บทที่ 3

การประเมินความเชื่อถือได้

ตามที่กล่าวไว้ในบทที่ 1 ว่าวิธีการคำนวณความเชื่อถือได้ของระบบไฟฟ้ากำลังตลอดจนสถานีไฟฟ้ากำลังนั้นประกอบด้วยวิธีการวิเคราะห์ และวิธีการจำลองเหตุการณ์ แต่ละวิธีมีข้อเสียและข้อด้อยต่างกัน ในบทนี้จะเป็นการนำเสนอถึงรายละเอียดของวิธีการทั้งสองดังต่อไปนี้

3.1 วิธีการวิเคราะห์ (Analytical method)

วิธีวิเคราะห์เป็นวิธีที่อาศัยแบบจำลองการทำงานของอุปกรณ์แล้วคำนวณดัชนีความเชื่อถือได้ตามสมการคณิตศาสตร์ และถือเป็นวิธีที่ให้ผลถูกต้องแม่นยำ [5] สำหรับวิธีการวิเคราะห์ที่นำมาใช้ในการคำนวณดัชนีความเชื่อถือได้ของสถานีไฟฟ้านั้นพอแบ่งออกได้เป็น 4 วิธีการหลักคือ

1. วิธีการลดทอนเครือข่าย
2. วิธีความน่าจะเป็นแบบมีเงื่อนไข
3. วิธีมินิมัลคัทเซต
4. วิธีการวิเคราะห์แผนภาพต้นไม้

แต่ละวิธีมีรายละเอียดดังนี้

3.1.1 วิธีลดทอนเครือข่าย (Network reduction method) [6,18]

วิธีนี้อาศัยหลักของการต่อแบบอนุกรมและขนาน ในระบบที่มีอุปกรณ์ต่ออนุกรมกันดังรูปที่ 3.1 ก) จะใช้งานได้เมื่ออุปกรณ์ทุกตัวใช้งานได้พร้อมกัน นั่นคือ

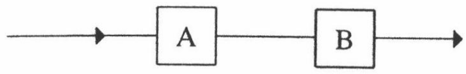
$$R_{sys} = \prod R = R_A * R_B \qquad (3.1)$$

โดย R_{sys} คือความเชื่อถือได้ของระบบ

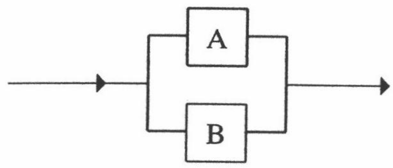
R_A คือความเชื่อถือได้ของอุปกรณ์ A

และ R_B คือความเชื่อถือได้ของอุปกรณ์ B

ระบบที่มีอุปกรณ์ต่อขนานกันดังรูปที่ 3.1 ข) จะขัดข้องเมื่ออุปกรณ์ทุกตัวเกิดขัดข้องพร้อมกันนั่นคือ



ก) ระบบอนุกรม



ข) ระบบขนาน

รูปที่ 3.1 ระบบอนุกรมและขนาน

$$Q_{sys} = \prod Q = Q_A * Q_B \tag{3.2}$$

โดย Q_{sys} คือความเสี่ยงของระบบ

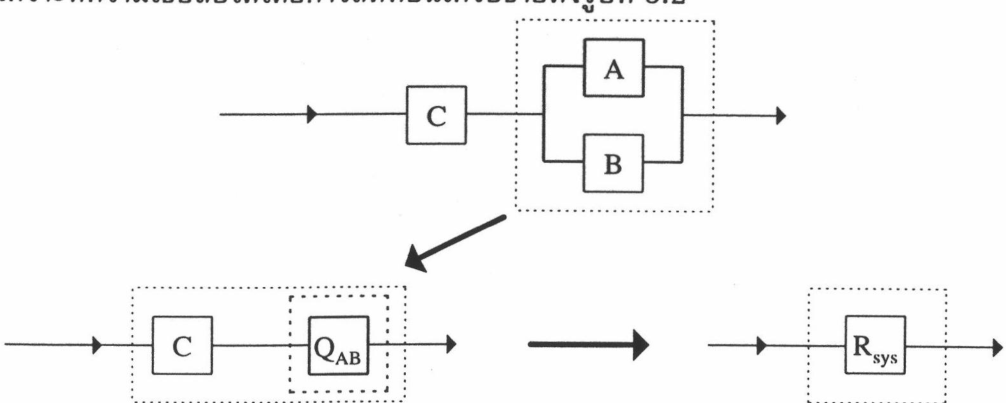
Q_A คือความเสี่ยงของอุปกรณ์ A

และ Q_B คือความเสี่ยงของอุปกรณ์ B

และทั้งระบบอนุกรมและขนานจะสามารถหา R_{sys} หรือ Q_{sys} ได้ดังนี้

$$Q_{sys} = 1 - R_{sys} \tag{3.3}$$

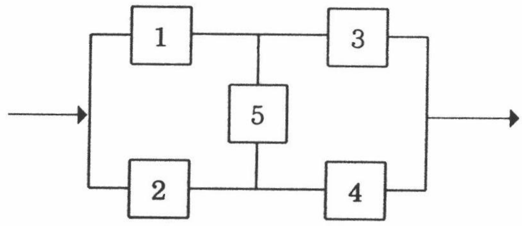
สำหรับระบบที่ประกอบด้วยการต่ออนุกรมและขนานผสมกันอยู่นั้นสามารถวิเคราะห์ความเชื่อถือได้โดยการลดทอนเครือข่ายดังรูปที่ 3.2



$Q_{AB} = Q_A * Q_B$ $\qquad$ $R_{sys} = 1 - [\, R_C * (1 - Q_{AB}) \,]$

รูปที่ 3.2 ตัวอย่างการยุบส่วนของระบบที่ต่อแบบขนานและอนุกรม

อย่างไรก็ตามในระบบที่ซับซ้อนขึ้นเช่นในรูปที่ 3.3 จะไม่สามารถวิเคราะห์โดยวิธีลดทอนเครือข่ายนี้ได้



รูปที่ 3.3 ระบบซับซ้อน

3.1.2 วิธีความน่าจะเป็นแบบมีเงื่อนไข (Conditional probability method) [6]

การวิเคราะห์ระบบที่ซับซ้อนเช่นในรูปที่ 3.3 สามารถกระทำได้โดยอาศัยทฤษฎีความน่าจะเป็นแบบมีเงื่อนไข หากให้ P คือความน่าจะเป็นจะได้ว่า

P(ระบบใช้งานได้หรือล้มเหลว) = P(ระบบใช้งานได้หรือล้มเหลวถ้าอุปกรณ์ ‘x’ ดี) * P(อุปกรณ์ ‘x’ ดี) + P(ระบบใช้งานได้หรือล้มเหลวถ้าอุปกรณ์ ‘x’ เลว) * P(อุปกรณ์ ‘x’ เลว) (3.4)

ตัวอย่างเช่นระบบในรูปที่ 3.3 จะสามารถวิเคราะห์ R_{sys} ได้รูปที่ 3.4 และได้สมการดังนี้

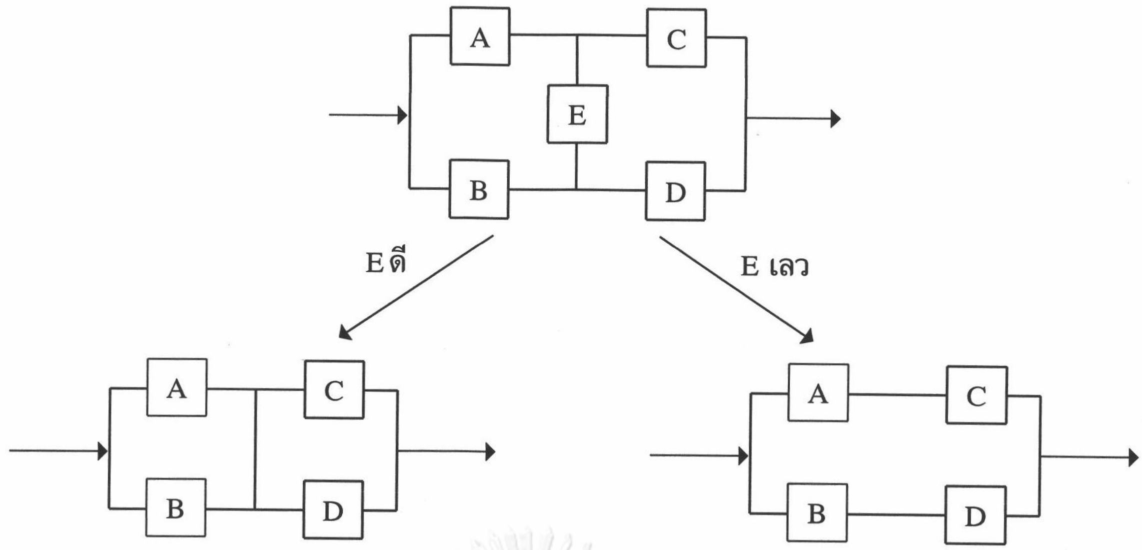
$$R_{sys}=R_{sys}(\text{ถ้า E ดี})R_E + R_{sys}(\text{ถ้า E เลว})Q_E \quad (3.5)$$

เงื่อนไข : ให้ E ดี จะได้ $R_{sys1} = (1-Q_AQ_B)(1-Q_CQ_D)$ (3.6)

เงื่อนไข : ให้ E เลว จะได้ $R_{sys2} = (1-R_AR_C)(1-R_BR_D)$ (3.7)

$$\begin{aligned}\therefore\ R_{sys} &= (1-Q_AQ_B)(1-Q_CQ_D)R_E + (1-(1-R_AR_C)(1-R_BR_D))Q_E \\ &= R_AR_C + R_BR_D + R_AR_DR_E + R_BR_CR_E - R_AR_BR_CR_D - R_AR_CR_DR_E \\ &\quad - R_AR_BR_CR_E - R_BR_CR_DR_E - R_AR_BR_DR_E + 2R_AR_BR_CR_DR_E\end{aligned}$$

(3.8)



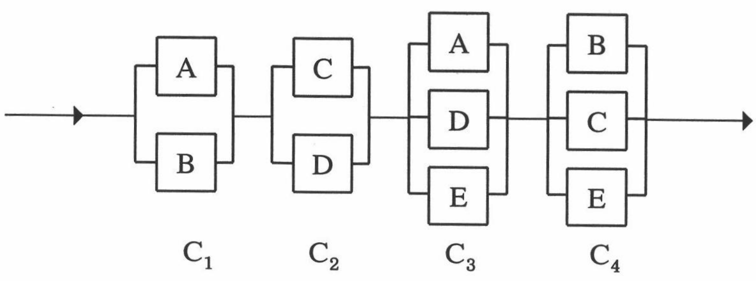
รูปที่ 3.4 การแยกเงื่อนไขเพื่อวิเคราะห์ระบบซับซ้อน

วิธีความน่าจะเป็นแบบมีเงื่อนไขนี้นับเป็นเครื่องมือที่มีประโยชน์มากในการวิเคราะห์ความเชื่อถือได้ในด้านของความแม่นยำ แต่วิธีดังกล่าวไม่เหมาะสมต่อการนำไปพัฒนาใช้กับโปรแกรมคอมพิวเตอร์สำหรับแก้ไขปัญหาที่เป็นกรณีทั่ว ๆไป

3.1.3 <u>วิธีมินิมัลคัตเซต (Minimal cut set method)</u> [6,18]

คัตเซต คือ กลุ่มอุปกรณ์ของระบบซึ่งเมื่อเกิดการล้มเหลวหรือขัดข้องแล้วทำให้ระบบล้มเหลวหรือไม่สามารถทำงานได้ตามไปด้วย ส่วนมินิมัลคัตเซต คือ คัตเซตที่เล็กที่สุดที่เป็นกลุ่มอุปกรณ์ของระบบซึ่งเมื่อเกิดการล้มเหลวขึ้นแล้วทำให้ระบบล้มเหลวด้วย และหากอุปกรณ์ตัวใดตัวหนึ่งในกลุ่มนั้นใช้งานได้ ระบบก็จะไม่ล้มเหลว หรือกล่าวได้ว่าอุปกรณ์ทุกตัวในมินิมัลคัตเซตจะต้องล้มเหลวทั้งหมดจึงจะทำให้ระบบล้มเหลว

ตัวอย่างเช่นระบบในรูปที่ 3.3 จะมีมินิมัลคัตเซตทั้งหมด 4 ชุดดังนี้ AB, CD, AED และ BEC และจากนิยามของมินิมัลคัตเซตกับหลักการของระบบอนุกรมและขนานจะได้ดังรูปที่ 3.5 โดย C_1 C_2 C_3 และ C_1 คือมินิมัลคัตเซตที่ 1 2 3 และ 4 ตามลำดับ



รูปที่ 3.5 มินิมัลคัตเซตของระบบในรูปที่ 3.3

ความเสี่ยงคำนวณได้จาก

$$Q_{sys} = P(C_1 \cup C_2 \cup C_3 \ldots \cup C_n) \tag{3.9}$$

$$\begin{aligned} Q_{sys} &= P(C_1 \cup C_2 \cup C_3 \ldots \cup C_n) \\ &= P(C_1)+P(C_2)+P(C_3)+P(C_4)-P(C_1 \cap C_2)-P(C_1 \cap C_3) \\ &\quad -P(C_1 \cap C_4)-P(C_2 \cap C_3)-P(C_2 \cap C_4)-P(C_3 \cap C_4) \\ &\quad +P(C_1 \cap C_2 \cap C_3)+P(C_1 \cap C_2 \cap C_4)+P(C_1 \cap C_3 \cap C_4) \\ &\quad +P(C_2 \cap C_3 \cap C_4)-P(C_1 \cap C_2 \cap C_3 \cap C_4) \end{aligned} \tag{3.10}$$

โดย $P(C_1) = Q_A Q_B$

$$\begin{aligned} P(C_2) &= Q_C Q_D \\ P(C_3) &= Q_A Q_D Q_E \\ P(C_4) &= Q_B Q_C Q_E \\ P(C_1 \cap C_2) &= P(C_1)P(C_2) = Q_A Q_B Q_C Q_D \\ P(C_1 \cap C_3) &= P(C_1)P(C_3) = Q_A Q_B Q_D Q_E \\ P(C_1 \cap C_4) &= P(C_1)P(C_4) = Q_A Q_B Q_C Q_E \\ P(C_2 \cap C_3) &= P(C_2)P(C_3) = Q_A Q_C Q_D Q_E \\ P(C_2 \cap C_4) &= P(C_2)P(C_4) = Q_B Q_C Q_D Q_E \\ P(C_3 \cap C_4) &= P(C_3)P(C_4) = Q_A Q_B Q_C Q_D Q_E \\ P(C_1 \cap C_2 \cap C_3) &= P(C_1 \cap C_2 \cap C_4) \\ &= P(C_1 \cap C_3 \cap C_4) \\ &= P(C_2 \cap C_3 \cap C_4) \\ &= P(C_1 \cap C_2 \cap C_3 \cap C_4) \\ &= Q_A Q_B Q_C Q_D Q_E \end{aligned}$$

ดังนั้น

$$\begin{aligned} Q_{sys} &= Q_A Q_B + Q_C Q_D + Q_A Q_D Q_E + Q_B Q_C Q_E - Q_A Q_B Q_C Q_D \\ &\quad - Q_A Q_B Q_D Q_E - Q_A Q_B Q_C Q_E - Q_A Q_C Q_D Q_E \\ &\quad - Q_B Q_C Q_D Q_E + 2 Q_A Q_B Q_C Q_D Q_E \end{aligned} \tag{3.11}$$

ในระบบที่มีจำนวนมินิมัลคัตเซตมาก วิธีการดังกล่าวจะไม่ค่อยสะดวกนัก จึงได้มีการประมาณจากวิธีนี้ จากสมการที่ 3.9 เมื่อวิเคราะห์โดยประมาณจะได้ว่า

$$Q_{sys} = P(C_1) + P(C_1) + P(C_1) + \dots + P(C_i) + \dots + P(C_n)$$

$$= \sum_{i=1}^{n} P(C_i) \qquad (3.12)$$

จากตัวอย่างในสมการที่ 3.11 ถ้าใช้วิธีโดยประมาณจะเหลือดังสมการที่ 3.13

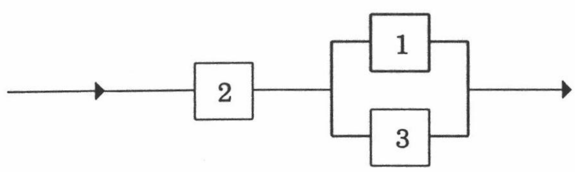
$$Q_{sys} = Q_A Q_B + Q_C Q_D + Q_A Q_D Q_E + Q_B Q_C Q_E \qquad (3.13)$$

3.1.4 วิธีวิเคราะห์แผนภาพต้นไม้แสดงการล้มเหลว (Fault tree analysis method) [1,2]

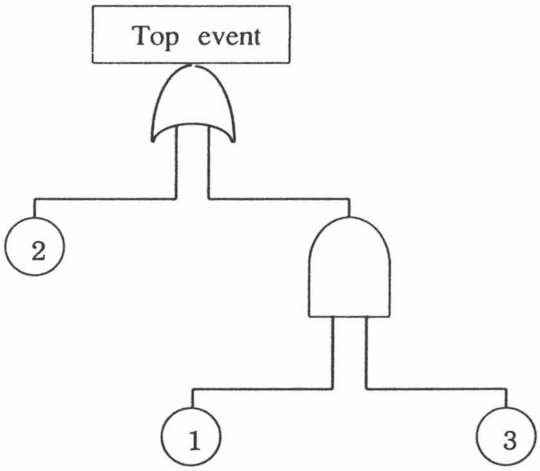
เริ่มต้นจากการวิเคราะห์ Fault tree ซึ่งอาศัยหลักการของ Logic gate โดยมีการประยุกต์เข้ากับความรู้ทางการคำนวณความเชื่อถือได้ดังรูปที่ 3.6 ข้อมูลเข้าเกทแต่ละตัวคือเหตุการณ์พื้นฐาน (Basic event) k เหตุการณ์ โดยมีผลลัพธ์ (Output) คือการขาดพลังงานไฟฟ้าทางด้านขาออกของสถานีไฟฟ้า ซึ่งเรียกว่า Top event หรือ Fault event ดังตัวอย่างในรูปที่ 3.7 และ 3.8

$U = U_1 U_2 U_3$ $\qquad$ $U = 1 - (1-U_1)(1-U_2)(1-U_3)$

รูปที่ 3.6 การใช้ OR gate และ AND gate ในการคำนวณค่า U



รูปที่ 3.7 ระบบตัวอย่าง



รูปที่ 3.8 แผนภาพต้นไม้แสดงการล้มเหลวของระบบ

โดยที่ หมายถึง ผลลัพธ์ที่ได้จากเกท (กรณีที่ใช้กับ Logic gates)

หมายถึง เหตุการณ์พื้นฐาน (กรณีที่ใช้กับ Logic gates)

หมายถึง เกท 'OR'

หมายถึง เกท 'AND'

และ U หมายถึง ความไม่พร้อมมูล

ทั้งนี้แผนภาพสัญลักษณ์ใน Fault tree จะมีความหมายในตัวเองที่นอกเหนือจากแผนภาพในทาง Logic gate เป็นต้นว่า รูปสี่เหลี่ยมผืนผ้าจะหมายถึงผลลัพธ์ของเกทโดยไม่จำเป็นต้องเป็นผลลัพธ์ของระบบทั้งหมด รูปวงกลมจะหมายถึงความล้มเหลวของอุปกรณ์ซึ่งไม่สามารถแตกกระจายลงไปได้อีกหรือก็คือเหตุการณ์พื้นฐานนั่นเอง และรูปอื่น ๆซึ่งไม่ได้แสดงไว้ในที่นี้ วิธีการนี้มีข้อดีในด้านความเป็นระบบในการวิเคราะห์ กล่าวคือเมื่อมีข้อมูลเข้าก็สามารถใส่ในแผนผังแล้วสามารถคำนวณค่าได้ทันที แต่มีข้อเสียคือในกรณีที่ไม่ทราบแผนภาพต้นไม้การล้มเหลวจะทำให้การวิเคราะห์ไม่สะดวกนักเนื่องจากจะต้องมาสร้างแผนภาพดังกล่าว และหากระบบซับซ้อนก็จะสร้างแผนภาพต้นไม้ได้ลำบาก

ในงานวิจัยนี้ได้ใช้วิธีมินิมัลคัตเซตเนื่องจากวิธีนี้เป็นวิธีที่สะดวกในการพัฒนาโปรแกรมคอมพิวเตอร์ ไม่ถูกจำกัดด้วยความยากและความซับซ้อนของระบบดังเช่นในวิธีอื่น ๆ ดังที่นำเสนอไปแล้วข้างต้น

3.2 วิธีจำลองเหตุการณ์ (Simulation method)

วิธีจำลองเหตุการณ์เป็นการอาศัยโปรแกรมคอมพิวเตอร์ในการสุ่มผ่านแบบจำลองความน่าจะเป็นซ้ำกันหลาย ๆครั้ง โดยทั่วไปวิธีนี้เป็นมีความสะดวกและสามารถใช้กับระบบที่ซับซ้อนหรือมีการทำงานที่ขนาดใหญ่มาก ๆ และวิธีการตัดสินใจจากประสบการณ์ (Deterministic method) ไม่สามารถหรือไม่สะดวกที่จะใช้ในการแก้ปัญหา [1, 11, 12]

วิธีจำลองเหตุการณ์ซึ่งมักอาศัยวิธีการมอนติคาร์โล [1, 5, 7] นั้นพอจะแบ่งออกเป็น 3 วิธีการย่อยคือ

1. Direct Monte Carlo Simulation
2. Restriced-Sampling Monte Carlo Simulation
3. Dagger Sampling Monte Carlo Simulation

สัญลักษณ์สำหรับใช้ในการอธิบายวิธีจำลองเหตุการณ์และรายละเอียดของแต่ละวิธีของการจำลองเหตุการณ์มีดังต่อไปนี้

k = จำนวนเหตุอุปกรณ์

x_i = สถานะของเหตุการณ์พื้นฐาน (State of basic event i)

= $\begin{cases} 1, & \text{ถ้าเหตุการณ์ } i \text{ เกิด} \\ 0, & \text{ถ้าเหตุการณ์ } i \text{ ไม่เกิด} \end{cases}$

$\mathbf{x}$ = $(x_1 ,..., x_k)$ เป็นเวกเตอร์แสดงสถานะของเหตุการณ์พื้นฐาน (Basic event state vector)

$\mathbf{b}$ = $(b_1 ,..., b_k)$ เป็นเวกเตอร์ไบนารี (Binary vector)

$\psi(\mathbf{x})$ = ฟังก์ชันไบนารีซึ่งแสดง Top event ของ Fault tree

= $\begin{cases} 1, & \text{ถ้า Top event เกิด} \\ 0, & \text{ถ้า Top event ไม่เกิด} \end{cases}$

ψ_L = ฟังก์ชันไบนารีของขีดจำกัดล่าง

ψ_U = ฟังก์ชันไบนารีของขีดจำกัดบน

N = จำนวนครั้งที่สุ่มในวิธีมอนติคาร์โล

n = จำนวนครั้งที่ระบบล้มเหลว

c_i = ตัวอย่างที่สุ่มขึ้นมาอย่างอิสระ (Independent sample) ในครั้งการสุ่มที่ i

$\mathbf{c}_V$ = เวกเตอร์ของตัวอย่างที่สุ่มขึ้นมาของการสุ่มครั้งที่ V สำหรับวิธี Direct Monte Carlo Simulation

s_V = เวกเตอร์ของตัวอย่างที่สุ่มขึ้นมาของการสุ่มครั้งที่ V สำหรับวิธี Restriced - Sampling Monte Carlo Simulation

z_V = เวกเตอร์ของตัวอย่างที่สุ่มขึ้นมาของการสุ่มครั้งที่ V สำหรับวิธี Dagger Sampling Monte Carlo Simulation

U = ค่าความไม่พร้อมมูล (Unavailability)

U_s = ค่าความไม่พร้อมมูลของระบบ

U_k = ค่าความไม่พร้อมมูลของเหตุการณ์พื้นฐานที่ k

U_U = ค่าความไม่พร้อมมูลที่ขีดจำกัดบน

U_L = ค่าความไม่พร้อมมูลที่ขีดจำกัดล่าง

Pr(i) = ความน่าจะเป็นที่เป็นฟังก์ชันของ i

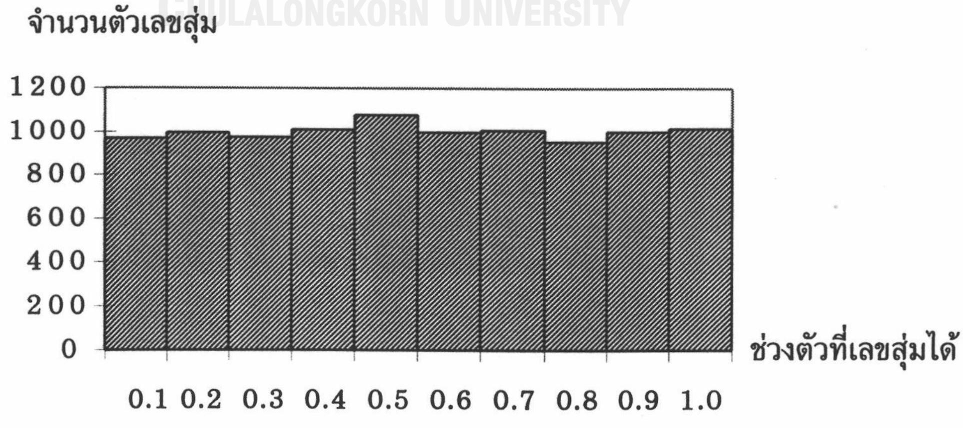
ฟังก์ชันพื้นฐานที่ใช้ในการจำลองเหตุการณ์ที่สำคัญได้แก่ฟังก์ชันที่ใช้ในการสุ่มซึ่งต้องให้การกระจายอย่างสม่ำเสมอตลอดช่วงที่สุ่ม (Uniform distribution) ตามสมการแสดงการกระจายที่ 3.14 หรือแสดงเป็นแผนภูมิดังรูปที่ 3.9

$$f(x) = a \tag{3.14}$$

โดยที่ f(x) คือ ฟังก์ชันการกระจาย

x คือ ค่าตัวแปรสุ่ม

และ a คือ ค่าคงที่



รูปที่ 3.9 แผนภูมิแท่งแสดงการกระจายของตัวเลขสุ่ม

จากรูปที่ 3.9 นี้แสดงการกระจายของตัวเลขที่สุ่มได้ โดยตัวเลขสุ่มมีจำนวน 10,000 ตัว จากการสุ่มในช่วง 0.0 ถึง 1.0 ซึ่งแบ่งออกเป็น 10 ช่วง ๆ ละ 0.1 ดังนั้นหากการกระจายมีความสม่ำเสมอค่าที่สุ่มได้ในแต่ละช่วงควรมีจำนวน 1,000 ตัวตามที่แสดงในรูปข้างต้น ข้อมูลที่แสดงในรูปนี้จากการใช้ฟังก์ชันสุ่ม random() ซึ่งอยู่ในภาษาซี

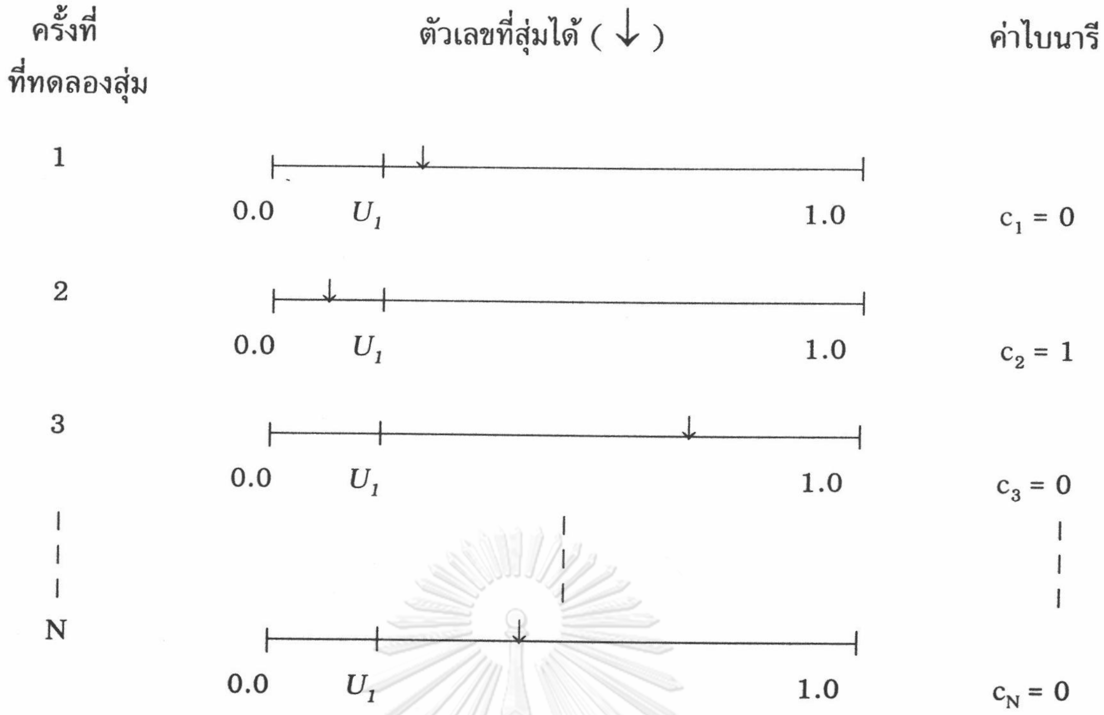
จากการกระจายอย่างสม่ำเสมอของตัวเลขสุ่มนี้ทำให้การสุ่มสามารถทำในช่วงใดก็ได้ ตัวอย่างเช่น ถ้ากล่าวถึงความน่าจะเป็นที่มีค่า 0.1 แล้ว อาจหมายถึงอัตราส่วนระหว่างจำนวนตัวเลขที่สุ่มได้ในช่วง 0.3 ถึง 0.4 ต่อจำนวนตัวเลขทั้งหมดที่สุ่ม หรืออาจหมายถึงอัตราส่วนระหว่างจำนวนตัวเลขที่สุ่มได้ในช่วง 0.8 ถึง 0.9 ต่อจำนวนตัวเลขทั้งหมดที่สุ่ม แต่เพื่อความสะดวกในการตรวจสอบเงื่อนไขในการพัฒนาโปรแกรมคอมพิวเตอร์จึงควรใช้ช่วงตั้งแต่ 0 ถึงค่าความน่าจะเป็นนั้น

โดยทั่ว ๆไปการจำลองเหตุการณ์แบบมอนติคาร์โลแบ่งได้ 3 วิธี [1] ดังนี้

3.2.1 Direct Monte Carlo Simulation

เป็นวิธีการสุ่มโดยตรงในแต่ละอุปกรณ์ (ที่มีอยู่ k ตัว) โดยให้คอมพิวเตอร์สุ่มตัวเลข (U)ในช่วง [0,1] ขึ้นมา N ครั้ง หากตัวเลขสุ่มนั้นมีค่ามากกว่า U_k จะได้ $c_i = 0$ มิเช่นนั้น $c_i = 1$ เพราะถ้า U_k มีค่ามากกว่าตัวเลขที่สุ่มขึ้นมาซึ่งตัวเลขสุ่มนี้หมายถึงค่าดัชนีที่แสดงถึงภาวะความเสี่ยงของเหตุการณ์ที่จำลองว่าเกิดขึ้นกับอุปกรณ์นั้น ๆ ก็จะหมายความว่าอุปกรณ์ตัวที่พิจารณาในแต่ละเหตุการณ์นั้นมีความเสี่ยงสูงกว่าโอกาสที่เหตุการณ์นั้นจะทำให้อุปกรณ์นั้นล้มเหลวจึงถือว่าเกิดการล้มเหลวของอุปกรณ์ขึ้น ซึ่งแทนด้วย $c_i = 1$ ดังแสดงในรูปที่ 3.9 และจาก k เหตุการณ์จะสามารถสร้าง $\mathbf{c}_1 ,..., \mathbf{c}_N$ ได้ โดยจะสุ่มทั้งหมด N*k ครั้ง จากนั้นสร้าง $\psi(\mathbf{c}_i)$ แต่ละตัว (N ตัว) โดยการพิจารณาการเกิด Top event ของแต่ละ $\mathbf{c}_i$ ทำให้สามารถคำนวณค่า U_s ได้จากสมการที่ 3.15

$$U_s = \frac{\sum_{v=1}^{N} \psi(c_v)}{N} \tag{3.15}$$



รูปที่ 3.10 การสร้างตัวอย่าง N ตัวอย่างสำหรับอุปกรณ์ตัวที่ 1

ในการประยุกต์ใช้เข้ากับระบบการจัดเรียงบัสในสถานีไฟฟ้าเช่นการจัดเรียงบัสแบบ Single bus แบบ Ring bus หรือแบบอื่นที่จะกล่าวต่อไปนั้นในแต่ละรอบ (Iteration) จะประกอบด้วยการสุ่มเชิง 2 มิติ ในที่นี้ให้เป็นแนวระดับและแนวตั้งอันหมายถึงการสุ่มเชิงจำนวนครั้งและการสุ่มเชิงอุปกรณ์ตามลำดับ ตัวอย่างเช่นถ้าระบบประกอบด้วยอุปกรณ์ 3 ตัว ได้แก่ บัส หม้อแปลง และเซอร์กิตเบรคเกอร์ ดังตารางที่ 3.1 ในการสุ่มแต่ละครั้งสำหรับอุปกรณ์แต่ละตัวจะได้ผลลัพธ์ว่าอุปกรณ์ตัวนั้นๆจะล้มเหลวหรือไม่ ถ้าล้มเหลวก็ใส่ x ในตาราง ตามตัวอย่างที่แสดงจะต้องสุ่มถึง 3N ครั้ง

ตารางที่ 3.1 แสดงตัวอย่างการจำลองเหตุการณ์โดยที่ x และ y หมายถึงการล้มเหลว

ครั้งที่สุ่ม →	1	2	3	4	5	6	7	8	9	...	N
บัส	x			x				x		...	
หม้อแปลง		x		x	x				x	...	
เซอร์กิตเบรคเกอร์	x				x		x			...	
ระบบรวม	y			y	y			y		...	

จากตารางที่ 3.1 นี้ค่า x หรือสมาชิกของเวคเตอร์ไบนารี c_V มิติ 1*N ซึ่งแสดงถึงการล้มเหลวของอุปกรณ์ ส่วนค่า y คือผลการล้มเหลวของระบบ โดยทั้ง x และ y มีค่าเป็น 1 ดังนั้น $\psi(c_V)$ จากสมการที่ 3.14 ก็คือฟังก์ชันที่ให้ผลการล้มเหลวของระบบโดยมีพารามิเตอร์คือผลการล้มเหลวของอุปกรณ์ต่าง ๆนั่นเอง ซึ่งครั้งการล้มเหลวของอุปกรณ์ที่สุ่มได้นั้นอาจทำให้ระบบรวมล้มเหลวหรือไม่ก็ได้ เช่นการล้มเหลวของหม้อแปลงเพียงตัวเดียวเช่นในครั้งสุ่มที่ 2 และ 7 ไม่ทำให้ระบบล้มเหลว แต่การล้มเหลวของบัสเพียงตัวเดียวก็ทำให้ระบบล้มเหลวได้ เช่นในครั้งสุ่มที่ 8 หรือการล้มเหลวของเซอร์กิตเบรคเกอร์ต้องล้มเหลวร่วมกับหม้อแปลงจึงจะทำให้ระบบรวมล้มเหลวเช่นในครั้งสุ่มที่ 5 กรณีเหล่านี้จะได้กล่าวละเอียดต่อไปในบทที่ 4

3.2.2. Restriced-Sampling Monte Carlo Simulation

วิธีนี้ใช้หลักการสุ่มแบบมีเป้าหมายโดยการกำหนดขีดจำกัดบน (U_U) และขีดจำกัดล่าง (U_L) กล่าวคือการสุ่มจะทำในช่วงที่ทราบค่าเบื้องต้นแทนการสุ่มตลอดช่วง [0,1] จากขีดจำกัดบน และขีดจำกัดล่างนี้จะได้ว่า

$$0 < (U_L) \leq (U_S) \leq (U_U) \leq 1 \tag{3.16}$$

วิธีนี้เป็นวิธีที่ไม่ค่อยมีประสิทธิภาพนักในการที่จะหาช่วงการสุ่ม ทั้งนี้การคำนวณ U_U และ U_L จะกระทำได้ด้วยวิธีการวิเคราะห์ จากนั้นเอาค่า U'_s ที่คำนวณได้ในเบื้องต้นนี้นำไปรวมกับ U_L จะทำให้ได้ U_s ตามต้องการ ดังนี้

$$U_s = U_s + U'_L \tag{3.17}$$

โดยที่

$$U'_s = [\, U_U - U_L \,] * \frac{\sum_{V=1}^{N} \psi(s_V)}{N}$$

3.2.3 Dagger Sampling Monte Carlo Simulation

วิธีนี้ได้รับการพัฒนามาจากวิธีที่ 1 ทำให้จำนวนครั้งในการสุ่มลดลง ตัวอย่างเช่น จากตัวอย่างที่ 1 ในเหตุการณ์ที่ 1 คอมพิวเตอร์ต้องสุ่มถึง N ครั้ง ถ้าสมมุติว่า N = 100 ครั้ง และ $U_1 = 0.01$ เราพบว่าในการสุ่ม 100 ครั้งจะพบเหตุการณ์ที่เกิดการขัดข้องจำนวน 1 ครั้ง ดังนั้นเราจึงเปลี่ยนเป้าหมายในการสุ่มจากเดิมที่เราเคยสุ่มเพื่อหาการเกิดการขัดข้องใน

ทุก ๆครั้งที่สุ่มเป็นจำนวน 100 ครั้งในเหตุการณ์ที่ 1 มาเป็นการสุ่มเพื่อหาครั้งที่ที่เกิดการขัดข้องในเหตุการณ์ที่ 1 แทนซึ่งจะมีเพียง 1 ครั้งเท่านั้น ทำให้เราสุ่มเพียงแค่ครั้งเดียวแทนที่เคยสุ่มถึง 100 ครั้งในแต่ละเหตุการณ์ และคำนวณความไม่พร้อมมูลของระบบได้ดังสมการที่ 3.18

$$U_s = \frac{\sum_{v=1}^{N} \psi(z_v)}{N} \tag{3.18}$$

ในวิทยานิพนธ์นี้ได้เลือกใช้วิธีจำลองเหตุการณ์แบบสุดท้ายตามสมการที่ 3.16 เนื่องจากวิธีดังกล่าวช่วยลดจำนวนในการสุ่ม และช่วยให้สามารถคำนวณผลลัพธ์ได้เร็วขึ้นหรือจำนวนที่วนรอบน้อยลง เนื่องจากถ้าเป็นการสุ่มโดยตรงตามแบบแรกนั้นถ้าความน่าจะเป็นมีค่า 0.01 ไม่ได้หมายความว่าในการสุ่ม 1,000 ครั้งจะพบเหตุการณ์ล้มเหลว 10 ครั้งเสมอไป แต่ถ้าเป็นการจำลองเหตุการณ์ในแบบสุดท้ายนี้จะแน่นอนว่าจะพบเหตุการณ์ล้มเหลว 10 ครั้ง ตัวอย่างแสดงการจำลองเหตุการณ์เดียวกับในตารางที่ 3.1 แสดงดังตารางที่ 3.2 จะเห็นว่าสามารถลดการทำงานของคอมพิวเตอร์ลงไปได้มากเนื่องจากไม่ต้องสุ่มทุกครั้งที่ต้องการหาผลการล้มเหลวแต่สุ่มหาครั้งที่ที่อุปกรณ์ล้มเหลวแทน

ตารางที่ 3.2 ตัวอย่างการจำลองเหตุการณ์ตามวิธี Dagger Sampling Monte Carlo

อุปกรณ์	ครั้งที่ที่ล้มเหลวถึงครั้งที่ N					
บัส	1	4	8	...		
หม้อแปลง	2	4	5	9	...	
เซอร์กิตเบรคเกอร์	1	5	7	...		
ระบบรวม	1	4	5	8	...	