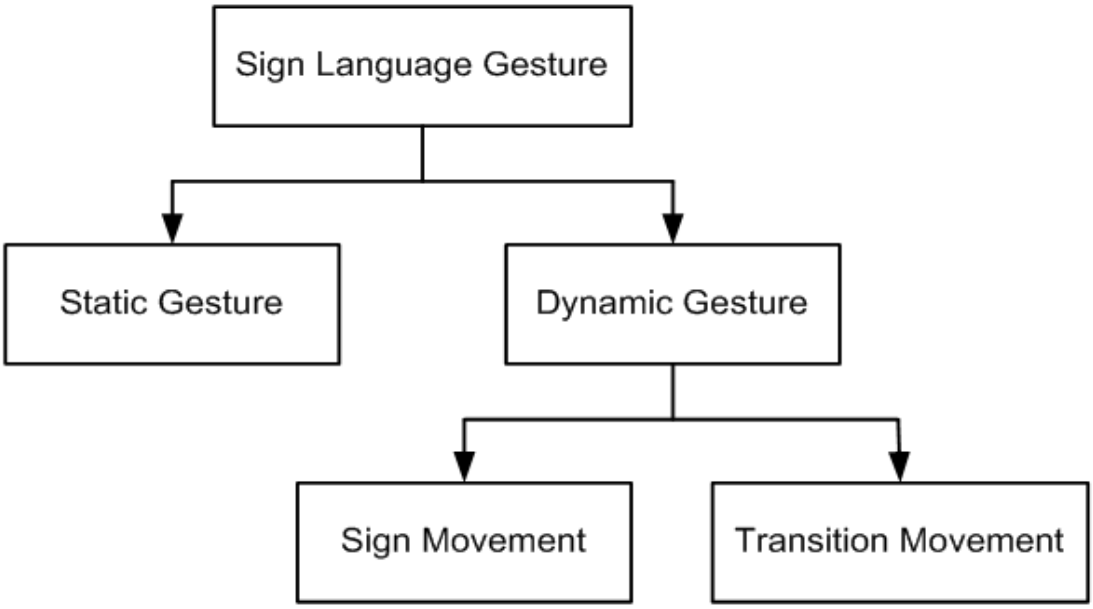
บทที่ 4
การรู้จำท่ามือแบบต่อเนื่องในภาษามือไทย

4.1 ระบบการแปลภาษามือไทย

ภาษามือไทยจะประกอบไปด้วย รูปมือ, ตำแหน่งของมือ, การเคลื่อนไหว, การหันของฝ่ามือ และการแสดงออกทางสีหน้า ซึ่งสามารถแบ่งส่วนประกอบของลักษณะการแสดงท่ามือได้ดังต่อไปนี้



รูปที่ 4.1 การแบ่งแยกรูปแบบของการแสดงท่าทางในระบบภาษามือไทย

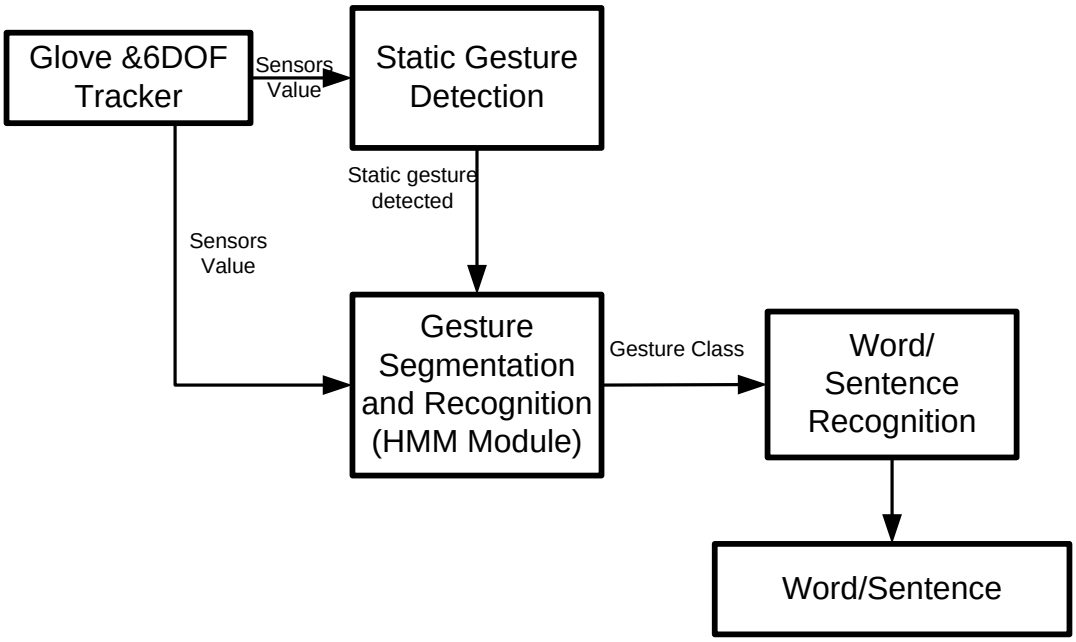
ท่านิ่ง (Static gesture, Posture) คือ การแสดงท่ามือที่ไม่มีการเคลื่อนไหวของมือและตำแหน่งต่างๆ

ท่าเคลื่อนไหว (Dynamic gesture) คือ การแสดงท่ามือที่มีการเคลื่อนไหวมือ หรือตำแหน่งของมือ ซึ่งสามารถแบ่งออกได้ คือ

- การเคลื่อนไหวที่มีความหมาย (Sign Movement) คือ การเคลื่อนไหวที่เป็นส่วนที่มีความหมายในภาษามือไทย

- การเคลื่อนไหวระหว่างท่ามือ (Transition Movement) คือ การเคลื่อนไหวในระหว่างการเปลี่ยนจากท่ามือหนึ่งไปสู่อีกท่าหนึ่ง

ซึ่งระบบการแปลภาษามืออัตโนมัติสามารถแสดงดังแผนภาพต่อไปนี้



รูปที่ 4.2 ระบบรู้จำภาษามือไทยชนิดต่อเนื่องอัตโนมัติ

ระบบการรู้จำภาษามือไทยต่อเนื่องอัตโนมัติ ดังรูปที่ 4.2 ประกอบไปด้วยส่วนสำคัญ 4 ส่วน คือ ส่วนแรกจะมีการรับข้อมูลจากถุงมืออิเลคทรอนิคส์และอุปกรณ์ตรวจวัดตำแหน่งแบบ 6 แกน ส่วนที่สอง คือ การตรวจจับท่านิ่ง ซึ่งผลจากส่วนนี้จะถูกนำมาใช้เพื่อช่วยให้ HMM recognizer สามารถลดเส้นทางในการหาเส้นด้วย Viterbi algorithm ได้ (เรียกว่าการ Pruning) ส่วนที่สาม คือ ส่วนของการรู้จำท่าทางด้วยฮิดเดนมาร์คอฟโมเดลและส่วนสุดท้าย คือ การรวมส่วนประกอบต่างๆ เข้าด้วยกันเป็นคำหรือประโยค ซึ่งในส่วนของโมดูล Static Gesture Detection นั้นเป็นการดำเนินงานจากงานวิจัยก่อนหน้า [6] โดยกระบวนการทำงานของระบบ สามารถแยกออกเป็นส่วนการทำงานได้ดังต่อไปนี้

การรับข้อมูลจากถุงมืออิเลคทรอนิคส์และอุปกรณ์ตรวจวัดตำแหน่งแบบ 6 แกน (Glove & 6DOF Tracker) เป็นการรับข้อมูลจากเซนเซอร์ต่างๆที่ติดไว้ที่ถุงมืออิเลคทรอนิคส์ ซึ่งในงานวิจัยนี้ใช้เซนเซอร์จากถุงมือจำนวน 18 เซนเซอร์และจากอุปกรณ์ตรวจวัดตำแหน่งแบบ 6 แกนจำนวน 6 เซนเซอร์

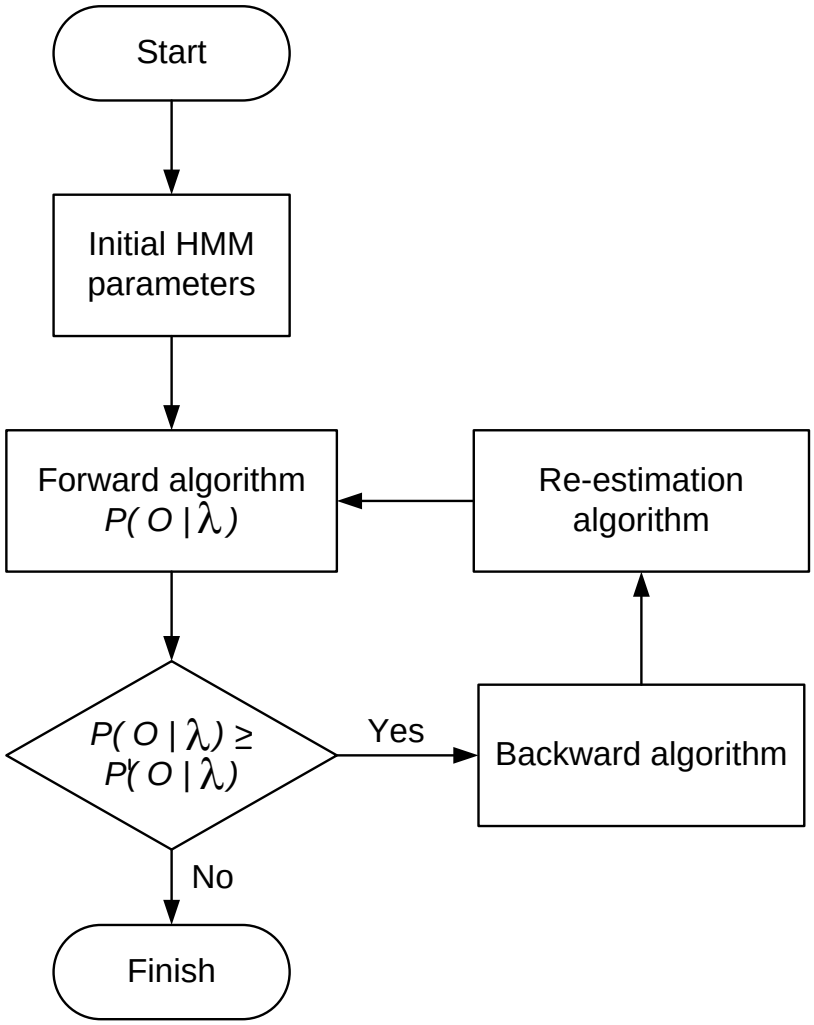
การแบ่งแยกท่านิ่งจากท่าเคลื่อนไหว (Static Gesture Detection) กระบวนการนี้จะทำหน้าที่แยกท่านิ่งออกจากท่าเคลื่อนไหว จะใช้วิธีการของ Bayesian Estimator เนื่องจากเป็นวิธีการที่ดีที่สุดในเชิงสถิติสำหรับการรู้จำท่านิ่ง

การแบ่งแยกท่าเคลื่อนไหว (Gesture Segmentation and Recognition: HMM Module) เป็นกระบวนการรู้จำท่าเคลื่อนไหวที่มีความหมายออกจากการเคลื่อนไหวระหว่างท่ามือออกมาได้ โดยใช้ฮิดเดนมาร์คอฟโมเดล

การรวมส่วนประกอบต่างๆ เข้าด้วยกันเป็นประโยค (Word/Sentence Recognition) เป็นการนำผลที่ได้จากการรู้จำในส่วนต่างๆมาประกอบกันเป็นประโยค

4.1.1 กระบวนการสอนฮิดเดนมาร์คอฟโมเดล และการใช้งานฮิดเดนมาร์คอฟโมเดล

ในหัวข้อนี้จะนำเสนอถึงกระบวนการสอนฮิดเดนมาร์คอฟโมเดล และการใช้งานฮิดเดนมาร์คอฟโมเดล ซึ่งในการสอนฮิดเดนมาร์คอฟโมเดลให้รู้จำ หรือแบ่งแยกสัญญาณที่ไม่ทราบ ทำได้โดยการนำสัญญาณที่ไม่ทราบซึ่งอยู่ในรูปของลำดับของค่าที่ปรากฏ นำมาสอนให้แก่ฮิดเดนมาร์คอฟโมเดล โดยกระทำตามขั้นตอนดังรูปที่ 4.3 โดยมีขั้นตอนในการสอนดังนี้



รูปที่ 4.3 กระบวนการสอนฮิดเดนมาร์คอฟโมเดล

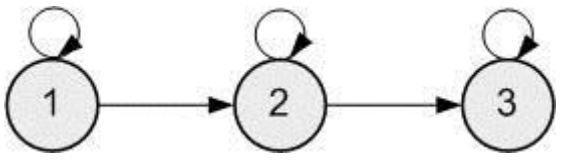
- เริ่มต้นด้วยการสุ่มค่าพารามิเตอร์ของฮิดเดนมาร์คอฟโมเดล ได้แก่ A, B, π

- คำนวณหาค่าความน่าจะเป็น โดยใช้กระบวนการไปข้างหน้า ซึ่งจะได้ค่าความน่าจะเป็นของโมเดลออกมา
- พิจารณาค่าความน่าจะเป็นใหม่กับค่าความน่าจะเป็นก่อนหน้า จะทำการปรับค่าพารามิเตอร์ของฮิดเดนมาร์คอฟโมเดล โดยใช้กระบวนการย้อนกลับ แล้วใช้บาม-เวลส์อัลกอริทึม (Baum-Welch algorithm) ในการปรับค่าพารามิเตอร์ A, B, π ให้ได้ค่าที่เหมาะสม แล้วย้อนกลับไปทำกระบวนการไปข้างหน้าอีกครั้งเพื่อหาค่าความน่าจะเป็นของโมเดล แต่ถ้าค่าความน่าจะเป็นใหม่ที่ได้มีค่าลดลง หรือมีค่าไม่เปลี่ยนแปลงเป็นระยะเวลานาน จึงจะหยุดวงจรการคำนวณ แล้วเก็บพารามิเตอร์ A, B, π โดยพารามิเตอร์เหล่านี้เป็นตัวแทนของลักษณะลำดับของค่าที่ปรากฏที่ไม่รู้จัก โดยวิธีการดังกล่าวแสดงดังรูปที่ 4.3

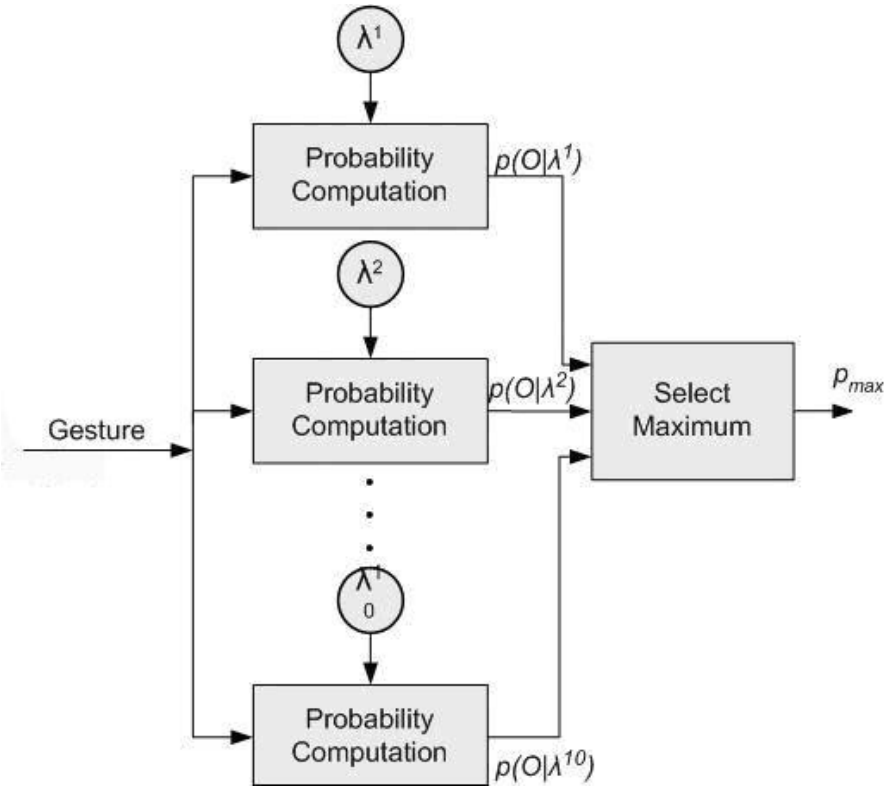
4.1.2 การใช้ฮิดเดนมาร์คอฟในการรู้จำท่าในภาษามือแบบไม่ต่อเนื่อง

ในการใช้ฮิดเดนมาร์คอฟในการรู้จำท่าในภาษามือแบบไม่ต่อเนื่อง (Isolated sign) สามารถอธิบายได้ดังรูปที่ 4.5 ซึ่งจากรูปแต่ละบล็อคของ Probability Computation นั้นแทนฮิดเดนมาร์คอฟโมเดล ซึ่งแต่ละโมเดลถูกสร้างจากท่ามือ 1 ท่า สำหรับโมเดลที่ i^{th} เราจะให้ λ^i แทนโมเดลของฮิดเดนมาร์คอฟโมเดล ซึ่งมีพารามิเตอร์คือ A^i, B^i, π^i ซึ่ง A^i คือ เซตของ state transition probability , B^i คือเซตของ observation probability และ π^i คือเซตของ initial state probability โดยที่สเตทของการสอนฮิดเดนมาร์คอฟโมเดลจะมีการประมาณค่าพารามิเตอร์ดังกล่าว โดยใช้บาม-เวลส์อัลกอริทึม

ในกรณีของการจำแนก เซตของข้อมูลหรือที่เรียกว่า Observation data จะถูกคัดออกจากข้อมูลที่รับมาทั้งหมดโดยถุงมืออิเล็กทรอนิกส์ ซึ่งจะเป็นข้อมูลเข้าในโมดูล Gesture Segmentation and Recognition (HMM Module) การประมาณค่าความน่าจะเป็นสูงสุดได้ใช้ Viterbi algorithm สำหรับแต่ละโมเดลที่ถูกเปรียบเทียบกับโมเดลอื่นๆ ค่าที่มีค่าสูงสุดแทนด้วย p_{max} จะถูกเลือก



รูปที่ 4.4 ตัวอย่างโทโปโลจีของแต่ละฮิดเดนมาร์คอฟโมเดลแบบ 3 สเตทซึ่ง 1 โมเดลแทน 1 คำ



รูปที่ 4.5 การใช้งานฮิดเดนมาร์คอฟโมเดล

4.2 วิธีการรู้จำภาษามือไทยชนิดต่อเนื่อง

4.2.1 การเลือกจำนวนสเตทของฮิดเดนมาร์คอฟโมเดล

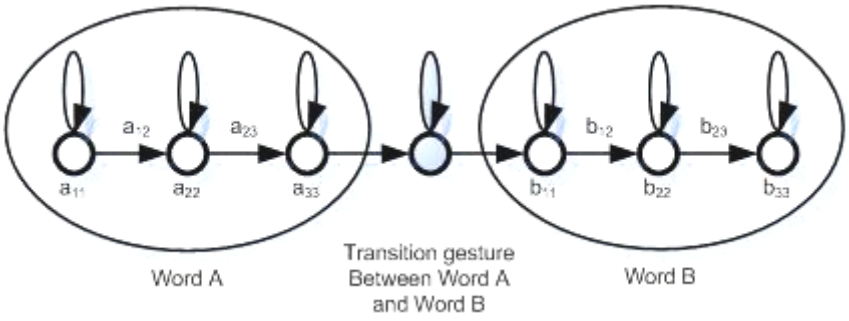
การเลือกจำนวนสเตทของฮิดเดนมาร์คอฟโมเดลนั้น จะพบปัญหา 2 กรณี คือ หากมีจำนวนสเตทน้อยเกินไป จะทำให้การรู้จำท่าที่มีความคล้ายคลึงนั้นเกิดความรู้จำผิดพลาดได้ และหากมีจำนวนสเตทมากเกินไป (Over fit) ก็จะทำให้การรู้จำท่าเดียวกันผิดพลาดได้ เนื่องจากการทำท่ามือในแต่ละครั้งในส่วนของการทำท่ามือ การหมุนมือ การเคลื่อนไหวของมืออาจจะไม่ได้เหมือนกันอย่างพอดีทุกครั้ง จากปัญหาดังกล่าวจึงได้นำการแบ่งข้อมูลเค-มีน (K-means Clustering) มาประยุกต์ใช้ในการเลือกจำนวน สเตท การแบ่งข้อมูลเค-มีนนั้นจะต้องกำหนดจำนวนกลุ่ม (K) ไว้ล่วงหน้า ซึ่งในงานวิจัยนี้ได้ทำการแบ่งกลุ่มข้อมูลในระดับคำ โดยแต่ละคำจะนำข้อมูลมาทำการเลือกการแบ่งกลุ่มออกเป็น 2, 3 และ 4 ทำการหาคู่อันดับของระหว่าง 2 กลุ่มที่มีค่าที่ใกล้ที่สุดหรือระยะห่างน้อยที่สุด (Euclidean distance) แล้วนำมาหาระยะทางระหว่างจุดสองจุด หลังจากนั้นหารด้วยผลบวกของค่า Standard Deviation ระหว่างจุดของจุดข้างต้น ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$Q = \frac{d}{STD_1 + STD_2} \tag{4.1}$$

เมื่อได้ค่า Q ของแต่ละชุดของข้อมูลในการแบ่งกลุ่ม 2, 3 และ 4 กลุ่มแล้ว จึงนำค่าทั้งหมดมาทำการหา Threshold 2 ค่า เพื่อทำการแบ่งกลุ่มค่า Q ที่ได้นี้ออกเป็น 3 กลุ่ม หลังจากนั้นทำการทดสอบกับข้อมูลในระดับคำ เพื่อทดสอบว่ากลุ่มใดที่มีความถูกต้องมากที่สุด แล้วจึงนำกลุ่มที่มีความถูกต้องมากที่สุดนี้ ไปทำการทดสอบการรู้จำในระดับประโยคต่อไป

4.2.2 การสร้างโมเดลของการเคลื่อนไหวระหว่างท่ามือ

เนื่องจากการเคลื่อนไหวระหว่างท่ามือนั้นเป็นท่าที่ไม่มีความหมาย ซึ่งเป็นการเคลื่อนไหวของมือระหว่างท่าที่มีความหมาย 2 ท่า แสดงได้ดังรูปที่ 4.6 จากงานวิจัย [2] ต้องทำการเก็บข้อมูลของการเคลื่อนไหวระหว่างท่ามือลงในฐานข้อมูล แล้วทำการสอนให้กับระบบ ซึ่งทำให้ฐานข้อมูลมีขนาดใหญ่ขึ้น และคำนวณซับซ้อนขึ้นเนื่องจากต้องสอนให้กับระบบทุกๆท่าเคลื่อนไหวระหว่างมือ



รูปที่ 4.6 ท่าเคลื่อนไหวระหว่างท่ามือซึ่งอยู่ระหว่างคำ A และ คำ B

ซึ่งในงานวิจัยนี้จึงขอนำเสนอ วิธีในการสร้างฮิดเดนมาร์คอฟโมเดล ของการเคลื่อนไหวระหว่างท่ามือ (Transition movement) ด้วยการคำนวณ โดยสร้างโมเดลแบบสเตทเดียว เพื่อการลดความจำเป็นในการที่ต้องเก็บข้อมูลลงในฐานข้อมูล ซึ่งทำให้ต้องเสียพื้นที่ในการจัดเก็บข้อมูล อีกทั้งการเก็บข้อมูลในแต่ละท่านั้นต้องใช้เวลาและต้องนำมาสอนให้แก่ระบบ หากโมเดลที่ได้นั้นยังไม่เหมาะสมก็ต้องทำการคำนวณใหม่ ซึ่งทำให้การคำนวณนั้นซับซ้อนมากขึ้น และการสร้างโมเดลจากการเก็บข้อมูลเข้าไปนั้น ข้อมูลที่สร้างจะอยู่ในฐานข้อมูลตลอดเวลา ซึ่งวิธีการที่งานวิจัยนี้ได้นำเสนอนั้น จะทำการสร้างโมเดลของการเคลื่อนไหวระหว่างท่าด้วยคำนวณพารามิเตอร์จากสเตทสุดท้ายของท่าก่อนหน้าร่วมกับสเตทแรกของท่าถัดไป ซึ่งในขณะที่โปรแกรมทำงานก็จะทำการสร้างขึ้นเป็นการชั่วคราว เพื่อนำมาใช้ในการหาเส้นทางที่ดีที่สุดสำหรับการรู้จำภาษามือไทยชนิดต่อเนื่องอัตโนมัติ และหลังจากนั้นระบบก็จะทำการลบทิ้งไป ทำให้ประหยัดพื้นที่ในการเก็บข้อมูลและหน่วยความจำ

ซึ่งการสร้างโมเดลนั้นประกอบด้วยพารามิเตอร์ (π, A, μ, **U**) โดย π คือ ค่าเริ่มต้นของการเปลี่ยน สเตทจากจุดแรก, A คือ เมตริกซ์ความน่าจะเป็นในการเปลี่ยนสเตท, μ คือ เวกเตอร์เฉลี่ย และ **U** คือ เมตริกซ์ Covariance กำหนดให้ (π, A, μ, **U**) เป็นพารามิเตอร์ของฮิดเดนมาร์คอฟโมเดลของคำก่อนหน้า, (π', A′, μ',**U**′) เป็นพารามิเตอร์ของฮิดเดนมาร์คอฟโมเดลของคำถัดไป และ ($\overline{\pi}$, $\overline{A}$, $\overline{\mu}$, $\overline{U}$) เป็นพารามิเตอร์ของฮิดเดนมาร์คอฟโมเดลของการเคลื่อนไหวระหว่างมือที่ได้จากการคำนวณ ซึ่งในการสร้างโมเดลของท่าเคลื่อนไหวระหว่างท่ามือจะกำหนดค่าต่างๆดังต่อไปนี้

ค่าเริ่มต้นของการเปลี่ยนสเตทจะได้

$$\overline{\pi} = 1 \tag{4.2}$$

เนื่องจากโมเดลเป็นแบบสเตทเดียวดังนั้นค่าความน่าจะเป็นในการเปลี่ยนสเตทจะได้

$$\overline{A} = 1 \tag{4.3}$$

ค่าเวกเตอร์เฉลี่ยคำนวณจากค่าเวกเตอร์เฉลี่ยในสเตทสุดท้ายของคำก่อนหน้า บวกกับค่าเวกเตอร์เฉลี่ยในสเตทแรกของคำถัดไปแล้วหารด้วย 2 จะได้

$$\overline{\mu} = \frac{\mu_K + \mu'_1}{2} \tag{4.4}$$

$$\overline{U}_{11} = \frac{U_{K,11} + U'_{1,11}}{2} \tag{4.5}$$

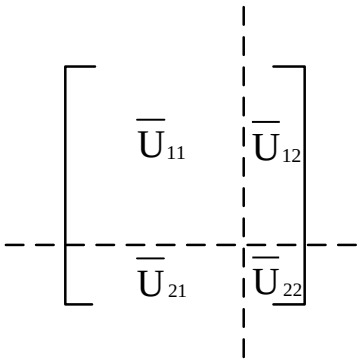
$$\overline{U}_{12} = \overline{U}_{21}^{T} = 0 \tag{4.6}$$

$$\overline{U}_{22} = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_L^2 \end{bmatrix} \tag{4.7}$$

จากสมการที่ 4.5 $\overline{U}_{11}$ คือ $M \times M$ top-left submatrix ของ $\overline{U}$ ซึ่ง M คือจำนวนข้อมูลซึ่งได้จากถุงมืออิเลคทรอนิคส์ในที่นี้คือ 18 ในทำนองเดียวกัน $U_{3,11}$ และ $U'_{1,11}$ คือ $M \times M$ top-left submatrix ของ U_3 และ U'_1 ตามลำดับ ในสมการที่ 4.6 และ 4.7 $\overline{U}_{12}$, $\overline{U}_{21}$, และ $\overline{U}_{22}$ คือ submatrice ที่เหลือของ $\overline{U}$ แสดงดังรูปที่ 4.7 ขนาด L ของ $\overline{U}_{22}$ นั้นเท่ากับจำนวนข้อมูลซึ่งได้จากอุปกรณ์ตรวจวัดตำแหน่งแบบ 6 แกนซึ่งในที่นี้คือ 6 และค่าความแปรปรวนในสมการที่ 4.7 สามารถแสดงได้ดังสมการต่อไปนี้

$$\sigma_i^{\,2} = \{\left|\mu_{3,M+i} - \mu'_{1,M+i}\right| / \alpha\}^2 \tag{4.8}$$

ซึ่ง $\mu_{3,M+i}$ คือ ค่าเฉลี่ยที่อยู่ในโมเดลซึ่งเป็นสเตทสุดท้ายของคำแรก เช่นเดียวกับข้อมูลลำดับที่ i^{th} ที่ได้จากเซนเซอร์ และ $\mu'_{1,M+i}$ ก็เช่นเดียวกัน

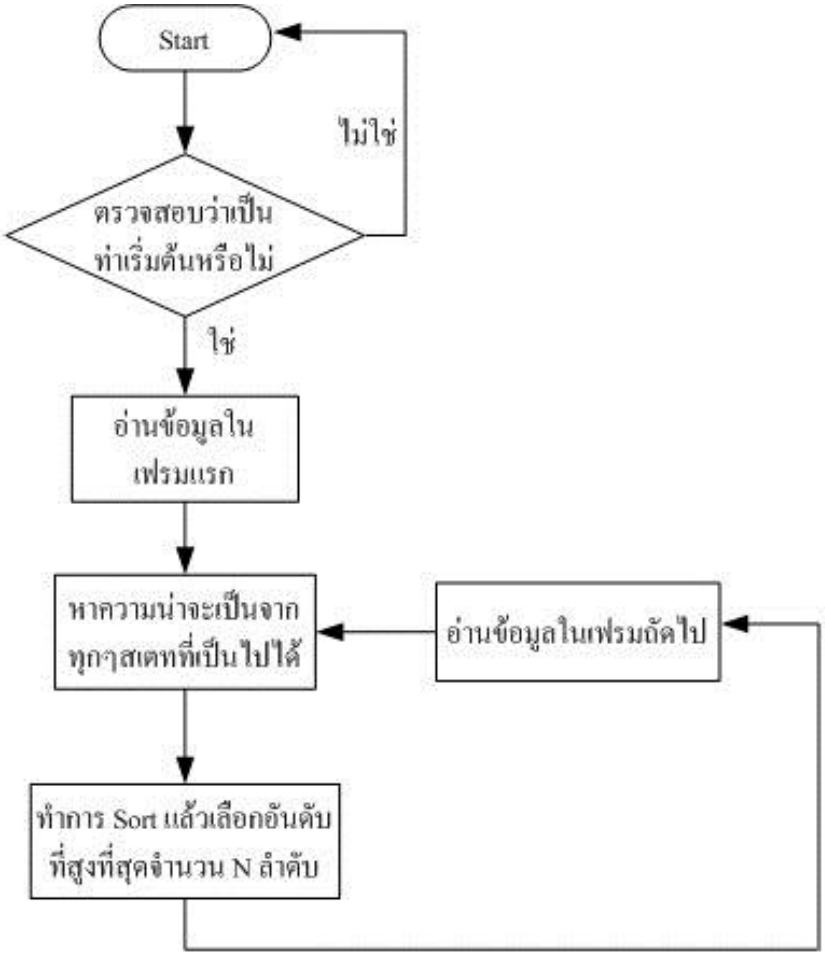


รูปที่ 4.7 แสดงการจำแนกของ $\overline{U}$ ออกเป็น 4 submatrice

4.2.3 การรู้จำด้วย Viterbi algorithm

การจดจำภาษามือนั้น เป็นการหาลำดับของภาษามือที่เหมาะสมกับข้อมูลภาษามือ โดยใช้ข้อมูลโมเดลที่ได้จากการประมาณจากขั้นตอนการสอน ซึ่งจะทำการหาลำดับของภาษามือที่มีความน่าจะเป็นที่สูงที่สุด ในการหาลำดับของภาษามือที่น่าจะเป็นไปได้ จะทำการหาได้โดยพิจารณาลำดับที่เป็นไปได้ของทุกสเตทและทุกโมเดล ซึ่งเป็นการทำให้การประมวลผลในขั้นตอนนี้มีจำนวนมากและจะทำให้กระบวนการในรู้จำนั้นช้า ดังนั้นเพื่อที่จะทำการลดปัญหาจำนวนในการค้นหาจึงใช้ Viterbi algorithm ซึ่งเป็นหลักการพื้นฐานช่วยในการเลือกเส้นทางที่ดีที่สุด ทำให้ไม่ต้องค้นหาในทุกๆเส้นทางได้ ซึ่งขั้นตอนนั้นเริ่มแรกจะทำการพิจารณาข้อมูลภาษามือที่เวลา t=1 ว่ามีความน่าจะเป็นอยู่ในสเตทแรกของโมเดลภาษามือโมเดลใด ต่อไปจะทำการพิจารณาข้อมูลภาษามือที่เวลา t ต่อไป โดยคำนวณจนกระทั่งครบจำนวนข้อมูลภาษามือที่เวลา t_n โดยพิจารณาจากสเตทก่อนหน้าซึ่งการทำแบบนี้จะมีลักษณะต่อเนื่องเป็นกราฟต้นไม้ (Tree graph) และเมื่อมาถึงเฟรมสุดท้าย ณ เวลา t_n ของการค้นหา จะทำการเลือกเส้นทางที่มีความน่าจะเป็นที่สูงที่สุด นอกจากนี้ Viterbi ยังลดเส้นทางด้วยการลบเส้นทางที่มีปลายทางเหมือนกันอีกด้วย โดยจะเลือกจากลำดับสเตทที่มีความน่าจะเป็นสูงที่สุด

จากรูปที่ 4.7 ได้แสดงการรู้จำด้วย Viterbi algorithm ซึ่งในงานวิจัยนี้ได้ทำการกำหนดจำนวนลำดับที่จะจัดเก็บลงในหน่วยความจำ โดยกำหนดให้ N แทนจำนวนลำดับดังกล่าว เพื่อช่วยในการลดการจัดเก็บในหน่วยความจำ โดยทำการหาความน่าจะเป็นจากทุกๆสเตทที่เป็นไปได้ในแต่ละเฟรม แล้วมาทำการจัดเรียง (Sort) และเลือกอันดับที่สูงที่สุด N ลำดับ แล้วจึงทำการการอ่านข้อมูลในเฟรมถัดไป ทำเช่นนี้ไปจนหมดข้อมูล



รูปที่ 4.8 แผนภาพแสดงการรู้จำด้วย Viterbi algorithm