

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

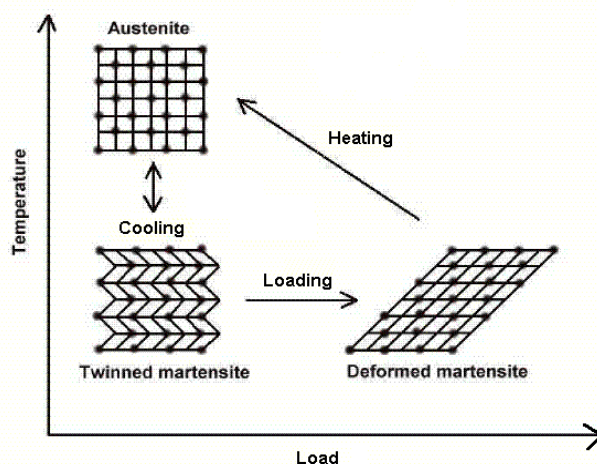
2.1 บทนำ

ในบทนี้จะอธิบายถึงทฤษฎีต่างๆที่มีความสำคัญในการทำวิจัยสำหรับการวินิจฉัยความล้าที่เกิดบนวาล์วควบคุมที่ใช้โลหะผสมจำรูปเป็นตัวขับเคลื่อนแอกชูเอเตอร์ ส่วนแรกกล่าวถึงแอกชูเอเตอร์ที่ใช้เพื่อกระตุ้นวาล์วควบคุมซึ่งมีเส้นลวดโลหะผสมจำรูปเป็นส่วนประกอบสำคัญ ส่วนที่สองกล่าวถึงความล้าจากการใช้งานโลหะผสมจำรูปในการขับเคลื่อนแอกชูเอเตอร์ในวาล์วควบคุม และส่วนที่สามกล่าวถึงการตรวจวัดความล้าของโลหะผสมจำรูปที่ใช้เป็นตัวขับเคลื่อนแอกชูเอเตอร์บนวาล์วควบคุม

2.2 แอกชูเอเตอร์ที่ขับเคลื่อนด้วยโลหะผสมจำรูปสำหรับวาล์วควบคุม

แอกชูเอเตอร์ คือ อุปกรณ์ที่มีคุณสมบัติในการกระตุ้น สามารถตอบสนองต่อสิ่งเร้าให้กำเนิดแรงได้ โดยอาศัยหลักการเปลี่ยนรูปพลังงานอื่นได้เช่น พลังงานทางไฟฟ้าเปลี่ยนเป็นพลังงานทางกล หรือความร้อนเปลี่ยนเป็นการเคลื่อนที่เชิงกลของวัสดุ แอกชูเอเตอร์ที่จะใช้ในงานวาล์วควบคุมมักใช้ลม (pneumatic) มอเตอร์ และไฮดรอลิก เป็นส่วนใหญ่ ในงานวิจัยนี้ได้ประยุกต์เอาโลหะผสมจำรูปมาเป็นตัวแอกชูเอเตอร์ในวาล์วควบคุมแทนระบบที่ใช้ลม มอเตอร์ หรือไฮดรอลิก

2.2.1 โลหะผสมจำรูป

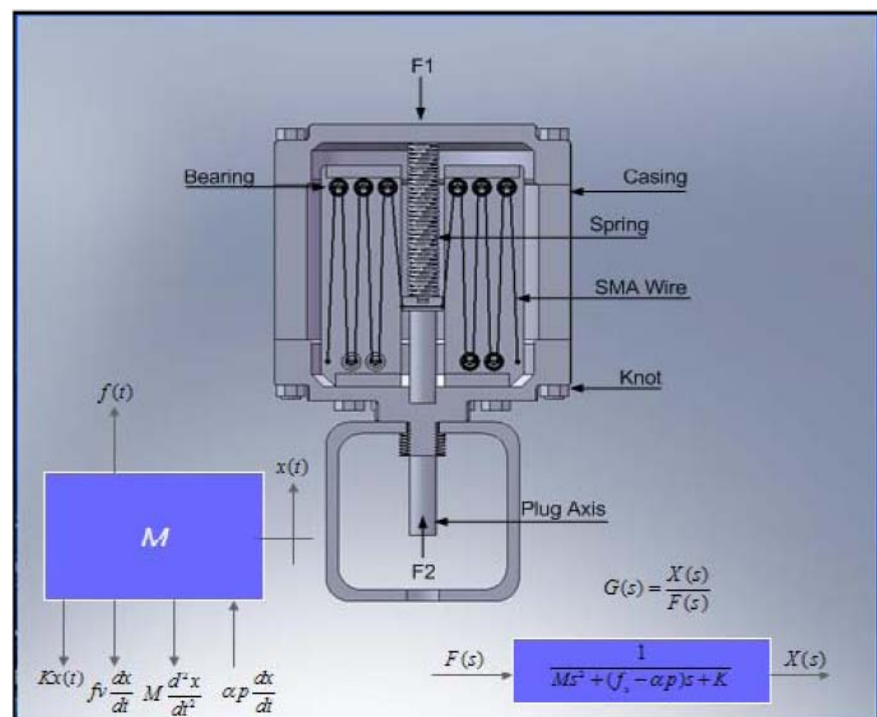


รูปที่ 2.1 แผนภาพกลไกอย่างง่ายของการจำรูปของโลหะผสมจำรูป (Ryhanen, Jorma, 1999)

โลหะผสมจำรูปจัดเป็นวัสดุฉลาดเนื่องจากสามารถเปลี่ยนแปลงรูปร่างได้เมื่ออุณหภูมิสูงขึ้น ในขณะที่ปกติหรือก่อนการเปลี่ยนแปลงรูปร่างโลหะผสมจำรูปมีโครงสร้างแบบทวินด์มาร์เทนไซต์ (twin

martensite) เมื่อทำการดัดแปลงรูปร่างของชิ้นงาน โลหะผสมจำรูป นอกจากลักษณะของชิ้นวัสดุที่ปรากฏจะเปลี่ยนไปแล้ว โครงสร้างภายในจะเปลี่ยนไปเป็นมาร์เทนไซต์ที่ถูกเปลี่ยนร่าง (deformed martensite) เมื่อทำให้ชิ้นโลหะร้อนขึ้นจนกระทั่งอุณหภูมิสูงถึงค่าหนึ่งซึ่งเรียกว่า อุณหภูมิเปลี่ยนรูป (transformation Temperature) โครงสร้างจะเปลี่ยนเป็นออสเทนไนต์ (austenite) โดยรูปร่างของชิ้นโลหะที่ปรากฏต่อสายตาจะมีลักษณะเหมือนกับรูปร่างของชิ้นโลหะก่อนการแปรรูป แม้ว่าออสเทนไนต์และทวินด์มาร์เทนไซต์จะมีโครงสร้างต่างกันก็ตาม เมื่อปล่อยให้ชิ้นงานเย็นตัวลงมา โครงสร้างของชิ้นงานจะเปลี่ยนเป็นทวินด์มาร์เทนไซต์อีกครั้ง จะเห็นว่าการเปลี่ยนแปลงโครงสร้างที่เกิดขึ้นตลอดทั้งกระบวนการ ทำให้โลหะผสมจำรูปสามารถกลับคืนสู่สภาพเดิมก่อนถูกเปลี่ยนรูปร่างดังรูปที่ 2.1

2.2.2 แอคชูเอเตอร์



รูปที่ 2.2 โครงสร้างภายในของแอคชูเอเตอร์

วาล์วควบคุมที่ใช้ในงานวิจัยนี้เป็นการวิจัยต่อยอดจาก อุกฤษณ์ (อุกฤษณ์, 2008) ซึ่ง ได้ออกแบบแอคชูเอเตอร์สำหรับวาล์วควบคุมมีโครงสร้างดังรูป 2.2 ซึ่ง โครงสร้างที่เห็นในรูปทำมาจากอลูมิเนียมถูกออกแบบให้แขวนเส้นลวดผ่านเบร้งเพื่อจัดการกับความยาวของเส้นลวดที่มีมากกว่าขนาดของแอคชูเอเตอร์และยึดปลายด้านนอกของเส้นลวดทั้งสองเส้นให้ติดกับโครง ส่วนปลายด้านในทั้งสองเส้นต่อ

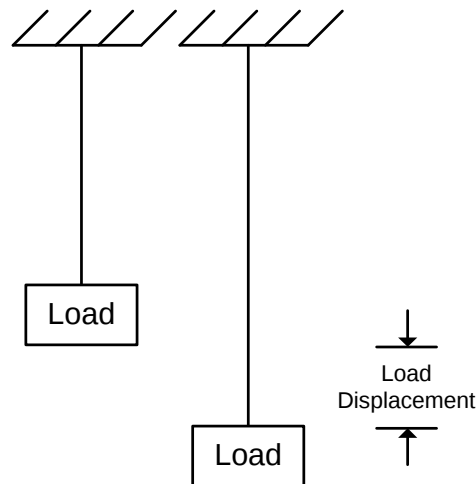
กับ plug stem ของแอกชูเอเตอร์ก่อนจะนำไปยึดเข้ากับก้านวาล์ว โดยด้านปลายของเส้นลวดทั้งสอง เส้นนั้นจะเป็นจุดที่จ่ายกระแสไฟฟ้าและกระแสไฟฟ้าจะไหลผ่านเส้นลวดโลหะผสมจำรูปลงกราวด์ เมื่อกระแสไฟฟ้าไหลผ่านเส้นลวดโลหะทั้งสองเส้นก็จะทำให้เส้นลวดนั้นหดตัวและดึงตัว plug stem ขึ้นตามระยะที่กำหนดได้ด้วยวิธีการควบคุมกระแสไฟฟ้าผ่านเส้นลวดโลหะผสมจำรูป

2.3 ความล้าจากการใช้งานโลหะผสมจำรูปที่ใช้ขับเคลื่อนแอกชูเอเตอร์ในวาล์ว

ควบคุม

ความล้าในโลหะผสมจำรูปสามารถแบ่งได้เป็นสองประเภทได้แก่ ความล้าทางโครงสร้าง (structural fatigue) ความล้าจากการใช้งาน (functional fatigue) ความล้าทางโครงสร้างหมายถึงโลหะผสมจำรูปที่ได้รับจำนวนรอบของโหลดทางกลที่สูงจะทำให้โลหะผสมจำรูปเสียหาย มีการแสดงผลกระทบจากการเกิดความล้าด้วยรอยร้าวที่เกิดขึ้นบน โครงสร้างทางกายภาพของเส้นลวดโลหะผสมจำรูป (Gall, K., 2001) และความล้าจากการใช้งานเกิดจากการถูกกระทำด้วยโหลดทางกลและความร้อน ซึ่งทำให้โลหะผสมจำรูปมีคุณสมบัติเปลี่ยนแปลงไป เช่น มีการเปลี่ยนแปลงความกว้างของอัสเทอริซิส หรือการเปลี่ยนแปลงระยะยี่ด (Farias Nascimento, M.M.S., 2006; Meier, H., 2009) ซึ่งความล้าจากการใช้งานของโลหะผสมจำรูปทำให้เกิดการเสื่อมสภาพไปจากสถานะปกติ ความล้าที่ศึกษาในงานวิจัยนี้เป็นความล้าที่เกิดจากการใช้งาน โดยหมายถึงการสูญเสียความสามารถในการคืนรูปของเส้นลวดโลหะผสมจำรูปที่เกิดจากพลังงานความร้อนเมื่อเทียบกันระหว่างการใช้งานครั้งแรกกับครั้งถัดไป เมื่อผ่านภาระทางกลและความร้อนในปริมาณที่มากขึ้น

ระดับความล้าสามารถอธิบายได้ดังนี้ เมื่อเปิด/ปิดการทำงานของโลหะผสมจำรูปในแต่ละครั้ง โดยการป้อนกระแสและตัดกระแสที่จ่ายให้กับโลหะผสมจำรูป จะเกิดระยะการยืดหดของโลหะผสมจำรูป (รูปที่ 2.3) โดยสันนิษฐานว่าเมื่อเกิดความล้ามากขึ้น ระยะในการยืดหดของโลหะผสมจำรูป ในแต่ละรอบการทำงานจะมีแนวโน้มลดลง



รูปที่ 2.3 ผลกระทบจากความล้าที่ทำให้ระยะการยืดหดของโลหะผสมจำรูปเปลี่ยนแปลงที่จำนวนรอบต่างๆ

ในงานวิจัยนี้นิยามความล้าเชิงใช้งานดังสมการที่ 2.1

$$\text{Functional fatigue} = \left(1 - \frac{D(N)}{D(1)}\right) \times 100\% \quad (2.1)$$

กำหนดให้

$D(1)$ คือ ระยะทางของโลหะผสมจำรูปเมื่อมีการหัดตัวรอบที่ 1

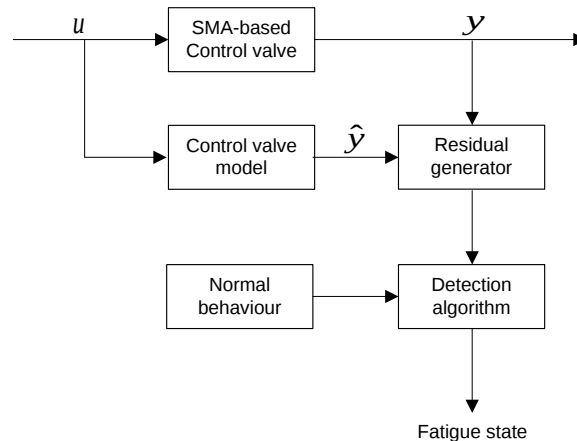
$D(N)$ คือ ระยะทางของโลหะผสมจำรูปเมื่อมีการหัดตัวรอบที่ N

การสร้างนิยามความล้าเชิงใช้งานเป็นปริมาณเชิงตัวเลข เพื่อให้มีเกณฑ์ในการบ่งบอกระดับของความล้าที่เกิดขึ้นบนวาล์วควบคุมที่ทดสอบ

2.4 การตรวจจับความล้าในโลหะผสมจำรูปที่ใช้ขับเคลื่อนแอกชูเอเตอร์ในวาล์วควบคุม

ดังที่ได้กล่าวในบทที่ 1 งานวิจัยนี้เลือกใช้วิธีการ model-based ในการตรวจจับความล้าของวาล์วทดสอบ โดยมีแผนภาพการทำงานแสดงดังรูปที่ 2.4 มีการใช้แบบจำลองทางคณิตศาสตร์ที่สามารถอธิบายลักษณะการตอบสนองของระบบเพื่อสร้างสัญญาณตกค้าง (residual signal) และตรวจสอบระดับความล้าโดยการวิเคราะห์การเปลี่ยนแปลงลักษณะของสัญญาณตกค้าง

การจัดระดับและระบุความล้ามีการศึกษาวิธีการทั้งแบบออฟไลน์และแบบออนไลน์ โดยพยายามแบ่งความล้าออกเป็น 3 ระดับ คือ ระดับปกติ ระดับความล้าปานกลาง และระดับความล้าสูง ซึ่งหมายถึงสมควรต้องมีการเปลี่ยนลวด การจัดกลุ่มความล้าแบบออฟไลน์นั้นเลือกใช้วิธี k-means clustering ในขณะที่การระบุความล้าแบบออนไลน์ใช้เทคนิค CUSUM มาตรวจจับการเปลี่ยนแปลงของสัญญาณตกค้าง



รูปที่ 2.4 ระบบตรวจจับความล้าในวาล์วควบคุม

สัญญาณตกค้างที่เหมาะสมได้มาจากโมเดลที่สามารถอธิบายระบบวาล์วควบคุมได้เป็นอย่างดี ดังนั้นการเลือกรูปแบบของโมเดลที่ใช้จึงมีความสำคัญอย่างมากสำหรับงานวิจัยนี้ การเลือกโมเดลที่เหมาะสมควรพิจารณาจากความง่ายของโครงสร้างโมเดล ความสะดวกในการหาตัวแปรในโมเดล และความเหมาะสมในการประยุกต์ใช้พัฒนาวาล์วควบคุมให้เป็นวาล์วควบคุมแบบอัจฉริยะที่มีฟังก์ชันการตรวจสอบตนเอง ซึ่งในที่นี้เลือกใช้โมเดลแบบ NARX

2.4.1 โมเดลแบบ NARX (Nonlinear Auto Regressive eXogenous)

Nonlinear Auto Regressive eXogenous model (NARX) เป็นโมเดลแบบ linear-in-the-parameters ซึ่งเป็นโมเดลที่ได้รับความนิยมโดยเฉพาะในงานด้านวิศวกรรม โดยใช้ฟังก์ชัน polynomial เป็น kernel function (polynomial NARX modelling)

โมเดลแบบ NARX เป็นรูปแบบหนึ่งของโมเดลแบบ NARMAX (Leontaritis, I.J. & Billings, S.A., 1985a,b) ซึ่งมีสมการดังนี้

$$y(k) = F(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u), \xi(k-1), \dots, \xi(k-n_\xi)) + \xi(k) \quad (2.2)$$

โดยที่ $f(\cdot)$ คือฟังก์ชันแบบไม่เชิงเส้น $u(\cdot)$ เป็นอินพุตของโมเดล $y(\cdot)$ เป็นเอาต์พุตของโมเดล และ $\xi(\cdot)$ เป็นเทอมของสัญญาณรบกวนแบบ white noise และมี n_y, n_u และ n_ξ เป็น ค่า maximum lag ของ $y(\cdot), u(\cdot)$ และ $\xi(\cdot)$ ตามลำดับ

เมื่อตัดเทอมของ moving average ออกไปจะได้โมเดลแบบ NARX ซึ่งสามารถเขียนได้ดังนี้

$$y(k) = F(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) + \xi(k) \quad (2.3)$$

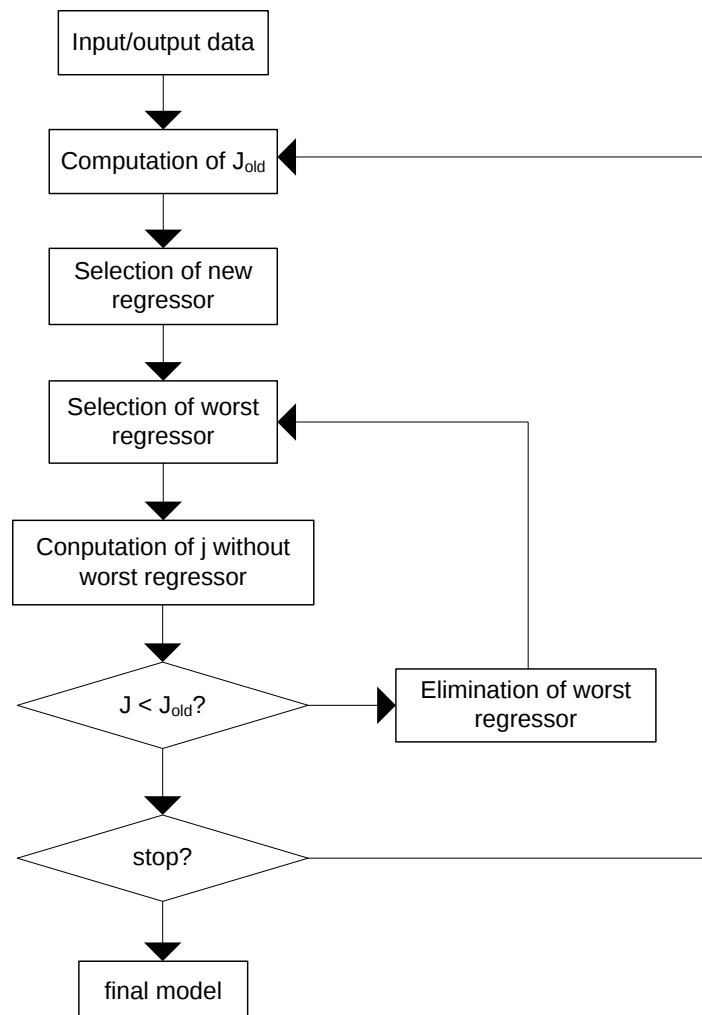
เนื่องจากจำนวนเทอมที่เป็นไปได้ของโมเดลแบบ NARX มีจำนวนมาก โดยเมื่อค่า maximum lag กำหนดมีค่ามากจำนวนเทอมดังกล่าวก็จะมากตามไปด้วย การสร้างโมเดลที่ดีจะต้องเลือกโมเดลที่มีโครงสร้างน้อยที่สุดที่จะสามารถอธิบายระบบให้ดีขึ้นเพื่อลดความยุ่งยากในการนำโมเดลไปใช้งาน ดังนั้นการเลือกโครงสร้างโมเดล (model structure selection) จึงมีความสำคัญมาก ในงานนี้หาโครงสร้างของโมเดลแบบ NARX โดยใช้อัลกอริทึม SEMP (Piroddi, L. & Spinelli, W., 2003) ซึ่งมีส่วนการทำงานหลักสองส่วนได้แก่ การเพิ่มเทอมที่เป็นไปได้ลงไปในโมเดล และการตัดเทอมที่มีความไม่เหมาะสมกับระบบหรือมีอิทธิพลต่อระบบน้อยออกไป ขั้นตอนการเพิ่มเทอมที่เป็นไปได้ลงไปในโมเดลพิจารณาจากค่า SRR (simulation error reduction ratio) หาได้ดังสมการ

$$[SRR]_j = \frac{MSSE(\mathcal{M}_i) - MSSE(\mathcal{M}_{i+1})}{\frac{1}{N} \sum_{t=1}^N y^2(t)} \quad (2.4)$$

เมื่อ $\mathcal{M}_i$ คือโมเดลที่ได้จาก iteration ที่ i และ $\mathcal{M}_{i+1}$ คือ candidate model ที่ได้จาก iteration ที่ $i+1$ เมื่อเพิ่มเทอม j เข้าไปในโมเดล และ MSSE คือ mean square simulation error

สำหรับอัลกอริทึม SEMP จะดำเนินการเลือกเทอมทั้งหมดที่เป็นไปได้ทีละเทอม จากเทอมที่เป็นไปได้ทั้งหมดมาเพิ่มเข้าในสมการโมเดล โดยเลือกจากเทอมที่ทำให้เกิดค่า SRR มากที่สุด ซึ่งแสดงให้เห็นถึงความแตกต่างระหว่างค่า $MSSE(\mathcal{M}_i)$ ของโมเดลปัจจุบันเปรียบเทียบกับค่า $MSSE(\mathcal{M}_{i+1})$ เมื่อเพิ่ม candidate term j เข้าไป หากค่า SRR มีค่าเป็นบวกมากแสดงให้เห็นว่าเทอม regressor ใหม่ที่เพิ่มเข้าไปทำให้โมเดลดีขึ้น ซึ่งเทอม regressor ที่มีค่า SRR มากที่สุดของเทอมทั้งหมดที่เป็นไปได้ คือ

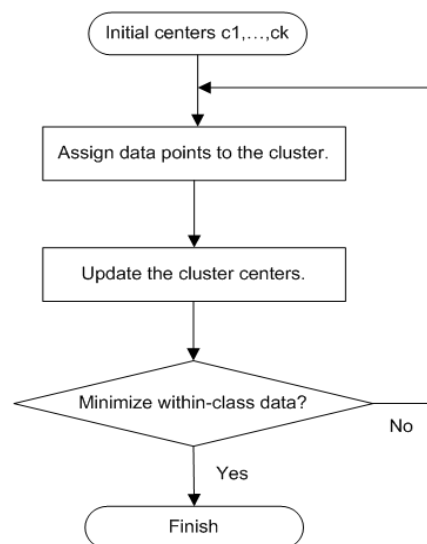
เทอม regressor ที่ดีที่สุดที่ควรเพิ่มให้แก่โมเดล หลังจากมีการเพิ่มเทอมเข้าไป ขั้นตอนถัดไปคือการตัดเทอม regressor ที่มีความไม่เหมาะสมกับระบบหรือมีอิทธิพลต่อระบบน้อยออกไปด้วยวิธีการ pruning เนื่องจากเมื่อเพิ่มเทอม regressor บางเทอมเข้าไปใน โมเดล เทอม regressor ที่มีอยู่เดิมอาจจะลดความสำคัญและอิทธิพลลง ซึ่งสมควรตัดเทอม regressor นั้นออกจากโมเดล เพื่อให้เหลือเพียงเทอมที่มีความสำคัญเท่านั้น ผลลัพธ์ที่ได้จะช่วยทำให้โมเดลมีขนาดเล็กลงเหลือเพียงเทอมที่น้อยที่สุดและมีประสิทธิภาพที่สุดในการอธิบายระบบเพื่อให้ง่ายแก่การนำไปประยุกต์ใช้ ขั้นตอนการทำงานของอัลกอริธึมดังแสดงรูปที่ 2.5



รูปที่ 2.5 โฟลว์ไคอะแกรมของการหาโมเดลด้วยวิธี SEMP เมื่อ J และ J_{old} คือ objective function ที่ใช้ในการพิจารณาเลือกจัดเทอม ในที่นี้ใช้ SRR เป็น objective function (Piroddi, L., 2003)

2.4.2 การวิเคราะห์ความคล้ายแบบออฟไลน์ด้วยวิธี k-means clustering

สำหรับการวิเคราะห์ความคล้ายแบบออฟไลน์ ได้เลือกใช้การจัดกลุ่มข้อมูลแบบ k-means clustering เป็นวิธีการในการแบ่งข้อมูลที่มีออกเป็น k กลุ่ม มีข้อดีที่วิธีการนี้เป็นวิธีการแบบ unsupervised learning ซึ่งไม่จำเป็นต้องใช้ข้อมูลต้นแบบในการเปรียบเทียบ สามารถแบ่งกลุ่มได้จากข้อมูลที่มีอยู่เพียงชุดเดียวจึงเหมาะสมกับการพิจารณาในขั้นตอนนี้ของการวิจัย k-means clustering จะพิจารณาจากการวัดความเหมือนของข้อมูลจากระยะห่างระหว่างข้อมูลกับจุดศูนย์กลางของกลุ่ม (centroid) ในการแบ่งกลุ่ม โดยจุดศูนย์กลางซึ่งเป็นตัวแทนของกลุ่มคำนวณได้จากค่าเฉลี่ยของข้อมูลทั้งหมดในกลุ่มนั้นๆ หลังจากจัดข้อมูลเข้ากลุ่มแล้ว ทำการคำนวณค่าจุดศูนย์กลางของกลุ่มใหม่ ขั้นตอนดังกล่าวจะกระทำจนกว่าผลรวมของระยะระหว่างข้อมูลในกลุ่ม (within-class data) กับจุดศูนย์กลางกลุ่มมีค่าต่ำที่สุด มีโฟลว์ไดอะแกรมของอัลกอริธึม kmeans clustering ดังรูปที่ 2.6



รูปที่ 2.6 โฟลว์ไดอะแกรมของอัลกอริธึม kmeans clustering

ผลรวมของระยะห่างระหว่างข้อมูลภายในกลุ่มกับจุดศูนย์กลางของกลุ่ม ($W(C)$) คำนวณในรูปของ Euclidian distance ดังนี้

$$W(C) = \sum_{j=1}^k N_j \sum_{C(i)=j} \|x_i - c_j\|^2 \quad (2.5)$$

เมื่อ x_i เมื่อ ชุดข้อมูลภายในกลุ่มของ iteration ที่ i , c_j คือ จุดศูนย์กลางของกลุ่มที่ j , $C(i)$ แทนหมายเลขคลัสเตอร์ใน iteration ที่ i , N_j คือ จำนวนข้อมูลของคลัสเตอร์ที่ j และ k คือ จำนวนคลัสเตอร์ทั้งหมดที่ต้องการจัดกลุ่ม

ในงานวิจัยนี้ นำ k-means clustering มาวิเคราะห์ความล้าแบบออนไลน์ โดยได้ทำการแบ่งกลุ่มของความล้าของวาล์วเป็น 3 ระดับ ได้แก่ความล้าระดับต่ำ ระดับกลาง และระดับสูง โดยนำค่าตัวแปรต่างๆ ของโมเดลร่วมกับค่าทางสถิติของสัญญาณตกค้างที่มีนัยสำคัญในการแสดงลักษณะผลกระทบต่อความล้าได้ชัดเจนมาใช้เป็นลักษณะสำคัญ (features) ในการจัดกลุ่มความล้าและจะกล่าวถึงโดยละเอียดอีกครั้งในบทที่ 4

2.4.3 การวิเคราะห์ความล้าแบบออนไลน์ด้วยอัลกอริธึม CUSUM

การวิเคราะห์ความล้าแบบออนไลน์ในวาล์วควบคุมได้เลือกใช้ Cumulative sum (CUSUM) เป็นเทคนิคที่ใช้ตรวจจับการเปลี่ยนแปลงคุณสมบัติทางสถิติของสัญญาณ (Page, E.S., 1954) ซึ่งงานวิจัยนี้เลือกใช้ Wald's CUSUM Sequential Probability Ratio Test (SPRT) ในการตรวจจับการเกิดการเปลี่ยนแปลงค่าเฉลี่ย (mean) หรือ ค่าความแปรปรวน (variance) ของสัญญาณตกค้างที่นำมาวิเคราะห์ เนื่องจากความง่ายในการสร้างและสามารถใช้เป็นแนวทางในการพัฒนาเป็นฟังก์ชันการตรวจสอบตัวเองในวาล์วควบคุมแบบออนไลน์ได้

พิจารณาให้ $X = \{x_1, x_2, \dots, x_N\}$ เป็นชุดข้อมูลที่ต้องการตรวจจับการเปลี่ยนแปลง ซึ่งในที่นี้หมายถึงสัญญาณตกค้างที่ใช้พิจารณาความล้า การเปลี่ยนแปลงความล้าพิจารณาจากค่า cumulative sum (g_k) ของ log-likelihood ratio (s_i) ที่มีการเปลี่ยนแปลงของตัวแปร θ ซึ่งใช้เป็นตัวบอกความแตกต่างระหว่างสมมติฐานของสถานะของระบบในสภาวะปกติ $H_0 : \theta = \theta_0$ กับสถานะของระบบในช่วงที่เกิดการเปลี่ยนแปลงของตัวแปรทางสถิติที่พิจารณา $H_1 : \theta = \theta_1$ ซึ่ง CUSUM ของ log-likelihood ratio คำนวณได้ดังนี้

$$g_k = \max_{1 \leq j \leq k} \sum_{i=j}^k s_i \quad \text{โดยที่} \quad s_i = \log \left(\frac{p_{\theta_1}(x_i)}{p_{\theta_0}(x_i)} \right) \quad (2.6)$$

ถ้าในสภาวะปกติความน่าจะเป็นของ $p_{\theta_0}(x_i)$ จะมีค่ามากกว่า $p_{\theta_1}(x_i)$ ซึ่งทำให้ g_k มีค่าเป็นลบ เนื่องจากผลหารของตัวตั้งต้นที่มีค่าน้อยกว่าตัวหาร หากในสภาวะที่มีการเปลี่ยนแปลงของสัญญาณที่นำมาใช้จะทำให้ g_k มีค่าเป็นบวกและเพิ่มมากขึ้นเรื่อยๆ เนื่องจากผลหารของตัวตั้งต้นที่มีค่ามากกว่าตัวหาร โดยที่เมื่อ i คือ ข้อมูลตัวที่ i และ p_θ คือฟังก์ชันการแจกแจงความน่าจะเป็นของ θ ซึ่งสมมุติว่าเป็นแบบปกติและมีรูปแบบดังนี้

$$p_\theta(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-u)^2}{2\sigma^2}} \quad (2.7)$$

เมื่อ σ คือค่าความเบี่ยงเบนมาตรฐาน, μ คือเฉลี่ยของสัญญาณที่พิจารณา และ y คือสัญญาณที่นำมาตรวจจับการเปลี่ยนแปลงทางสถิติ

รูปแบบของอัลกอริธึม CUSUM เขียนแบบ recursive ได้ดังนี้

$$g_k = \max(g_{k-1} + s_k, 0) \quad (2.8)$$

สำหรับการพิจารณาการเปลี่ยนแปลงค่าเฉลี่ย ภายใต้สมมติฐานที่ไม่มีการเปลี่ยนแปลงค่าความเบี่ยงเบนมาตรฐาน σ หรือมีค่าคงที่ และค่าเฉลี่ยมีการเปลี่ยนแปลงในสถานะปกติและสถานะที่เกิดการเปลี่ยนแปลงเป็น μ_0 และ μ_1 ตามลำดับ สามารถคำนวณ s_i ได้ดังนี้

$$s_i = \frac{u_1 - u_0}{\sigma^2} (y_i - \frac{u_1 + u_0}{2}) \quad (2.9)$$

โดยอาศัยแนวคิดดังกล่าว อัลกอริธึมแบบ CUSUM สามารถนำมาใช้พิจารณาการเปลี่ยนแปลงค่าความแปรปรวนของสัญญาณได้เช่นเดียวกัน โดยสมมติให้ค่าเฉลี่ยมีค่าคงที่เป็น μ ค่าความเบี่ยงเบนมาตรฐานในสถานะปกติเป็น σ_0 และสถานะหลังการเปลี่ยนแปลงเป็น σ_1 สามารถคำนวณค่า CUSUM ของ log-likelihood ratio ได้ดังนี้

$$s_i = \log\left(\frac{\sigma_1}{\sigma_0}\right) + \left(\frac{1}{\sigma_0^2} + \frac{1}{\sigma_1^2}\right) \frac{(y_k - \mu)^2}{2} \quad (2.10)$$

สัญญาณเตือน (alarm) จะเกิดขึ้นเมื่อค่า g_k มีค่าเพิ่มขึ้นมากกว่าค่า threshold (T) เวลาที่เกิดสัญญาณเตือนสามารถหาได้ดังนี้

$$t_a = \min\{k : g_k \geq T\} \quad (2.11)$$

โดยที่ T คือ threshold และ t_a คือเวลาที่ตรวจพบความผิดปกติ ซึ่งแสดงการตรวจจับโดยใช้เวลาที่น้อยที่สุดที่ค่า CUSUM จะมีค่ามากกว่าระดับ threshold

ตัวอย่างที่ 2.1: การทำงานของอัลกอริธึม CUSUM

ตัวอย่างนี้เป็นการทดสอบการตรวจจับการเปลี่ยนแปลงค่าความแปรปรวนด้วยอัลกอริธึม CUSUM เพื่อพิจารณาข้อดีข้อเสียของอัลกอริธึมเริ่มจากการสร้างสัญญาณแบบ Gaussian ที่มีค่าความเบี่ยงเบนมาตรฐานในสถานะเริ่มต้นเป็น 1 ที่ 1-1000 ข้อมูล ต่อด้วยค่าเบี่ยงเบนมาตรฐานในสถานะที่เกิดการ

เปลี่ยนแปลงเป็น 1.5 ที่ 1001-2000 ข้อมูล และค่าเบี่ยงเบนมาตรฐานในสถานะที่เกิดการเปลี่ยนแปลงเป็น 2 ที่ 2001-3000 ข้อมูล ดังรูป 2.7 (a) ตามลำดับ

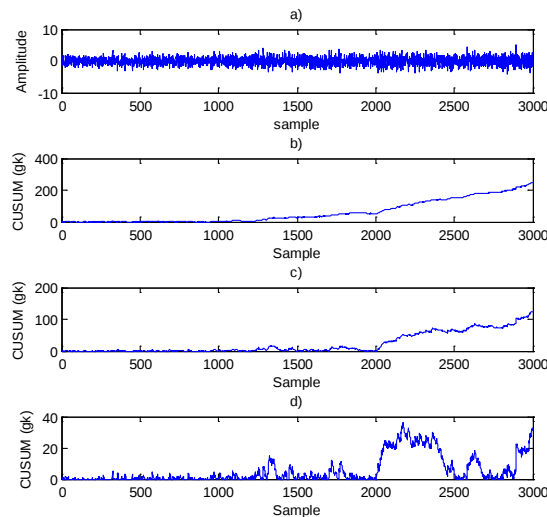
นำสัญญาณที่สร้างขึ้นมาทดสอบกับอัลกอริทึม CUSUM ซึ่งมีการคำนวณค่า CUSUM โดยใช้สมการที่ 2.10 ซึ่งจำเป็นต้องกำหนดตัวแปร μ , σ_0 และ σ_1 จากนั้นคำนวณแบบ recursive ดังสมการที่ 2.8

ในการใช้งานจะกำหนดให้ μ และ σ_0 ได้จากการประมาณข้อมูลในช่วงเริ่มต้นจำนวน M ข้อมูล โดยเลือกใช้ M ที่ 250 มีรูปแบบสมการดังสมการที่ 2.12 และ 2.13

$$\mu = \frac{1}{M} \sum_{i=1}^M r_i \quad (2.12)$$

$$\sigma_0 = \sqrt{\frac{\sum_{i=1}^M (r_i - \mu)^2}{M}} \quad (2.13)$$

จากนั้นสังเกตอิทธิพลในการตรวจจับค่าการเปลี่ยนแปลงความเบี่ยงเบนมาตรฐานที่เกิดจากการกำหนด σ_1 ด้วยวิธีการต่างๆ ดังรูปที่ 2.7 (b), (c) และ (d)



รูปที่ 2.7 ผลการตรวจจับการเปลี่ยนแปลงความเบี่ยงเบนมาตรฐานด้วย CUSUM

- (a) สัญญาณแบบ Gaussian ที่มีการเปลี่ยนแปลงค่าเบี่ยงเบนมาตรฐาน
- (b) ค่า CUSUM กรณี $\sigma_1 = 1.5\sigma_0$
- (c) ค่า CUSUM กรณี $\sigma_1 = 2\sigma_0$
- (d) ค่า CUSUM กรณี $\sigma_1 = 3\sigma_0$

จากรูปที่ 2.7 (b), (c) และ (d) แสดงผลลัพธ์ของสัญญาณทดสอบด้วยการกำหนดตัวแปร σ_1 ที่ 1.5 เท่า, 2 เท่า และ 2.5 เท่าของ σ_0 ตามลำดับ เห็นได้ว่าการกำหนดตัวแปรดังกล่าวมีอิทธิพลโดยตรงต่อประสิทธิภาพในการตรวจจับ กล่าวคือหากค่า σ_1 ไม่เหมาะสมกับขนาดความเบี่ยงเบนมาตรฐานจะทำให้ผลลัพธ์ของค่า CUSUM ไม่ชัดเจน ดังแสดงในรูป 2.7 กราฟย่อยที่ 3 และหากค่า σ_1 มีขนาดใหญ่เกินกว่าความเป็นจริงมากจะทำให้ค่า CUSUM ที่ได้ไม่สามารถนำมาใช้บ่งชี้การเปลี่ยนแปลงที่เกิดขึ้นได้ ดังแสดงในรูป 2.7 กราฟย่อยที่ 4 ซึ่งการกำหนดตัวแปรนี้ให้มีความเหมาะสมมักจะกำหนดโดยการทดสอบโดยใช้ตัวแปร σ_1 ที่ละค่า จนได้เอาท์พุตที่ดีที่สุด แสดงให้เห็นว่าการกำหนดตัวแปรนี้มีความเซนซิทิฟเป็นอย่างมาก

2.4.4 การวิเคราะห์ความล่าแบบออนไลน์ด้วยอัลกอริธึม adaptive CUSUM

ปัญหาในการใช้งานอัลกอริธึมแบบ CUSUM คือ ต้องมีการกำหนดค่าตัวแปรหลายตัวในการใช้งาน ได้แก่ ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานในสถานะก่อนและหลังการเปลี่ยนแปลง รวมถึงการกำหนดค่า threshold ที่ใช้ตรวจจับการเปลี่ยนแปลงด้วย การกำหนดค่าดังกล่าวส่วนใหญ่ไม่มีข้อกำหนดตายตัวและกำหนดได้จากทดสอบแบบ trial and error ในงานวิจัยนี้จึงนำเสนอการเลือกค่าตัวแปรดังกล่าวอย่างมีเหตุผลผลและกำหนดค่าต่างๆ โดยอัตโนมัติ โดยให้ตัวแปรต่างๆที่ใช้ในอัลกอริธึมปรับตัวให้เหมาะสมในการคำนวณเพื่อให้สามารถประยุกต์ใช้ CUSUM ได้กับงานที่สัญญาณมีการเปลี่ยนแปลงลักษณะทางสถิติมากกว่าหนึ่งครั้งในระหว่างขั้นตอนการตรวจจับ (Alippi, C., 2006) การกำหนดตัวแปรต่างๆที่ใช้ในอัลกอริธึมมีความสำคัญมากดังที่ได้กล่าวถึงปัญหาในข้างต้น ดังนั้นจึงมีวิธีการกำหนดตัวแปรที่ใช้ดังหัวข้อต่อไปนี้

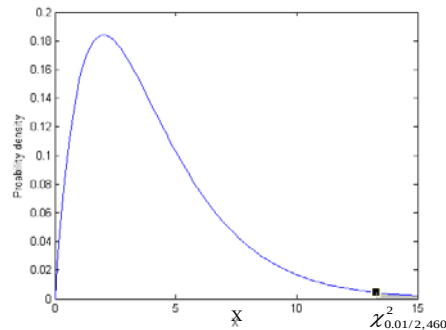
2.4.4.1 การกำหนดค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานภายหลังการเปลี่ยนแปลง

เริ่มจากเก็บข้อมูลที่เป็นตัวแทนของสถานะที่พิจารณาในปัจจุบัน (เช่น ขณะระบบทำงานปกติ) ทั้งหมด M ข้อมูล แล้วนำแซมเปิลดังกล่าวมาประมาณค่าเฉลี่ย μ_0 และค่าเบี่ยงเบนมาตรฐาน σ_0 ของการกระจายข้อมูลของสัญญาณที่สถานะนั้น จากนั้นประมาณค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานภายหลังการเปลี่ยนแปลง $\{\mu_1, \sigma_1\}$ จากการพิจารณาช่วงความเชื่อมั่นที่ค่าความเชื่อมั่น $100(1-\alpha)\%$ ในที่นี้กำหนดความเชื่อมั่นที่ 99% นั่นคือ $\alpha = 0.01$ จะได้

$$\mu_1 = \mu_0 + \frac{2.58}{\sqrt{M}} \sigma_0 \quad (2.14)$$

$$\sigma_1 = \sqrt{\frac{M\sigma_0^2}{\chi_{\alpha/2}^2}} \quad (2.15)$$

เมื่อ $\chi^2_{\alpha/2}$ คือ ค่า upper critical value จากตาราง chi-square หรือหาค่าได้จากพื้นที่ใต้กราฟดังรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างการกระจายตัวแบบ chi square กรณีความเชื่อมั่นที่ 99% และ degrees of freedom = 4

2.4.4.2 การกำหนดค่า threshold

หลังจากได้ค่าประมาณของค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานภายหลังการเปลี่ยนแปลงจากหัวข้อที่ผ่านมา นำค่าดังกล่าวไปคำนวณ CUSUM เพื่อหา test statistics (s_i) จากนั้นกำหนดค่า threshold (T) จาก

$$T = \beta \max_{1 \leq i \leq M} s_i \quad (2.16)$$

เมื่อ β เป็นค่าคงที่ที่มีค่าเป็นบวกและกำหนดโดยผู้ใช้ การกำหนดค่า β นั้น ส่วนใหญ่ได้จากการทดลอง แต่มีหลักการคือ ถ้าค่า β มีค่ามาก ก็จะทำให้มีโอกาสที่จะเกิด missed alarm มากขึ้น ในทางตรงกันข้ามถ้าค่า β มีค่าน้อย ก็จะทำให้มีโอกาสที่จะเกิดการแจ้งเตือนผิดพลาด (false alarm) มากขึ้น

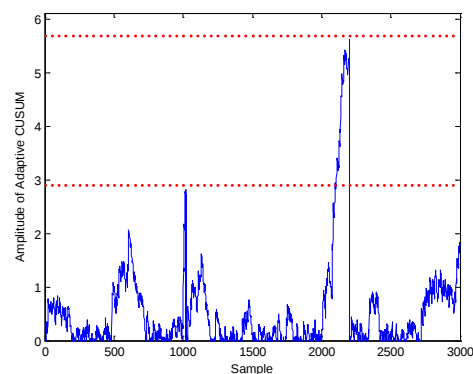
ภายหลังจากได้ค่าตัวแปรของ CUSUM ทั้งหมดที่ได้กล่าวมาแล้ว ขั้นตอนต่อไปคือการใช้ adaptive CUSUM ในการตรวจจับการเปลี่ยนแปลงของสัญญาณ โดยมีหลักการดังนี้ เมื่อค่า CUSUM มีค่ามากเกินระดับ threshold ซึ่งคำนวณจาก β เท่าของค่าสูงสุดของ CUSUM ในช่วง M ข้อมูล จะทำการระบุว่าเกิดการเปลี่ยนแปลงขึ้น จากนั้นจะเริ่มต้น คำนวณตัวแปร $\{\mu_0, \mu_1, \sigma_0, \sigma_1, T\}$ ใหม่อีกครั้งโดยใช้ข้อมูล M ตัวถัดไป รวมถึงตั้งค่า CUSUM เป็นศูนย์ใหม่อีกครั้งหนึ่ง

ตัวอย่างที่ 2.2: การทำงานของอัลกอริทึม adaptive CUSUM

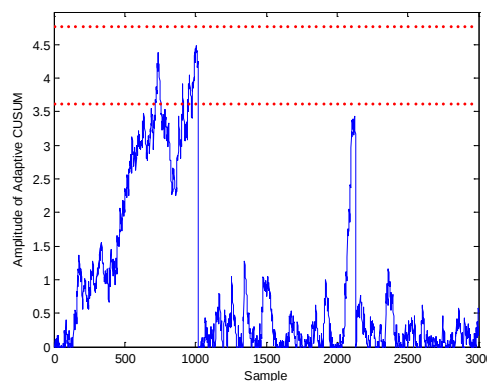
ตัวอย่างนี้เป็นการทดสอบการตรวจจับการเปลี่ยนแปลงค่าความเบี่ยงเบนมาตรฐานด้วยอัลกอริทึม adaptive CUSUM เพื่อพิจารณาข้อดีข้อเสียของอัลกอริทึมนี้เพื่อเปรียบเทียบกับอัลกอริทึม CUSUM

เริ่มจากการสร้างสัญญาณแบบ Gaussian เพื่อใช้เป็นสัญญาณทดสอบที่มีลักษณะทางสถิติเหมือนกับสัญญาณทดสอบในตัวอย่างที่ 1 ดังรูปที่ 2.7 (a)

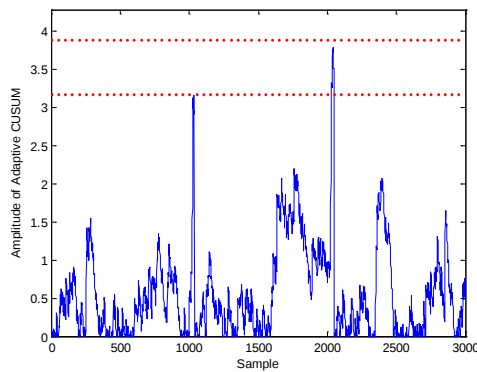
นำสัญญาณทดสอบที่สร้างขึ้นมาทดสอบกับอัลกอริธึม adaptive CUSUM ซึ่งมีการคำนวณค่า CUSUM โดยใช้สมการที่ 2.10 จากนั้นคำนวณแบบ recursive ดังสมการที่ 2.8 โดยกำหนดให้ μ และ σ_0 คำนวณจากการประมาณข้อมูลในช่วงเริ่มต้นจำนวน M ข้อมูล โดยเลือกใช้ M ที่ 250 ดังสมการที่ 2.12 และ 2.13 และคำนวณหาค่า σ_1 ด้วยสมการที่ 2.15 ซึ่งอัลกอริธึมจะคำนวณตัวแปรใหม่ทั้งหมดเมื่อมีการ alarm ขึ้นจากสมการที่ 2.16 ซึ่งระดับการตรวจจับกำหนดได้จากค่า beta ซึ่งการทดสอบนี้เลือกใช้ beta ที่ 3.5 ซึ่งเลือกมาจากการทดสอบค่า beta ไปจนกระทั่งได้ค่าที่เหมาะสมที่สุด ซึ่งให้ผลการตรวจจับที่เวลาใกล้เคียงกับการเปลี่ยนแปลงค่าเบี่ยงเบนมาตรฐานที่เกิดขึ้นมากที่สุด แสดงผลการทดสอบการตรวจจับค่าการเปลี่ยนแปลงความแปรปรวนจำนวน 3 ครั้ง เพื่อศึกษาว่าลักษณะสัญญาณที่ถูกสุ่มขึ้นมาใหม่จะมีผลต่อการตรวจจับอย่างไร ดังรูปที่ 2.9 - 2.11 ตามลำดับ



รูปที่ 2.9 ผลการทดสอบด้วยอัลกอริธึม adaptive CUSUM ครั้งที่ 1 (alarm ที่ 1021 และ 2201)



รูปที่ 2.10 ผลการทดสอบด้วยอัลกอริธึม adaptive CUSUM ครั้งที่ 2 (alarm ที่ 1019 และ 2130)



รูปที่ 2.11 ผลการทดสอบด้วยอัลกอริธึม adaptive CUSUM ครั้งที่ 3 (alarm ที่ 1036 และ 2049)

เมื่อทดสอบอัลกอริธึมด้วยสัญญาณทดสอบที่สร้างขึ้นใหม่ทุกครั้งที่มีรูปแบบสัญญาณเป็น gaussian ที่มีค่าทางสถิติเหมือนกันทุกครั้ง ซึ่งผลลัพธ์การทดสอบทั้ง 3 ครั้งที่มีความแตกต่างกันเล็กน้อย แสดงให้เห็นว่าการใช้อัลกอริธึมในการทดสอบแต่ละครั้ง ให้ความสามารถในการตรวจจับที่ไม่เท่ากันซึ่งเกิดจากความแปรปรวนของสัญญาณทดสอบแต่ละครั้ง แต่แนวโน้มของเวลาในการตรวจจับทั้ง 3 ครั้งนั้น มีค่าใกล้เคียงกัน ซึ่งหมายความว่าอัลกอริธึมนี้สามารถใช้ได้กับข้อมูลที่มีค่าทางสถิติเหมือนกันถึงแม้จะเป็นข้อมูลคนละชุดก็ตาม แต่ delay time ที่เกิดขึ้นนั้นภายหลังที่มีการเปลี่ยนแปลงค่าไปแล้วนั้นเกิดจากความจำเป็นของอัลกอริธึมที่ต้องมีการสะสมค่าให้มากขึ้นจนกว่าจะการระดับ threshold ที่กำหนด ดังนั้นการกำหนดตัวแปร beta จึงเป็นตัวแปรเพียงตัวเดียวเท่านั้นที่ผู้ใช้อาจกำหนดเองจากการทดสอบเพื่อให้ได้ผลลัพธ์ที่เหมาะสม