

การจำลองผลตอบสนองของวงจรทอนระดับแรงดัน
RESPONSE SIMULATION OF STEP-DOWN CONVERTERS

นายทัศนกร ชมวงศ์ รหัส 49364066

นายอิศเรศ ธรรมดูล รหัส 49364387

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 19/๗.ค. 2555
เลขทะเบียน..... 1575806 3
เลขเรียกหนังสือ..... 2/6.
มหาวิทยาลัยนครสวรรค์ ๓ 363 ๗

2552

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์


ปีการศึกษา 2552

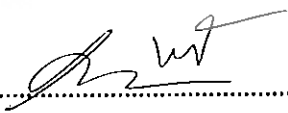


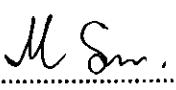
ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ การจำลองผลตอบสนองของวงจรทอนระดับแรงดัน
ผู้ดำเนินโครงการ นายทัศนกร ชมวงศ์ รหัส 49364066
 นายอิศเรศ ธรรมกุล รหัส 49364387
ที่ปรึกษาโครงการ ดร. นิพัทธ์ จันทรมินทร์
สาขาวิชา วิศวกรรมไฟฟ้า
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2552

.....
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า


.....ที่ปรึกษาโครงการ
(ดร. นิพัทธ์ จันทรมินทร์)


.....กรรมการ
(ดร. สุภวรรณ พลพิทักษ์ชัย)


.....กรรมการ
(ดร. มุกษิตา สงฆ์จันทร์)

ชื่อหัวข้อโครงการ	การจำลองผลตอบสนองของวงจรถอนระดับแรงดัน	
ผู้ดำเนินโครงการ	นายทัศนกร ชมวงศ์	รหัส 49364066
	นายอิศเรศ ธรรมคุณ	รหัส 49364387
ที่ปรึกษาโครงการ	ดร. นิพัทธ์ จันทรมินทร์	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2552	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอโครงการเกี่ยวกับการจำลองผลตอบสนองของวงจรถอนระดับแรงดันกระแสตรงด้วยการเขียนโปรแกรมใน MATLAB ผลตอบสนองของวงจรที่สนใจในโครงการนี้คือกระแสไฟฟ้าของตัวเหนี่ยวนำ และแรงดันไฟฟ้าของตัวเก็บประจุซึ่งเป็นแรงดันด้านออกของวงจร เนื้อหาในปริญญานิพนธ์ประกอบด้วยหลักการทํางานของวงจรถอนระดับแรงดัน การหาอัตราขยายแรงดัน การวิเคราะห์ผลตอบสนองด้วยวิธีเฉลี่ยปริภูมิสถานะ การเขียนโปรแกรมใน MATLAB โดยอาศัยระเบียบวิธีของออยเลอร์สร้างอัลกอริทึมเพื่อจำลองผล รวมถึงในส่วนท้ายของโครงการได้ดำเนินการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อให้เกิดความสะดวกในใช้งานโปรแกรมที่สร้างขึ้น ผลงานที่ได้จากโครงการนี้สามารถช่วยในการออกแบบวงจร นั่นคือการเลือกค่าความเหนี่ยวนำ ความจุไฟฟ้า ค่าความถี่สวิตชิง เพื่อให้เกิดผลตอบสนองชั่วคราว และความพลัวของผลตอบสนองตามต้องการ

Project title Response Simulation of Step-Down Converters
Name Mr. Tassanakorn Chomvong ID. 49364066
Mr. Isaret Tammadool ID. 49364387
Project advisor Mr. Niphat Jantharamin, Ph.D.
Major Electrical Engineering
Department Electrical and Computer Engineering
Academic year 2009

Abstract

This thesis presents a project that deals with response simulation of dc step-down converters, aka buck converters, by means of MATLAB. The responses in question are the inductor current and the capacitor voltage, which is the converter output voltage. The thesis describes operating principles of the converter, voltage-gain derivation, response analysis via the state-space averaging technique. Based on Euler's method, a program in MATLAB for response simulation was developed. Finally, the graphical user interface (GUI) was built in order to allow the user to interact with the program more comfortably. The project outcome is useful in the converter design stage. The developed program helps to choose values of inductance, capacitance and switching frequency that give the desired transient behaviour and ripples of the responses.

กิตติกรรมประกาศ

คณะผู้ดำเนินโครงการขอกราบขอบพระคุณอาจารย์ ดร. นิพัทธ์ จันทรมินทร์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการที่ได้คอยชี้แนะแนวทางตลอดการทำโครงการ

นอกจากนี้ขอขอบพระคุณคุณอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินโครงการ รวมทั้งขอขอบคุณคุณคุณพงษ์นาวิน เบ็ญพรม ที่ให้คำปรึกษาในส่วนของ การสร้าง ส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI)

สุดท้ายนี้ ขอกราบขอบพระคุณบิดามารดาของที่ให้ความรัก ความหวังดีกำลังใจ และคอยสนับสนุนในทุกๆด้านตลอดมา



นายทัศนกร ชมวงศ์

นายอิศเรศ ธรรมกุล

สารบัญ

	หน้า
ใบรับรองปริญญาโท.....	ก
บทคัดย่อภาษาไทย.....	ข
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญรูป.....	ฉ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตโครงการ.....	1
1.4 ขั้นตอนการดำเนินงานและแผนการดำเนินงานตลอดโครงการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 งบประมาณ.....	3
บทที่ 2 วงจรทอนระดับแรงดัน.....	4
2.1 วงจรสวิตชิง.....	4
2.2 เงื่อนไขการทำงานของวงจรทอนระดับแรงดัน.....	7
2.3 หลักการทำงานของวงจรทอนระดับแรงดัน.....	7
2.4 การหาค่าอัตราขยายแรงดัน.....	9
2.5 การหาค่าความเหนี่ยวนำที่เล็กที่สุด.....	11
2.6 ค่าความพลิวของแรงดันไฟฟ้าด้านออก.....	13
2.7 การวิเคราะห์ผลตอบสนองของวงจรทอนระดับแรงดัน.....	16
บทที่ 3 การจำลองผลการตอบสนองของวงจรทอนระดับแรงดัน.....	20
3.1 ระเบียบวิธีของออยเลอร์.....	20
3.2 การหาสมการของผลตอบสนองเพื่อเขียนอัลกอริทึม.....	21
3.3 ผังการจำลองผลตอบสนองวงจรทอนระดับแรงดัน.....	23
3.3 การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้.....	25

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการดำเนินงาน.....	28
4.1 ผลตอบสนองของวงจรทอนระดับแรงดัน	28
4.2 การจำลองผลการตอบสนองของวงจรผ่าน GUI.....	30
บทที่ 5 สรุปผลการดำเนินงาน.....	34
5.1 สรุปผล.....	34
5.2 ปัญหาที่พบและแนวทางการแก้ไข.....	34
5.3 แนวทางการพัฒนาต่อไป.....	35
เอกสารอ้างอิง.....	36
ภาคผนวก.....	37
ประวัติผู้ดำเนินงาน.....	53

สารบัญรูป

รูปที่	หน้า
2.1 (ก) วงจรแปลงผันกำลังไฟฟ้ากระแสตรงเป็นกระแสตรงพื้นฐาน (ข) วงจรสมมูลของสวิตช์ (ค) แรงดันไฟฟ้าด้านออก.....	5
2.2 วงจรทอนระดับแรงดัน.....	8
2.3 วงจรสมมูลเมื่อสวิตช์นำกระแส.....	8
2.4 วงจรสมมูลเมื่อสวิตช์ไม่นำกระแส.....	9
2.5 รูปคลื่นการทำงานของวงจรทอนระดับแรงดัน (ก) แรงดันตกคร่อมตัวเหนี่ยวนำ (ข) กระแสของตัวเหนี่ยวนำ (ค) กระแสของตัวเก็บประจุ.....	12
2.6 รูปการทำงานของวงจรทอนระดับแรงดัน (ก) กระแสตัวเก็บประจุ (ข) ความพัวพันแรงดันที่ตัวเก็บประจุ.....	14
2.7 วงจรสมมูลเมื่อสวิตช์นำกระแส.....	16
2.8 วงจรสมมูลเมื่อสวิตช์ไม่นำกระแส.....	17
3.1 ระเบียบวิธีของออยเลอร์.....	20
3.2 วงจรทอนระดับแรงดันขณะสวิตช์นำกระแส.....	21
3.3 วงจรทอนระดับแรงดันขณะสวิตช์ไม่นำกระแส.....	22
3.4 ผังงานการจำลองผลตอบสนองของวงจรทอนระดับแรงดัน.....	24
3.5 หน้าต่างของ GUIDE.....	26
3.6 หน้าต่าง Untitled1.fig.....	26
3.7 การจำลองผลตอบสนองเพื่อสะดวกต่อผู้ใช้.....	27
3.8 การจำลองผลตอบสนองเพื่อสะดวกต่อผู้ใช้เมื่อยังไม่ป้อนค่า.....	27
4.1 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $L = 1 \text{ mH}$	28
4.2 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $L = 2 \text{ mH}$	29
4.3 การจำลองผลตอบสนองเพื่อสะดวกต่อผู้ใช้เมื่อป้อนค่า.....	30
4.4 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $L = 4.6 \text{ mH}$	31
4.5 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $R = 5.6 \Omega$	32
4.6 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $C = 4.8$	33

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในปัจจุบันวงจรแปลงผันกำลังไฟฟ้ากระแสตรงเป็นกระแสตรง (DC to DC converters) นิยมใช้เป็นอุปกรณ์อิเล็กทรอนิกส์ในอุตสาหกรรมสารสนเทศและคอมพิวเตอร์ โดยเป็นเครื่องมือที่ช่วยในกระบวนการแปลงแรงดันไฟฟ้ากระแสตรงให้อยู่ในระดับที่สอดคล้องกับการนำไปใช้งาน ความจำเป็นในการใช้วงจรแปลงผันกำลังไฟฟ้ากระแสตรงเป็นกระแสตรงเข้ามาเป็นส่วนประกอบในอุปกรณ์ไฟฟ้าหรืออิเล็กทรอนิกส์ คือ อุปกรณ์อิเล็กทรอนิกส์ในปัจจุบันมักต้องการใช้กำลังไฟที่สูงแต่ในขณะที่เคาน์เตอร์กึ่งตัวนำ (Semiconductor) ในวงจรกลับต้องการแรงดันไฟฟ้าที่ต่ำลง จึงทำให้ต้องป้อนกระแสไฟที่สูงขึ้นและเมื่อกระแสไฟวิ่งผ่านในสายลวดจะทำให้เกิดแรงดันซึ่งทำให้เกิดการสูญเสียแรงดันเป็นผลให้การทำงานของวงจรไม่ราบรื่นจึงทำให้มีความจำเป็นต้องติดตั้งวงจรแปลงผันกำลังไฟฟ้าไว้ใกล้กับวงจรใช้งานเพื่อแปลงแรงดันไฟฟ้าให้ต่ำลงมาอย่างคงที่ในขณะที่ยังสามารถป้อนกระแสไฟที่มีกำลังสูงได้

ในโครงการนี้ กลุ่มผู้ดำเนินโครงการได้ทำการจำลองผลการตอบสนองของวงจรทอนระดับแรงดันเพื่อให้ทราบลักษณะของผลตอบสนองล่วงหน้าก่อนที่จะนำไปสร้างวงจรจริง ทำให้การออกแบบวงจรมีประสิทธิภาพของอุปกรณ์ซึ่งเป็นส่วนประกอบสำหรับของวงจร อันได้แก่ ค่าความต้านทาน ค่าความเหนี่ยวนำ และค่าความจุมาจำลองผลตอบสนองชั่วคราว (Transient response) และผลตอบสนองอยู่ตัว (Steady-state response) ที่ต้องการก่อนที่จะทำการเลือกอุปกรณ์ได้อย่างถูกต้องและเหมาะสมก่อนนำไปใช้งาน

ถึงแม้ว่าในโครงการนี้จะเลือกศึกษาการจำลองผลตอบสนองของวงจรทอนระดับแรงดัน แต่ก็ยังสามารถนำไปเป็นแนวทางในการสร้างแบบจำลองของวงจรอื่น ๆ เช่น วงจรทอนระดับแรงดัน หรือวงจรทอนทอนระดับแรงดันได้เช่นกัน

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อจำลองผลตอบสนองของวงจรทอนระดับแรงดัน
- 2) เพื่อพัฒนาให้การจำลองผลตอบสนองดังกล่าวมีความสะดวกต่อผู้ใช้

1.3 ขอบเขตของโครงการ

- 1) เขียนโปรแกรมใน MATLAB เพื่อจำลองผลตอบสนองของวงจรทอนระดับแรงดัน
- 2) สร้างส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface: GUI)

1.4 ขั้นตอนและแผนการดำเนินงานตลอดโครงการ

รายละเอียด	ปี 2552							ปี 2553		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1) ศึกษาผลตอบสนองชั่วคราวของวงจรทอนระดับแรงดัน										
2) ศึกษาวิธีการใช้โปรแกรม MATLAB										
3) เขียนโปรแกรมเพื่อจำลองผลตอบสนองของวงจรทอนระดับแรงดัน										
4) ศึกษาและสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI)										
5) สรุปผลการดำเนินโครงการและจัดทำเล่มปริญญานิพนธ์										

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถใช้โปรแกรมที่สร้างขึ้นเพื่อออกแบบวงจร โดยเลือกอุปกรณ์ อัน ได้แก่ ตัวต้านทาน ตัวเหนี่ยวนำ และตัวเก็บประจุ รวมทั้งเลือกใช้ค่าความถี่สวิตชิง (Switching frequency) ที่จะทำให้เกิดผลตอบสนองของวงจรตามต้องการ
- 2) สามารถประยุกต์ใช้แนวทางการจำลองแบบที่ได้กับการจำลองแบบผลการตอบสนองของวงจรอื่น ๆ ได้

1.6 งบประมาณ

ค่าน้ำสื่อที่ใ้ประกอบการดำเนินโครงการ	1,000 บาท
ค่าถ่ายเอกสารและเช่าเล่มปริญญาบัตร	1,000 บาท
รวมเป็นเงินทั้งสิ้น (สองพันบาทถ้วน)	<u>2,000 บาท</u>
หมายเหตุ: ถัวเฉลี่ยทุกรายการ	



บทที่ 2

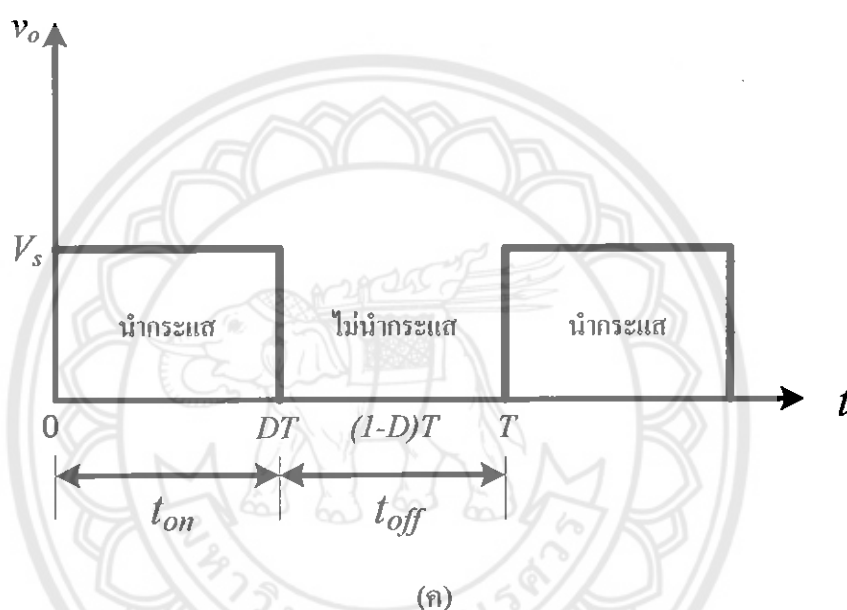
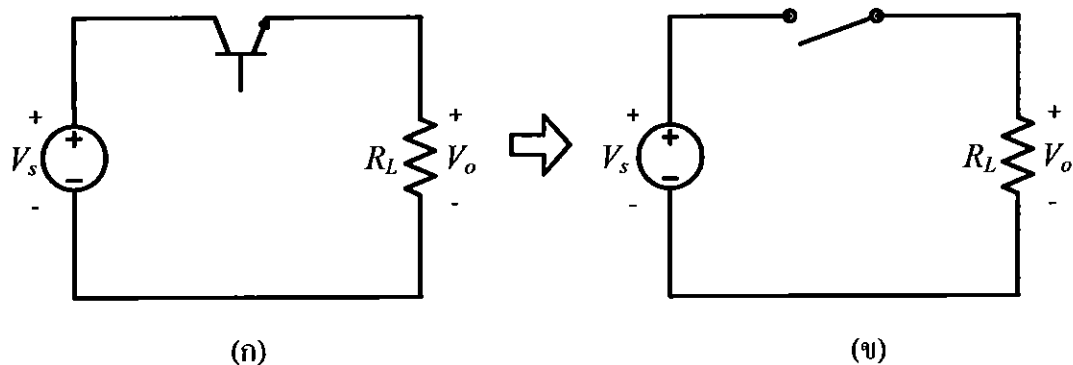
วงจรทอนระดับแรงดัน

วงจรทอนระดับแรงดัน (Step-down converter หรือ Buck converter) ที่ศึกษาในโครงการนี้เป็นตัวแปลงผันกำลังกระแสตรง (DC power converter) ที่ทำให้ระดับแรงดันไฟฟ้าด้านออกมีค่าต่ำกว่าแรงดันไฟฟ้าด้านเข้า และใช้ตัวกรองความถี่ต่ำผ่าน (Low-pass filter) เพื่อให้แรงดันไฟฟ้าด้านออกมีความเป็นกระแสตรงมาก ๆ ซึ่งนิยมเลือกใช้ตัวกรองที่สร้างจากตัวเหนี่ยวนำและตัวเก็บประจุ (LC filter) [1]

การแปลงผันกำลังไฟฟ้า (Power conversion) มีจุดประสงค์หลักคือต้องทำให้การแปลงผันกำลังไฟฟ้าง่ายกว่ามีประสิทธิภาพสูงสุดหรือมีการสูญเสียต่ำสุด เพื่อเมื่อการสูญเสียมากก็จำเป็นต้องมีกระบวนการระบายความร้อน ทำให้ราคาแพงขึ้น ขนาดและน้ำหนักก็มากขึ้น จากความต้องการดังกล่าวการแปลงผันกำลังไฟฟ้าแบบเชิงเส้น (Linear power conversion) จะไม่สามารถทำงานได้ตามเป้าหมายด้วยเหตุผลสองประการ ประการแรก คือการแปลงผันไฟฟ้าแบบเชิงเส้นอันหมายถึงการทำงานของสวิตช์ในย่านเชิงเส้น (Active region) จะมีประสิทธิภาพในการแปลงผันกำลังไฟฟ้าต่ำ ยกตัวอย่างเช่นการใช้งานทรานซิสเตอร์ในย่านเชิงเส้น ทรานซิสเตอร์จะทำตัวเหมือนกับเป็นตัวต้านทานปรับค่าได้ (Variable resistor) ทำให้เกิดกำลังสูญเสีย (Power losses) ในตัวทรานซิสเตอร์ตลอดเวลา จึงทำให้ประสิทธิภาพในการแปลงกำลังไฟฟ้าของวงจรมีค่าต่ำ ประการที่สอง คือขนาดและน้ำหนักของหม้อแปลงไฟฟ้าที่ใช้งานในย่านความถี่ต่ำ (50 หรือ 60 Hz) มีขนาดใหญ่ และน้ำหนักมาก โดยเหตุผลดังกล่าวข้างต้น ในปัจจุบันวงจรทอนระดับแรงดันจึงมีรูปแบบเป็นวงจรสวิตชิง (Switching converters)

2.1 วงจรสวิตชิง

ในวงจรสวิตชิงจะใช้อุปกรณ์โซลิดสเตต (Solid state devices) เช่น มอสเฟต (MOSFET) หรือ ไอจีบีที (IGBT) ทำหน้าที่เป็นสวิตช์อิเล็กทรอนิกส์ โดยมีช่วงการทำงานคือ ช่วงนำกระแส และช่วงหยุดนำกระแส โดยสมมติให้ มอสเฟต หรือ ไอจีบีที ที่ใช้เป็นสวิตช์อิเล็กทรอนิกส์ในอุดมคติดังในรูปที่ 2.1 หลักการทำงานคือ



รูปที่ 2.1 (ก) วงจรแปลงผันกำลังไฟฟ้ากระแสตรงเป็นกระแสตรงพื้นฐาน
(ข) วงจรสมมูลของสวิตช์ (ค) แรงดันไฟฟ้าด้านออก [1]

เมื่อสวิตช์นำกระแส แรงดันไฟฟ้าด้านออกจะมีค่าเท่ากับแรงดันไฟฟ้าด้านเข้า $v_o = V_s$ และเมื่อสวิตช์ไม่นำกระแส แรงดันไฟฟ้าด้านออกจะมีค่าเท่ากับศูนย์ หรือ $v_o = 0$

ช่วงเวลาในการนำกระแสและหยุดนำกระแสของสวิตช์ จะได้สัญญาณแรงดันไฟฟ้าด้านออกเป็นพัลส์ (Pulse) ดังในรูปที่ 2.1 ซึ่งสามารถหาค่าเฉลี่ยหรือค่าของแรงดันไฟฟ้ากระแสตรงด้านออกได้จากสมการที่ (2.1)

$$V_o = \frac{1}{T} \int_0^T v_o(t) dt$$

$$V_o = \frac{1}{T} \int_0^{DT} V_s dt$$

$$V_o = D \cdot V_s \quad (2.1)$$

เมื่อ V_s คือค่าแรงดันไฟฟ้ากระแสตรงของแหล่งกำเนิด (V)

D คือค่าดิวตี้ไซเคิล

จากสมการที่ (2.1) จะพบว่าแรงดันไฟฟ้ากระแสตรงทางด้านออกจะถูกควบคุมได้จากการปรับค่าดิวตี้ไซเคิล (Duty cycle หรือ Duty ratio: D) ซึ่งดิวตี้ไซเคิล หมายถึง อัตราส่วนของช่วงเวลาที่สวิตช์นำกระแสต่อช่วงเวลา 1 คาบสวิตซิง มีความสัมพันธ์ดังสมการที่ (2.2)

$$D = \frac{t_{on}}{t_{on} + t_{off}} = \frac{t_{on}}{T} \quad (2.2)$$

$$t_{on} = D \cdot T$$

$$t_{off} = (1 - D) \cdot T$$

โดยที่ t_{on} คือช่วงเวลานำกระแส (s) ของสวิตช์ในหนึ่งคาบสวิตซิง (Switching period)

t_{off} คือช่วงเวลาไม่นำกระแส (s) ของสวิตช์ในหนึ่งคาบสวิตซิง

T คือคาบสวิตซิง (s)

การกำหนดค่าดิวตี้ไซเคิลนิยมบอกเป็น 2 ลักษณะ คือเป็นเลขเต็มหนึ่งและบอกเป็นเปอร์เซ็นต์ ค่าต่ำสุดของดิวตี้ไซเคิล คือ 0 และค่าสูงสุดคือ 1 ในขณะที่ถ้าบอกเป็นเปอร์เซ็นต์ จะมีค่าต่ำสุดเป็น 0% และค่าสูงสุดคือ 100%

ตัวอย่างเช่น $D = 0.4$ หรือ $D = 40\%$ หมายถึงช่วงเวลาที่สวิตช์นำกระแสมีอยู่ 0.4 ส่วนจากช่วงเวลา 1 คาบ หรือหมายถึงช่วงเวลาที่สวิตช์นำกระแสมีอยู่ 40% จากช่วงเวลา 1 คาบ หรือ 100%

ค่าความสูญเสียของวงจรสวิตซิงประกอบด้วยความสูญเสียขณะนำกระแส (Conduction loss) ความสูญเสียในการสวิตซิง (Switching loss) ซึ่งเกิดจากการเปลี่ยนสถานะของสวิตช์ การที่สวิตช์ของวงจรสวิตซิงมีการนำและไม่นำกระแส (อย่างสมบูรณ์) ที่ความถี่สูงจึงทำให้มีความสูญเสียขณะนำกระแสมีค่าต่ำเมื่อเทียบกับการที่สวิตช์ทำงานในย่านเชิงเส้น อย่างไรก็ตามความสูญเสียในการสวิตซิงมีค่าแปรผันตามค่าความถี่สวิตซิง ดังนั้นการเลือกความถี่สวิตซิงจึงขึ้นอยู่กับ การพิจารณาระหว่างความสูญเสียในการสวิตซิง ราคาของหม้อแปลงไฟฟ้าและวงจรรองความถี่ จนอาจทำให้ประสิทธิภาพโดยรวมของวงจรสวิตซิงมีค่าต่ำกว่าวงจรอิเล็กทรอนิกส์เชิงเส้น [1]

2.2 เงื่อนไขการทำงานของวงจรทอนระดับแรงดัน

การวิเคราะห์การทำงานของวงจรทอนระดับแรงดันในช่วงที่สถานะอยู่ตัว (Steady state) จะมีการกำหนดเงื่อนไขการทำงานของวงจรทอนระดับแรงดันดังนี้

- 1) กระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำในแต่ละคาบจะมีค่าเท่ากัน

$$i_L(t+T) = i_L(t) \quad (2.3)$$

หมายถึง กระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำจะมีค่าเท่ากัน ณ ตำแหน่งเดียวกันทุกคาบเวลา

- 2) แรงดันไฟฟ้าเฉลี่ยตกคร่อมตัวเหนี่ยวนำแต่ละคาบมีค่าเท่ากับศูนย์

$$V_L = \frac{1}{T} \int_0^{T} v_L(\lambda) d\lambda = 0 \quad (2.4)$$

หมายถึง ผลรวมของผลคูณระหว่างแรงดันไฟฟ้าที่ตกคร่อมตัวเหนี่ยวนำกับเวลา (Volt-second) ในแต่ละคาบจะมีค่าเท่ากับศูนย์

- 3) กระแสไฟฟ้าเฉลี่ยของตัวเก็บประจุในแต่ละคาบจะมีค่าเท่ากับศูนย์

$$I_C = \frac{1}{T} \int_0^{T} i_C(t) dt = 0 \quad (2.5)$$

หมายถึง ประจุที่ถูกเก็บ (Charged) และที่คาย (Discharged) ในแต่ละคาบจะต้องเท่ากัน

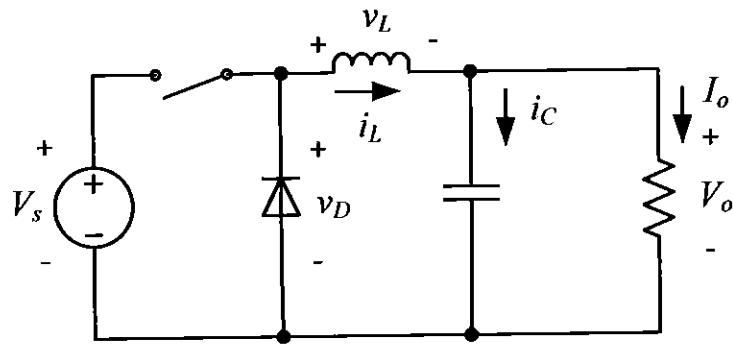
- 4) กำลังไฟฟ้าด้านเข้าเท่ากับกำลังไฟฟ้าด้านออก

$$P_s = P_o \quad (2.6)$$

กรณีนี้กำหนดให้อุปกรณ์ทุกตัวมีคุณสมบัติเป็นอุดมคติ ทำให้ไม่มีการสูญเสียเนื่องจากการทำงานของวงจรประสิทธิภาพของวงจรจึงเท่ากับ 100%

2.3 หลักการทำงานของวงจรทอนระดับแรงดัน

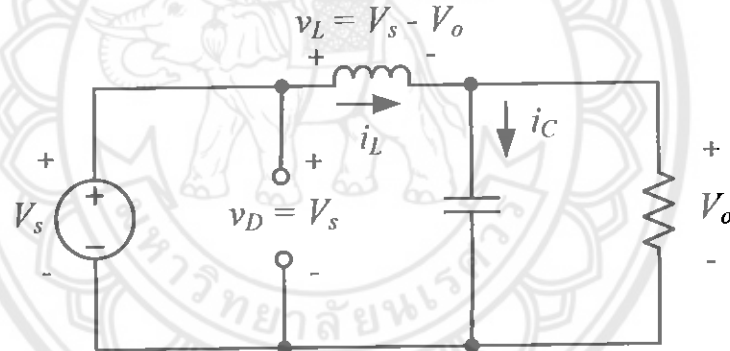
แผนภาพของวงจรทอนระดับแรงดันแสดงได้ดังรูปที่ 2.2 ในการออกแบบให้ได้แรงดันไฟฟ้าด้านออกตามที่ต้องการ จะเริ่มจากการพิจารณาเงื่อนไขที่ว่าแรงดันไฟฟ้าเฉลี่ยที่ตกคร่อมตัวเหนี่ยวนำในแต่ละคาบมีค่าเท่ากับศูนย์ การวิเคราะห์การทำงานของสวิตช์ในแต่ละโหมดจะต้องวิเคราะห์ในสถานะอยู่ตัวดังนี้



รูปที่ 2.2 วงจรทอนระดับแรงดัน [1]

ก) ช่วงที่สวิตช์นำกระแส วาดวงจรสมมูลได้ดังรูปที่ 2.3

กระแสไฟฟ้าจากแหล่งกำเนิดไฟฟ้ากระแสตรงจะไหลผ่านสวิตช์ ผ่านตัวเหนี่ยวนำไปยังโหลด โดยที่กระแสไฟฟ้าส่วนหนึ่งจะไปเก็บที่ตัวเก็บประจุ จากกฎของเคอร์ชอฟฟ์สำหรับแรงดัน (Kirchhoff's voltage law: KVL) สามารถเขียนเป็นสมการแรงดันไฟฟ้าได้ดังนี้



รูปที่ 2.3 วงจรสมมูลเมื่อสวิตช์นำกระแส [1]

$$-V_s + v_L + V_o = 0$$

$$v_L = V_s - V_o \quad (2.7)$$

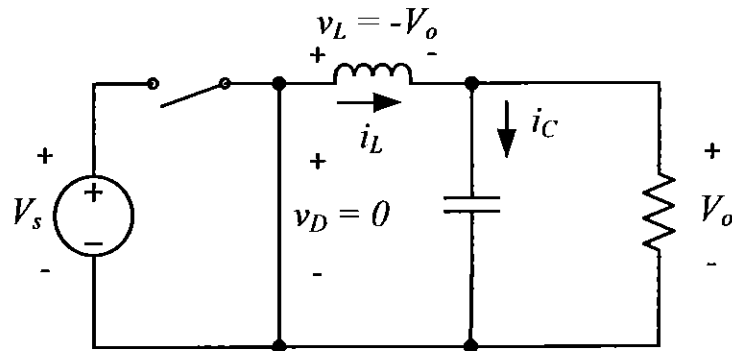
$$v_L = L \frac{di_L}{dt} \quad (2.8)$$

$$\frac{di_L}{dt} = \left(\frac{V_s - V_o}{L} \right) \quad (2.9)$$

dt ในกรณีอยู่ในช่วงเวลานำกระแส คือ $dt = DT$

$$\Delta i_{L,on} = \left(\frac{V_s - V_o}{L} \right) DT \quad (2.10)$$

ข) ช่วงที่สวิตช์ไม่นำกระแส วาดวงจรสมมูลได้ดังรูปที่ 2.4



รูปที่ 2.4 วงจรสมมูลเมื่อสวิตช์ไม่นำกระแส [1]

เมื่อสวิตช์ไม่นำกระแส ไดโอดจะถูกไบแอสไปหน้า (Forward bias) ให้นำกระแส ทำให้กระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำเกิดการไหลอย่างต่อเนื่อง จากกฎของเคอร์ชอฟฟ์สำหรับแรงดัน สามารถเขียนเป็นสมการแรงดันไฟฟ้าได้ดังนี้

$$v_L + V_o = 0 \quad (2.11)$$

$$v_L = -V_o = L \frac{di_L}{dt} \quad (2.12)$$

$$\frac{di_L}{dt} = \frac{\Delta i_{L,off}}{\Delta t} = -\frac{V_o}{L} \quad (2.13)$$

จากสมการที่ (2.13) คือ กระแสที่ไหลผ่านตัวเหนี่ยวนำจะค่อย ๆ มีค่าลดลง เพราะการเปลี่ยนแปลงกระแสตัวเหนี่ยวนำต่อเวลามีค่าเป็นลบ

dt ในกรณีอยู่ในช่วงเวลาหยุดนำกระแส คือ $dt = (1-D) \cdot T$ ดังนั้นสมการที่ (2.13) สมการจัดรูปใหม่ได้เป็น

$$\Delta i_{L,off} = -\frac{V_o}{L} (1-D) \cdot T \quad (2.14)$$

2.4 การหาค่าอัตราขยายแรงดัน

จากเงื่อนไขที่กำหนดไว้ในสมการที่ (2.3) กล่าวคือ ในช่วงสถานะอยู่ตัวการเปลี่ยนแปลงของกระแสที่ไหลผ่านตัวเหนี่ยวนำในช่วงที่สวิตช์นำกระแส จะมีการเปลี่ยนแปลงที่เพิ่มขึ้นเท่ากับการเปลี่ยนแปลงที่ลดลงของกระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำในช่วงที่สวิตช์ไม่นำกระแส หรือ

อาจจะกล่าวได้ว่าการเปลี่ยนแปลงของกระแสที่ไหลผ่านตัวเหนี่ยวนำสุทธิมีค่าเป็นศูนย์ ดังนั้นจากสมการที่ (2.10) และ (2.14) จะได้ว่า

$$\Delta i_{L,on} + \Delta i_{L,off} = 0 \quad (2.15)$$

$$\left(\frac{V_s - V_o}{L}\right)DT + \left(-\frac{V_o}{L}(1-D)T\right) = 0 \quad (2.16)$$

$$(V_s - V_o)D - V_o(1-D) = 0$$

$$V_s D - V_o D - V_o + V_o D = 0$$

$$\frac{V_o}{V_s} = D \quad (2.17)$$

ในสมการที่ (2.17) อัตราขยายแรงดันจะมีค่าสูงสุดไม่เกิน 100% ตามค่า D วงจรทอนระดับแรงดันเป็นวงจรที่ทำให้แรงดันไฟฟ้าด้านออกน้อยกว่าหรือเท่ากับแรงดันไฟฟ้าด้านเข้า ทั้งนี้ขึ้นอยู่กับค่า D โดยการปรับค่า D สามารถปรับได้ตั้งแต่ศูนย์ถึงหนึ่ง เช่น แรงดันไฟฟ้าเท่ากับ 100 V ซึ่งในทางทฤษฎีวงจรทอนระดับแรงดันก็จะสามารถปรับแรงดันไฟฟ้าด้านออกได้ตั้งแต่ 0 ถึง 100 V แต่ในทางปฏิบัติจะไม่สามารถปรับได้ถึง 100 V เพราะเกิดแรงดันตกคร่อมสวิตช์และตัวเหนี่ยวนำทำให้แรงดันไฟฟ้าด้านออกที่ได้จริงมีค่าน้อยกว่า 100 V เล็กน้อย

การหาความสัมพันธ์ของอัตราส่วนของแรงดันด้านออกต่อแรงดันด้านเข้า หรือที่เรียกว่า “อัตราขยายแรงดัน” (Voltage gain) สามารถหาได้อีกวิธีหนึ่ง โดยที่กล่าวไว้ว่าแรงดันไฟฟ้าเฉลี่ยที่ตกคร่อมตัวเหนี่ยวนำในแต่ละคาบจะเท่ากับศูนย์ดังสมการที่ (2.4)

$$V_{L,av} = v_{L,on}(t_{on}) + v_{L,off}(t_{off}) = 0 \quad (2.18)$$

$$V_{L,av} = (V_s - V_o)(t_{on}) + (-V_o)(t_{off}) = 0 \quad (2.19)$$

$$(V_s - V_o)(DT) - V_o(1-D)T = 0$$

$$\frac{V_o}{V_s} = D$$

โดยที่ $V_{L,av}$ คือแรงดันเฉลี่ยตกคร่อมตัวเหนี่ยวนำ

$v_{L,on}$ คือแรงดันเฉลี่ยตัวเหนี่ยวนำขณะสวิตช์นำกระแส

$v_{L,off}$ คือแรงดันเฉลี่ยตัวเหนี่ยวนำขณะสวิตช์ไม่นำกระแส

2.5 การหาค่าความเหนี่ยวนำที่เล็กที่สุด

การหาค่าตัวเหนี่ยวนำที่เล็กที่สุดที่จะทำให้วงจรทอนระดับแรงดันทำงานในโหมดกระแสต่อเนื่อง โดยเริ่มต้นจากการหากระแสเฉลี่ยที่ไหลผ่านตัวเหนี่ยวนำซึ่งจะเท่ากับกระแสเฉลี่ยที่ไหลผ่านโหลด เนื่องจากอยู่ในสถานะการทำงานอยู่ตัว กระแสเฉลี่ยที่ไหลผ่านตัวเก็บประจุจะมีค่าเท่ากับศูนย์ ดังนั้นจะได้

$$I_L = I_o = \frac{V_o}{R} \quad (2.20)$$

จากรูปที่ 2.5 (ข) จะสามารถหากระแสที่ไหลผ่านตัวเหนี่ยวนำที่มีค่าสูงสุดและต่ำสุด ได้จากสมการในช่วงที่สวิตช์ไม่นำกระแส ดังนี้

$$I_{L,\max} = I_L + \frac{\Delta i_L}{2} \quad (2.21)$$

$$I_{L,\max} = \frac{V_o}{R} + \frac{1}{2} \left[\frac{V_o}{L} (1-D)T \right]$$

หรือเขียนให้ง่ายขึ้นดังนี้

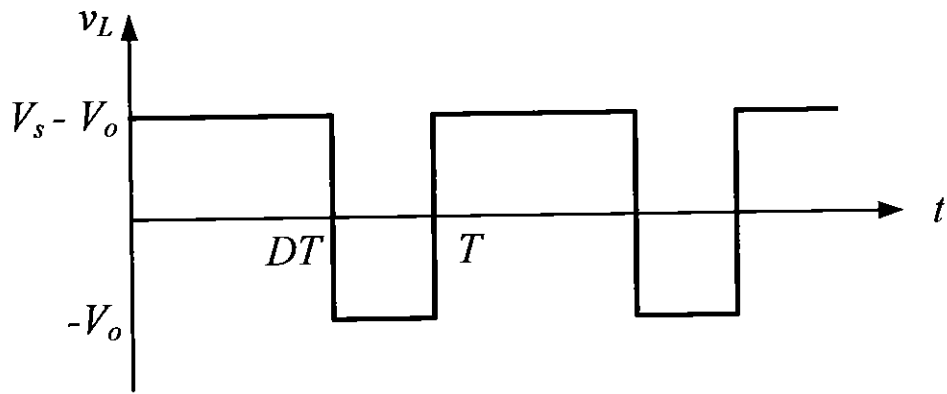
$$I_{L,\max} = V_o \left[\frac{1}{R} + \frac{(1-D)}{2Lf} \right] \quad (2.22)$$

ดังนั้นค่าต่ำสุดของกระแสของตัวเหนี่ยวนำคือ

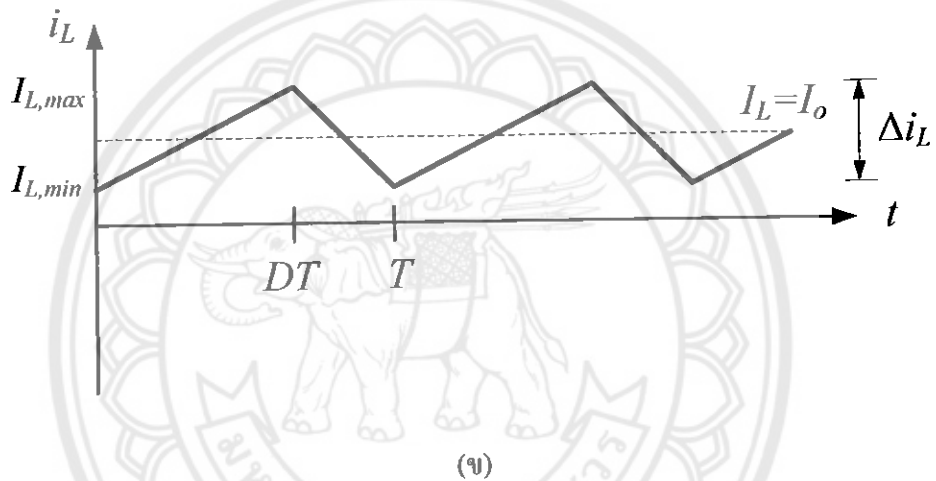
$$I_{L,\min} = I_L - \frac{\Delta i_L}{2} \quad (2.23)$$

$$I_{L,\min} = \frac{V_o}{R} - \frac{1}{2} \left[\frac{V_o}{L} (1-D)T \right]$$

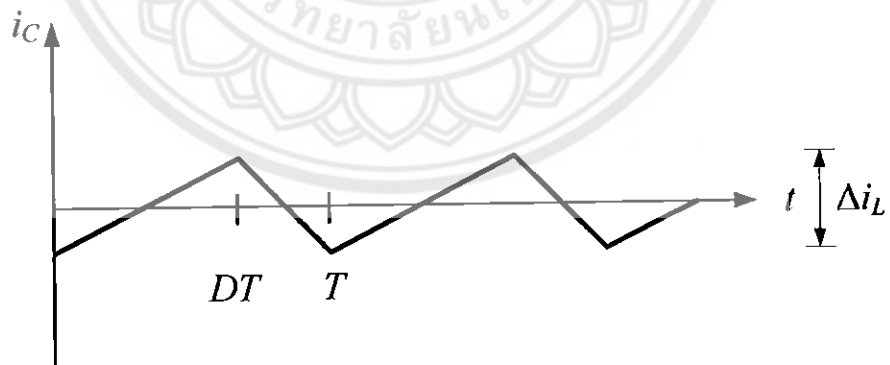
$$I_{L,\min} = V_o \left[\frac{1}{R} - \frac{(1-D)}{2Lf} \right] \quad (2.24)$$



(ก)



(ข)



(ค)

รูปที่ 2.5 รูปคลื่นการทำงานของวงจรทอนระดับแรงดัน [1]

(ก) แรงดันตกคร่อมตัวเหนี่ยวนำ

(ข) กระแสของตัวเหนี่ยวนำ

(ค) กระแสของตัวเก็บประจุ

เงื่อนไขที่สำคัญสำหรับการทำงานในโหมดกระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำแบบต่อเนื่อง คือกระแสที่ไหลผ่านตัวเหนี่ยวนำจะต้องเป็นกระแสบวกเสมอ สมการที่ (2.24) สามารถนำมาใช้นำมาใช้ในการหาค่าความเหนี่ยวนำและความถี่ในการสวิตชิง หาก $I_{L,\min}$ เท่ากับ ศูนย์พอดี คือ เป็นช่วงต่อระหว่างกระแสไฟฟ้าที่ไหลผ่านตัวเหนี่ยวนำเป็นแบบต่อเนื่องและไม่ต่อเนื่อง

$$I_{L,\min} = V_o \left[\frac{1}{R} - \frac{(1-D)}{2Lf} \right] = 0 \quad (2.25)$$

$$\left[\frac{1}{R} - \frac{(1-D)}{2Lf} \right] = 0$$

$$\frac{1}{R} = \frac{(1-D)}{2Lf}$$

ดังนั้นจะได้ว่า

$$L_{\min} = \frac{(1-D)R}{2f} \quad (2.26)$$

$$f = \frac{(1-D)R}{2L_{\min}} \quad (2.27)$$

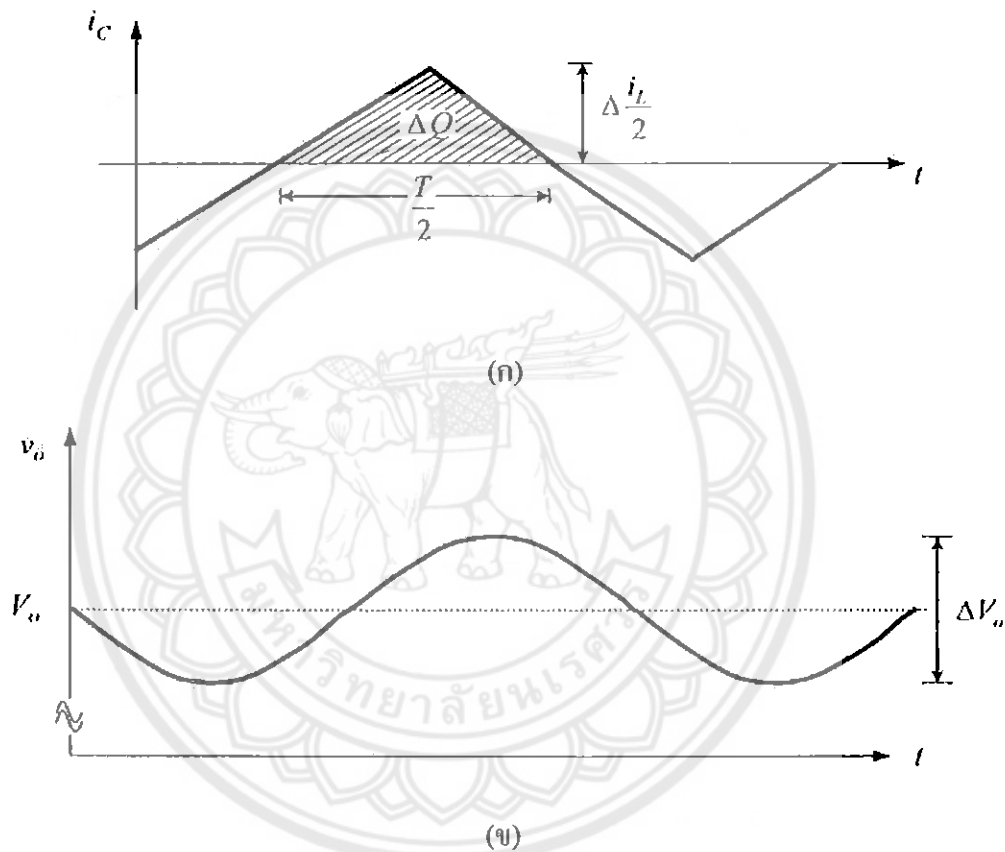
สมการที่ (2.25) ใช้ในการหาค่าความเหนี่ยวนำเล็กที่สุดที่ทำให้วงจรยังสามารถทำงานในโหมดกระแสต่อเนื่อง ซึ่งจะได้ค่าความเหนี่ยวนำที่เล็กที่สุดดังสมการที่ (2.26) ดังนั้นหากต้องการจะออกแบบให้ได้ค่าความเหนี่ยวนำที่เล็กที่สุดโดยที่วงจรยังทำงานในโหมดกระแสต่อเนื่อง ก็จะสามารถเลือกออกแบบได้สองแนวทางคือ กรณีที่หนึ่งการพยายามทำให้เศษในสมการที่ (2.26) มีค่าน้อยที่สุดโดยการทำให้ค่า D มีค่าเข้าใกล้หนึ่งหรือการทำให้ไหลลด R มีค่าน้อย ๆ แต่ผลกระทบที่เกิดขึ้น คือกระแสไหลลดมีค่าสูง ส่วนอีกกรณีหนึ่งคือพยายามทำให้ค่าส่วนหรือต้องการให้สมการที่ (2.26) มีค่ามาก ๆ เช่น ความถี่สวิตชิงเพิ่มขึ้นจาก 50 kHz เป็น 100 kHz ก็จะทำให้ขนาดของตัวเหนี่ยวนำที่ออกแบบเล็กลงสองเท่าได้เช่นกัน ส่วนสมการที่ (2.27) ใช้สำหรับการพิจารณาหาความถี่สวิตชิงที่เหมาะสม เมื่อตัวเหนี่ยวนำถูกกำหนดค่าความเหนี่ยวนำ

2.6 ค่าความพลีของแรงดันด้านออก

โดยปกติวงจรกรองความถี่ต่ำผ่าน หรือวงจรกรองแบบ LC ที่อยู่ในวงจรทอนระดับแรงดันจะมีตัวเก็บประจุที่มีขนาดใหญ่ เพื่อรักษาให้แรงดันไฟฟ้าด้านออกมีค่าคงที่ แต่ในทาง

ปฏิบัติจะไม่สามารถเลือกใช้ตัวเก็บประจุที่มีขนาดใหญ่มาก ๆ ได้ เนื่องจากมีราคาแพง และใช้พื้นที่มาก จึงควรเลือกใช้ตัวเก็บประจุที่มีขนาดเหมาะสมและให้ค่าความพลีวของแรงดันไฟฟ้าค่านอกอยู่ในระดับที่ยอมรับได้ สำหรับการกำหนดหาค่าความพลีวของแรงดันไฟฟ้าค่านอก สามารถหาได้จากความสัมพันธ์ของแรงดันกับกระแสของตัวเก็บประจุดังแสดงในรูปที่ 2.6 ซึ่งกระแสในตัวเก็บประจุหาได้จาก

$$i_c = i_L - i_o \quad (2.28)$$



รูปที่ 2.6 รูปการทำงานของวงจรทอนระดับแรงดัน
(ก) กระแสตัวเก็บประจุ (ข) ความพลีวแรงดันที่ตัวเก็บประจุ [1]

เมื่อกระแสไฟฟ้าที่ไหลผ่านตัวเก็บประจุมีค่าบวก ซึ่งช่วงเวลาดังกล่าวตัวเก็บประจุจะสะสมประจุโดยสามารถคำนวณจาก

$$Q = CV_o \quad (2.29)$$

$$\Delta Q = C \Delta V_o$$

$$\Delta V_o = \frac{\Delta Q}{C} \quad (2.30)$$

- เมื่อ ΔQ คือค่าประจุที่เปลี่ยนแปลงน้อย ๆ
- ΔV_o คือค่าแรงดันของแรงดันไฟฟ้าด้านออกที่เปลี่ยนแปลงน้อย ๆ ระหว่างขอดถึงขอด
- ΔQ คือพื้นที่สามเหลี่ยมที่เกิดจากกระแสตัวเก็บประจุคูณกับเวลาจะได้

$$\Delta Q = \frac{1}{2} \left(\frac{T}{2} \right) \left(\frac{\Delta i_L}{2} \right) = \frac{T \Delta i_L}{8} \quad (2.31)$$

แทนค่าสมการที่ (2.31) ลงในสมการที่ (2.30) จะได้

$$\Delta V_o = \frac{T \Delta i_L}{8C} \quad (2.32)$$

จากสมการที่ (2.14) แทนค่า $\Delta i_{L,avg} = \frac{V_o}{L} (1-D)T$ (โดยจะพิจารณาเฉพาะขนาด) ลงในสมการที่ (2.32) จะได้

$$\Delta V_o = \frac{T}{8C} \frac{V_o}{L} (1-D)T \quad (2.33)$$

$$\frac{\Delta V_o}{V_o} = \frac{T}{8C} \left(\frac{1}{L} \right) (1-D)T \quad (2.34)$$

ดังนั้นจะได้

$$\frac{\Delta V_o}{V_o} = \frac{1-D}{8LCf^2} \quad (2.35)$$

สมการที่ (2.35) เรียกว่าอัตราความพลัว (Ripple ratio) มักใช้แสดงเป็นอัตราส่วนของความพลัวแรงดันไฟฟ้าด้านออกระหว่างขอดถึงขอดกับแรงดันไฟฟ้าด้านออก ในที่นี้หมายถึงเฉพาะแรงดันไฟฟ้า หากต้องการทำให้อัตราความพลัวแรงดันไฟฟ้าของวงจรทอนระดับแรงดันมีค่าน้อยลง (อัตราความพลัวยิ่งน้อยหมายถึงการที่ได้แรงดันไฟฟ้ากระแสตรงที่เรียบมากขึ้น) สามารถเลือกได้สองแนวทางคือกรณีที่หนึ่งการพยายามทำให้เศษในสมการที่ (2.35) มีค่าน้อยที่สุด โดยการทำให้ค่า D มีค่าเข้าใกล้หนึ่ง ส่วนอีกกรณีหนึ่งคือการพยายามทำให้ค่าในสมการที่ (2.35) มีค่ามาก ๆ เช่น การเพิ่มค่าตัวเหนี่ยวนำ ตัวเก็บประจุหรือความถี่สวิทชิงให้มากขึ้น โดยการเพิ่มความถี่สวิทชิง เช่น เพิ่มขึ้นจาก 50 kHz เป็น 100 kHz ค่าอัตราความพลัวจะมีค่าลดลงสี่เท่า ซึ่งความถี่สวิทชิงของวงจรทอนระดับแรงดันนับว่ามีความไวสูงในการควบคุมอัตราความพลัวของแรงดันไฟฟ้าด้านออก

ประเด็นที่น่าสนใจอย่างหนึ่งของวงจรทอนระดับแรงดันคือ กำลังไฟฟ้าที่จ่ายจากแหล่งกำเนิดไฟฟ้ากำลังจะเท่ากับกำลังไฟฟ้าที่โหลดได้รับ เงื่อนไขนี้คือสมมติการสูญเสียเป็นศูนย์จะได้

$$P_s = P_o \quad (2.36)$$

โดยที่ P_s คือกำลังไฟฟ้าที่จ่ายจากแหล่งกำเนิดไฟฟ้ากำลัง

P_o คือกำลังไฟฟ้าที่โหลด

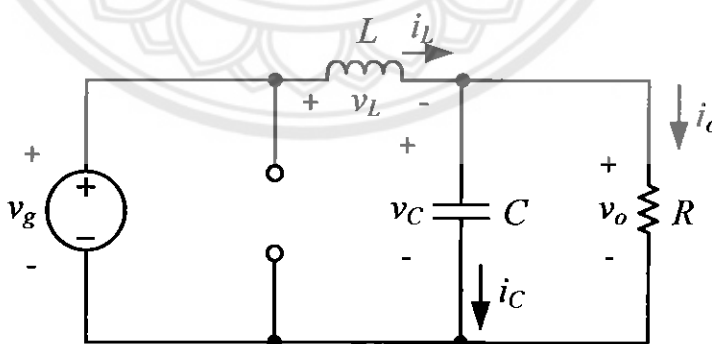
$$V_s I_s = V_o I_o \quad (2.37)$$

จากสมการที่ (2.37) จะเห็นได้ว่ามีความสัมพันธ์ของแรงดันกับกระแสไฟฟ้าเหมือนกับกรณีของหม้อแปลงไฟฟ้ากระแสสลับ จึงอาจกล่าวได้ว่า วงจรทอนระดับแรงดันมีวงจรสมมูลเปรียบเทียบได้กับหม้อแปลงไฟฟ้ากระแสตรง

2.7 การวิเคราะห์ผลตอบสนองของวงจรทอนระดับแรงดัน

เมื่อพิจารณาให้อุปกรณ์ในวงจรมีคุณสมบัติเป็นอุดมคติ และวงจรทำงานในโหมดการนำกระแสต่อเนื่อง (Continuous conduction mode: CCM) เราสามารถเขียนสมการอนุพันธ์ของกระแสของตัวเหนี่ยวนำและแรงดันของตัวเก็บประจุในแต่ละช่วงการทำงานของสวิตช์ได้ดังนี้

ในช่วงที่สวิตช์นำกระแส การทำงานของวงจรสามารถอธิบายด้วยวงจรสมมูลในรูปที่ 2.7



รูปที่ 2.7 วงจรสมมูลเมื่อสวิตช์นำกระแส

จากกฎเคอร์ชอฟฟ์สำหรับแรงดัน เราสามารถหาสมการอนุพันธ์ของกระแสของตัวเหนี่ยวนำในช่วงที่สวิตช์นำกระแส ได้ดังนี้

$$v_L = v_g - v_C$$

$$L \frac{di_L}{dt} = v_g - v_C$$

$$\frac{di_L}{dt} = -\frac{1}{L}v_C + \frac{1}{L}v_g \quad (2.38)$$

จากกฎเคอร์ชอฟฟ์สำหรับกระแส (Kirchhoff's current law: KCL) เราสามารถหาสมการอนุพันธ์ของแรงดันของตัวเก็บประจุในช่วงที่สวิตช์นำกระแส ได้ดังนี้

$$i_C = i_L - i_o$$

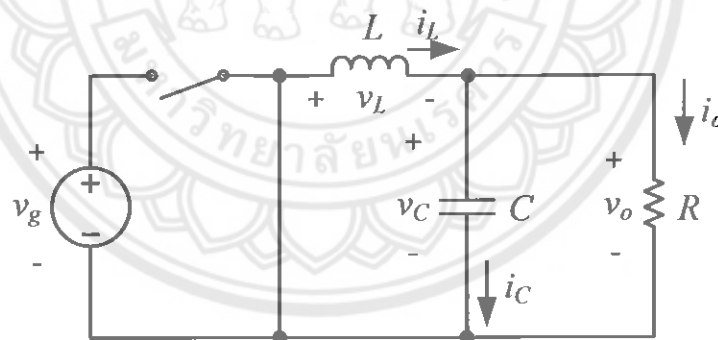
$$C \frac{dv_C}{dt} = i_L - \frac{v_C}{R}$$

$$\frac{dv_C}{dt} = \frac{1}{C}i_L - \frac{1}{RC}v_C \quad (2.39)$$

เมื่อนำสมการที่ (2.38) และ (2.39) มาเขียนในรูปของเมทริกซ์ จะได้

$$\begin{bmatrix} \frac{d}{dt} i_L \\ \frac{d}{dt} v_C \end{bmatrix} = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} v_g \quad (2.40)$$

ในช่วงที่สวิตช์ไม่นำกระแส สามารถอธิบายด้วยวงจรสมมูลในรูปที่ 2.8



รูปที่ 2.8 วงจรสมมูลเมื่อสวิตช์ไม่นำกระแส

จากกฎเคอร์ชอฟฟ์สำหรับแรงดัน เราสามารถหาสมการอนุพันธ์ของกระแสของตัวเหนี่ยวนำในช่วงที่สวิตช์ไม่นำกระแส ได้ดังนี้

$$v_L = -v_C$$

$$L \frac{di_L}{dt} = -v_C$$

$$\frac{di_L}{dt} = -\frac{1}{L}v_C \quad (2.41)$$

จากกฎเคอร์ชอฟฟ์สำหรับกระแส เราสามารถหาสมการอนุพันธ์ของแรงดันของตัวเก็บประจุในช่วงที่สวิตช์ไม่นำกระแส ได้ดังนี้

$$\begin{aligned} i_c &= i_L - i_o \\ C \frac{dv_c}{dt} &= i_L - \frac{v_c}{R} \\ \frac{dv_c}{dt} &= \frac{1}{C} i_L - \frac{1}{RC} v_c \end{aligned} \quad (2.42)$$

จะเห็นว่าสมการที่ (2.39) และ (2.42) เป็นสมการเดียวกัน

เมื่อนำสมการที่ (2.41) และ (2.42) มาเขียนในรูปของเมตริกซ์ จะได้

$$\begin{bmatrix} \frac{d}{dt} i_L \\ \frac{d}{dt} v_c \end{bmatrix} = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \begin{bmatrix} i_L \\ v_c \end{bmatrix} \quad (2.43)$$

หลังจากนำสมการที่ (2.40) และ (2.43) ไปวิเคราะห์ด้วยวิธีเฉลี่ยปริภูมิสถานะ (State-space averaging technique) [2] จะได้แบบจำลองทางคณิตศาสตร์สำหรับค่ากระแสตรง (DC model) ของวงจรทอนระดับแรงดันคือ

$$\begin{bmatrix} I_L \\ V_C \end{bmatrix} = \begin{bmatrix} 1/R \\ 1 \end{bmatrix} D V_g \quad (2.44)$$

ซึ่งสามารถเขียนได้ดังนี้

$$I_L = \frac{D \cdot V_g}{R} = \frac{V_C}{R} = I_o \quad (2.45)$$

$$V_C = D \cdot V_g \quad (2.46)$$

จะพบว่า สมการที่ (2.45) ยืนยันว่าค่าเฉลี่ยของกระแสของตัวเหนี่ยวนำมีค่าเท่ากับค่าเฉลี่ยของกระแสโหลด ในขณะที่สมการที่ (2.46) แสดงให้เห็นว่าค่าเฉลี่ยของแรงดันของตัวเก็บประจุ (ซึ่งเท่ากับแรงดันด้านออกของวงจร) มีค่าต่ำกว่าแรงดันด้านเข้า โดยขึ้นอยู่กับค่าคิวตี้ไซเคิล D

เมื่อนำสมการที่ (2.40) และ (2.43) ไปวิเคราะห์ด้วยวิธีเฉลี่ยปริภูมิสถานะ [2] เพื่อหาแบบจำลองทางคณิตศาสตร์สำหรับสัญญาณขนาดเล็ก (Small-signal model) จะได้ฟังก์ชันถ่ายโอนจากด้านเข้าไปด้านออก (Input-to-output transfer function) คือ

$$\frac{\tilde{v}_c}{\tilde{v}_g}(s) = D \cdot \frac{\frac{1}{LC}}{s^2 + s \frac{1}{RC} + \frac{1}{LC}} \quad (2.47)$$

โดยที่ \tilde{v}_c คือการเปลี่ยนแปลงเพียงเล็กน้อย (Small perturbation) ของแรงดันด้านออก

\tilde{v}_g คือการเปลี่ยนแปลงเพียงเล็กน้อยของแรงดันด้านเข้า

และได้ฟังก์ชันถ่ายโอนจากการควบคุมไปด้านออก (Control-to-output transfer function)

คือ

$$\frac{\tilde{v}_c}{\tilde{d}}(s) = V_g \cdot \frac{\frac{1}{LC}}{s^2 + s \frac{1}{RC} + \frac{1}{LC}} \quad (2.48)$$

โดยที่ \tilde{d} คือการเปลี่ยนแปลงเพียงเล็กน้อยของค่าความถี่พัลส์

เมื่อเปรียบเทียบสมการที่ (2.47) และ (2.48) กับรูปมาตรฐานของฟังก์ชันถ่ายโอนที่มีอัตราขยายเท่ากับหนึ่ง (Unity-gain transfer function) ของระบบอนุพันธ์อันดับที่สอง (Second-order systems) จะพบว่า ค่าความถี่ธรรมชาติ (Natural frequency, ω_n) ของวงจรถอนระดับแรงดันมีค่าเท่ากับ

$$\omega_n = \frac{1}{\sqrt{LC}} \quad (2.49)$$

และตัวประกอบหน่วง (Damping factor, ζ) มีค่าเท่ากับ

$$\zeta = \frac{1}{2R} \sqrt{\frac{L}{C}} \quad (2.50)$$

สมการที่ (2.49) และ (2.50) แสดงให้เห็นว่าผลตอบสนองชั่วคราว (Transient responses) ของวงจรถอนระดับแรงดันจะขึ้นอยู่กับค่าความหน่วงนำ ค่าความจุไฟฟ้า และค่าความต้านทานของโหลด

เมื่อคำนวณหาค่าความถี่ธรรมชาติและตัวประกอบหน่วงแล้ว เราสามารถประมาณค่าเวลาสู่สมดุล (Settling time, t_s) และค่าพุ่งเกินสูงสุด (Maximum overshoot, M_p) [3] ได้ ในกรณีที่ให้ผลตอบสนองมีค่าต่างจากค่าสุดท้าย 5% ค่าเวลาสู่สมดุลหาได้จาก

$$t_s = \frac{3}{\zeta \omega_n} \quad (2.51)$$

และค่าพุ่งเกินสูงสุดในหน่วยเปอร์เซ็นต์ หาได้จาก

$$M_p = 100 \exp[-\zeta \pi / \sqrt{1 - \zeta^2}] \quad (2.52)$$

บทที่ 3

การจำลองผลตอบสนองของวงจรถอนระดับแรงดัน

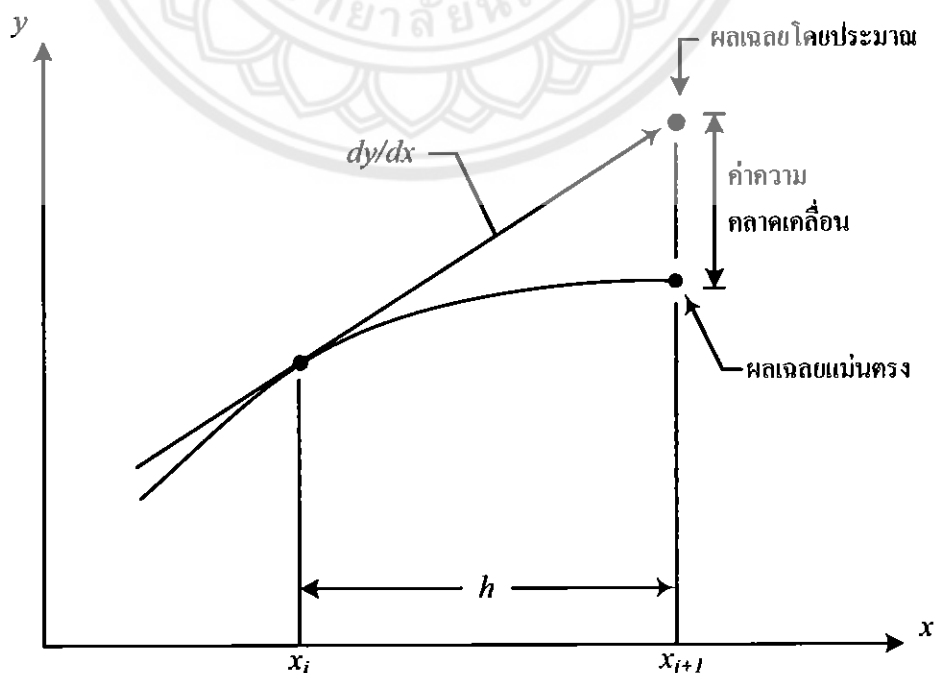
ผลตอบสนองของวงจรถอนระดับแรงดันที่ต้องการจำลองผลในโครงการนี้คือกระแสของตัวเหนี่ยวนำและแรงดันของตัวเก็บประจุ (หรือแรงดันด้านออกของวงจรถอน) การจำลองผลจะใช้การเขียนโปรแกรมใน MATLAB โดยอาศัยระเบียบวิธีของออยเลอร์ (Euler's method) เพื่อสร้างอัลกอริทึมของโปรแกรมโดยพิจารณาหลักการการทำงานของวงจรถอน เนื่องจากวงจรถอนระดับแรงดันเป็นวงจรถอนสวิทชิง จึงแยกพิจารณาการทำงานของวงจรถอนออกเป็นสองช่วง คือช่วงที่สวิทช์นำกระแสและช่วงที่สวิทช์ไม่นำกระแส

3.1 ระเบียบวิธีของออยเลอร์

ระเบียบวิธีการของออยเลอร์จัดได้ว่าเป็นระเบียบวิธีที่ง่ายแก่การทำความเข้าใจมากที่สุดสำหรับการแก้สมการเชิงอนุพันธ์สามัญ [4] ซึ่งอยู่ในรูปแบบโดยทั่วไปดังนี้

$$\frac{dy}{dx} = f(x, y) \quad (3.1)$$

หลักการที่ใช้ในการแก้สมการเชิงอนุพันธ์สามัญ สามารถอธิบายได้โดยพิจารณารูปที่ 3.1



รูปที่ 3.1 ระเบียบวิธีของออยเลอร์ [3]

จากรูปที่ 3.1 สามารถหาค่าผลเฉลยโดยประมาณ y_{i+1} ที่ x_{i+1} จากผลลัพธ์ y_i ซึ่งรู้ค่าที่ x_i โดยใช้ค่าของความชันที่ x_i ดังนี้

$$\frac{dy}{dx} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{h} \quad (3.2)$$

โดยที่ $h = x_{i+1} - x_i$ คือความกว้างช่วง (Step size) ที่ใช้ในการคำนวณ ทำการแทนค่าของความชันที่ x_i จากสมการที่ (3.2) ลงในสมการที่ (3.1) จะได้

$$\frac{y_{i+1} - y_i}{h} = f(x, y) \quad (3.3)$$

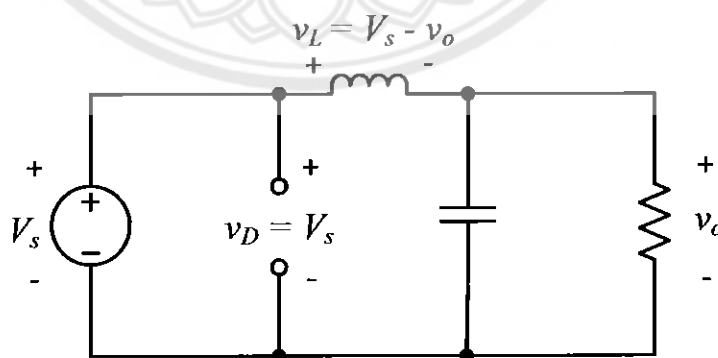
$$\text{นั่นคือ} \quad y_{i+1} = y_i + f(x, y) \quad (3.4)$$

ซึ่งหมายความว่า เราสามารถทำการคำนวณโดยเริ่มจากเงื่อนไขเริ่มต้นของ y_i ที่ x_i แล้วคำนวณค่า y_{i+1} ใหม่จากความกว้างช่วง h ที่กำหนดให้ และจากรูปที่ 3.1 จะเห็นว่าความเที่ยงตรงของผลเฉลยโดยประมาณนั้นขึ้นอยู่กับค่า h ที่ใช้ในการคำนวณนี้ กล่าวคือ ยิ่งใช้ h มีค่าน้อยเท่าใด ก็จะได้ผลลัพธ์ที่มีความเที่ยงตรงมากยิ่งขึ้นเท่านั้น

3.2 การหาสมการของผลตอบสนองเพื่อเขียนอัลกอริทึม

สมการของกระแสของตัวเหนี่ยวนำและแรงดันของตัวเก็บประจุ (หรือแรงดันด้านออกของวงจร) สามารถหาได้โดยแยกพิจารณาเป็นสองช่วงตามหลักการทำงานของวงจรสวิตชิงดังนี้

ก) ช่วงที่สวิตช์นำกระแส วาดวงจรสมมูลได้ดังรูปที่ 3.2



รูปที่ 3.2 วงจรทอนระดับแรงดันขณะสวิตช์นำกระแส [1]

จากการใช้ KVL เราสามารถหาค่าสมการ i_L ได้ดังนี้

$$-V_s + v_L + v_o = 0$$

จาก
$$v_L = V_s - v_o$$

$$v_L = L \frac{di_L}{dt} = L \frac{\Delta i_L}{\Delta t}$$

แทนค่า จะได้
$$L \frac{\Delta i_L}{\Delta t} = V_s - v_o$$

$$\Delta i_L = \frac{1}{L} [V_s - v_o] \cdot \Delta t$$

$$i_{L,new} - i_{L,old} = \left[\frac{V_s - v_{o,old}}{L} \right] \cdot \Delta t$$

$$i_{L,new} = i_{L,old} + \left[\frac{V_s - v_{o,old}}{L} \right] \cdot \Delta t \tag{3.5}$$

และใช้กฎของเกออร์ชอฟฟ์สำหรับกระแส (Kirchhoff's current law: KCL) เราสามารถหาสมการของ v_o ได้ดังนี้

$$i_C = i_L - i_o$$

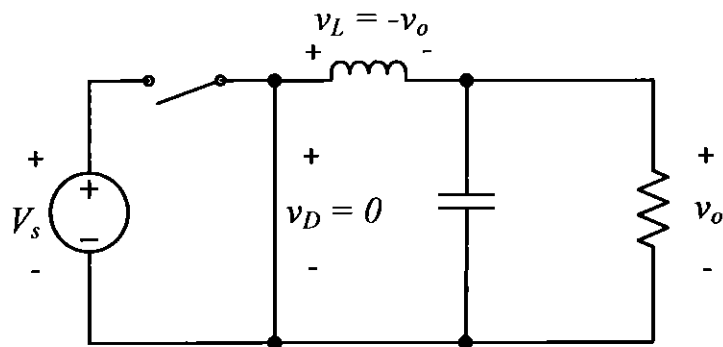
$$i_C = C \frac{dv_o}{dt} = C \frac{\Delta v_o}{\Delta t} = i_L - \frac{v_o}{R}$$

$$\Delta v_o = \frac{1}{C} \left[i_L - \frac{v_o}{R} \right] \cdot \Delta t$$

$$v_{o,new} - v_{o,old} = \frac{1}{C} \left[i_{L,new} - \frac{v_{o,old}}{R} \right] \cdot \Delta t \tag{3.6}$$

$$v_{o,new} = v_{o,old} + \frac{1}{C} \left[i_{L,new} - \frac{v_{o,old}}{R} \right] \cdot \Delta t \tag{3.6}$$

ข) ช่วงที่สวิตช์ไม่นำกระแส วาดวงจรสมมูลได้ดังรูปที่ 3.2



รูปที่ 3.3 วงจรทอนระดับแรงดันขณะสวิตช์ไม่นำกระแส [1]

จากการใช้ KVL เราสามารถหาค่าสมการ i_L ได้ดังนี้

$$v_L + v_o = 0$$

$$v_L = -v_o$$

$$L \frac{\Delta i_L}{\Delta t} = -v_o$$

$$\Delta i_L = -\frac{v_o}{L} \Delta t$$

$$i_{L,new} - i_{L,old} = -\frac{v_o}{L} \Delta t$$

$$i_{L,new} = i_{L,old} - \frac{v_o}{L} \Delta t \quad (3.7)$$

และใช้ KCL เราสามารถหาสมการของ v_o ได้ดังนี้

$$i_C = i_L - i_o$$

จาก $i_C = C \frac{dv_o}{dt} = C \frac{\Delta v_o}{\Delta t}$

แทนค่า จะได้ $C \frac{\Delta v_o}{\Delta t} = i_L - \frac{v_o}{R}$

$$\Delta v_o = \frac{1}{C} \left[i_L - \frac{v_o}{R} \right] \cdot \Delta t$$

$$v_{o,new} - v_{o,old} = \frac{1}{C} \left[i_{L,new} - \frac{v_{o,old}}{R} \right] \cdot \Delta t$$

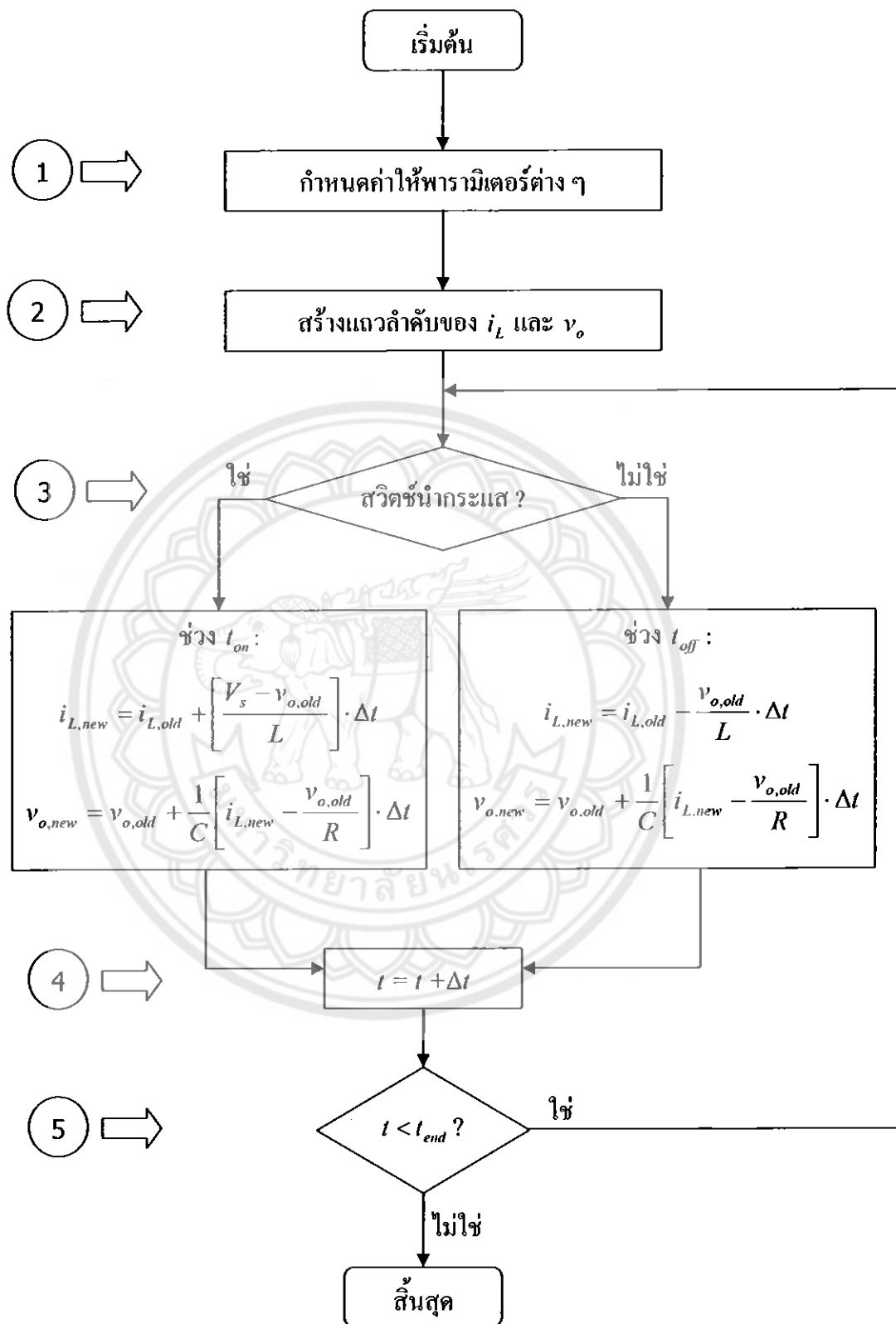
$$v_{o,new} = v_{o,old} + \frac{1}{C} \left[i_{L,new} - \frac{v_{o,old}}{R} \right] \cdot \Delta t \quad (3.8)$$

จะสังเกตเห็นว่าสมการที่ (3.6) และ (3.8) เป็นสมการเดียวกัน

3.3 ฟังก์ชันการจำลองผลตอบสนองของวงจรทอนระดับแรงดัน

ลำดับการเขียนโปรแกรมเพื่อจำลองผลตอบสนองของวงจรทอนระดับแรงดันแสดงได้ดังรูปที่ 3.4 โดยอธิบายเป็นขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 ก่อนเริ่มจำลองผลตอบสนอง ต้องกำหนดค่าให้พารามิเตอร์ของการสวิตชิงของวงจร และของการจำลองผลเสียก่อน



รูปที่ 3.4 ฝั่งงานการจำลองผลตอบสนองของวงจรทอนระดับแรงดัน

ขั้นตอนที่ 2 สร้างแถวลำดับ (Array) ของตัวแปร i_L และ v_o เพื่อนำค่าของตัวแปรทั้งสองที่คำนวณได้ในแต่ละรอบของการจำลองผลมาเก็บไว้

ขั้นตอนที่ 3 ตรวจสอบสถานะของสวิตช์ว่าเป็นช่วงที่สวิตช์นำกระแส (t_{on}) หรือช่วงที่สวิตช์ไม่นำกระแส (t_{off}) ถ้าเป็นช่วงที่สวิตช์นำกระแส ค่าของ i_L และ v_o จะถูกคำนวณตามสมการที่ (3.5) และ (3.6) ตามลำดับ ในขณะที่ถ้าเป็นช่วงที่สวิตช์ไม่นำกระแส ค่าของ i_L และ v_o จะถูกคำนวณตามสมการที่ (3.7) และ (3.8) ตามลำดับ

ขั้นตอนที่ 4 ปรับระยะเวลาที่จำลองผล โดยใช้ Δt ตามสมการ $t = t + \Delta t$

ขั้นตอนที่ 5 ตรวจสอบระยะเวลาที่จำลองผล เมื่อครบระยะเวลาที่เรากำหนด คือ t_{end} ถือเป็นอันสิ้นสุดการจำลองผล

หลังจากเสร็จสิ้นการจำลองผลแล้ว ข้อมูลของค่ากระแสของตัวเหนี่ยวนำและค่าแรงดันของตัวเก็บประจุ (หรือค่าแรงดันค้ำออกของวงจร) ที่มีการเปลี่ยนแปลงตลอดระยะเวลาที่ทำการจำลองผลได้ถูกจัดเก็บไว้ในแถวลำดับของตัวแปร i_L และ v_o ที่สร้างขึ้นในโปรแกรม จากนั้นเราจะนำข้อมูลที่ได้จากการจำลองผลนี้ไปวาดกราฟซึ่งได้แสดงไว้ในบทที่ 4

3.4 การสร้างส่วนต่อประสานกราฟิกกับผู้ใช้

เพื่อให้การใช้งาน โปรแกรมการจำลองผลที่สร้างขึ้นในโครงการนี้สะดวกสำหรับผู้ใช้ คณะผู้ดำเนินโครงการได้สร้างส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical user interface: GUI) ซึ่งสามารถสร้างขึ้นด้วยอ็อบเจกต์กราฟิก (Graphical object) แบบต่าง ๆ เช่น ช่องรับค่า ช่องแสดงค่า ปุ่มกด และแถบเลื่อนหน้าจอ เป็นต้น [5] โดยทั่วไปผู้ใช้คอมพิวเตอร์ส่วนใหญ่เข้าใจถึงความหมายและวิธีการใช้อ็อบเจกต์เหล่านี้เป็นอย่างดีอยู่แล้ว สิ่งสำคัญที่จะกล่าวถึงคือ หลังจากผู้ใช้ได้มีการกดปุ่มเมาส์ เลื่อนแถบเลื่อนหน้าจอ หรือเลือกเมนู เราจะมีวิธีการกำหนดให้เกิดขั้นตอนต่อ ๆ ไปตามที่เราต้องการได้อย่างไร การประยุกต์ใช้งานต่าง ๆ ที่ออกแบบมาเป็น GUI จะพบว่าสามารถทำให้ผู้ใช้เข้าใจการใช้งานนั้นได้อย่างรวดเร็ว เพราะแทบจะไม่มีชุดคำสั่งใดให้จดจำ โดยการทำงานจะเกิดขึ้นทันทีที่ได้รับข้อมูลค้ำเข้า (Input) จากผู้ใช้

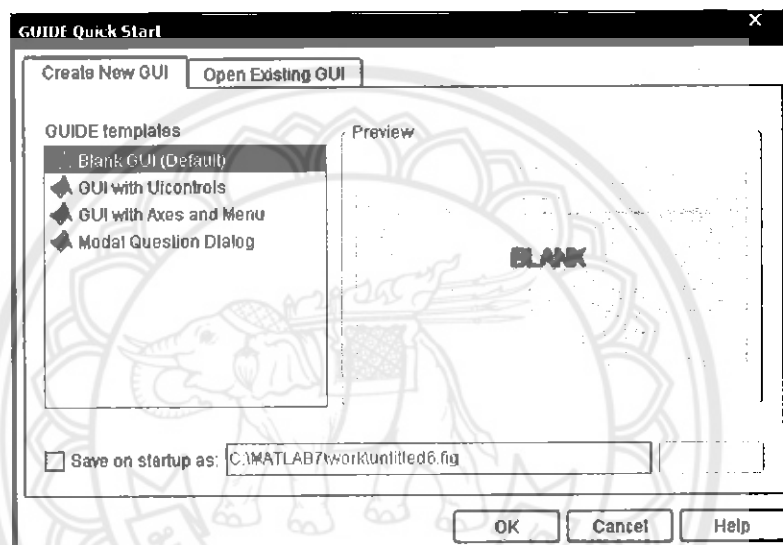
ส่วนประกอบของส่วนต่อประสานกราฟิกผู้ใช้ จะประกอบด้วย 2 ส่วนหลัก คือ

- 1) ไฟล์รูปภาพ (Figure file) ซึ่งจะบรรจุรายละเอียดของอ็อบเจกต์ต่าง ๆ ที่เป็นองค์ประกอบอยู่ในหน้าต่างรูปภาพที่เป็น GUI ซึ่งเราได้สร้างขึ้น

15758063
ร.ร.
ท369D
2552

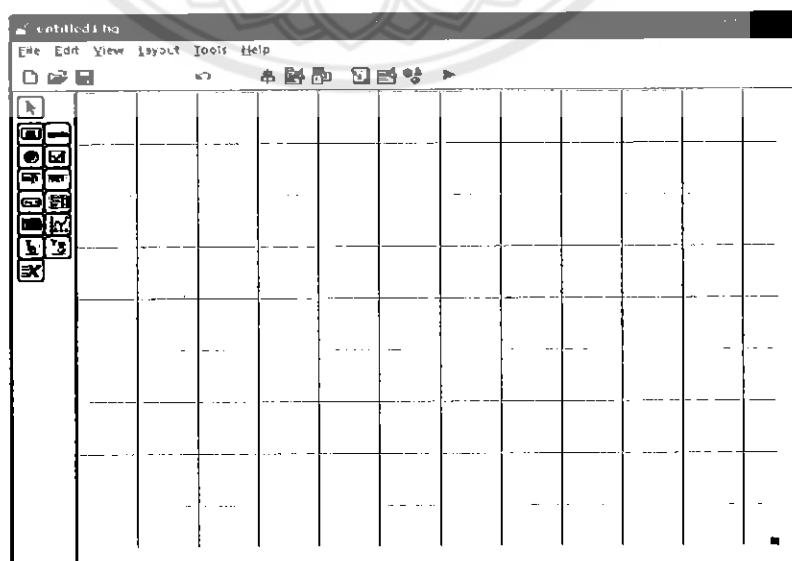
- 2) ไฟล์คำสั่ง (M-file) ซึ่งบรรจุฟังก์ชันที่กำหนดการทำงานของ GUI ที่สร้างขึ้นรวมถึงฟังก์ชันที่ตอบกลับ (Callbacks) ทั้งหมด โดยฟังก์ชันที่ตอบกลับเหล่านี้จะบรรจุเป็นฟังก์ชันย่อย (Subfunction) อยู่ในไฟล์คำสั่ง

การพัฒนา GUI ขึ้นใน MATLAB มีขั้นตอนการสร้างโดยเริ่มต้นจากการพิมพ์คำว่า `guide` ลงในหน้าต่างคำสั่งของ MATLAB จะปรากฏหน้าต่างดังรูปที่ 3.5



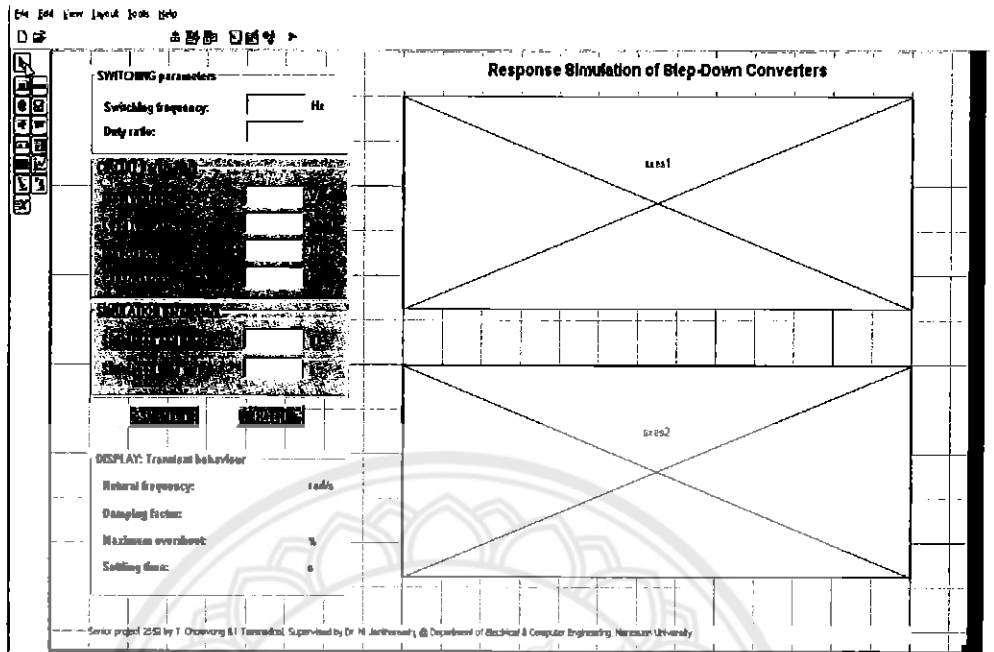
รูปที่ 3.5 หน้าต่างของ GUIDE

เมื่อกด OK แล้วจะได้หน้าต่างดังรูปที่ 3.6



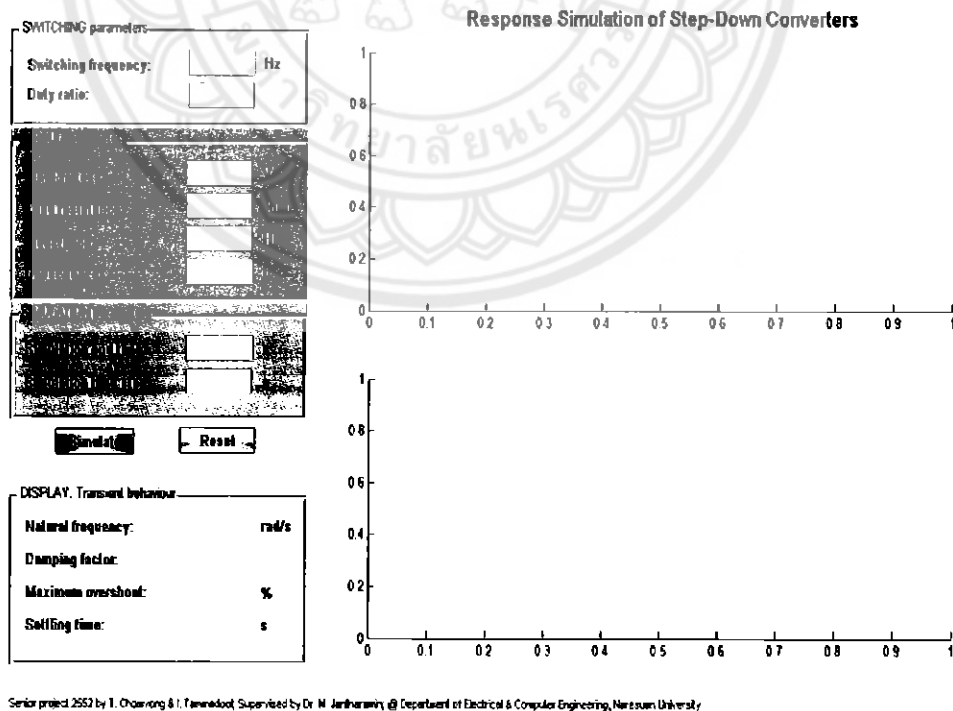
รูปที่ 3.6 หน้าต่าง Untitled1.fig

คณะผู้ดำเนินโครงการได้ออกแบบและสร้าง GUI ตามที่ต้องการ ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 การจำลองผลตอบสนองเพื่อสะดวกต่อผู้ใช้

หลังจากนั้น กด RUN จะปรากฏหน้าต่างดังรูปที่ 3.8 ซึ่งพร้อมสำหรับการรับค่าพารามิเตอร์ต่าง ๆ จากผู้ใช้งานก่อนทำการจำลองผลตอบสนองของวงจรต่อไป



รูปที่ 3.8 การจำลองผลตอบสนองเพื่อสะดวกต่อผู้ใช้เมื่อยังไม่ป้อนค่า

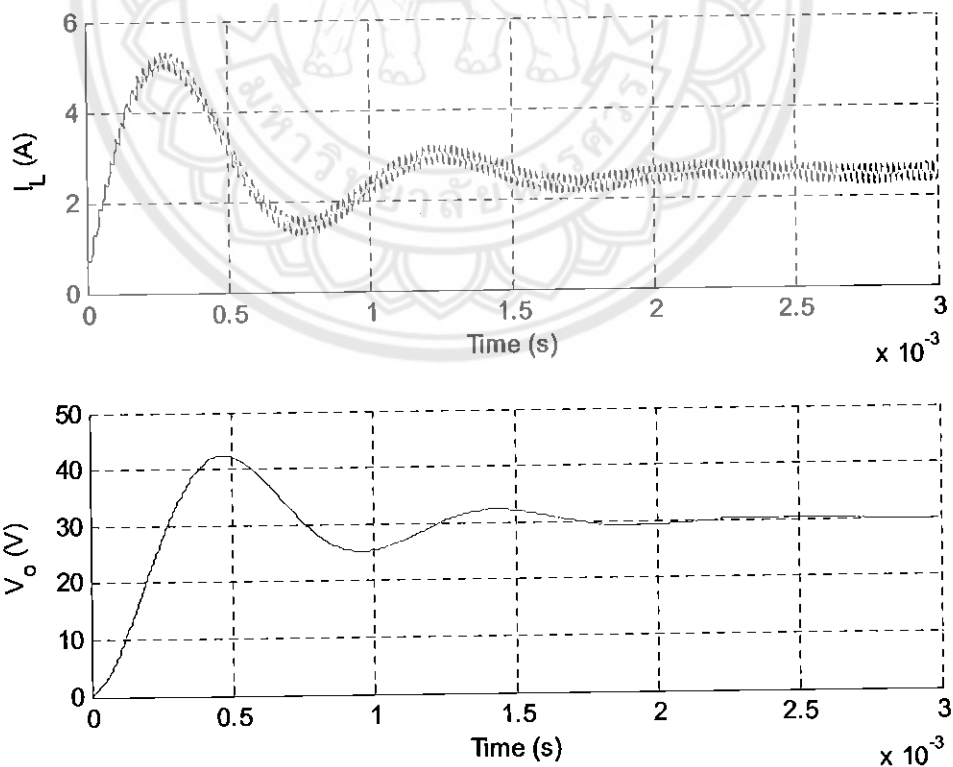
บทที่ 4

ผลการดำเนินโครงการ

เมื่อได้ทำการจำลองผลตอบสนองของวงจรทอนระดับแรงดันเป็นที่เรียบร้อยแล้วก็จะสามารถวิเคราะห์ผลตอบสนองว่าสัญญาณของแรงดันด้านออกและกระแสของตัวเหนี่ยวนำว่ามีผลตรงกับสมการที่คำนวณหรือไม่ ส่งผลกระทบต่อลักษณะของผลตอบสนองหรือเปล่าและหลังจากนั้นก็ได้อำลองผลตอบสนองให้มีความสะดวกแก่ผู้ใช้งาน

4.1 ผลตอบสนองของวงจรทอนระดับแรงดัน

เมื่อกำหนดให้ $V_s = 100 \text{ V}$; $R = 12 \text{ } \Omega$; $L = 1 \text{ mH}$; $C = 22 \text{ } \mu\text{F}$; $f_s = 50 \text{ kHz}$; $D = 0.3$ และกำหนดขั้นเวลาในการจำลองผล $\Delta t = 1 \text{ } \mu\text{s}$ ซึ่งผลตอบสนองที่จำลองได้ถูกแสดงในรูปที่ 4.1



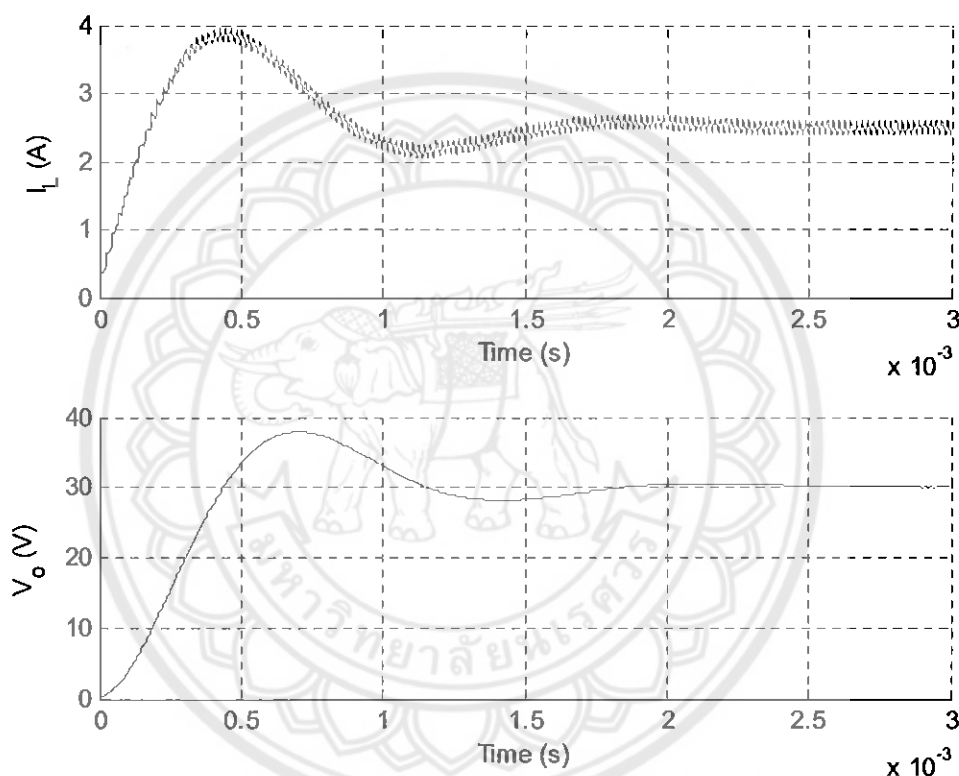
รูปที่ 4.1 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $L = 1 \text{ mH}$

จากสมการที่ (2.17) $V_o = D \cdot V_g = 0.3 \times 100 = 30 \text{ V}$

จากสมการที่ (2.20) $I_L = V_o / R = 30 / 12 = 2.5 \text{ A}$

จะเห็นว่าในสถานะอยู่ตัว ค่าเฉลี่ยของแรงดันด้านออกและกระแสของตัวเหนี่ยวนำที่ได้จากการจำลองผลมีค่าตรงกับที่คำนวณจากสมการที่ (2.17) และ (2.20) ตามลำดับ

เมื่อกำหนดให้ $L = 2 \text{ mH}$ ในขณะที่ค่าตัวแปรและพารามิเตอร์อื่น ๆ คงเดิม จะได้ผลตอบสนองดังรูปที่ 4.2



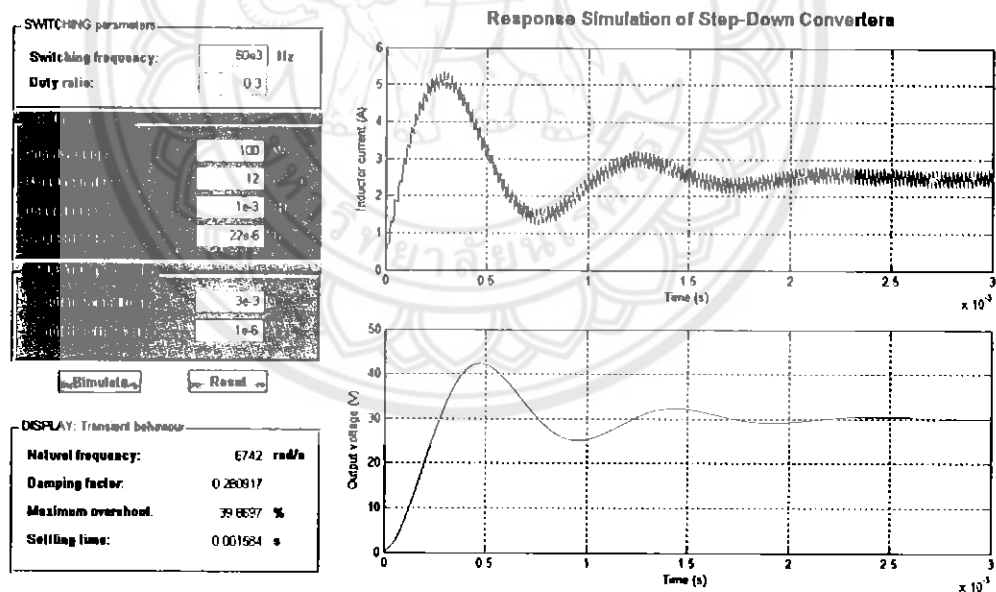
รูปที่ 4.2 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $L = 2 \text{ mH}$

จากรูปที่ 4.2 จะเห็นว่าการเปลี่ยนค่า L ไม่ส่งผลต่อค่าเฉลี่ยของแรงดันด้านออกและกระแสของตัวเหนี่ยวนำให้เปลี่ยนไป แต่ส่งผลต่อลักษณะของผลตอบสนอง นั่นคือการเพิ่มค่าความเหนี่ยวนำจะลดการแกว่ง (Oscillation) ของผลตอบสนอง และทำให้ค่าความพลีวของสัญญาณกระแสของตัวเหนี่ยวนำลดลง นอกจากนี้ยังทำให้ค่าพุ่งเกินสูงสุด (Overshoot) ของผลตอบสนองต่ำลง แต่ทำให้วงจรตอบสนองช้าลง

4.2 การจำลองผลการตอบสนองของวงจรผ่าน GUI

เมื่อทำการจำลองผลการตอบสนองของวงจรทอนระดับแรงดันในหัวข้อที่ 4.1 โดยที่ยังไม่ได้ทำการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ หลังจากนั้นได้ทำการสร้างส่วนต่อประสานกราฟิกผู้ใช้จะเห็นได้ว่าผู้ใช้สามารถป้อนค่าพารามิเตอร์ต่าง ๆ ได้ง่ายและสะดวกกว่าโดยที่ไม่ต้องเข้าไปเปลี่ยนค่าที่โปรแกรม เมื่อสร้างส่วนต่อประสานกราฟิกผู้ใช้สำเร็จแล้ว ผู้ดำเนินโครงการได้ทดลองเปลี่ยนค่าพารามิเตอร์ต่าง ๆ ไม่ว่าจะเป็น ค่าความต้านทาน, ค่าความเหนี่ยวนำ, ค่าความจุ, คิวดีไซเคิล และค่าความถี่สวิตซิง จากนั้นวิเคราะห์ผลการตอบสนองของวงจรทอนระดับแรงดันซึ่งจะได้ผลการตอบสนองดังเช่นรูปที่ 4.3, 4.4, 4.5 และ 4.6

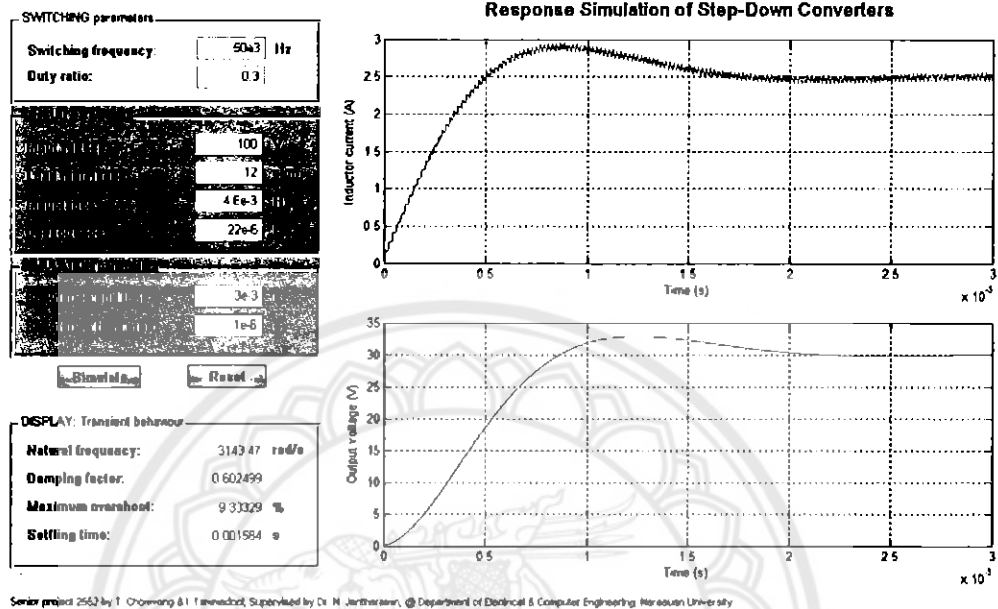
ทำการป้อนค่าต่างๆลงในรูปที่ 3.8 โดยค่าพารามิเตอร์ต่างๆ มีค่าเท่ากับกับในหัวข้อ 4.1 เสร็จเรียบร้อยแล้วกดปุ่ม Simulate ก็จะได้ ค่าความถี่ธรรมชาติ, ตัวประกอบการหน่วง, ค่าพุงเกินสูงสุด และเวลาสู่สมดุล จากการคำนวณแสดงผลบนหน้าจอแล้วแสดงผลตอบสนองของกระแสของตัวเหนี่ยวนำกับแรงดันด้านออก แสดงผลออกมาเป็นกราฟดังรูปที่ 4.3



Senior project 2552 by T. Chonwong & T. Tanwadol, Supervised by Dr. H. Jiratharwan, @ Department of Electrical & Computer Engineering, Maejoor University.

รูปที่ 4.3 การจำลองผลตอบสนองเพื่อสะดวกต่อผู้ใช้เมื่อป้อนค่าแล้ว

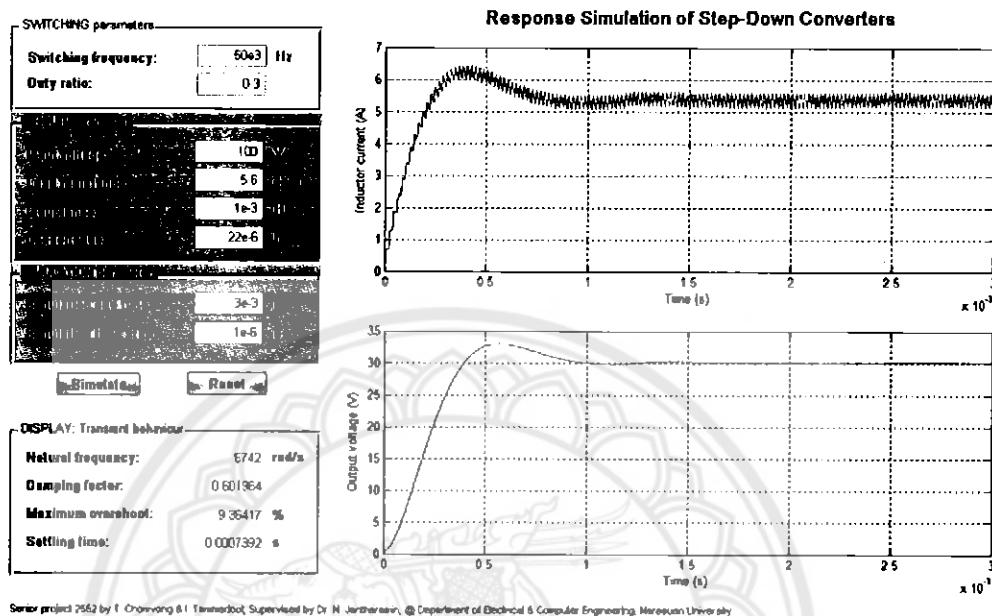
เมื่อกำหนดให้ $L = 4.6 \text{ mH}$ ในขณะที่ค่าตัวแปรและพารามิเตอร์อื่น ๆ คงเดิม จะได้ผลตอบสนองดังรูปที่ 4.4



รูปที่ 4.4 สัญญาณกระแสของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $L = 4.6 \text{ mH}$

จากรูปที่ 4.4 จะเห็นว่า การเปลี่ยนค่า L ไม่ส่งผลต่อค่าเฉลี่ยของแรงดันด้านออกและกระแสของตัวเหนี่ยวนำให้เปลี่ยนไป แต่ส่งผลต่อลักษณะของผลตอบสนอง นั่นคือการลดค่าความเหนี่ยวนำจะลดการแกว่งของผลตอบสนอง และทำให้ค่าความพลีวของสัญญาณกระแสของตัวเหนี่ยวนำลดลง นอกจากนี้ยังทำให้ค่าพุ่งเกินสูงสุดของผลตอบสนองลดลง และทำให้วงจรตอบสนองช้าลง เนื่องจากผลของค่าความถี่ธรรมชาติมีค่าลดลง

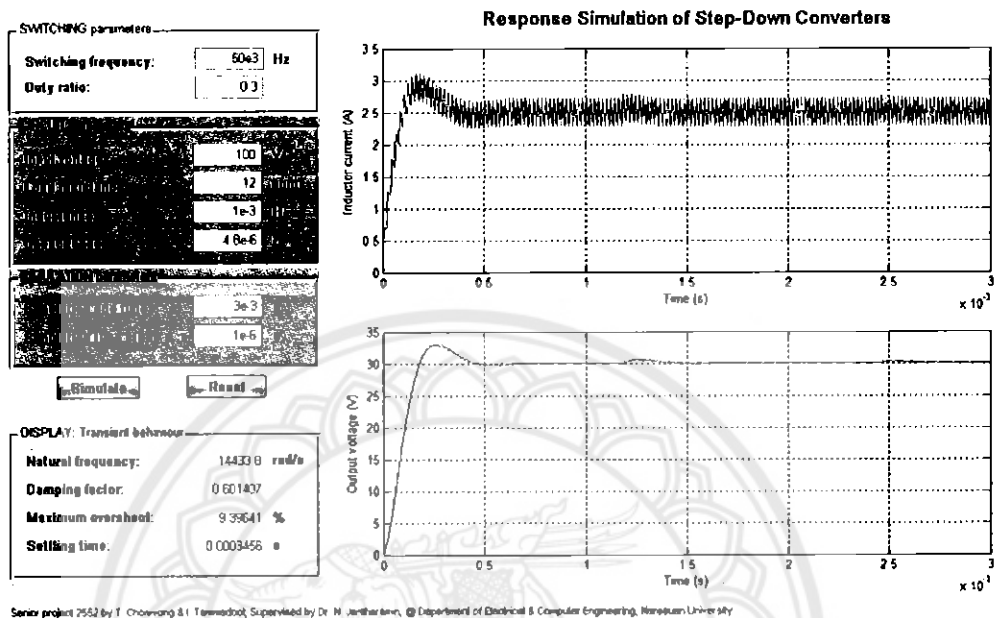
เมื่อกำหนดให้ $R = 5.6 \Omega$ ในขณะที่ค่าตัวแปรและพารามิเตอร์อื่น ๆ คงเดิม จะได้ผลตอบสนองดังรูปที่ 4.5



รูปที่ 4.5 ลักษณะกระแสของตัวเหนี่ยวนำและแรงดันค้ำด้านออก เมื่อ $R = 5.6 \Omega$

จากรูปที่ 4.5 จะเห็นว่า การเปลี่ยนค่า R ไม่ส่งผลต่อค่าเฉลี่ยของแรงดันค้ำด้านออก แต่ส่งผลต่อค่าเฉลี่ยของกระแสของตัวเหนี่ยวนำ และส่งผลต่อลักษณะของผลตอบสนอง นั่นคือการเพิ่มความเหนี่ยวนำจะลดการแกว่งของผลตอบสนอง นอกจากนี้ยังทำให้ค่าพุงเกินสูงสุดของผลตอบสนองลดลง แต่ไม่ส่งผลต่อการตอบสนองของวงจร เนื่องจากค่าความถี่ธรรมชาติไม่เปลี่ยนแปลง

เมื่อกำหนดให้ $C = 4.8 \mu F$ ในขณะที่ค่าตัวแปรและพารามิเตอร์อื่น ๆ คงเดิม จะได้ผลตอบสนองดังรูปที่ 4.6



รูปที่ 4.6 ลักษณะการตอบสนองของตัวเหนี่ยวนำและแรงดันด้านออก เมื่อ $C = 4.8 \mu F$

จากรูปที่ 4.6 จะเห็นว่า การเปลี่ยนค่า C ไม่ส่งผลต่อค่าเฉลี่ยของแรงดันด้านออกและกระแสของตัวเหนี่ยวนำให้เปลี่ยนไป แต่ส่งผลต่อลักษณะของผลตอบสนอง นั่นคือการเพิ่มค่าความเหนี่ยวนำจะลดการแกว่งของผลตอบสนอง นอกจากนี้ยังทำให้ค่าพุงเกินสูงสุดของผลตอบสนองต่ำลง และทำให้วงจรตอบสนองเร็วขึ้น เนื่องจากผลของค่าความถี่ธรรมชาติมีค่าเพิ่มขึ้น

บทที่ 5

สรุปผลการดำเนินโครงการ

5.1 สรุปผล

ในโครงการนี้ คณะผู้ดำเนินโครงการได้ทำการศึกษาการทำงานของวงจรทอนระดับแรงดันกระแสตรง และการวิเคราะห์ผลตอบสนองด้วยวิธีเฉลี่ยปริภูมิสถานะ จากนั้นได้ทำการจำลองผลตอบสนองของวงจรทอนระดับแรงดัน นั่นคือ ค่ากระแสของตัวเหนี่ยวนำ และค่าแรงดันของตัวเก็บประจุ (หรือแรงดันคั่นออกของวงจร) โดยการเขียน โปรแกรมใน MATLAB นอกจากนี้ในส่วนท้ายของการดำเนินโครงการ คณะผู้ดำเนินโครงการได้สร้างส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) เพื่อให้เกิดความสะดวกในการใช้งาน โปรแกรมการจำลองผลที่สร้างขึ้น

โปรแกรมการจำลองผลตอบสนองของวงจรที่สร้างขึ้นช่วยให้ทราบถึงผลของพารามิเตอร์ต่าง ๆ ที่มีต่อพฤติกรรมของผลตอบสนองล่วงหน้าก่อนที่จะนำไปสร้างเป็นวงจรจริง จึงช่วยให้ในกระบวนการออกแบบวงจรนั้นเราสามารถนำค่าความถี่สวิทชิง และค่าของอุปกรณ์ต่าง ๆ ซึ่งเป็นส่วนประกอบสำคัญของวงจร ได้แก่ ค่าความต้านทาน ค่าความเหนี่ยวนำ และค่าความจุไฟฟ้า มาใช้จำลองผลเพื่อให้เกิดผลตอบสนองตามที่ต้องการก่อน ส่งผลให้เราสามารถเลือกค่าของอุปกรณ์ได้อย่างเหมาะสมก่อนนำไปใช้งานจริงได้

5.2 ปัญหาที่พบ

- 1) เนื่องจากรูปสัญญาณใน GUI ที่ได้สร้างขึ้นไม่สามารถปรับขนาดให้ใหญ่ขึ้นหรือเล็กลงได้ ทำให้ในบางกรณีไม่สามารถดูพฤติกรรมของสัญญาณ และเกิดความยุ่งยากในการคำนวณหาความพลັวของสัญญาณที่สนใจได้
- 2) หน่วยการวัด (Scale) ในแนวแกนตั้ง (แกน y) ของรูปสัญญาณใน GUI ที่สร้างขึ้นจะเปลี่ยนโดยอัตโนมัติเมื่อมีการจำลองผลใหม่ทุกครั้ง จึงอาจทำให้เกิดความสับสนได้ในการวิเคราะห์สัญญาณ

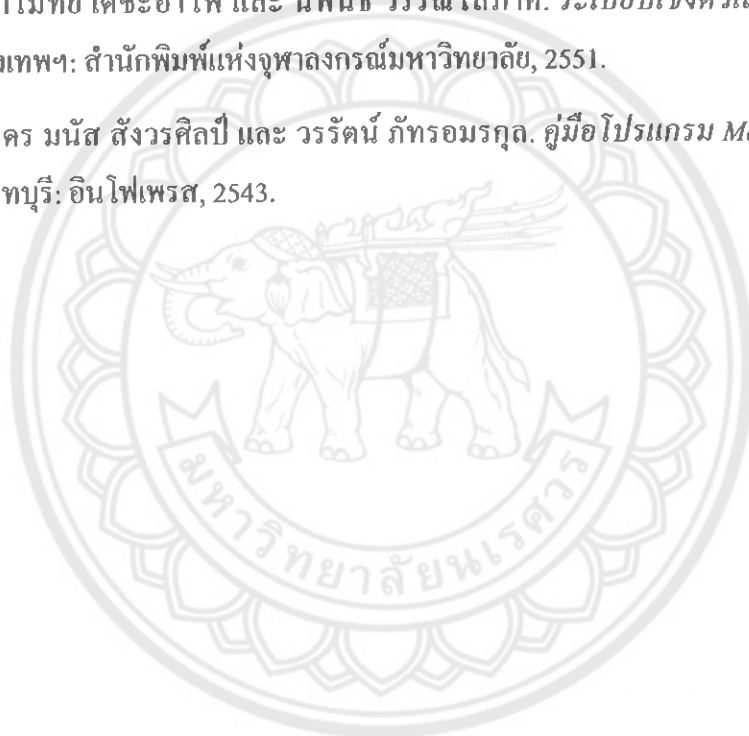
5.3 แนวทางในการพัฒนาต่อไป

- 1) ถึงแม้ในโครงการนี้ได้ดำเนินการจำลองผลตอบแทนของวงจรทอนระดับแรงดัน อย่างไรก็ตาม แนวทางการดำเนินงานสามารถนำไปใช้กับการจำลองผลของตัวแปลงผันกำลังชนิดอื่นได้ เช่น วงจรทอนระดับแรงดัน วงจรทอนทอนระดับแรงดัน เป็นต้น
- 2) สามารถปรับปรุงและเพิ่มในส่วนของการสร้างส่วนต่อประสานกราฟิกกับผู้ใช้ เช่น การเพิ่มแถบเลื่อน (Scroll bar) และรายการเลือกแบบดึงลง (Drop-down menu) เพื่อให้สามารถใช้งานได้ง่ายยิ่งขึ้น



เอกสารอ้างอิง

- [1] รศ.ดร.ธีระเชษฐ์ ชันเงิน และ วุฒิพล ธาราธิรเศรษฐ์. *อิเล็กทรอนิกส์กำลัง*. กรุงเทพฯ: วี.เจ.พรินติ้ง, 2550.
- [2] R. D. Middlebrook and S. Cuk, "A general unified approach to modelling switching-converter power stages," *IEEE Power Electronics Specialists Conference*, 1976, pp. 18-34.
- [3] Katsuhiko Otaka, "Modern control engineering," *Prentice Hall International, Inc*, 1997.
- [4] ปราโมทย์ เคชออำไพ และ นิพนธ์ วรรณโสภาค. *ระเบียบเชิงตัวเลขในงานวิศวกรรม*. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2551.
- [5] รศ.ดร มนัส สังวรศิลป์ และ วรรัตน์ ภัทรอมรกุล. *คู่มือโปรแกรม Matlab ฉบับสมบูรณ์*. นนทบุรี: อินโฟเพรส, 2543.



ภาคผนวก

รหัสต้นฉบับในการจำลองผลตอบแทนของวงจร และการสร้าง GUI

```

function varargout = SOB(varargin)

% SOB M-file for SOB.fig

% SOB, by itself, creates a new SOB or raises the existing
% singleton*.

% H = SOB returns the handle to a new SOB or the handle to
% the existing singleton*.

% SOB('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in SOB.M with the given input arguments.

% SOB('Property','Value',...) creates a new SOB or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before SOB_OpeningFunction gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to SOB_OpeningFcn via varargin.

% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".

% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help SOB

% Last Modified by GUIDE v2.5 05-Apr-2010 13:46:18

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

```

```

gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @SOB_OpeningFcn, ...
                  'gui_OutputFcn', @SOB_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before SOB is made visible.

function SOB_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% varargin   command line arguments to SOB (see VARARGIN)

% Choose default command line output for SOB

handles.output = hObject;

```

```
% Update handles structure

guidata(hObject, handles);

% UIWAIT makes SOB wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = SOB_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function resistor_Callback(hObject, eventdata, handles)

% hObject handle to resistor (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of resistor as text

% str2double(get(hObject,'String')) returns contents of resistor as a double

resistor = str2double(get(hObject, 'String'));

if isnan(resistor)

    set(hObject, 'String', 0);

    errordlg('Input must be a number','Error');

end

% Save the new density value
```

```

handles.metricdata.resistor = resistor;

guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.

function resistor_CreateFcn(hObject, eventdata, handles)

% hObject    handle to resistor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function inductor_Callback(hObject, eventdata, handles)

% hObject    handle to inductor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of inductor as text
%       str2double(get(hObject,'String')) returns contents of inductor as a
double

inductor = str2double(get(hObject, 'String'));

if isnan(inductor)

    set(hObject, 'String', 0);

```



```

        errordlg('Input must be a number','Error');
    end

    % Save the new density value

    handles.metricdata.inductor = inductor;

    guidata(hObject,handles)

    % --- Executes during object creation, after setting all properties.

    function inductor_CreateFcn(hObject, eventdata, handles)

    % hObject    handle to inductor (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    %         See ISPC and COMPUTER.
    usewhitebg = 1;
    if usewhitebg
        set(hObject,'BackgroundColor','white');
    else
        set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
    end

    function capacitor_Callback(hObject, eventdata, handles)

    % hObject    handle to capacitor (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of capacitor as text
    %         str2double(get(hObject,'String')) returns contents of capacitor as a
double

```

```
capacitor = str2double(get(hObject, 'String'));

if isnan(capacitor)

    set(hObject, 'String', 0);

    errorDlg('Input must be a number','Error');

end

% Save the new density value

handles.metricdata.capacitor = capacitor;

guidata(hObject,handles)

% --- Executes during object creation, after setting all properties.

function capacitor_CreateFcn(hObject, eventdata, handles)

% hObject    handle to capacitor (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.

usewhitebg = 1;

if usewhitebg

    set(hObject,'BackgroundColor','white');

else

    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));

end

% --- Executes on button press in simulate.

function simulate_Callback(hObject, eventdata, handles)

% hObject    handle to simulate (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

L1=get(handles.inductor,'string');

L=str2num(L1);

R1=get(handles.resistor,'string');

R=str2num(R1);

C1=get(handles.capacitor,'string');

C=str2num(C1);

tend1=get(handles.edit7,'string');

tend=str2num(tend1);

dt1=get(handles.edit4,'string');

dt=str2num(dt1);

D1=get(handles.edit8,'string');

D=str2num(D1);

Vs1=get(handles.edit6,'string');

Vs=str2num(Vs1);

fs1=get(handles.edit5,'string');

fs=str2num(fs1);

% Switching period

Ts = 1/fs ;

%%%Parameter

n = 1;

t = 0;

IL_old = 0;
```

```

Vo_old = 0;

%%%Program

while (t < tend)

    if mod(t,Ts) < D*Ts

        IL_new = IL_old + (Vs-Vo_old)/L*dt;

        if IL_new < 0; IL_new = 0; end

        Vo_new = Vo_old + (IL_old-Vo_old/R)/C*dt;

    else

        IL_new = IL_old - (Vo_old)/L*dt;

        if IL_new < 0; IL_new = 0; end

        Vo_new = Vo_old + (IL_old-Vo_old/R)/C*dt;

    end

    %%%Data save

    time(n) = t;

    lind(n) = IL_new;

    Vout(n) = Vo_new;

    IL_old = IL_new;

    Vo_old = Vo_new;

    t = t + dt;

    n = n + 1;

end

%%%Wn & S Answer

Wn = 1/(sqrt(L*C)) ;

S = (1/(2*R))*(sqrt(L/C)) ;

```

```

Mp = 100*exp(-pi*S/sqrt(1-S^2)); % Percent max. overshoot

ts = 3/(Wn*S); % Settling time (converge within 5% of final value)

set(handles.Wn, 'String', Wn);

set(handles.S, 'String', S);

set(handles.Mp, 'String', Mp);

set(handles.ts, 'String', ts);

%%%data plot

axes(handles.axes1)

plot(time,lind,'blue','linewidth',1);

xlabel('Time (s)');

ylabel('Inductor current (A)');

grid on

axes(handles.axes2)

plot(time,Vout,'r','linewidth',1);

xlabel('Time (s)');

ylabel('Output voltage (V)');

grid on

% --- Executes on button press in reset.

function reset_Callback(hObject, eventdata, handles)

% hObject    handle to reset (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

cla(handles.axes1,'reset')

cla(handles.axes2,'reset')

```

```
set(handles.inductor,'string',' ');

set(handles.resistor,'string',' ');

set(handles.capacitor,'string',' ');

set(handles.edit7,'string',' ');

set(handles.edit4,'string',' ');

set(handles.edit8,'string',' ');

set(handles.edit6,'string',' ');

set(handles.edit5,'string',' ');

set(handles.Wn,'string',' ');

set(handles.S,'string',' ');

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2

function edit4_Callback(hObject, eventdata, handles)
```

```
% hObject handle to edit4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text

% str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.

function edit4_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit5_Callback(hObject, eventdata, handles)

% hObject handle to edit5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text

% str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
```

```
function edit5_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit5 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.

if ispc

    set(hObject,'BackgroundColor','white');

else

    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));

end

function edit6_Callback(hObject, eventdata, handles)

% hObject    handle to edit6 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text

%    str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.

function edit6_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit6 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.
```



```
if ispc

    set(hObject,'BackgroundColor','white');

else

    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));

end

function edit7_Callback(hObject, eventdata, handles)

% hObject    handle to edit7 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of edit7 as a double
% --- Executes during object creation, after setting all properties.

function edit7_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit7 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%         See ISPC and COMPUTER.

if ispc

    set(hObject,'BackgroundColor','white');

else

    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));

end

function edit8_Callback(hObject, eventdata, handles)
```

```

% hObject  handle to edit8 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles  structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text

%   str2double(get(hObject,'String')) returns contents of edit8 as a double

% --- Executes during object creation, after setting all properties.

function edit8_CreateFcn(hObject, eventdata, handles)

% hObject  handle to edit8 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles  empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%   See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit9_Callback(hObject, eventdata, handles)

% hObject  handle to edit9 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles  structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text

%   str2double(get(hObject,'String')) returns contents of edit9 as a double

% --- Executes during object creation, after setting all properties.

```

```
function edit9_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit9 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.

if ispc

    set(hObject,'BackgroundColor','white');

else

    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));

end

function edit16_Callback(hObject, eventdata, handles)

% hObject    handle to edit16 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text

%    str2double(get(hObject,'String')) returns contents of edit16 as a double

% --- Executes during object creation, after setting all properties.

function edit16_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit16 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.
```

```
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

