บทที่ 2

ทฤษฎีพื้นฐาน

สำหรับการจำลองการทำงานของวงจรไฟฟ้านั้นมีทฤษฎีที่เกี่ยวข้องมากมาย ดังนั้นในบทนี้จะนำเสนอทฤษฎีที่เกี่ยวข้องในการทำวิทยานิพนธ์นี้อย่างพอสังเขป โดยจะแยกอธิบายเป็น 3 หัวข้อใหญ่ๆ คือ *ทฤษฎีในการวิเคราะห์วงจรไฟฟ้า*, *ทฤษฎีทางด้านวิธีเชิงตัวเลข (Numerical Method)* และ *วงจรเชิงเส้นแบบท่อน* โดยที่จะนำเสนอเนื้อหาตั้งแต่ การกำหนดตัวแปรของวงจร การสร้างสมการวงจร และการแก้ปัญหาวงจร ซึ่งขั้นตอนวิธีทั้งหมดนี้เป็นวิธีจำลองเชิงตัวเลขที่ถูกเลือกใช้ในโปรแกรมวิเคราะห์วงจรไฟฟ้า “เล็ก 6.0” และเป็นขั้นตอนวิธีเดียวกับที่ใช้ในโปรแกรมจำลองเชิงตัวเลขอื่นทั่วไป เช่น SPICE

2.1 <u>ทฤษฎีในการวิเคราะห์วงจรไฟฟ้า</u>

2.1.1 ตัวแปรของวงจร

ค่าตัวแปรอิสระทุกตัวที่โปรแกรมจำลองการทำงานของวงจรอิเล็กทรอนิกส์ใช้ในการเฉลยเพื่อหาคำตอบของวงจร เรียกว่า ตัวแปรวงจร ซึ่งมีหลายประเภท ได้แก่

- แรงดันปม (Node voltage)
- แรงดันกิ่ง (Branch voltages)
- กระแสกิ่ง (Branch currents)
- กระแสวงรอบ (Loop currents) ฯลฯ

แต่ตัวแปรวงจรแบบที่นิยมใช้กันมากที่สุดคือ แรงดันโนด โดยที่แรงดันโนดคือ ค่าแรงดันระหว่างโนดนั้นและกราวด์โนด (Ground Node) เนื่องจากโดยทั่วไปแล้วแรงดันโนดคือตัวแปรที่ผู้ใช้ต้องการทราบค่า

2.1.2 การสร้างสมการเมตริกซ์ของวงจรด้วยวิธีโมดิฟายด์โนดัล

ในทางทฤษฎีวงจร (Circuit theory) วงจรไฟฟ้าวงจรหนึ่งอาจมีสมการวงจรได้หลายแบบขึ้นอยู่กับการกำหนดตัวแปรของวงจรและกฎทางไฟฟ้าที่ใช้ในการสร้างสมการ วิธีสร้างสมการที่ใช้กันอยู่ทั่วไปมี 4 วิธีตามตารางที่ 2.1

ชื่อวิธี	ตัวแปรของวงจร	กฎทางไฟฟ้า
Node Analysis	Node voltage	กฎกระแส (Kirchoff current Law)
Mesh Analysis	Mesh current	กฎแรงดัน (Kirchoff voltage Law)
Loop Analysis	Loop current	กฎแรงดัน
Cut-set Analysis	Tree branch voltage	กฎกระแส

ตารางที่ 2.1 ตารางเปรียบเทียบวิธีต่างๆในการสร้างสมการวงจร

ดังที่ได้กล่าวมาแล้วว่าวิธี Node Analysis เป็นวิธีที่ใช้กันมากที่สุดแต่วิธีนี้มีจุดอ่อนตรงที่ไม่สามารถนำไปใช้โดยตรงกับวงจรบางชนิดได้ เช่นวงจรที่มีแหล่งกำเนิดแรงดันต่ออยู่ ดังตัวอย่างวงจรในรูปที่ 2.1 การสร้างสมการวงจรของวงจรในรูปที่ 2.1 ด้วยวิธี Node Analysis นั้นจำเป็นต้องอาศัยการแปลงแหล่งกำเนิดแรงดันให้เป็นแหล่งกำเนิดกระแสก่อนจึงสามารถสร้างสมการได้ จากสาเหตุข้างต้นจึงได้มีผู้คิดค้นวิธีสร้างสมการแบบอื่นๆ โดยอาศัยการดัดแปลงจากวิธี Node Analysis และให้ชื่อว่า วิธีโมดิฟายด์โนดัล (Modified Nodal) วิธีนี้จะยอมให้กระแสของตัวเหนี่ยวนำเป็นตัวแปรของวงจรได้โดยไม่ต้องมีการแปลงวงจรเดิมเสียก่อน ทำให้สะดวกต่อการนำไปเขียนโปรแกรม ดังแสดงไว้ในหัวข้อถัดไป

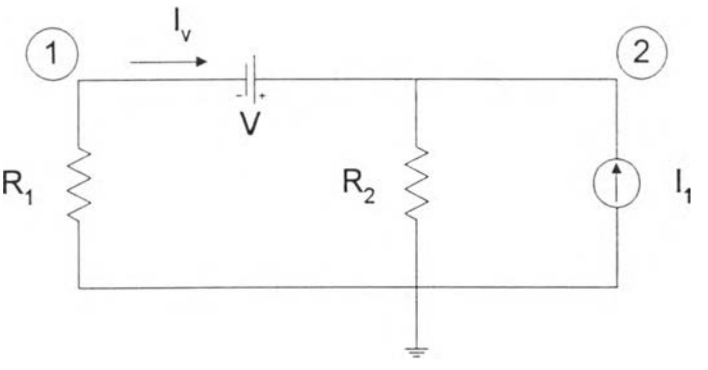
2.1.3 ตัวอย่างในการสร้างสมการเมตริกซ์ของวงจรไฟฟ้า

จากวงจรในรูปที่ 2.1 ซึ่งเป็นวงจรขนาด 2 โนดที่ไม่สามารถใช้ได้โดยตรงกับวิธี Node Analysis ถ้าใช้วิธีโมดิฟายด์โนดัล (Modified Nodal) ในการสร้างสมการไฟฟ้าซึ่งเป็นวิธีที่ยอมให้ค่ากระแสผ่านแหล่งกำเนิดแรงดันเป็นตัวแปรของวงจรด้วยนั้น จะกำหนดให้มีตัวแปรวงจร 3 ตัว คือ ค่าแรงดันที่โนด 1 (V_1) ค่าแรงดันที่โนด 2 (V_2) และค่ากระแสที่ไหลผ่านแหล่งกำเนิดแรงดัน (I_V) และสมการวงจรที่สร้างโดยวิธีโมดิฟายด์โนดัลนั้นสามารถเขียนได้ดังนี้

KCL_1: $$\frac{V_1}{R_1} + I_V = 0 \qquad (2.1)$$

KCL_2: $$\frac{V_2}{R_2} - I_V = I_1 \qquad (2.2)$$

BR_V: $$-V_1 + V_2 = V \qquad (2.3)$$



รูปที่ 2.1 ตัวอย่างวงจรที่ไม่สามารถใช้วิธี Node Analysis สร้างสมการโดยตรงได้

จากตัวอย่างนี้จะพบว่าเราสามารถใช้วิธีโมดิฟายด์โนดัลในการสร้างสมการวงจรได้ และหลังจากที่สามารถสร้างสมการวงจรได้แล้ว ต่อไปจะเป็นขั้นตอนในการแก้สมการ โดยที่เนื้อหาโดยรวมจะเป็นเนื้อหาในส่วนของทฤษฎีทางด้านเชิงตัวเลข (Numerical Methods) ซึ่งจะได้กล่าวไว้ในหัวข้อถัดไป

2.2 <u>ทฤษฎีทางด้านเชิงตัวเลข</u>

สำหรับในการแก้ปัญหาวงจรนั้น เราจะสามารถเขียนสมการวงจรให้อยู่ในรูปของสมการเมตริกซ์ Ax=b ได้เสมอ โดยที่ x เป็นเวกเตอร์ของตัวแปรวงจร ส่วนเมตริกซ์ A และเวกเตอร์ b นั้นเป็นค่าคงที่ เพื่อช่วยให้เข้าใจที่มาของสมการวงจรได้ง่ายขึ้น เราอาจจะเขียนสมการวงจรให้อยู่ในรูปของสมการเมตริกซ์ได้เป็นดังรูปที่ 2.2

$$[\mathbf{A}]_{nxn}[\mathbf{x}]_{nx1}=[\mathbf{b}]_{nx1}$$

รูปที่ 2.2 รูปแบบการเขียนสมการวงจรในรูปสมการเมตริกซ์

และเพื่อให้เข้าใจถึงการแปลงสมการวงจรให้อยู่ในรูปสมการเมตริกซ์มากยิ่งขึ้น จึงขอยกตัวอย่างสมการวงจร (2.1)-(2.3) ของวงจรในรูปที่ 2.1 ที่สร้างโดยวิธีโมดิฟายด์โนดัลดังที่ได้อธิบายมาแล้วนั้น นำมาเขียนใหม่ให้อยู่ในรูปของสมการเมตริกซ์ และผลลัพธ์ที่ได้นั้นได้แสดงไว้ในรูปที่ 2.3

$$\begin{bmatrix} 1/R_1 & 0 & 1 \\ 0 & 1/R_2 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_v \end{bmatrix} = \begin{bmatrix} 0 \\ I_1 \\ V \end{bmatrix}$$

รูปที่ 2.3 สมการเมตริกซ์ของวงจรในรูปที่ 2.1

หลังจากที่สามารถจัดรูปของสมการวงจรให้อยู่ในรูปสมการเมตริกซ์ Ax=b ได้แล้วนั้น ต่อไปจะเป็นขั้นตอนในการแก้สมการเมตริกซ์ ซึ่งมีมากมายหลายวิธี เช่น Gaussian Elimination [9] ซึ่งเป็นวิธีที่รู้จักกันแพร่หลายและเป็นที่นิยมใช้กันทั่วไป แต่ในวิทยานิพนธ์นี้จะนำเสนอวิธีในการแก้สมการเมตริกซ์อีกวิธีหนึ่ง ซึ่งเป็นวิธีที่ถูกเลือกใช้ในโปรแกรม “เล็ก 6.0” คือ วิธีการแยกตัวประกอบแอล-ยู (LU Decomposition) [9]

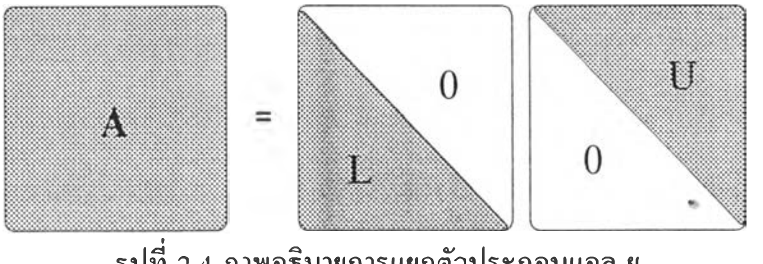
2.2.1 การแก้สมการเมตริกซ์ด้วยวิธีแยกตัวประกอบแอล-ยู

แก้สมการเมตริกซ์ Ax=b นั้นมีด้วยกันหลายวิธี แต่วิธีหนึ่งที่น่าสนใจ คือวิธีการแยกตัวประกอบแบบแอล-ยู ซึ่งจะแบ่งขั้นตอนในการแก้ปัญหาออกเป็น 2 ขั้นตอนดังนี้

ขั้นตอนที่1. Factorization จะเป็นส่วนของการคำนวณที่จะแยกเมตริกซ์ A ให้เป็นผลคูณของเมตริกซ์ L และ เมตริกซ์ U โดยที่ เมตริกซ์ L คือเมตริกซ์สามเหลี่ยมข้างล่าง และ U คือเมตริกซ์สามเหลี่ยมข้างบนที่มีค่าในแนวเส้นทะแยงมุมเป็น 1 ทั้งหมด

ขั้นตอนที่2. Substitution จะเป็นการแก้สมการ LUx=b เพื่อหาค่าเวกเตอร์ x ที่เป็นตัวแปรของวงจร โดยที่จะแบ่งขั้นตอนในการแก้สมการเป็น 2 ขั้นตอนด้วยกันคือ

- Forward sequence: Ly = b
- Backward sequence: Ux = y



รูปที่ 2.4 ภาพอธิบายการแยกตัวประกอบแอล-ยู

วิธีการแก้สมการด้วยวิธีแยกตัวประกอบแอล-ยูนี้มีข้อได้เปรียบวิธีอื่นในกรณีที่ต้องมีการแก้สมการ Ax=b หลายครั้ง โดยที่เมตริกซ์ A มีค่าคงที่ แต่เวกเตอร์ b มีค่าเปลี่ยนแปลงไป ทั้งนี้เนื่องจากไม่มีความจำเป็นต้องแยกตัวประกอบแอล-ยูใหม่ซึ่งเป็นขั้นตอนที่เสียเวลามากที่สุด (รายละเอียดเรื่องเวลาที่ใช้ไปในแต่ละขั้นตอนของการคำนวณจะแสดงไว้ในบทที่ 3) แต่อย่างไรก็ดีการใช้วิธีแยกตัวประกอบแอล-ยูในการแก้ปัญหาก็ยังมีจุดอ่อนอยู่ก็คือ สำหรับบางปัญหาแล้วการแก้สมการด้วยวิธีแยกตัวประกอบแอล-ยูนี้ อาจให้ผลลัพธ์จากการคำนวณผิดพลาดไปจากคำตอบจริง ซึ่งความผิดพลาดนี้เกิดขึ้นจากค่าผิดพลาดเนื่องจากการปัดเศษของคอมพิวเตอร์ที่ไม่สามารถเก็บรายละเอียดของตัวเลขไว้ได้ทั้งหมด อย่างไรก็ดี ความผิดพลาดนี้สามารถทำให้ลดลงได้ด้วยการใช้เทคนิคพิเศษที่เรียกว่า การทำ Complete Pivoting [10]

2.2.2 การประมาณสมการอนุพันธ์ให้เป็นสมการพีชคณิตโดยวิธี Backward Euler formula

การแก้สมการอนุพันธ์ด้วยวิธีเชิงตัวเลขเพื่อหาผลตอบสนองเชิงเวลาของวงจรอิเล็กทรอนิกส์ตั้งแต่เวลา 0 ถึงเวลา T นั้น จะเริ่มต้นด้วยการกำหนดจุดเวลา (time point) ทั้งหมดจำนวน N จุด ได้แก่ $t_0 < t_1 < t_2 < t_3 < \dots < t_N$ โดยที่ $t_0 = 0$ และ $t_N = T$ ในขั้นแรกนี้เราจะกำหนดให้จุดเวลาเหล่านี้อยู่ห่างเท่าๆกัน เท่ากับ h ซึ่งเรียกว่า ขั้นเวลา (time step) กล่าวคือ

$$t_1 - t_0 = t_2 - t_1 = t_3 - t_2 = \dots = t_N - t_{N-1} = h = T/N$$

การหาผลตอบสนองเชิงเวลาก็หมายถึงการแก้สมการอนุพันธ์ของวงจรเพื่อหาคำตอบที่จุดเวลาต่างๆเหล่านี้ ดังแสดงในรูปที่ 2.5

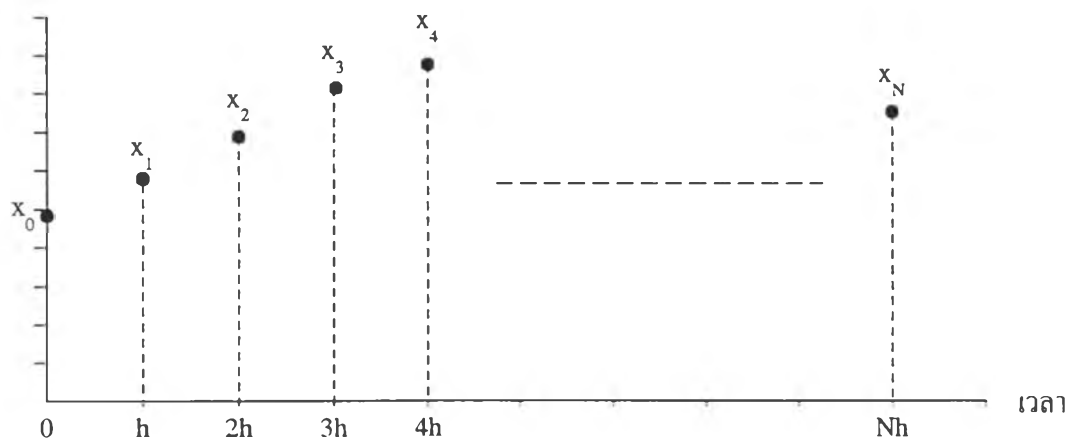
วิธีที่เราใช้ในการแก้สมการอนุพันธ์ $\dot{x}(t) = f(x,t)$ ที่จุดเวลาต่างๆนี้ ใช้การประมาณอนุพันธ์ตามสูตรที่เรียกกันทั่วไปว่า Backward Euler formula ดังต่อไปนี้

$$\dot{x}(t_i) \approx \frac{x(t_i) - x(t_{i-1})}{h}; i = 1,2,3,\dots,N \tag{2.4}$$

โดยการแทนอนุพันธ์ของตัวแปรทุกตัวในสมการวงจรด้วยการประมาณแบบ Backward Euler จะทำให้สมการอนุพันธ์ของวงจรเปลี่ยนเป็นสมการพีชคณิตล้วนๆ ตัวอย่างเช่น จากสมการที่ (2.5) ซึ่งเป็นสมการอนุพันธ์ที่พบเห็นได้ทั่วไป หลังจากผ่านการประมาณอนุพันธ์ตามสูตรของ Backward Euler formula ดังในสมการที่ (2.4) แล้วจะกลายเป็นสมการพีชคณิตดังที่แสดงไว้ในสมการที่ (2.6)

$$C\frac{dV}{dt}=\frac{V}{R} \tag{2.5}$$

$$\frac{C}{h}\left[V(t_i)-V(t_{i-1})\right]=\frac{V}{R} \tag{2.6}$$



รูปที่ 2.5 กราฟแสดงผลตอบสนองเชิงเวลาของตัวแปร x ที่จุดเวลาต่างๆ

2.2.3 การปรับขนาดขั้นเวลาอัตโนมัติ

จากในหัวข้อที่แล้วเราได้แนะนำถึงวิธีที่ใช้ในการประมาณอนุพันธ์อันดับหนึ่งให้เป็นสมการเชิงเส้นด้วยวิธี Backward Euler formula นั้น การประมาณจะมีความแม่นยำมากขึ้นเมื่อ ขนาดขั้นเวลา (h) มีขนาดเล็กลง แต่ก็หมายถึงจำนวนจุดเวลาจะเพิ่มมากขึ้น ซึ่งจะทำให้เวลาที่ใช้ในการทำการจำลองก็จะเพิ่มมากขึ้นไปด้วย ดังนั้นการกำหนดขนาดของค่า h ให้พอเหมาะจึงเป็นสิ่งที่สำคัญ คือหากเลือกขนาดขั้นเวลาที่โตเกินไปถึงแม้ว่าจะใช้เวลาในการคำนวณน้อย แต่ความแม่นยำของการประมาณจะต่ำ ในทางกลับกันถ้าเลือกขนาดขั้นเวลาที่เล็กลง ก็จะทำให้คำตอบที่ได้มีความแม่นยำมากขึ้น แต่จะต้องเสียเวลาในการคำนวณมากขึ้นตามไปด้วย

ทางเลือกทางหนึ่งที่ทำให้ได้ความแม่นยำของคำตอบสูง และการจำลองก็ใช้เวลาน้อยคือการเลือกใช้ขั้นเวลาที่เปลี่ยนได้เองแบบอัตโนมัติ ซึ่งขนาดขั้นเวลาที่เหมาะสมจะขึ้นกับลักษณะการเปลี่ยนแปลงตามเวลาของตัวแปรสถานะ x(t) ที่เราประมาณอนุพันธ์นั่นเอง กล่าวคือ ถ้า x(t) เปลี่ยนแปลงอย่างรวดเร็ว h ก็ควรมีค่าน้อยๆ แต่ถ้า x(t) เปลี่ยนแปลงช้าๆ h ก็อาจจะมีค่ามากๆได้

หลักการที่ใช้ในการปรับขนาดขั้นเวลาแบบอัตโนมัติจะพิจารณาจากความชันและอนุพันธ์อันดับสูงของตัวแปรสถานะวงจร ซึ่งมีชื่อเรียกว่า "Local Truncation Error (LTE)" โดยที่ LTE คือค่าความผิดพลาดที่เกิดจากการประมาณอนุพันธ์ให้เป็นเชิงเส้นในแต่ละจุดเวลา

เนื่องจากไม่มีวิธีเชิงเลขใดที่สามารถหาค่า x(t) จากสมการ $\dot{x}(t) = f(x,t)$ ที่ถูกต้องจริงๆ ออกมาได้ ถ้าสมมติให้ x'_{n+1} คือค่าคำตอบจริงของสมการ $\dot{x}(t) = f(x,t)$ ที่เวลา t_{n+1} และ x_{n+1} คือคำตอบของสมการอนุพันธ์ดังกล่าวที่ได้จากการคำนวณด้วยเครื่องคอมพิวเตอร์โดยใช้วิธีเชิงตัวเลขประมาณอนุพันธ์ ซึ่งจะได้ ค่าความผิดพลาดที่เกิดจากการคำนวณ (ε_{n+1}) ดังสมการ (2.7)

$$\varepsilon_{n+1} = \left\| x'_{n+1} - x_{n+1} \right\| \tag{2.7}$$

โดยที่ค่า ε_{n+1} จะแทนความผิดพลาดทั้งหมดของการประมาณ (total error) ที่เวลา t_{n+1} ซึ่งความผิดพลาดนี้จะประกอบด้วย 2 ส่วนคือ

- *round off error* คือความผิดพลาดที่เกิดจากการคำนวณของคอมพิวเตอร์ที่ไม่สามารถเก็บรายละเอียดของตัวเลขไว้ได้ทั้งหมด
- *truncation error* คือความผิดพลาดที่เกิดจากการใช้วิธีเชิงตัวเลขประมาณอนุพันธ์ให้เป็นสมการเชิงเส้น

เนื่องจากความผิดพลาดในส่วนที่เกิดจาก round off error นั้นไม่สามารถลดลงได้ ประกอบกับความผิดพลาดที่เกิดจาก round off error มีค่าน้อยมากเมื่อเทียบกับความผิดพลาดที่เกิดจาก truncation error ทำให้ ค่าความผิดพลาดจากการคำนวณ (ε_{n+1}) ส่วนใหญ่ในสมการ (2.7) นั้นเกิดจาก truncation error ดังนั้นเราจึงอาจประมาณค่าความผิดพลาด (ε_{n+1}) ได้ว่า $LTE_{n+1} \approx \varepsilon_{n+1}$ และเนื่องจากวิธี Backward Euler เป็นวิธีประมาณอนุพันธ์อันดับหนึ่ง (first order approximation) จึงสามารถพิสูจน์ [9] ได้ว่า

$$LTE_{n+1} = \left\| x'_{n+1} - x_{n+1} \right\| \approx -\frac{1}{2} h^2 \ddot{x}_{n+1} \tag{2.8}$$

$$LTE_{n+1} \approx x_{n+1} - x_n - h\dot{x}_n \tag{2.9}$$

นั่นคือในการทำการจำลองเราจะต้องลดขนาด h ลงหากว่าค่าความผิดพลาด LTE มากเกินไป แต่ถ้าค่าความผิดพลาด LTE น้อยหรืออยู่ในเกณฑ์ที่เหมาะสม เราก็อาจจะเพิ่มขนาด h เพื่อเร่งความเร็วในการจำลองได้

2.3 วงจรเชิงเส้นแบบท่อน

วงจรเชิงเส้นแบบท่อน (piecewise linear circuit) หมายถึงวงจรที่ประกอบด้วยอุปกรณ์เชิงเส้น (linear device) ซึ่งได้แก่ ตัวต้านทาน ตัวเก็บประจุ ตัวเหนี่ยวนำ แหล่งกำเนิดแรงดัน และแหล่งกำเนิดกระแส เป็นต้น และอุปกรณ์เชิงเส้นแบบท่อน (piecewise linear device) ได้แก่ ไดโอด สวิตช์ ทรานซิสเตอร์ ออปแอมป์ เป็นต้น โดยที่อุปกรณ์เชิงเส้นแบบท่อนเหล่านี้ แท้ที่จริงมีลักษณะสมบัติเป็นแบบไม่เชิงเส้น (nonlinear characteristics) แต่ถูกประมาณให้มีลักษณะสมบัติเป็นเชิงเส้นแบบท่อน ซึ่งมีความพิเศษตรงที่ว่า เราสามารถแบ่งความสัมพันธ์ (relation) ของตัวแปรในลักษณะสมบัติเชิงเส้นแบบท่อนออกเป็นย่าน (region) หลายย่าน หรือท่อน (piece) หลายท่อน โดยที่ความสัมพันธ์ในแต่ละท่อนนั้นจะเป็นแบบเชิงเส้นธรรมดา

ในการประมาณอุปกรณ์ไม่เชิงเส้นให้มีคุณสมบัติเป็นเชิงเส้นแบบท่อนนั้น ได้อาศัยตัวแปรเลขเต็ม (integer variable) ที่เรียกว่า “ตัวแปรสถานะการทำงานของอุปกรณ์ (device's status variable)” ซึ่งจะเรียกสั้นๆว่า ตัวแปรสถานะช่วยในการหาคำตอบ โดยที่จำนวนตัวแปรสถานะของแต่ละอุปกรณ์ก็แตกต่างกันไปดังนี้

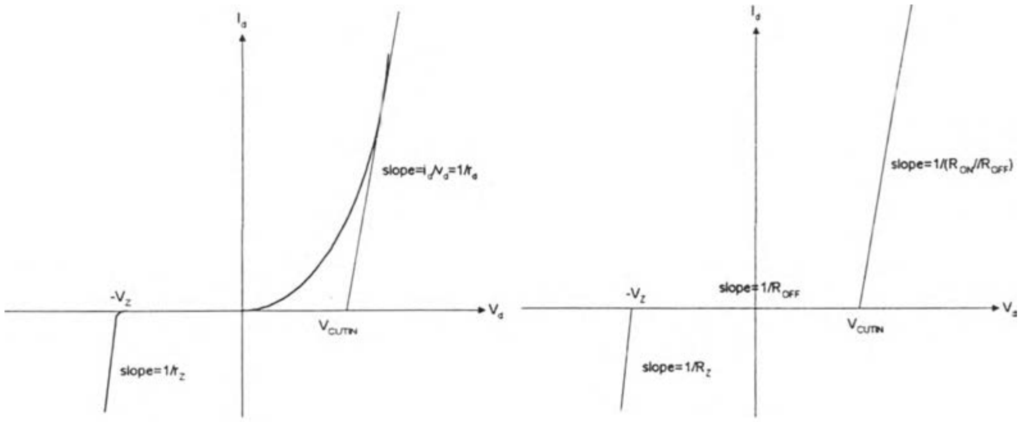
- ไดโอด มี 3 สถานะการทำงาน
- ทรานซิสเตอร์ มี 4 สถานะการทำงาน
- ออปแอมป์ มี 9 สถานะการทำงาน
- สวิตช์ มี 2 สถานะการทำงาน

2.3.1 ตัวอย่างของอุปกรณ์เชิงเส้นแบบท่อน

อุปกรณ์ไม่เชิงเส้นที่เรามักจะประมาณให้เป็นอุปกรณ์เชิงเส้นแบบท่อนนี้ได้แก่ ไดโอด ทรานซิสเตอร์ และออปแอมป์ เป็นต้น เพื่อให้เข้าใจถึงรายละเอียดของการประมาณอุปกรณ์ไม่เชิงเส้นให้เป็นอุปกรณ์เชิงเส้นแบบท่อนนั้น จะขอยกตัวอย่าง ไดโอด ซึ่งมีกราฟลักษณะสมบัติของแรงดัน-กระแสเป็นดังรูปที่ 2.6 (ก) ซึ่งเราอาจประมาณให้ไดโอดมีการทำงาน 3 สถานะด้วยกันคือ

1. สถานะพังทลาย (breakdown) เกิดเมื่อ $V_d < -V_b$
2. สถานะไม่นำกระแส (off) เกิดเมื่อ $-V_b \leq V_d < V_{on}$
3. สถานะนำกระแส (on) เกิดเมื่อ $V_d \geq V_{on}$

ซึ่งในแต่ละสถานะก็สามารถประมาณได้ด้วยลักษณะสมบัติเป็นเชิงเส้นดังในรูปที่ 2.6 (ข) ลักษณะที่ได้จากการประมาณเช่นนี้เราเรียกว่า “ลักษณะสมบัติเชิงเส้นแบบท่อน”



(ก) (ข)

รูปที่ 2.6 ลักษณะสมบัติของไดโอด (ก) ไม่เชิงเส้น (ข) เชิงเส้นแบบท่อน

2.3.2 การสร้างสมการและแก้สมการวงจรเชิงเส้นแบบท่อน

การสร้างสมการวงจรเชิงเส้นแบบท่อนก็ยังคงใช้หลักการของโมดิฟายด์โนดัลเช่นเดิม แต่เนื่องจากอุปกรณ์เชิงเส้นแบบท่อนมีหลายสถานะ ดังนั้นในการสร้างสมการวงจรจึงต้องกำหนดเสียก่อนว่าอุปกรณ์เชิงเส้นแบบท่อนนั้นกำลังทำงานอยู่ในสถานะใด

ในการคำนวณค่าคำตอบของสมการวงจรที่จุดเวลา t_{i+1} ใดๆเราไม่สามารถทำนายได้ก่อนว่าที่จุดเวลานั้นอุปกรณ์เชิงเส้นแบบท่อนทำงานอยู่ในสถานะใด ดังนั้นวิธีการที่เรานำมาใช้ก็คือกำหนดให้สถานะการทำงานของอุปกรณ์เชิงเส้นแบบท่อนที่เวลา t_{i+1} เป็นสถานะเดียวกันกับที่เวลา t_i และสร้างสมการวงจรแล้วแก้สมการเพื่อหาคำตอบดังที่ได้กล่าวมาแล้ว หลังจากนั้นจะสามารถนำคำตอบที่ได้มาตรวจสอบว่าอุปกรณ์เชิงเส้นแบบท่อนนั้นควรทำงานอยู่ในสถานะไหน ดังนี้

- หากสถานะการทำงานยังอยู่ในสถานะเดิมที่เรากำหนด แสดงว่าค่าสถานะที่เรากำหนดในตอนแรกนั้นถูกต้อง ก็ให้เลื่อนไปคำนวณหาค่าคำตอบที่จุดเวลา t_{i+2} ต่อไป
- หากสถานะการทำงานไม่ตรงกับสถานะเดิม แสดงว่าสถานะที่เรากำหนดในตอนแรกนั้นไม่ถูกต้อง ให้ใช้สถานะใหม่ในการสร้างสมการวงจรเพื่อแก้สมการหาคำตอบจนกว่าค่าสถานะใหม่ที่ได้จะสอดคล้องกับค่าที่เรากำหนด จึงจะเลื่อนไปคำนวณที่จุดเวลา t_{i+2} ถัดไป

ตัวอย่างในการสร้างสมการ และแก้สมการวงจรเชิงเส้นแบบท่อนนี้ได้แสดงไว้ในบทที่ 3 ในการอธิบายเพื่อช่วยให้เข้าใจถึงการทำงานกับวงจรเชิงเส้นแบบท่อนได้ดียิ่งขึ้น