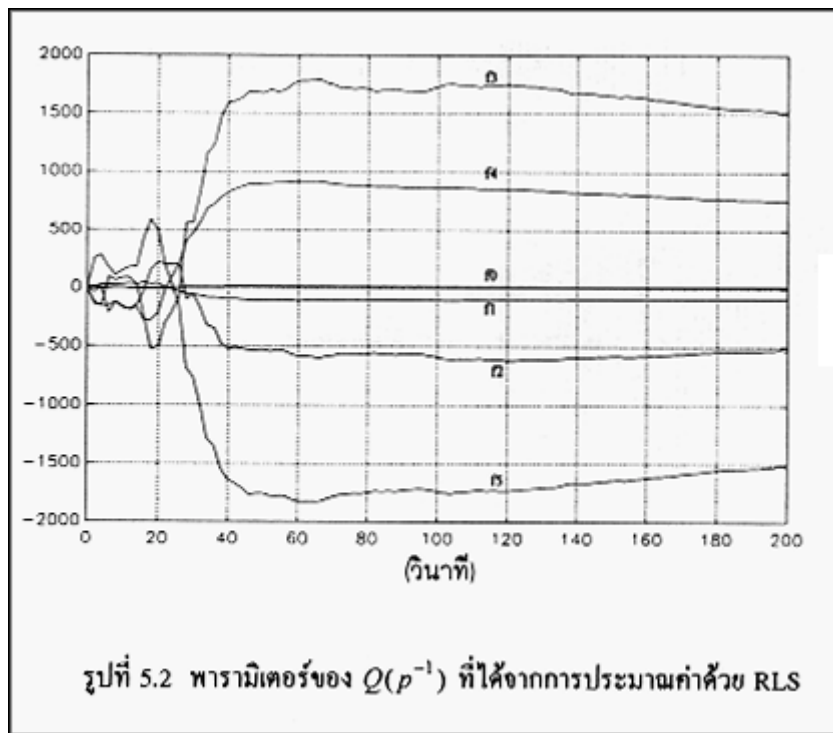
สมรรถนะของระบบดีขึ้นอย่างเห็นได้ชัดเมื่อเราใช้การปรับ Q ด้วยโอเปอเรเตอร์ p ซึ่งได้ค่าตรรกะนิสมรรถนะเป็น 10.24 และ 2104 สำหรับกระบวนการจริง 1 และกระบวนการจริง 2 ตามลำดับ นอกจากนี้จำนวนพารามิเตอร์ที่ใช้ก็น้อยกว่าของโอเปอเรเตอร์ q

รูปที่ 5.1 แสดงตัวอย่างเอาต์พุตของกระบวนการจริง 2 ในการทดสอบครั้งหนึ่ง โดยในรูปที่ 5.1 (ก) นั้นเราใช้ K_0 เพียงตัวเดียวไม่มีการเพิ่มลูบของ Q ซึ่งจะเห็นว่าระบบไร้เสถียรภาพ รูปที่ 5.1 (ข) และ (ค) เป็นกรณีที่ได้ปรับ Q ด้วยโอเปอเรเตอร์ q จำนวน 10 เทอม และโอเปอเรเตอร์ p จำนวน 6 เทอมตามลำดับ จะเห็นได้ว่าการปรับ Q ด้วยโอเปอเรเตอร์ p นั้นจะทำให้ระบบเข้าสู่สภาวะอยู่ตัวได้เร็วกว่า และให้ค่าตรรกะนิสมรรถนะน้อยกว่า โดยที่ใช้จำนวนพารามิเตอร์น้อยกว่า เมื่อเปรียบเทียบกับการใช้โอเปอเรเตอร์ q รูปที่ 5.2 แสดงพารามิเตอร์ของ $Q(p^{-1})$ ที่ประมาณได้จากอัลกอริทึม

6. สรุป

บทความนี้ได้เสนอแนวทางในการแก้ปัญหาของการออกแบบระบบควบคุมแบบปรับตัวเองในกรณีที่ใช้อัตราการซึกตัวอย่างสัญญาณที่สูง (high sampling rate) โดยการเอาหลักการออกแบบตัวควบคุมแบบปรับตัวเองมาเสริมสมรรถนะของตัวควบคุมแบบคงทน นั่นคือ การเอาข้อมูลที่ได้จากกระบวนการที่ระบุ (nominal plant) มาออกแบบตัวควบคุมแบบคงทน (robust control) จากนั้นจะเพิ่มลูบของฟังก์ชันถ่ายโอน Q พร้อมกับอัลกอริทึมในการปรับพารามิเตอร์ของ Q เพื่อเสริมสมรรถนะของระบบในกรณีที่กระบวนการจริงแตกต่างไปจากกระบวนการที่ระบุ ปัญหาที่เกิดขึ้นกับระบบที่ใช้อัตราการซึกตัวอย่างสัญญาณที่สูงโดยทั่วไปก็คือ โพลของระบบจะเข้ามาอยู่ใกล้จุด $z=1$ ในระนาบ z ซึ่งเป็นผลให้ประสิทธิภาพของอัลกอริทึมของการปรับพารามิเตอร์ของ Q ต่ำลง ปัญหานี้แก้ไขได้โดยการแปลงโคออดิเนตของ Q ก่อนที่จะทำการปรับพารามิเตอร์ ผลการทดลองแสดงให้เห็นว่าวิธีการนี้ทำให้สมรรถนะของระบบดีขึ้นอย่างเห็นได้ชัด





เอกสารอ้างอิง

- [1] Wittenmark, B, "Adaptive Stability Augmentation", Preprint, IFAC Symposium on Identification and Parameter Estimation, Aug 1988.
- [2] Astrom, K.J. and Wittenmark B. "On Self Tuning Regulators", Automatica, Vol. 9, 1973, pp.185-199.
- [3] Tay, T.T., J.B. Moore, "Enhancement of Fixed controller via Adaptive Disturbance Estimate Feedback", Automatica, Vol. 27, No 1, Jan 91, pp. 39-54.
- [4] Moore, J.B., TT Tay, Adaptive Control within the Class of Stabilizing Controllers for a Time-varying Nominal Plant", International Journal of Control, Vol. 50, No 1, July 1989, pp. 33-53.
- [5] Vidyasagar, M (1985). Control System Synthesis: A Factorization Approach, MIT Press USA
- [6] Goodwin, G, K.S. Sin, "Adaptive Filtering Prediction and Control", Prentice Hall, 1984

ประวัติผู้วิจัย



ชื่อ

สุขสันต์ นุ่นงาม

การศึกษา

Ph.D. (Control Engineering)

การทำงาน

อาจารย์ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ