โปรแกรมการออกแบบระบบควบคุม

ด้วยการกำหนดตำแหน่งโพล

CONTROL SYSTEM DESIGN

BY

POLE PLACEMENT TECHNIQUE

โดย

นาย กฤษฎา แซ่โง้ว

นาย สมศักดิ์ ถกลชัยศรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032631

ปริญญานิพนธ์ปีการศึกษา 2535

ภาควิชา วิศวกรรมระบบควบคุม

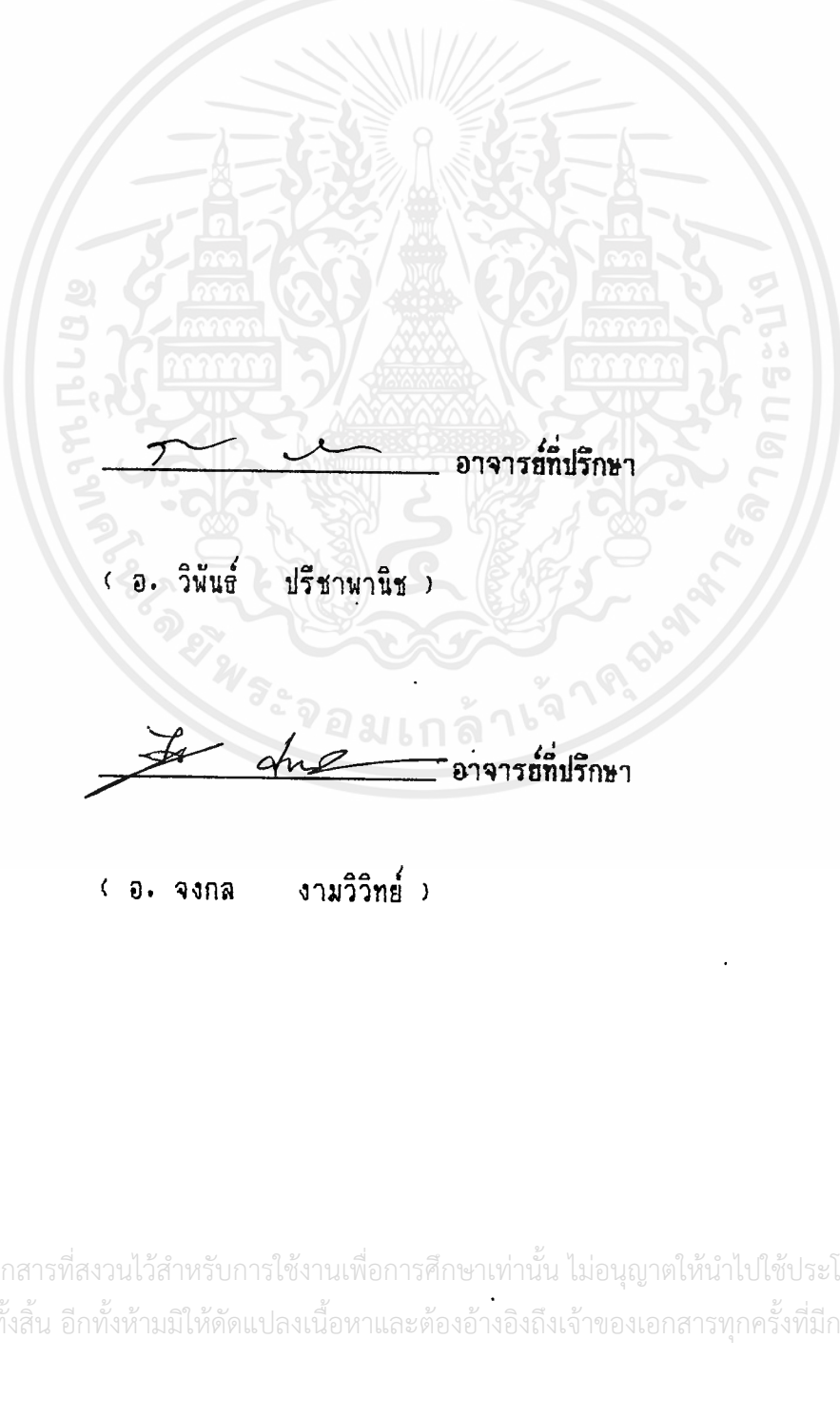
คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมการออกแบบระบบควบคุมด้วยการกำหนดตำแหน่งโพล

CONTROL SYSTEM DESIGN BY POLE PLACEMENT TECHNIQUE

ผู้จัดทำ 1. นาย กฤษฎา แซ่โง้ว 321009

2. นาย สมศักดิ์ ถกลชัยศรี 321356



_____ อาจารย์ที่ปรึกษา

(อ. วิพันธ์ ปรีชาพานิช)

_____ อาจารย์ที่ปรึกษา

(อ. จงกล งามวิวิทย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032631

โปรแกรมออกแบบระบบควบคุมเชิงเส้นด้วยการกำหนดตำแหน่งโพล

นาย กฤษฎา แซ่โง้ว 321009
นาย สมศักดิ์ ถกลชัยศรี 321356

อาจารย์ที่ปรึกษา
อ. วิพันธ์ ปรีชาพานิช
อ. จงกล งามวิวิทย์

บทคัดย่อ

ในการวิเคราะห์ออกแบบระบบควบคุมสมัยใหม่ นิยมเขียนในรูปสมการสเตท แล้วทำการออกแบบด้วยวิธีการต่าง ๆ การกำหนดตำแหน่งโพลให้กับระบบ เป็นวิธีการหนึ่งที่นิยมใช้กันมาก ทำได้โดยการกำหนดตำแหน่งโพลที่ต้องการ แล้วคำนวณค่าเมตริกซ์สเตทป้อนกลับ K สำหรับระบบที่ไม่สามารถทำการป้อนกลับจากตัวแปรสเตทได้ เนื่องจากตัวแปรสภาวะดังกล่าวไม่สามารถวัดค่าได้โดยตรง จึงจำเป็นต้องสร้างตัวสังเกตสเตทที่เหมาะสม ซึ่งจะสร้างตัวแปรสภาวะชุดใหม่ ที่สามารถวัดค่าได้โดยตรง และมีค่าใกล้เคียงกับตัวแปรสเตทชุดเดิมของระบบมากที่สุด การออกแบบตัวสังเกตนี้ ทำโดยกำหนดตำแหน่งโพลที่เหมาะสมให้กับตัวสังเกตสเตท แล้วหาค่าเมตริกซ์ K_e สำหรับตัวสังเกตสเตทนั้น

การออกแบบจะสะดวกและรวดเร็วขึ้นอย่างมากโดยการนำคอมพิวเตอร์เข้ามาช่วย ดังนั้นโครงงานที่ทำขึ้นจึงเน้นการออกแบบโดยเขียนโปรแกรมให้คอมพิวเตอร์ช่วยคำนวณค่าที่ต้องการ ซึ่งได้แก่ เมตริกซ์ K และ K_e ตามลำดับ และมีการคำนวณค่าตัวแปรสเตทและเอาท์พุทที่เวลาต่าง ๆ เพื่อนำมาแสดงกราฟ ทำให้ผู้ออกแบบสามารถเลือกตำแหน่งโพลที่เหมาะสมที่สุดได้จากลักษณะเส้นกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL SYSTEM DESIGN BY POLE PLACEMENT TECHNIQUE

CHRISADA SAE-NGOW 321009

SOMSAK THAKOLCHAISRI 321356

ADVISOR

WIPAN PREECHAPANICH

JONGKOL NGAMWIWIT

ABSTRACT

In modern control analysis and design of control systems,it is preferable to present the interested systems in the form of state eqautions, allowing many practical analysis and design methods to applied. Pole placement,being one of the most extensively utilized,requires the locations of poles to compute the corresponding state feedback gain matrix K. But for a system whose state variables can not directly be approached,that is,not physically measurable,a proper state observer is needed. An appropriate state observer will generate a new set of state variables which are directly measurable and are as close to those unmeasurable state variables of the same system as possible. Traditionally,design of a state observer can be done by assigning all the desired poles for the observer,and the observer gain matrix K_e can consequently be determined.

On account of the fact that the task of calculating K and K_e is quite complicated and labour intensive,a computer is inevitably introduced. Because of this,this project is concerned with design of systems via computers,on in other words,writing a programme for computing the values of K and K_e,respectively. another advantage of the use of computers here is that it is very convenient for the users to select the most suitable poles that give optimal responses which can also be simulated on computers.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ที่ปรึกษาโครงงาน อาจารย์ วินัธ ปรีชาพานิช และอาจารย์ จงกล งามวิวิทย์ ที่ได้ให้คำปรึกษาระหว่างทำโครงงานนี้ ขอบคุณเพื่อน ๆ ภาคระบบควบคุมทุกคนที่ได้ให้ข้อติชมต่าง ๆ เกี่ยวกับโครงงานนี้ จนโครงงานนี้ได้รับการปรับปรุงอย่างเหมาะสมและสำเร็จลุล่วงมาด้วยดี

และขอบคุณทุก ๆ ท่านที่มีส่วนให้โครงงานนี้สำเร็จมาด้วยดี ไม่ว่าจะเป็นด้านใดก็ตาม

****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
บทนำ	1
บทที่ 1 การวิเคราะห์และออกแบบระบบควบคุมโดยวิธีสเตทสเปซ	2
บทที่ 2 การออกแบบระบบควบคุม	5
2.1 การออกแบบระบบโดยการกำหนดตำแหน่งโพล	6
2.2 การออกแบบตัวสังเกตสเตท	7
บทที่ 3 ระบบควบคุมสัญญาณไม่ต่อเนื่อง	11
บทที่ 4 การ Simulation ระบบต่อเนื่องและไม่ต่อเนื่อง	12
4.1 กรณีสัญญาณไม่ต่อเนื่อง(Discrete)	12
4.2 กรณีสัญญาณต่อเนื่อง(Continuous)	12
บทที่ 5 การออกแบบโปรแกรม	14
บทที่ 6 โปรแกรม LINEAR.EXE	16
บทที่ 7 โปรแกรมกำหนด System	19
บทที่ 8 โปรแกรม Simulation	22
บทที่ 9 โปรแกรม MATRIX.EXE	25
บทที่ 10 โปรแกรมแสดงผลกราฟ	30
บทที่ 11 โปรแกรม ANALYSIS	36
11.1 โปรแกรมกำหนดตำแหน่งโพล [POLE.EXE]	36
11.2 โปรแกรมคำนวนตัวสังเกตสเตท [OBSERVER.EXE]	38
บทที่ 12 ตัวอย่าง	40
บทที่ 13 สรุปและวิจารณ์	44
บรรณานุกรม	45
ภาคผนวก	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

		หน้า
รูปที่ 1.	โครงสร้างรวมของโปรแกรม	14
รูปที่ 2.	แสดงชื่อโปรแกรมต่างๆ	15
รูปที่ 3.	แสดงแนวคิดการเรียกโปรแกรมต่างๆ	16
รูปที่ 4.	แผนผังแสดงรายละเอียดของโปรแกรม LINEAR.EXE	17
รูปที่ 5.	เมนูหลักของ LINEAR.EXE	18
รูปที่ 6.	แสดงการใช้เมนู Directory	18
รูปที่ 7.	แสดงการรับชื่อไฟล์ก่อนส่งให้ GRAPH.EXE ต่อไป	18
รูปที่ 8.	แผนผังแสดงรายละเอียดของโปรแกรม SYSTEM	19
รูปที่ 9.	แสดงเมนูหลักของ SYSTEM	20
รูปที่ 10.	แสดงการใช้เมนู Make และการรับค่าข้อมูล	20
รูปที่ 11.	แสดงการใช้เมนู Edit	21
รูปที่ 12.	แสดงการใช้เมนู Delete	21
รูปที่ 13.	flow chart แสดงการ simulation	23
รูปที่ 14.	แสดงการใส่อินพุต	24
รูปที่ 15.	แสดงการเลือกข้อมูลที่จะให้ save	24
รูปที่ 16.	แผนผังแสดงรายละเอียดของโปรแกรม MATRIX.EXE	26
รูปที่ 17.	แสดงเมนูหลัก ของโปรแกรม MATRIX.EXE	27
รูปที่ 18.	แสดงการแสดงข้อมูลเมตริกซ์ และ การใส่ comment	27
รูปที่ 19.	แสดงการใส่สมการพีชคณิต และ การใช้งานเมนู Cancel	28
รูปที่ 20.	แสดงการ save ข้อมูล	28
รูปที่ 21.	แสดงการใช้เมนู Directory	29
รูปที่ 22.	แสดงการใส่ค่าข้อมูลเมตริกซ์ และ การเปลี่ยนขนาดเมตริกซ์	29
รูปที่ 23.	flow chart ของการวาดกราฟ	31
รูปที่ 24.	แผนผังแสดงรายละเอียดของโปรแกรม GRAPH.EXE	32
รูปที่ 25.	แสดงเมนูหลักของ GRAPH.EXE	33

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 26. แสดงการวาดรูปกราฟ และการ List รายชื่อข้อมูล 33
รูปที่ 27. แสดงการเปลี่ยนรูปแบบการ plot และ แสดงเมนู X_axis 34
รูปที่ 28. แสดงการเปลี่ยนรูปแบบ carveและ แสดงเมนู Y_axis 34
รูปที่ 29. แสดงเมนู Plot_control 35
รูปที่ 30. แสดงการใช้เมนู Cursor ดูค่าข้อมูลที่จุดต่างๆ 35
รูปที่ 31. แสดงขั้นตอนการหา K 37
รูปที่ 32. แสดงการรับโพลใหม่ของ POLE.EXE (กรณีiอินพุต) 37
รูปที่ 33. แสดงขั้นตอนการหา Ke 39
รูปที่ 34. แสดงการรับ A ใหม่ของ OBSERVER.EXE (กรณีหลายเอาท์พุต) 39
รูปที่ 35. แสดงรูปกราฟของสเตทต่างๆ ของระบบที่ย้ายโพลแล้ว 41
รูปที่ 36. แสดงกราฟของ เอาท์พุตและ สเตท 43
เมื่อย้ายโพลไปอยู่บนแกนจริง(μ=-1,-3)
รูปที่ 37. แสดงกราฟของ เอาท์พุตและ สเตท 43
เมื่อย้ายโพลไปที่ตำแหน่ง μ=-1.8$\pm$2.4j
รูปที่ 38. แสดงกราฟของ เอาท์พุตและ สเตท 43
เมื่อย้ายโพลไปอยู่บนแกนจินตภาพ (μ=$\pm$2j)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

วิทยาการและเทคโนโลยีสาขาต่าง ๆ ในปัจจุบันนั้นมีความก้าวหน้าอย่างรวดเร็วมาก ไม่ว่าจะเป็นวิทยาการทางด้านไฟฟ้า เครื่องกล การสื่อสาร หรือคอมพิวเตอร์ ล้วนแล้วมีการพัฒนาขึ้นอย่างรวดเร็วกันทั้งสิ้น การวิเคราะห์และออกแบบระบบควบคุมใด ๆ ที่จะให้ผลเป็นที่พอใจนั้น จึงเป็นสิ่งที่น่าสนใจศึกษาเป็นอย่างยิ่ง และเนื่องจากการทำวิทยานิพนธ์ของนักศึกษารุ่นก่อน ๆ ที่ผ่านมา มักเป็นชิ้นงานซึ่งไม่ค่อยมีการสานต่อหรือพัฒนาต่อในรุ่นถัดไป ทั้งนี้เนื่องมาจากความยุ่งยากสำหรับผู้ที่จะมาสานต่อ เพราะจะต้องใช้เวลามากพอสมควรในการศึกษางานที่ทำขึ้นแล้วในรุ่นก่อน ดังนั้นจึงทำให้มีเวลาน้อยและจำกัดที่จะพัฒนาชิ้นงานนั้นให้ดีขึ้นไปอีก

ดังนั้น วิทยานิพนธ์ที่ทำขึ้นนี้ นอกจากจะเพื่อศึกษาและเขียนโปรแกรมอำนวยความสะดวกในการวิเคราะห์และออกแบบระบบควบคุมโดยวิธีการทาง Modern Control แล้ว ยังมีความหวังอย่างยิ่งที่จะให้โปรแกรมที่เขียนขึ้นนี้ ได้รับสานต่อหรือพัฒนาต่อ ทั้งนี้เพราะโปรแกรมที่เขียนขึ้นได้ออกแบบไว้โดยผู้ที่จะมาทำการพัฒนาต่อนั้นไม่ต้องศึกษารายละเอียดของโปรแกรมมากนัก เพียงศึกษาระบบฐานข้อมูลของโปรแกรมก็พอ ก็จะสามารถพัฒนาหรือสานต่อได้

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

การวิเคราะห์และออกแบบระบบควบคุมโดยวิธีสเตทสเปซ

ระบบควบคุมใด ๆ ไม่ว่าจะเป็นระบบทางวงจรไฟฟ้า ระบบทางเครื่องกลหรือระบบอื่น ๆ ล้วนแล้วสามารถเขียนให้อยู่ในรูปแบบจำลองทางคณิตศาสตร์ของระบบนั้น ๆ ได้ โดยแบบจำลองทางคณิตศาสตร์ของระบบสามารถแสดงในรูปสมการอนุพันธ์เชิงเส้น

$$y^{(n)}+a_1y^{(n-1)}+\ldots+a_{n-1}y+a_ny = b_0u^{(n)}+b_1u^{(n-1)}+\ldots+b_{n-1}u+b_nu \quad \ldots\ldots\ldots(ก)$$

เมื่อ u คือ อินพุทที่เข้าสู่ระบบ

y คือ เอาท์พุท

จากความรู้เรื่องการวิเคราะห์และออกแบบระบบแบบ Classical Control เราจะได้ทรานซเฟอร์ฟังก์ชันจากระบบอินพุทเดียวเอาท์พุทเดียวในสมการ (ก) ดังนี้

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0s^n + b_1s^{n-1} +\ldots+ b_{n-1}s + b_n}{s^n + a_1s^{n-1} +\ldots+ a_{n-1}s + a_n} \quad \ldots\ldots\ldots(ข)$$

แล้วเราจะทำการศึกษาคุณสมบัติ วิเคราะห์และออกแบบระบบจากทรานซเฟอร์ฟังก์ชันที่ได้ เพื่อให้ได้ระบบควบคุมที่มีคุณลักษณะตามที่ต้องการ

แต่ในการวิเคราะห์และออกแบบระบบควบคุมโดยแนว Modern Control เราจะเขียนสมการแสดงพฤติกรรม (Dynamic Equation) ของระบบให้อยู่ในรูปของระบบสมการเชิงอนุพันธ์อันดับหนึ่งของตัวแปรสภาวะ (State Variable) ดังนี้

$$\dot{X}(t) = AX(t) + Bu(t) \quad \ldots\ldots\ldots(ค)$$

$$y(t) = CX(t) + Du(t) \quad \ldots\ldots\ldots(ง)$$

เมื่อ X(t) คือ สเตทเวกเตอร์

u(t) คือ อินพุท

y(t) คือ เอาท์พุท

A,B,C,D เป็นเมตริกซ์ซึ่งสมาชิกแต่ละตัวภายในเมตริกซ์เป็นค่าคงที่และไม่แปรตามเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ระบบมีหลายอินพุทหลายเอาท์พุท ปริมาณ Scalar y(t) และ u(t) ก็จะเขียนอยู่ในรูปของเวกเตอร์ของเอาท์พุทและอินพุท ตามลำดับ ต่รงจุดนี้จะทำให้เห็นข้อดีของการวิเคราะห์และออกแบบในแนว Modern Control คือสามารถใช้ได้กับระบบหลายอินพุทหลายเอาท์พุทได้ด้วย

จากสมการ (ค) และ (ง) จะสามารถวิเคราะห์และออกแบบระบบในโดเมนของเวลาได้ และสมการดังกล่าว อาจจะได้มาจากการแปลงรูปสมการ (ก) หรือ (ข) ด้วยวิธีต่าง ๆ หรืออาจจะได้มาจากการเชื่อมโยงความสัมพันธ์ของตัวแปรหรือปริมาณต่าง ๆ ในระบบจริง แล้วจัดรูปสมการเสียใหม่ให้อยู่ในรูป (ค) และ (ง) ดังนั้นเราจะเห็นว่า เราสามารถจะเลือกกลุ่มหรือเซ็ตของตัวแปรสภาวะที่จะแสดงพฤติกรรมของระบบที่ต้องการได้ แต่เรานิยมเลือกตัวแปรสภาวะที่สามารถวัดหรือสังเกตได้จริง ๆ จากระบบ เพราะในการออกแบบระบบควบคุมป้อนกลับที่จะกล่าวถึงนี้ จะมีการนำค่าตัวแปรสภาวะเหล่านี้มาป้อนกลับด้วย

ก่อนที่จะกล่าวถึงการวิเคราะห์และออกแบบระบบที่เขียนในรูปสมการ (ค) และ (ง) จะขอแสดงความสัมพันธ์กันระหว่างการวิเคราะห์และออกแบบระบบควบคุมในโดเมนความถี่ (สมการ (ข)) และในโดเมนของเวลา (สมการ (ค) และ (ง)) ดังนี้

จากสมการ (ค) และ (ง) จะได้ว่า

$$sX(s) - AX(s) = Bu(s)$$

$$(sI-A)X(s) = Bu(s)$$

$$X(s) = (sI-A)^{-1}Bu(s)$$

$$Y(s) = CX(s) + Du(s)$$

$$G(s) = Y(s)/U(s) = C(sI-A)^{-1}B + D$$

ดังนั้น การวิเคราะห์และออกแบบในแนว Modern Control และ Classical Control ก็มีความสัมพันธ์กันอยู่ เพราะเราได้พิสูจน์แล้วว่า เราสามารถเขียนทรานซเฟอร์ฟังก์ชันของระบบ ในรูปของเมตริกซ์ A,B,C,D ในระบบที่มองในโดเมนของเวลาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ (ค) แสดงให้เห็นความสัมพันธ์ระหว่างตัวแปรสภาวะกับอินพุทที่ป้อนเข้ามา และหากเราสามารถป้อนสัญญาณควบคุมใด ๆ ซึ่งจะพาสภาวะเริ่มต้นใด ๆ ของระบบ ไปยังสภาวะสุดท้ายในช่วงเวลาที่จำกัด $t_0 < t < t_1$ แล้ว เราจะถือว่าเราควบคุมตัวแปรสภาวะเหล่านั้นได้ และถ้าตัวแปรสภาวะทุกตัวในระบบสามารถถูกควบคุมได้แล้ว เราจะเรียกระบบนั้นว่า ระบบซึ่งสามารถควบคุมได้โดยสมบูรณ์

เราสามารถตรวจสอบความสามารถในการควบคุมได้โดยสมบูรณ์ของระบบจาก Controllability Matrix

$$[\ B \mid AB \mid \ldots \mid A^{n-1}B\ ]$$

สำหรับระบบที่มีตัวแปรสภาวะ n ตัว ระบบจะสามารถควบคุมได้โดยสมบูรณ์ก็ต่อเมื่อ Rank ของ Controllability Matrix เท่ากับ n

และในทำนองเดียวกัน หากเราสามารถทำนายค่าเริ่มต้นของตัวแปรสภาวะทุกตัวของระบบได้จากการสังเกตค่าของเอาท์พุทของระบบในช่วงเวลาจำกัด $t_0 < t < t_1$ แล้ว เราจะถือว่าระบบดังกล่าว สังเกตได้โดยสมบูรณ์

สำหรับระบบที่มีตัวแปรสภาวะ n ตัว ระบบจะสังเกตได้โดยสมบูรณ์ ก็ต่อเมื่อ Rank ของ Observability Matrix

$$[\ C^T \mid A^TC^T \mid \ldots \mid (A^T)^{n-1}C^T\ ]$$

มีค่าเท่ากับ n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การออกแบบระบบควบคุม

ในการออกแบบระบบควบคุมในแนว Classical Control นั้น จะอาศัยหลักของการลองแล้วแก้ไข หรือ Trial and Error ซึ่งการออกแบบด้วยวิธีนี้มักจะมิได้ให้ระบบที่ดีที่สุด (Optimal System) แต่สำหรับการออกแบบโดยแนว Modern Control นั้น ผู้ออกแบบระบบสามารถออกแบบให้ระบบมีตำแหน่งของโพลของระบบปิดหรือมีสมการคุณลักษณะของระบบได้ตามที่ตนต้องการ หรือจะออกแบบด้วยการเลือกใช้ค่าดัชนี Performance Index ที่เหมาะสม ดังนั้นจึงทำให้การออกแบบด้วยวิธีทาง Modern Control จะทำให้ได้ระบบที่ดีที่สุดได้

นอกจากนี้ ถ้าหากมีความจำเป็นต้องใช้ค่าเริ่มต้นของระบบในการออกแบบด้วย ก็สามารถใช้วิธีทาง Modern Control ได้เช่นกัน แต่ในวิธีทาง Modern Control ก็มีข้อควรให้ความสนใจเป็นพิเศษ คือ ค่าตัวเลขต่าง ๆ ในสมการแสดงระบบจะต้องมีความถูกต้องมากพอสมควร การออกแบบจึงจะได้ผลที่ถูกต้องด้วย

ด้วยเหตุนี้ในการออกแบบระบบควบคุมป้อนกลับโดยในแนว Modern Control จึงนิยมใช้เครื่องคอมพิวเตอร์เข้ามาช่วยคำนวณค่าต่าง ๆ ในระบบ รวมถึงการทำ Simulation ระบบ เพื่อดูผลตอบสนองของเอาท์พุทหรือตัวแปรสภาวะต่าง ๆ

ในระบบควบคุมใด ๆ เมื่อมีอินพุทถูกป้อนเข้ามา ก็จะเกิดการเปลี่ยนแปลงค่าของเอาท์พุทและตัวแปรสภาวะต่าง ๆ ภายในระบบ ในโครงงานจึงมีการทำ Simulation ดูผลตอบสนองของเอาท์พุทและตัวแปรสภาวะ รวมถึงการออกแบบโปรแกรมเพื่ออำนวยความสะดวกแก่ผู้ใช้ในการวิเคราะห์และออกแบบระบบควบคุม โปรแกรมที่เขียนขึ้นจะวิเคราะห์ความควบคุมได้ และใช้ออกแบบระบบควบคุมโดยวิธีการกำหนดตำแหน่งโพลของระบบ (Pole Placement) ตลอดจนวิเคราะห์ความสามารถสังเกตได้ (observability) และหาค่าป้อนกลับให้ตัวสังเกตสเตท(state observer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 การออกแบบระบบโดยการกำหนดตำแหน่งโพล

จากระบบควบคุม ซึ่งมีตัวแปรสภาวะภายใน n ตัว

$$\dot{X}(t) = AX(t) + Bu(t)$$

เงื่อนไขที่สำคัญคือ "ระบบที่จะได้รับการออกแบบนั้นจะต้องเป็นระบบที่ควบคุมได้โดยสมบูรณ์" กล่าวคือ controllable matrix Q มี rank เท่ากับ n

และถ้าเราเลือก $u(t) = -KX(t)$

จะได้ $\dot{X}(t) = (A-BK)X(t)$

$$X(t) = e^{(A-BK)t}X(0)$$

จะพบว่า เราสามารถตรวจสอบเสถียรภาพของระบบ และหาค่าไอเกนของระบบที่ได้รับการออกแบบแล้วได้จากเมตริกซ์ (A-BK) ซึ่งนั้นก็หมายความว่า จะมีเมตริกซ์ K ที่เหมาะสมซึ่งจะสอดคล้องกับสมการ

$$\text{Det}(\, sI-A+BK\,) = (s-\mu_1)(s-\mu_2)\ldots(s-\mu_n)$$

เมื่อ $\mu_1,\mu_2,\ldots,\mu_n$ คือ ตำแหน่งหรือค่าของโพลของระบบควบคุมป้อนกลับที่เราจะออกแบบ

ดังนั้น ปัญหาในการออกแบบระบบควบคุม เมื่อเราจะกำหนดตำแหน่งโพลของระบบป้อนกลับที่จะออกแบบ ก็คือการหาเมตริกซ์ K ที่เหมาะสมนั้นเอง

2.1.1 กรณีระบบมีอินพุทเดียว

<u>วิธีที่1</u> จากตำแหน่งโพลที่ต้องการคือ $\mu_1,\mu_2,\ldots,\mu_n$ เราจะสามารถคำนวณหาค่าเมตริกซ์ K ได้จากสมการ

$$K = [\, \alpha_n-a_n \mid \alpha_{n-1}-a_{n-1} \mid \ldots \mid \alpha_2-a_2 \mid \alpha_1-a_1 \,]T^{-1}$$

โดย $\text{Det}(sI-A) = s^n + a_1s^{n-1} + \ldots + a_{n-1}s + a_n$

$$(s-\mu_1)(s-\mu_2)\ldots(s-\mu_n) = s^n + \alpha_1s^{n-1} + \ldots + \alpha_{n-1}s + \alpha_n$$

และ T คือเมตริกซ์ที่เหมาะสมซึ่งจะแปลงเมตริกซ์ A ให้อยู่ในรูปควบคุมได้จากการแปลง

$$X = TX'$$

ซึ่งจะทำให้สมการ $\dot{X}' = (T^{-1}AT)X' + T^{-1}Bu$

อยู่ในรูปควบคุมได้ (Controllable Form)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>วิธีที่2</u> วิธี Ackermann's formula

$$K = [\ 0\ 0\ \ldots\ 0\ 1][\ B\ |\ AB\ |\ldots|\ A^{n-1}B\ ]^{-1}\phi(A)$$

โดย $\phi(A) = A^n + \alpha_1 A^{n-1} + \ldots + \alpha_{n-1}A + \alpha_n I$

$$\phi(s) = (s-\mu_1)(s-\mu_2)\ldots(s-\mu_n)$$

$$= s^n + \alpha_1 s^{n-1} + \ldots + \alpha_{n-1}s + \alpha_n$$

เมื่อ $\mu_1, \mu_2, \ldots, \mu_n$ คือตำแหน่งโพลที่ต้องการ

2.1.2 กรณีหลายอินพุท

จากระบบ

$$X = AX + BU$$

ถ้าเราเลือก เมตริกซ์ P ที่เหมาะสมซึ่งจะแปลงความสัมพันธ์ $X = PX'$ หรือ $P^{-1}X = X'$ จะเขียนระบบได้ใหม่เป็น

$$X' = P^{-1}APX' + P^{-1}BU$$

จาก $U = -KX$ หรือ $U = -K'X'$

$$X' = (P^{-1}AP - P^{-1}BK')X'$$

ดังนั้น สเตทเมตริกซ์ของ X' เมื่อได้รับการป้อนกลับ $U' = -K'X'$ จะมีค่าเท่ากับ

$$A' = (P^{-1}AP - P^{-1}BK')$$

ฉนั้น สามารถหา K' ได้จากสมการนี้และ $K = K'P^{-1}$

2.2 การออกแบบตัวสังเกตสเตท

เงื่อนไขที่สำคัญคือ "ระบบที่จะได้รับการออกแบบตัวสังเกตสเตทนั้น ต้องเป็นระบบซึ่งสามารถสังเกตได้โดยสมบูรณ์"

กล่าวคือ Rank ของ Observable Matrix

$$[\ C^T\ |\ A^TC^T\ |\ \ldots\ |\ (A^T)C^T\ ]$$

มีค่าเท่ากับ n

เมื่อ n คือ จำนวนตัวแปรสภาวะของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบตัวสังเกตสเตท ในที่นี้ถ้าให้ x' เป็นเวกเตอร์ของตัวแปรสภาวะที่ถูกสังเกต (Observed State Vector) เวกเตอร์นี้จะถูกใช้ในการป้อนกลับผ่านเมตริกซ์ K ซึ่งได้มาจากการออกแบบด้วยวิธีการวางตำแหน่งโพลที่กล่าวมาก่อนหน้านี้

สำหรับระบบควบคุม

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

จะได้สมการแสดงตัวแปรสภาวะที่ถูกสังเกต คือ

$$\dot{x}' = Ax' + Bu + K_e(y - Cx')$$

ซึ่งเป็นสมการแสดงตัวสังเกตสเตท มี y และ u เป็นอินพุท และ x' เป็นเอาท์พุท

ตัวสังเกตสเตทที่เรากำลังสนใจอยู่นี้ จะเป็นแบบเต็มอันดับ (Full Order State Observer) กล่าวคือ อันดับของตัวสังเกตสเตทจะเท่ากับอันดับของระบบ

จากสมการที่กล่าวมาข้างต้น เราจะได้ว่า

$$\dot{x} - \dot{x}' = Ax - Ax' - K_e(Cx - Cx')$$

$$\dot{x} - \dot{x}' = (A - K_eC)(x - x')$$

กำหนดให้ผลต่างระหว่าง x และ x' เป็นเวกเตอร์ของค่าคลาดเคลื่อน(Error Vector) กล่าวคือ

$$e = x - x'$$

ดังนั้น จะได้ว่า

$$\dot{e} = (A - K_eC)e$$

จะเห็นได้ว่าพฤติกรรมของ Error Vector สามารถดูได้จากค่าไอเกนวาลูของเมตริกซ์ $A-K_eC$ และเรานิยมเลือกให้ค่าไอเกนวาลูของ $A - K_eC$ มีค่าที่เหมาะสมซึ่งจะทำให้ Error Vector ลู่เข้าสู่ค่าเริ่มต้นด้วยอัตราเร็วที่พอเหมาะ

ดังนั้น จากที่กล่าวมา จะสรุปได้ว่า ปัญหาของการออกแบบตัวสังเกตสเตท ก็คือการหาค่าเมตริกซ์ K_e ที่เหมาะสมที่จะทำให้พฤติกรรมของ Error Vector เป็นไปตามที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

2.2.1 การหาค่า K_e

กรณีที่จะกล่าวถึงต่อไปนี้ เป็นกรณีที่ระบบมีอินพุทเดียวและเอาท์พุทเดียว (Single Input Single Output System)

<u>วิธีที่1</u>จากตำแหน่งโพลของตัวสังเกตสเตทที่ต้องการคือ $\mu_1,\mu_2,\mu_3,\ldots,\mu_n$ สามารถหาค่าเมตริกซ์ K_e ได้จาก

$$K_e = Q\,[\,\alpha_n - a_n \mid \alpha_{n-1} - a_{n-1} \mid \ldots \mid \alpha_1 - a_1\,]^{-1}$$

โดย $\mathrm{Det}(sI - A) = s^n + a_1 s^{n-1} + \ldots + a_{n-1}s + a_n$

$$(s-\mu_1)(s-\mu_2)\ldots(s-\mu_n) = s^n + \alpha_1 s^{n-1} + \ldots + \alpha_{n-1}s + \alpha_n$$

Q คือ เมตริกซ์ที่เหมาะสมซึ่งจะแปลงเมตริกซ์ A ให้อยู่ในรูปควบคุมได้โดยการแปลง

$$x = x'$$

ซึ่งจะทำให้สมการ $x' = (T^{-1}AT)x' + T^{-1}Bu$

อยู่ในรูปควบคุมได้ (Controllable Form)

<u>วิธีที่ 2</u> วิธี Ackermann's formula

$$K = \phi(A)\,[\,C \mid CA \mid \ldots \mid CA^{n-1}\,]\,[\,0\;0\;\ldots\;1\,]^{-1}$$

โดย $\phi(A) = A^n + \alpha_1 A^{n-1} + \ldots + \alpha_{n-1}A + \alpha_n I$

$$\phi(s) = (s-\mu_1)(s-\mu_2)\ldots(s-\mu_n)$$

$$= s^n + \alpha_1 s^{n-1} + \ldots + \alpha_{n-1}s + \alpha_n$$

เมื่อ $\mu_1,\mu_2,\ldots,\mu_n$ คือ ตำแหน่งโพลที่ต้องการ

2.2.2 **คุณสมบัติ Duality**

ดังที่กล่าวมาแล้วว่า ปัญหาของการออกแบบตัวสังเกตสเตท ก็คือการหาค่า K_e ที่จะทำให้ Error Vector e โดยที่

$$\dot{e} = (A - K_e C)e$$

มีความเสถียรอย่างแอซิมโทต (Asymptotically Stable) และผลตอบสนองที่เร็วพอเหมาะ โดยเราจะต้องกำหนดตำแหน่งโพลที่ต้องการให้กับเมตริกซ์ $(A - K_e C)$ แล้วจึงคำนวณหาค่า K_e

วิธีการหนึ่งที่จะใช้หาค่า K_e ได้ ได้แก่การใช้คุณสมบัติ Duality เข้ามาช่วยพิจารณาซึ่งมีรายละเอียดดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032631

จากระบบ

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

จะสามารถทำการออกแบบ Full Order State Observer ของระบบได้โดยแก้ปัญหาการวางตำแหน่งโพล (Pole Placement) ของระบบคู่ Dual ของระบบเดิมคือ

$$\dot{z} = A^T z + C^T v$$

$$n = B^T z$$

และสมมติให้สัญญาณควบคุม คือ

$$v = -Kz$$

ถ้าระบบคู่ Dual ของระบบเดิมดังกล่าวสามารถควบคุมได้โดยสมบูรณ์ ดังนั้นสเตทเมตริกซ์ป้อนกลับ (State Feedback Gain Matrix) K สามารถหาได้จากเมตริกซ์ A^T-C^TK ซึ่งมีโพลอยู่ตรงตำแหน่งที่เรากำหนดไว้ กล่าวคือ ถ้าให้ $\mu_1,\mu_2,\ldots,\mu_n$ เป็นโพลที่เราจะกำหนดให้กับตัวสังเกตสเตทของระบบเดิม และให้โพลเหล่านี้ เป็นโพลซึ่งจะกำหนดให้กับสเตทเมตริกซ์ป้อนกลับ ของระบบคู่ Dual ของระบบเองด้วย

เราจะได้ว่า

$$\left| sI - (A^T-C^TK) \right| = (s-\mu_1)(s-\mu_2)\ldots(s-\mu_n)$$

จะสังเกตพบว่าไอเกนวาลูของ A^T-C^TK และของ $A-K^TC$ เป็นชุดเดียวกัน และจะได้ว่า

$$\left| sI - (A^T-C^TK) \right| = \left| sI - (A - K^TC) \right|$$

เปรียบเทียบโพลิโนเมียลคุณลักษณะ $\left|sI-(A-K^TC)\right|$ กับโพลิโนเมียลคุณลักษณะ $\left|sI-(A-K_eC)\right|$ ของตัวสังเกตสเตทของระบบเดิม จะพบว่า

$$K_e = K^T$$

จึงสรุปได้ว่า เราสามารถหาค่า K_e ของตัวสังเกตสเตทของระบบควบคุมใด ๆ ได้จากการหาค่าสเตทเมตริกซ์ป้อนกลับ (State Feedback Gain Matrix) K ของระบบคู่ Dual ของระบบนั้น โดยที่

$$K_e = K^T$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระบบควบคุมสัญญาณไม่ต่อเนื่อง

สำหรับระบบควบคุมที่เป็นสัญญาณไม่ต่อเนื่อง (Discrete Time Control System)

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k) + Du(k)$$

เราสามารถแสดงความสัมพันธ์ระหว่างการวิเคราะห์และออกแบบระบบควบคุมในโดเมนความถี่และโดเมนเวลา โดยใช้การแปลง z (z Transformation)

$$zX(z) - zX(0) = AX(z) + BU(z)$$

$$Y(z) = CX(z) + DU(z)$$

$$X(z) = (zI - A)^{-1}BU(z)$$

$$Y(z) = [C(zI - A)^{-1}B + D]U(z)$$

จะได้ $G(z) = Y(z)/U(z) = C(zI - A)^{-1}B + D$

G(z) เป็นทรานสเฟอร์ฟังชั่นของระบบปิดแบบสัญญาณไม่ต่อเนื่อง (Discrete Time Closed Loop Transfer Function)

ในทำนองเดียวกัน การออกแบบระบบสัญญาณไม่ต่อเนื่องก็คือการหาค่าเมตริกซ์K ป้อนกลับให้ระบบเช่นกัน กล่าวคือ กำหนดให้

$$U(k) = -Kx(k)$$

ปัญหาของการออกแบบก็คือ การหาค่า K ที่เหมาะสมเมื่อกำหนดตำแหน่งโพลของระบบมาให้ส่วนการออกแบบตัวสังเกตสเตทของระบบสัญญาณไม่ต่อเนื่อง ก็คือการหาค่าคงที่ K_e ให้กับตัวสังเกตสเตทเมื่อกำหนดโพลให้ตัวสังเกตสเตทนั่นเอง

วิธีการคำนวณหาค่าเมตริกซ์ K และ K_e ในกรณีระบบเป็นแบบสัญญาณไม่ต่อเนื่องนั้น ก็สามารถทำได้แบบเดียวกันกับกรณีที่ระบบเป็นแบบสัญญาณต่อเนื่องทุกประการ (ต่างกันเพียงว่า นกรณีสัญญาณต่อเนื่อง การมองตำแหน่งโพล จะดูบนระนาบเชิงซ้อน s ของการแปลงลาปลาซ ส่วนกรณีสัญญาณไม่ต่อเนื่อง จะมองตำแหน่งโพลบนระนาบ z ของการแปลง z แต่วิธีการคำนวณค่า K และ K_e ยังคงเหมือนกันทุกประการ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การ Simulation ระบบต่อเนื่องและไม่ต่อเนื่อง

SIMULATION คำนวณค่าเอาท์พุทและตัวแปรสเตทเพื่อดูผลการตอบสนองมีดังนี้

4.1 กรณีสัญญาณไม่ต่อเนื่อง(Discrete):

"Computational Method"

$$\begin{bmatrix} x_1(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} A_{n\times n} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} B_{n\times r} \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_r(k) \end{bmatrix}$$

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k) + Du(k)$$

จากการกำหนดค่า u(k) ที่เวลาใด ๆ และค่าเริ่มต้น x(0) เราสามารถคำนวณหาค่าตัวแปรสเตท x(k) และเอาท์พุท y(k) ที่เวลาใด ๆ โดยเริ่มต้นคำนวณค่าจาก k = 0,1,2,... ไปเรื่อย ๆ

4.2 กรณีสัญญาณต่อเนื่อง(Continuous):

"Runge Kutta of Fourth Order"

คำนวณคำตอบของสมการอนุพันธ์ เมื่อทราบค่าเริ่มต้นต่าง ๆ

$$\dot{x} = f(x,t) \quad ; \quad x(t_0) = x_0$$

โดย $t_1 = t_0 + h$, $t_2 = t_0 + 2h$, ... , $t_N = t_0 + Nh$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำการหาค่า x(t) ในช่วง [t_0 , t_N] โดยเริ่มจาก

INPUT: ค่าเริ่มต้น t_0 , x_0 , ช่วง step ที่จะคำนวณ =h , จำนวน step = N

output: ค่า x_{n+1} ซึ่งเป็นค่าประมาณของคำตอบ $x(t_{n+1})$ ที่เวลา $t_{n+1}=t_0+(n+1)h$

โดย n = 0 , 1 , 2 ,..., N-1

FOR n = 0 , 1 , 2 ,..., N-1 do:

$k_1 = h*f(t_n, x_n)$

$k_2 = h*f(t_n + h/2, x_n + k_1/2)$

$k_3 = h*f(t_n + h/2, x_n + k_2/2)$

$k_4 = h*f(t_n + h, x_n + k_3)$

$t_{n+1} = t_n + h$

$x_{n+1} = x_n + (k_1 + 2k_2 + 2k_3 + k_4)/6$

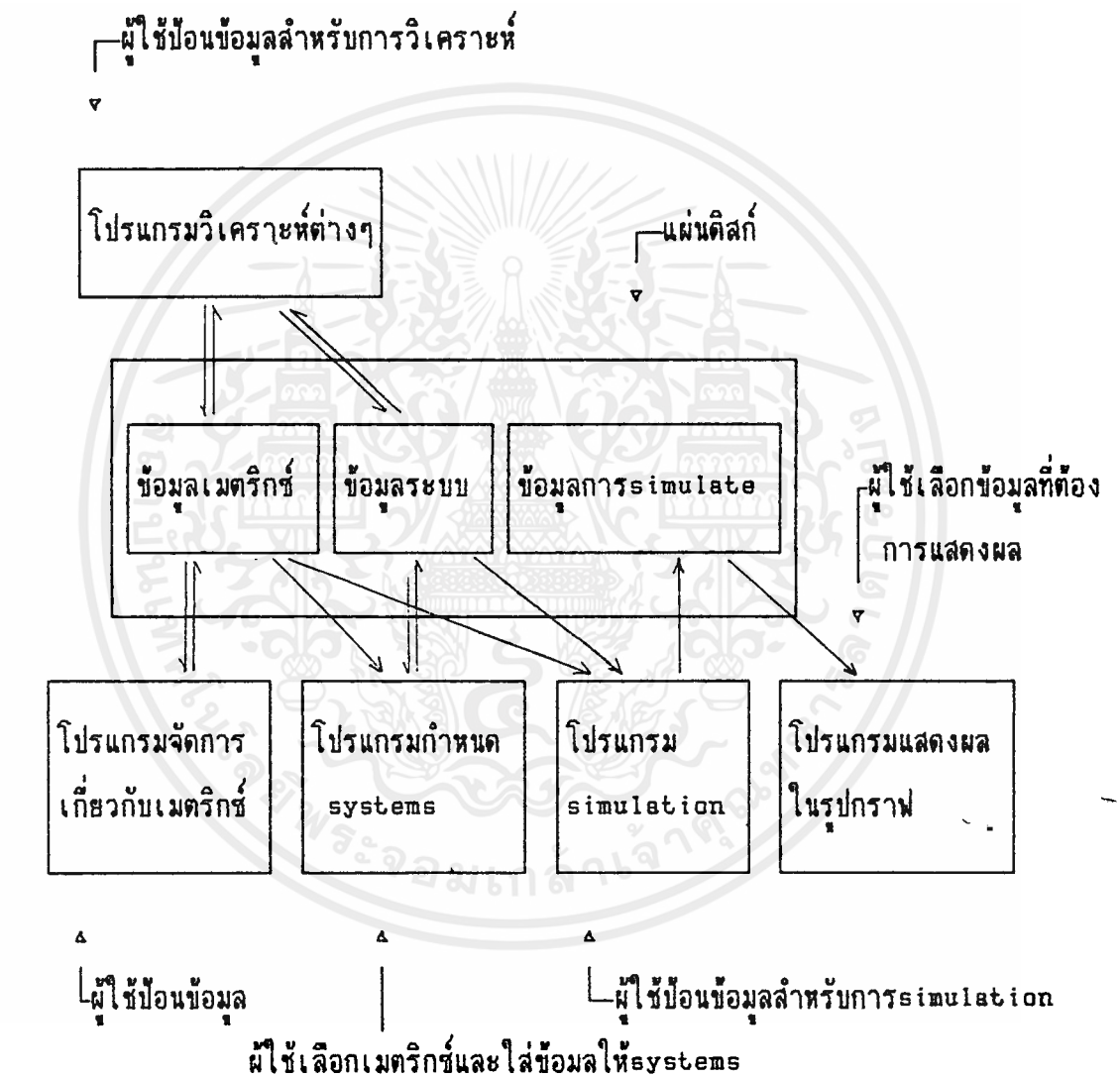
END OUTPUT t_{n+1}, x_{n+1}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบโปรแกรม

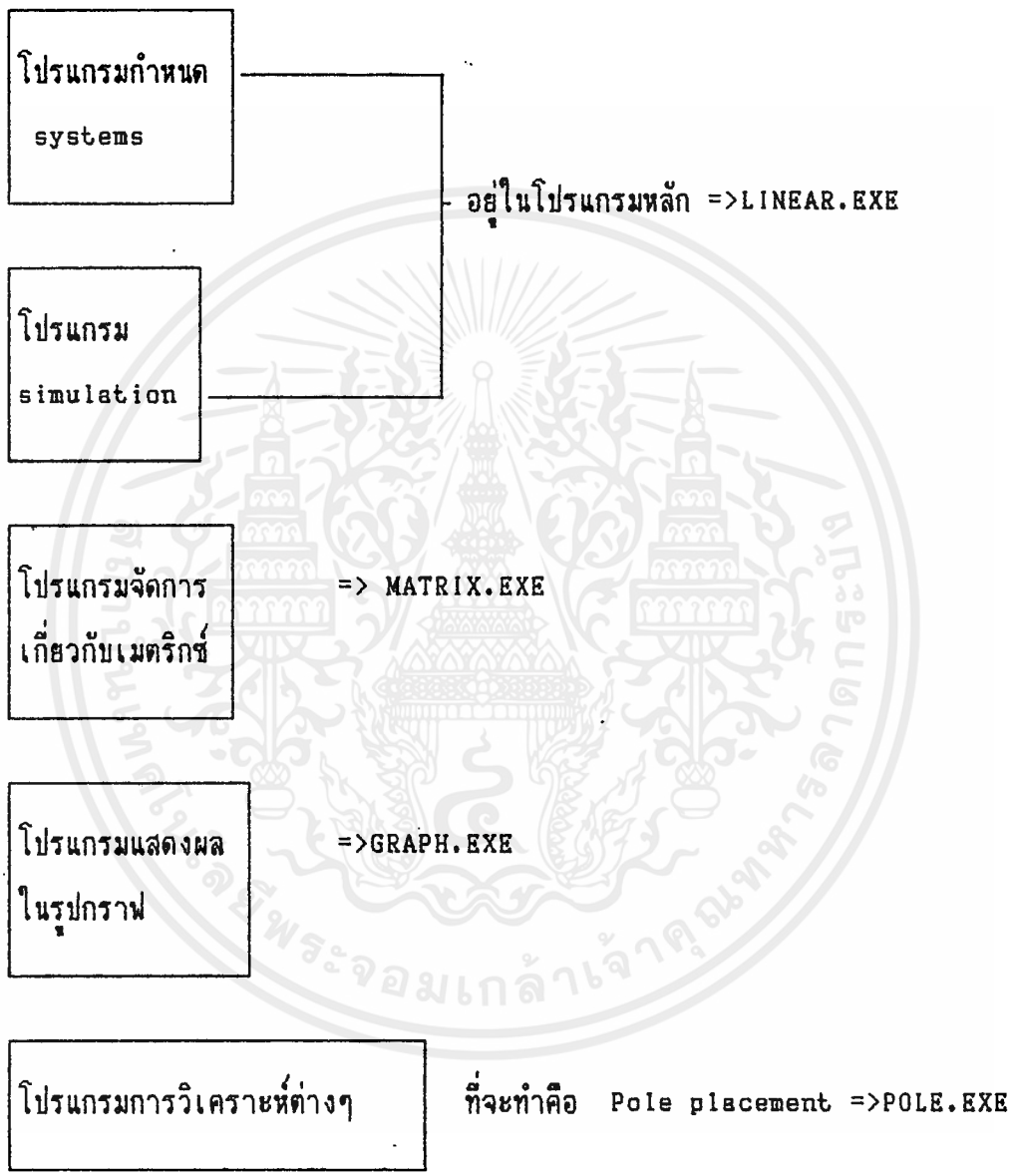
รูปแบบของข้อมูลและโปรแกรมแต่ละส่วนเป็นดังนี้



รูปที่ 1. โครงสร้างรวมของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากโปรแกรมบางตัวมีขนาดไม่ใหญ่และไม่ค่อยมีประโยชน์ในการ RUN เดี่ยวๆจึงจัดรวมอยู่ในโปรแกรมหลัก ได้โปรแกรมที่ออกแบบ มีดังต่อไปนี้



และที่ทำเพิ่มเติมคือ Observer แบบ Full Order =>OBSERVER.EXE

รูปที่ 2. แสดงชื่อโปรแกรมต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

โปรแกรม LINEAR.EXE

6.1 จุดประสงค์และหน้าที่

-เป็นโปรแกรมหลักที่จะคอยจัดการเรียกโปรแกรมอื่นขึ้นมาเพื่อให้ผู้ใช้ใช้งานได้สะดวกขึ้น

-สามารถเพิ่มรูปแบบการวิเคราะห์ได้โดยไม่ต้องแก้โปรแกรมส่วนอื่น (เขียนโปรแกรมการวิเคราะห์แล้วนำมาเชื่อมต่อกับโปรแกรมหลักได้เลย)

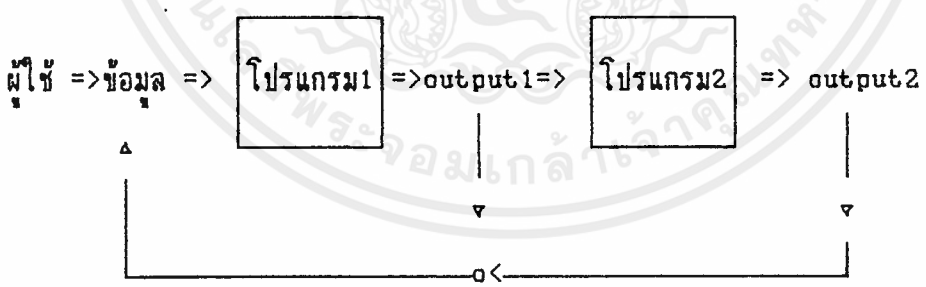
-มีส่วนใช้งานย่อยเพื่ออำนวยความสะดวก เช่น การแสดงข้อมูลออกทางเครื่องพิมพ์ การตั้งรากเพื่อไปหา โปรแกรมหรือข้อมูล

นอกจากนี้ยังรวมโปรแกรมกำหนด System และโปรแกรม Simulation ไว้เป็นโปรแกรมย่อย ซึ่งจะแยกกล่าวในหัวข้อต่อไป

6.2 แนวคิด

เมื่อผู้ใช้ป้อนข้อมูลแล้ว RUN โปรแกรม1ได้ output1 แล้วสามารถนำ output1 มาแก้ไขหรือจะส่ง output1 ไป RUN โปรแกรม2 ที่ output2 ก็เช่นเดียวกัน

รูปที่ 3. แสดงแนวคิดการเรียกโปรแกรมต่างๆ

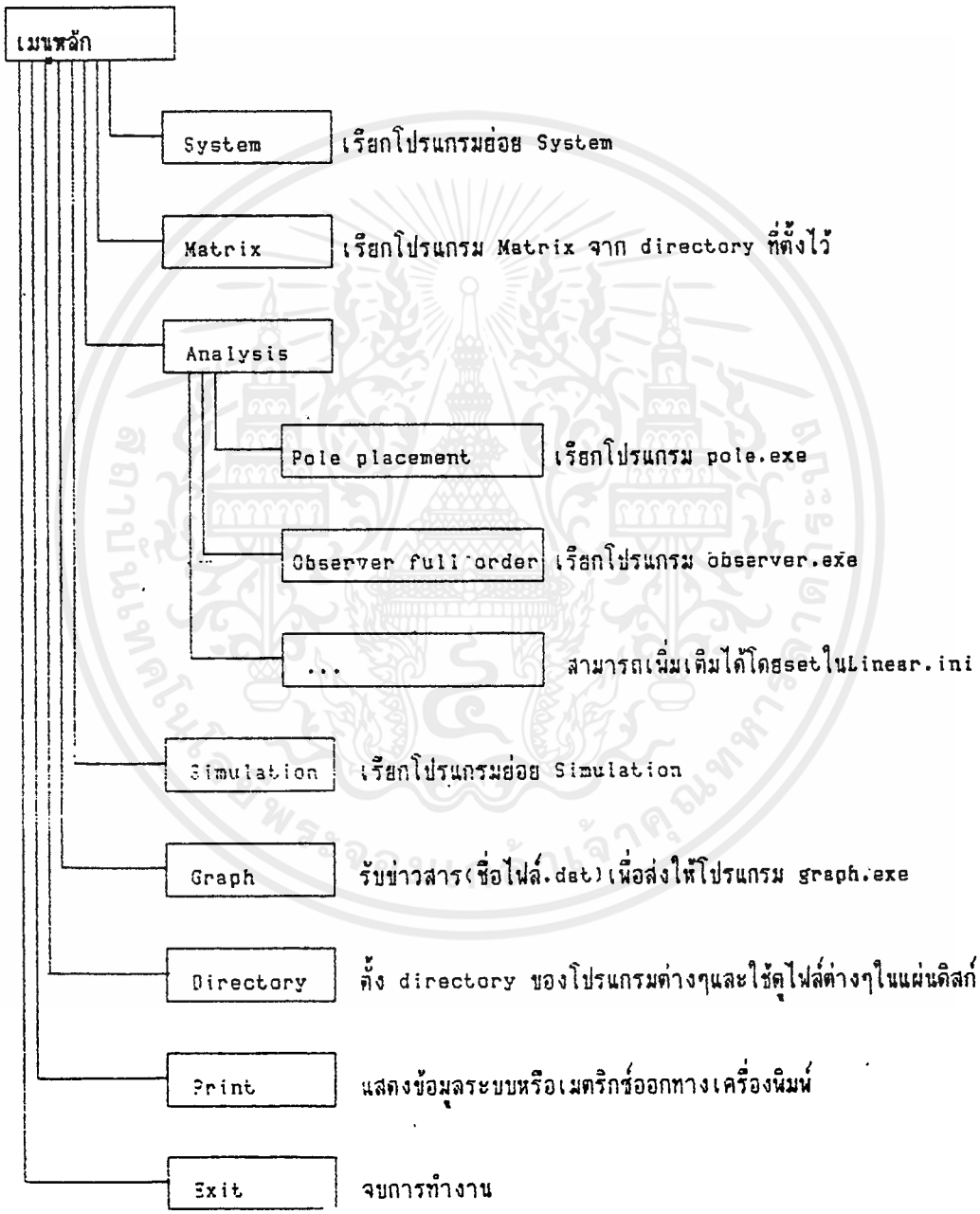


จึงต้องแบ่งโปรแกรมออกเป็นส่วนๆ และเพื่อให้สามารถใช้งานเฉพาะช่วงการวิเคราะห์จึงให้แต่ละส่วนเป็นโปรแกรมสำเร็จรูป ซึ่งจะสามารถเรียกโดยโปรแกรมหลักตัวเดียวกัน หรือสามารถเรียกมาใช้งานเดี่ยวๆได้ ทั้งนี้สามารถเพิ่มโปรแกรมการวิเคราะห์ได้ด้วยการเขียนโปรแกรมสำเร็จรูปเพิ่มขึ้นมาโดยไม่ต้องแก้ไขโปรแกรมเก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 รายละเอียด

รายละเอียดส่วนใหญ่เป็นการจัดการเรียกโปรแกรมอื่นขึ้นมาใช้งานจึงมีแต่การติดต่อกับผู้ใช้เพื่อเก็บข่าวสารส่งไปให้โปรแกรมที่เรียกขึ้นมา เช่น รับคีย์เลือกโปรแกรม รับชื่อไฟล์ข้อมูลเพื่อส่งให้โปรแกรมแสดงผลกราฟ เป็นต้น ดังแผนภาพ

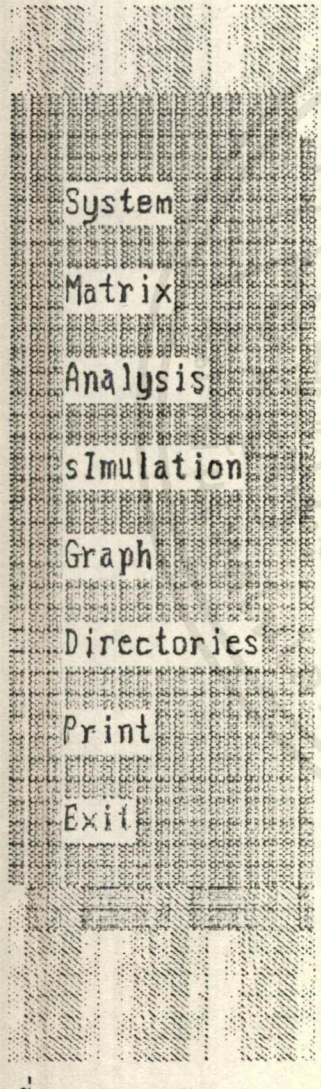


รูปที่ 4. แผนผังแสดงรายละเอียดของโปรแกรม LINEAR.EXE

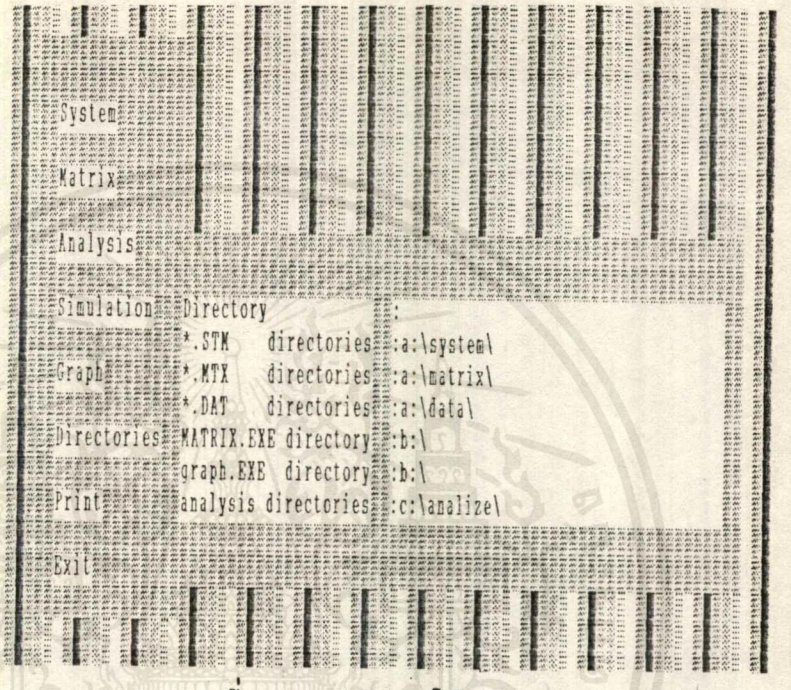
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 ผลการออกแบบ

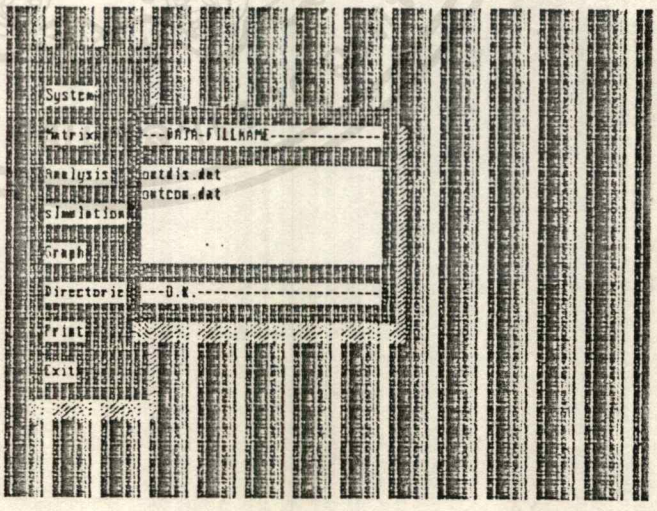
มีเมนูการใช้งานดังรูป



รูปที่ 5. เมนูหลักของ LINEAR.EXE



รูปที่ 6. แสดงการใช้เมนู Directory



รูปที่ 7. แสดงการรับชื่อไฟล์ก่อนส่งให้ GRAPH.EXE ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

โปรแกรมกำหนด System

7.1 จุดประสงค์และหน้าที่

-สร้างและรับค่าส่วนต่างๆของsystem

-สร้างขึ้นมาได้ทีละหลาย systems แล้วเลือกมาประมวลผลทีละหนึ่งsystem

7.2 แนวคิด

จากวัตถุประสงค์เราจะต้องสร้างฟังก์ชันหรือโปรแกรมย่อยขึ้นมาดังนี้

-ฟังก์ชันรับค่าข้อมูลระบบ

-ยกเลิกหรือลบข้อมูลระบบ

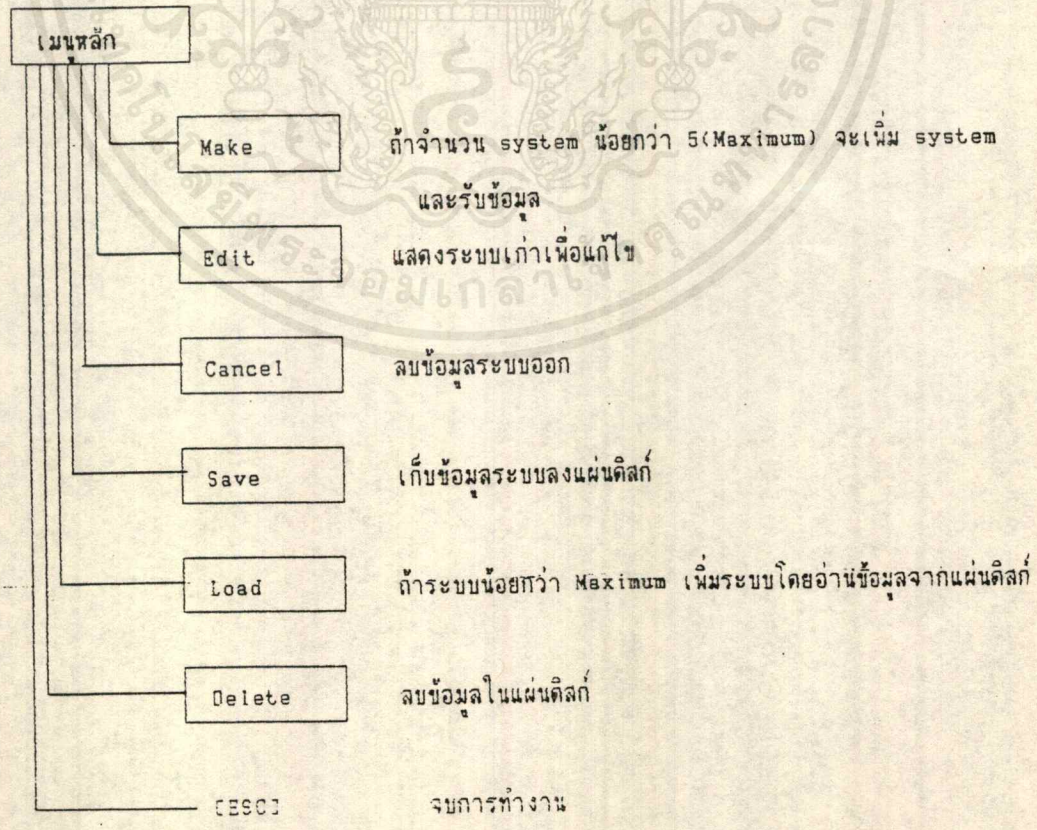
-เลือกระบบมาแก้ไขข้อมูล

-เก็บข้อมูลเมตริกซ์ลงแผ่นดิสก์(save)

-อ่านข้อมูลจากแผ่นดิสก์ขึ้นมาประมวลผล(load)

-ลบข้อมูลในแผ่นดิสก์

7.3 รายละเอียด

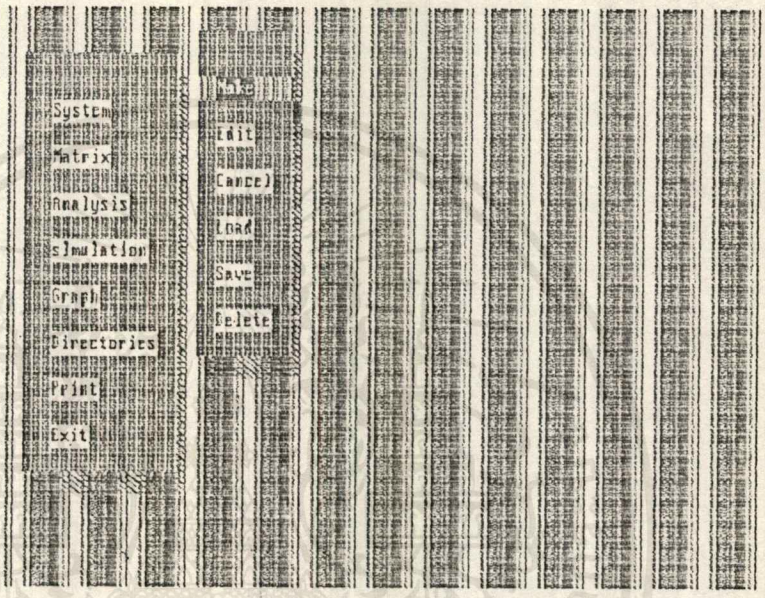


รูปที่ 8. แผนผังแสดงรายละเอียดของโปรแกรม SYSTEM

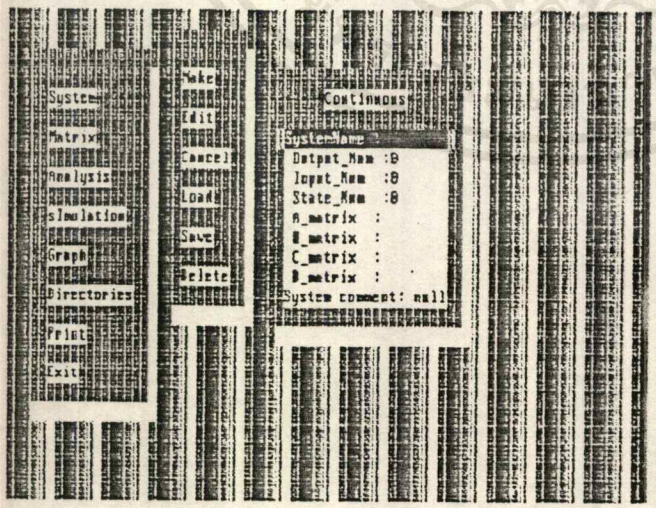
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 ผลการออกแบบ

เนื่องจากโปรแกรมจะรวมอยู่ในโปรแกรมหลัก จึงเป็นส่วนหนึ่งของโปรแกรมหลักโดยแยกมาจากเมนูอันหนึ่งในโปรแกรมหลัก ซึ่งมีเมนูย่อยตามการใช้งานดังต่อไปนี้



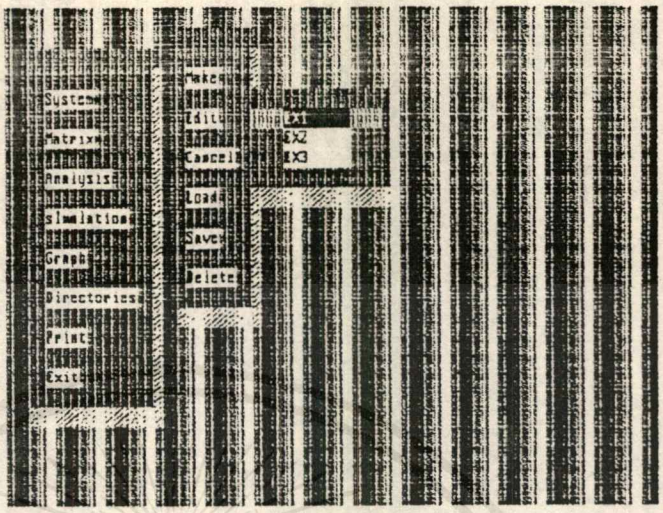
รูปที่ 9. แสดงเมนูหลักของ SYSTEM



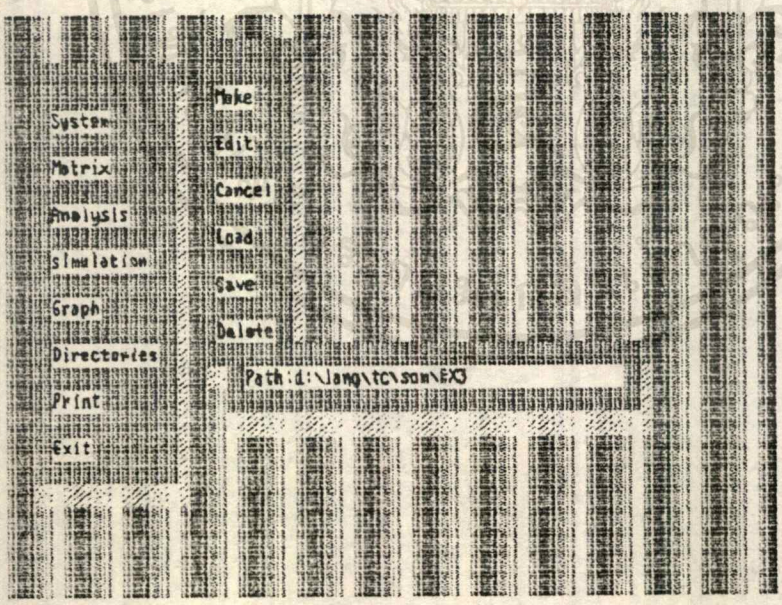
รูปที่ 10. แสดงการใช้เมนู Make และการรับค่าข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที 11. แสดงการใช้เมนู Edit



รูปที่ 12. แสดงการใช้เมนู Delete



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

โปรแกรม Simulation

8.1 จุดประสงค์และหน้าที่

-ใช้ Simulation ทั้ง ระบบ Discrete และContinue

-สามารถใส่ข้อมูล Input เป็นสมการของเวลาได้

-คำนวณหรือ Simulation ในช่วงเวลาที่ผู้ใช้ตั้งได้

-จะเก็บข้อมูล การ Simulation เป็นไฟล์

-สามารถเลือกเก็บเฉพาะข้อมูลที่สนใจได้

8.2 แนวคิด

รับข่าวสารที่เก็บรายละเอียดการ Simulation จากผู้ใช้ซึ่งจะรับค่าจนผู้ใช้ SET ค่าตามต้องการเสร็จ จึงนำข่าวสารนั้นไปตีความ เมื่อไม่มีข้อผิดพลาดจึง Simulate ข้อมูล ตามชนิดของระบบ ซึ่งได้อธิบาย Algorithm การ Simulation ดังที่กล่าวมาแล้วในหัวข้อการ Simulation ระบบต่อเนื่องและไม่ต่อเนื่อง

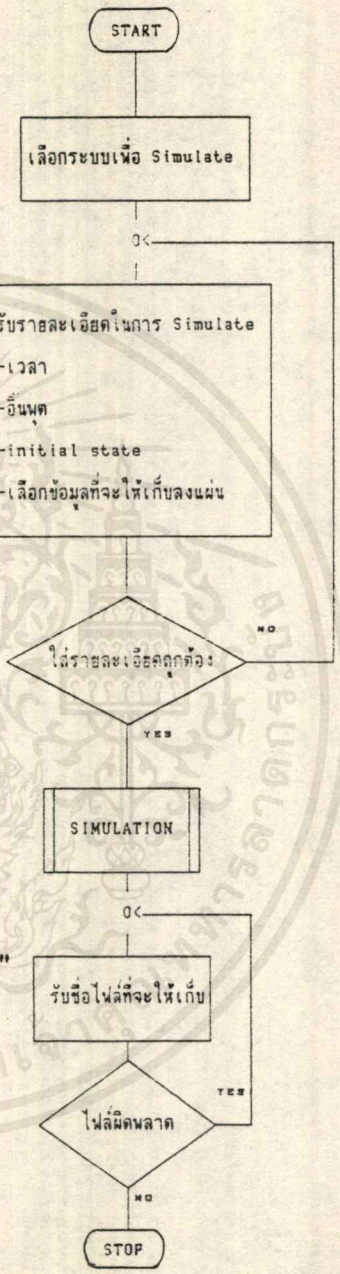
8.3 รายละเอียด

ข่าวสารที่รับจะมี เวลาที่ให้ Simulate, สเตทเริ่มต้น, ข่าวสารที่บอกว่าข้อมูลใดจะเก็บหรือไม่เก็บลงไฟล์, Input ซึ่งเป็นสมการของเวลา (ฟังก์ชันที่ใช้ได้มี +,-,*,/,^,sin( ),cos( ), tan( ), exp( ),log( ),abs( ),unit( ),ramp( ),impulse( ))

จะสามารถแสดงเป็นแผนภาพง่ายๆได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 13. flow chart แสดงการ simulation

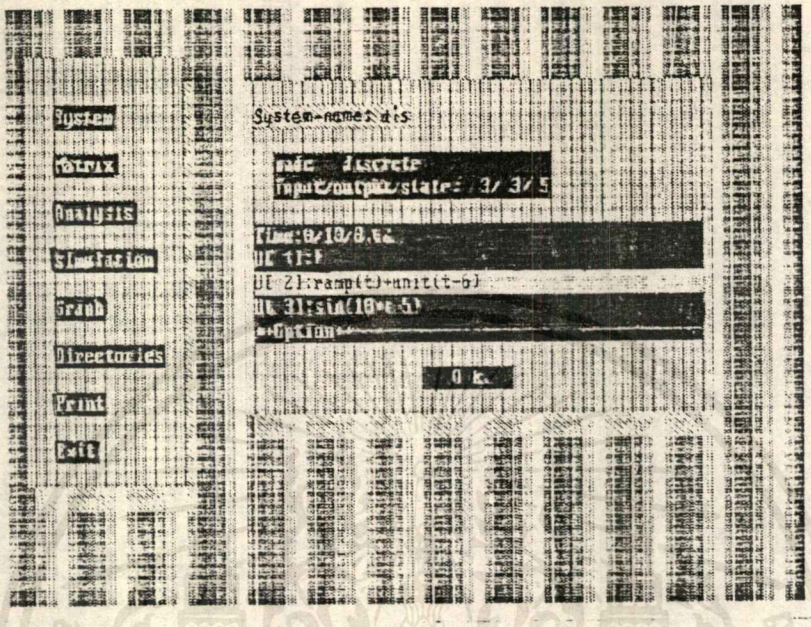


การ Simulate ของ continue ใช้
"Runge Kutta of Fourth Order"
ของ discrete จะใช้
"Computational Method"

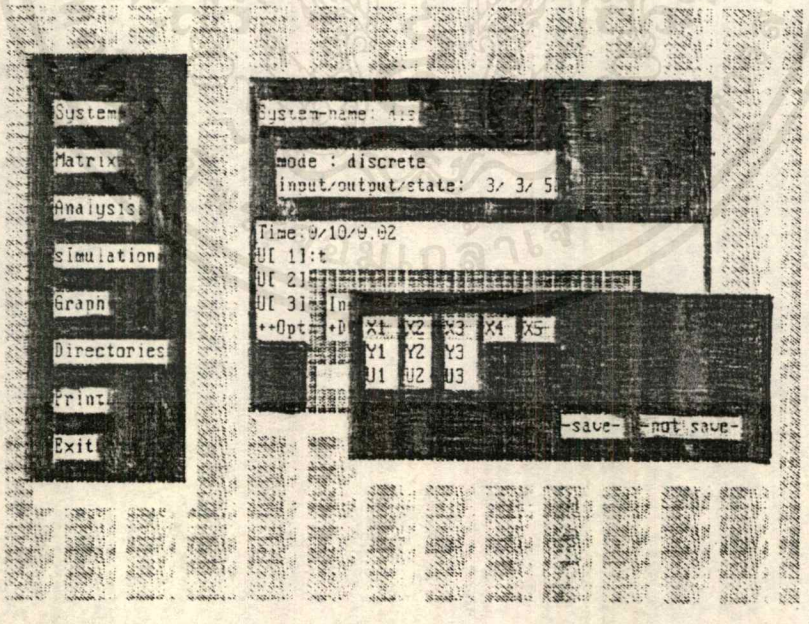
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.4 ผลการออกแบบ

มีดังนี้



รูปที่ 14. แสดงการใส่อินพุต



รูปที่ 15. แสดงการเลือกข้อมูลที่จะให้ save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

โปรแกรม MATRIX.EXE

9.1 จุดประสงค์และหน้าที่

-สร้างฐานข้อมูลของเมตริกซ์ โดยประหยัดหน่วยความจำ

-สามารถใส่ข้อมูลหรือประมวลผลส่วนไหนก่อนหรือหลังก็ได้

-คำนวณพีชคณิตของเมตริกซ์ได้

-แสดงผลได้ชัดเจน ใช้งานง่าย

-ประมวลผลได้หลายเมตริกซ์

9.2 แนวคิด

จากวัตถุประสงค์เราจะต้องสร้างฟังก์ชันหรือโปรแกรมย่อยขึ้นมาดังนี้

-ฟังก์ชันสร้างเมตริกซ์ขึ้นมาโดยจองหน่วยความจำขึ้นมาตามขนาดของเมตริกซ์ที่สั่ง(demand assigned)

-ยกเลิกเมตริกซ์ออกจากหน่วยความจำ

-เลือกเมตริกซ์มาประมวลผล(ใส่ข้อมูล,แก้ไข,คำนวณ)

-เก็บข้อมูลเมตริกซ์ลงแผ่นดิสก์(save)

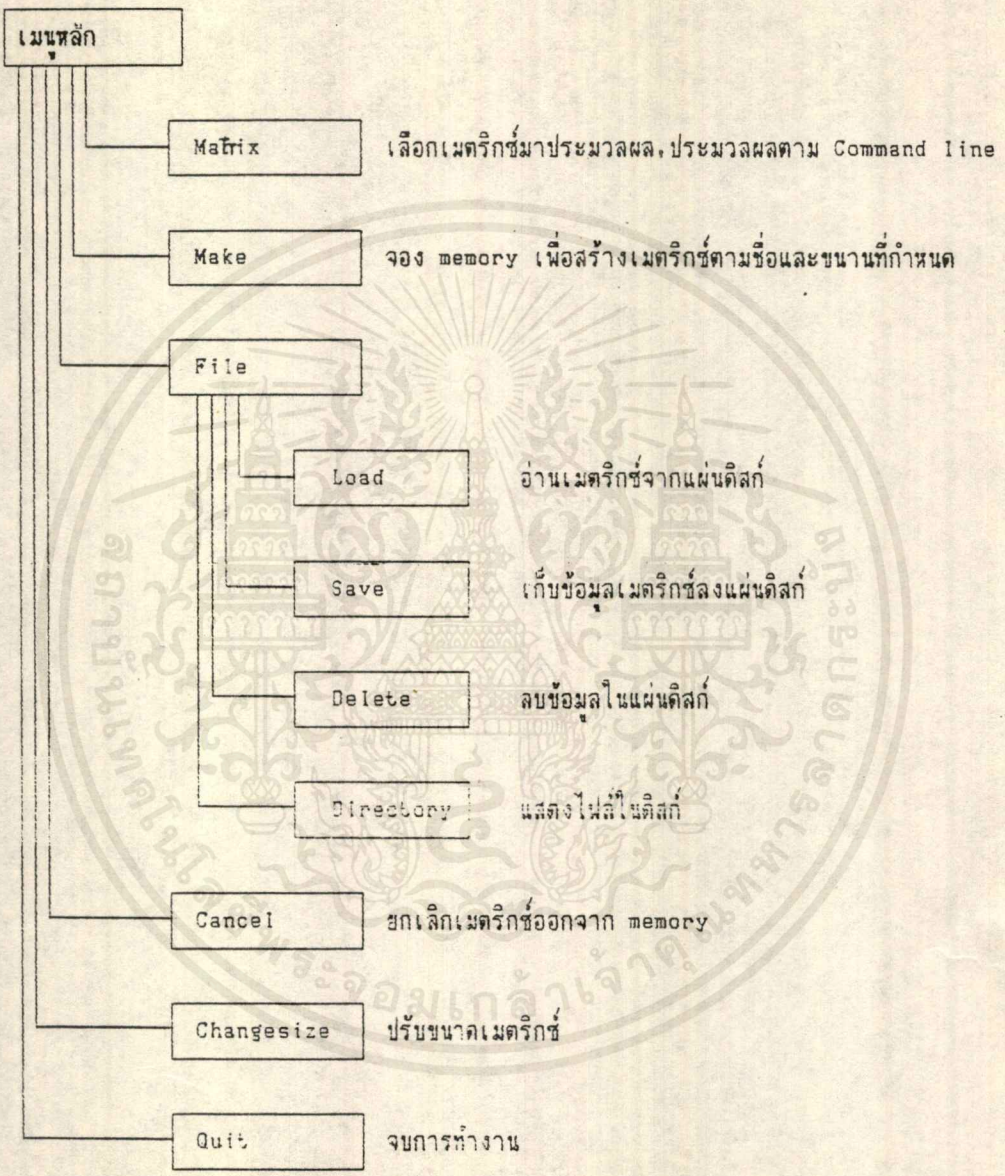
-อ่านข้อมูลจากแผ่นดิสก์ขึ้นมาประมวลผล(load)

-ลบข้อมูลในแผ่นดิสก์

นอกจากนี้ยังมีฟังก์ชันเสริมอื่นๆอีกเช่น เปลี่ยนขนาดของเมตริกซ์(จะลบแถวลบหลักที่เกินออก จะเติมแถว-หลักที่เกินด้วยศูนย์),directoryเพื่อตรวจสอบว่ามี file อะไรบ้าง(เมตริกซ์อะไรบ้าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.3 รายละเอียด

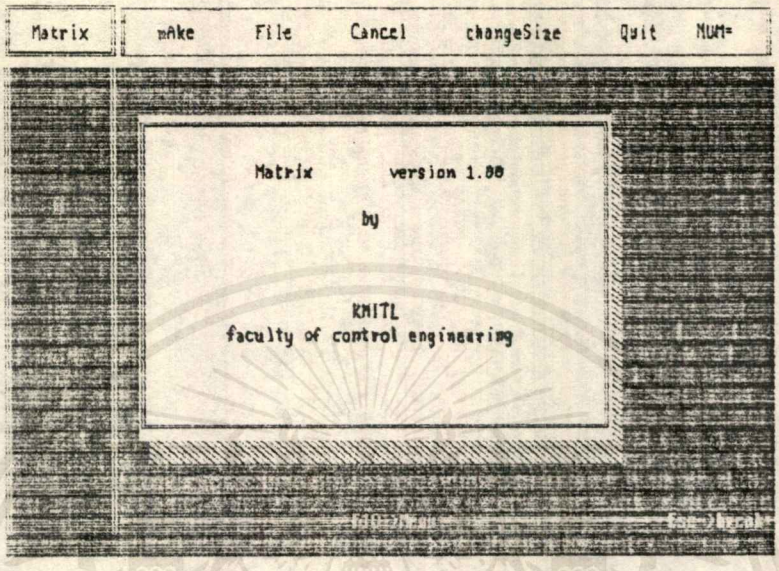


รูปที่ 16. แผนผังแสดงรายละเอียดของโปรแกรม MATRIX.EXE

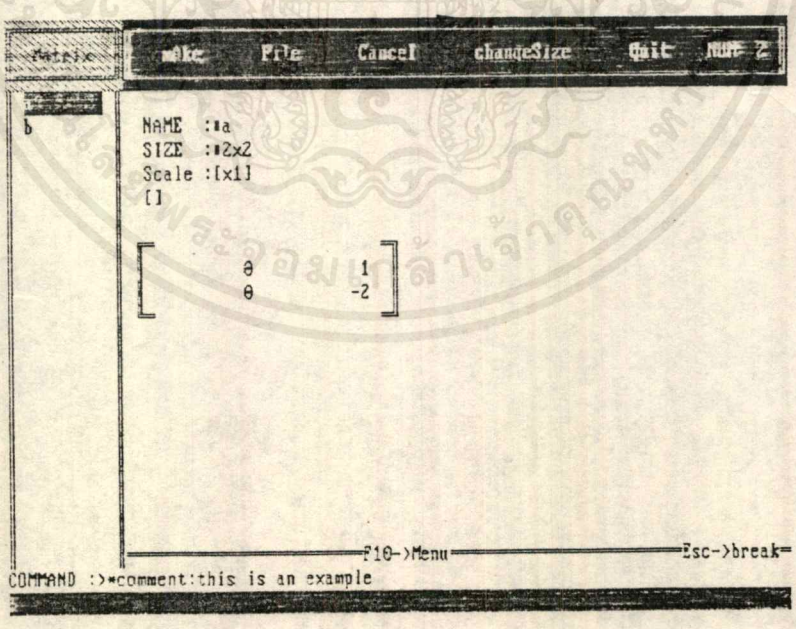
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.4 ผลการออกแบบ

มีฟังก์ชันการใช้งานเป็นเมนูและcommand line ดังนี้

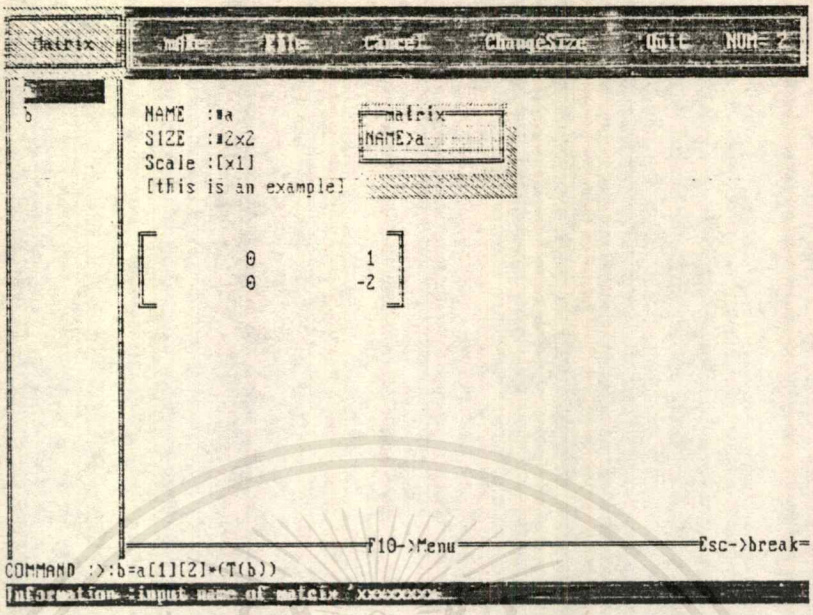


รูปที่ 17. แสดงเมนูหลัก ของโปรแกรม MATRIX.EXE

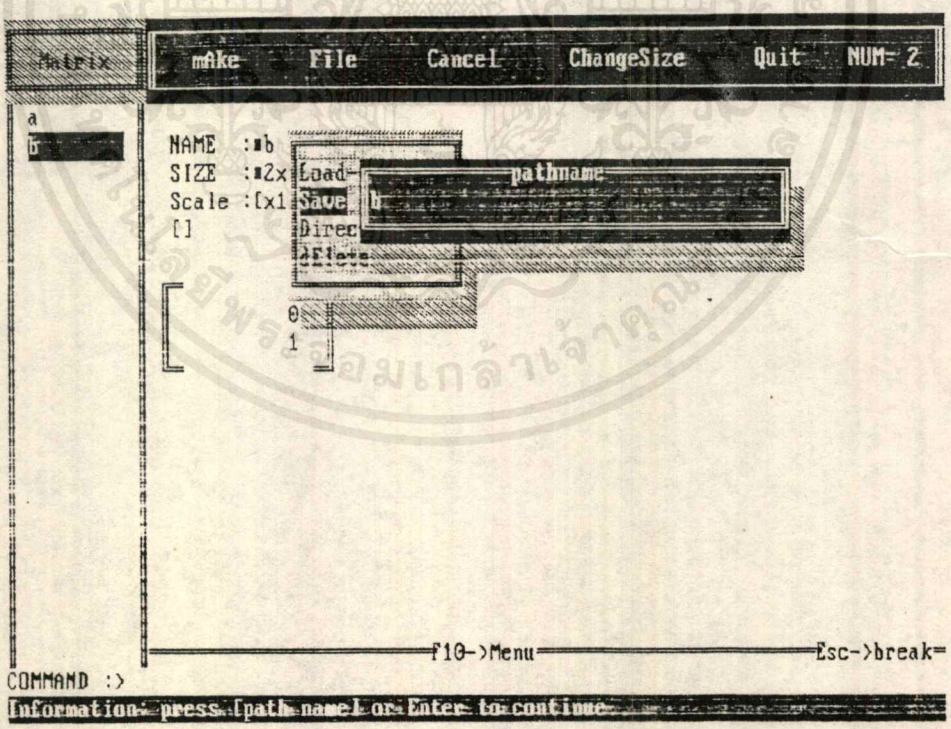


รูปที่ 18. แสดงการแสดงข้อมูลเมตริกซ์ และ การใส่ comment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

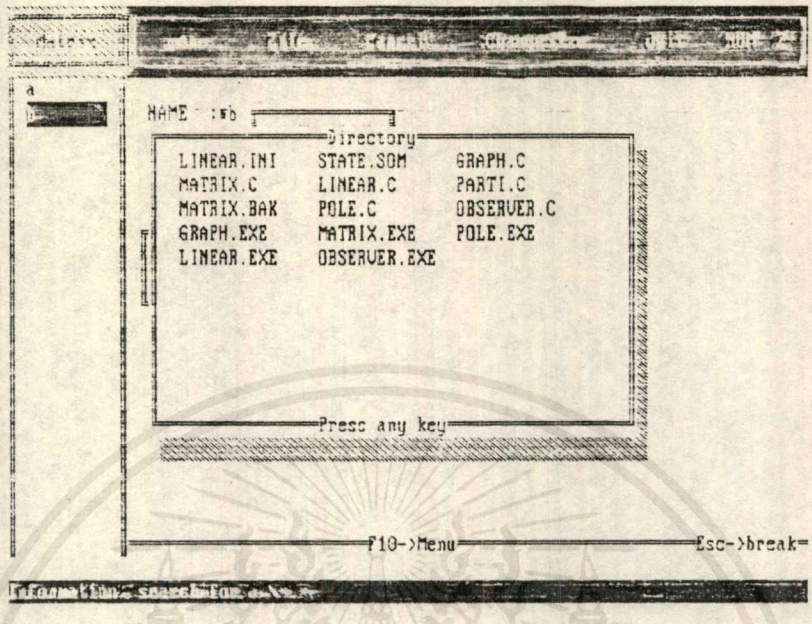


รูปที่ 19. แสดงการใส่สมการพีชคณิต และ การใช้งานเมนู Cancel

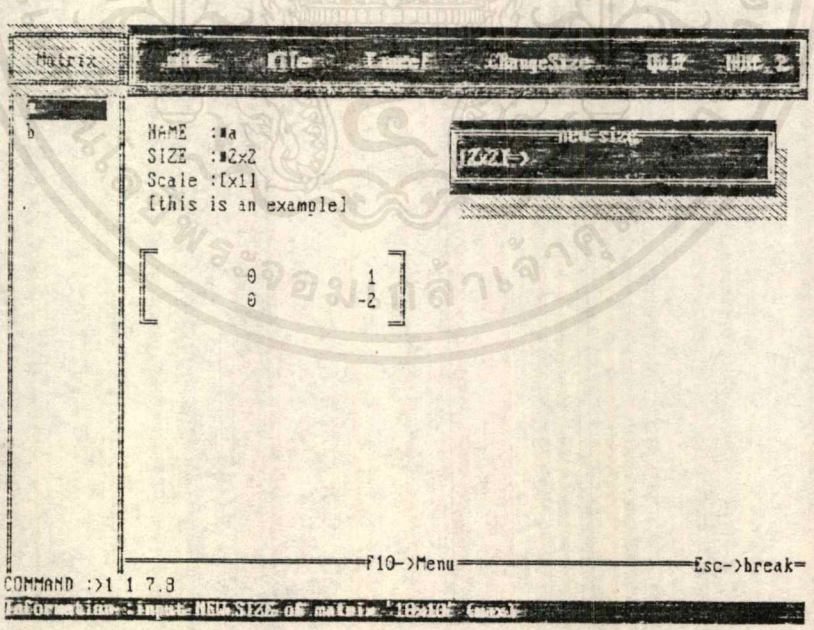


รูปที่ 20. แสดงการ save ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 21. แสดงการใช้เมนู Directory



รูปที่ 22. แสดงการใส่ค่าข้อมูลเมตริกซ์ และ การเปลี่ยนขนาดเมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10

โปรแกรมแสดงผลกราฟ

10.1 จุดประสงค์และหน้าที่

-อ่านข้อมูลการ simulation จากแผ่นดิสก์มาแสดงผลในรูปกราฟ

-นำข้อมูลจากหลาย systems มาแสดงผลเปรียบเทียบได้

-มีฟังก์ชันการใช้งานย่อย เพื่อช่วยในการแสดงรูปกราฟที่เหมาะสม

10.2 แนวคิด

รูปกราฟเป็น 2 มิติ ควรมีสเกลที่เหมาะสมเช่น ข้อมูลอยู่ตั้งแต่ 0.1-10.9 ก็ควรขีดเส้นสเกลที่ 1,2,3,..,9,10 มีฟังก์ชันย่อยอื่นที่จะช่วยในการแสดงผลเช่น กราฟหนึ่งสามารถแสดงผลได้หลายข้อมูล และแสดงได้หลายกราฟ สามารถปรับrangeได้ตามต้องการ เลือกได้ว่าจะให้แกน x,y เป็นข้อมูลอะไร แสดงในสเกล log ได้ ดูค่าข้อมูลแต่ละจุดในกราฟได้ แสดงผลออกทางเครื่องพิมพ์ได้

การออกแบบจะสร้างข้อมูลควบคุมการแสดงผลขึ้นมา ซึ่งการทำงานในแต่ละโปรแกรมย่อยจะขึ้นกับข้อมูลควบคุมนี้เท่านั้น จะทำให้แก้ไขเปลี่ยนแปลงแต่ละส่วนได้ง่ายขึ้นโดยไม่กระทบกระเทือนส่วนอื่น ซึ่งข้อมูลควบคุมมีดังต่อไปนี้

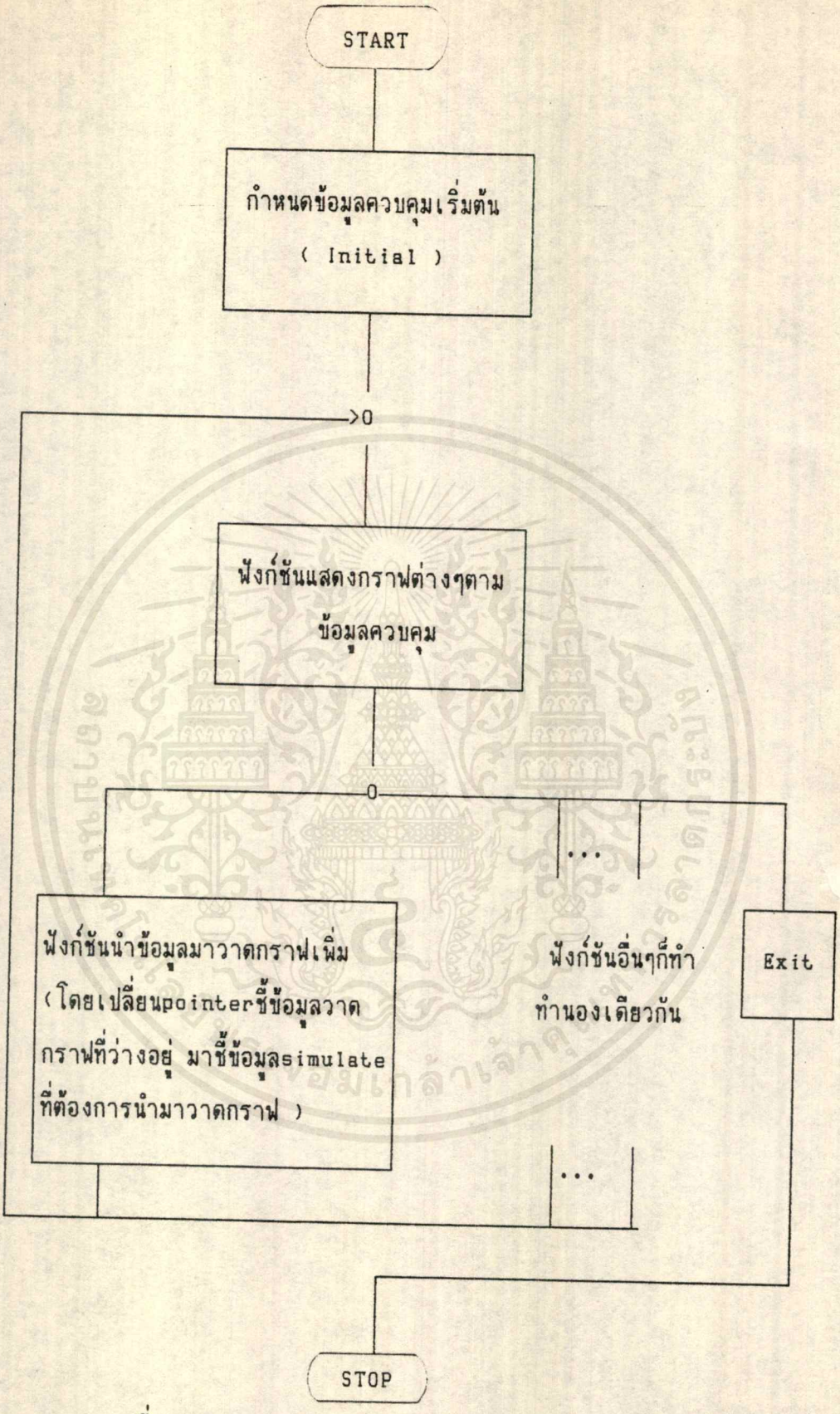
-ข้อมูลแสดงสถานะปัจจุบัน (จะเปลี่ยนไปตามผู้ใช้เลือก) เช่น จำนวนรูปกราฟที่กำลังแสดงอยู่, ตำแหน่งกราฟที่กำลังใช้งาน, รูปแบบเส้นกราฟที่ใช้อยู่ปัจจุบัน, จำนวน system ที่ load เข้ามา

-ข้อมูลที่คงที่(จะกำหนดเมื่อโปรแกรมถูก RUNและไม่เปลี่ยนตลอดการใช้งาน) เช่น ขนาดจอ, จำนวน sytems ที่ loadเข้ามา, จำนวนข้อมูลในแต่ละ system

-ข้อมูลกราฟ เช่น สเกลแกน x และ y,pointerที่ชี้ไปยังข้อมูลที่เป็นเส้นกราฟแต่ละเส้น, รูปแบบเส้นกราฟของแต่ละข้อมูล

เมื่อมีข้อมูลควบคุมก็สร้างฟังก์ชันที่วาดกราฟตามข้อมูลควบคุมขึ้นมา ฟังก์ชันอื่นๆก็เพียงแต่แก้ข้อมูลควบคุมให้มีการแสดงผลตามต้องการ ดัง flowchart

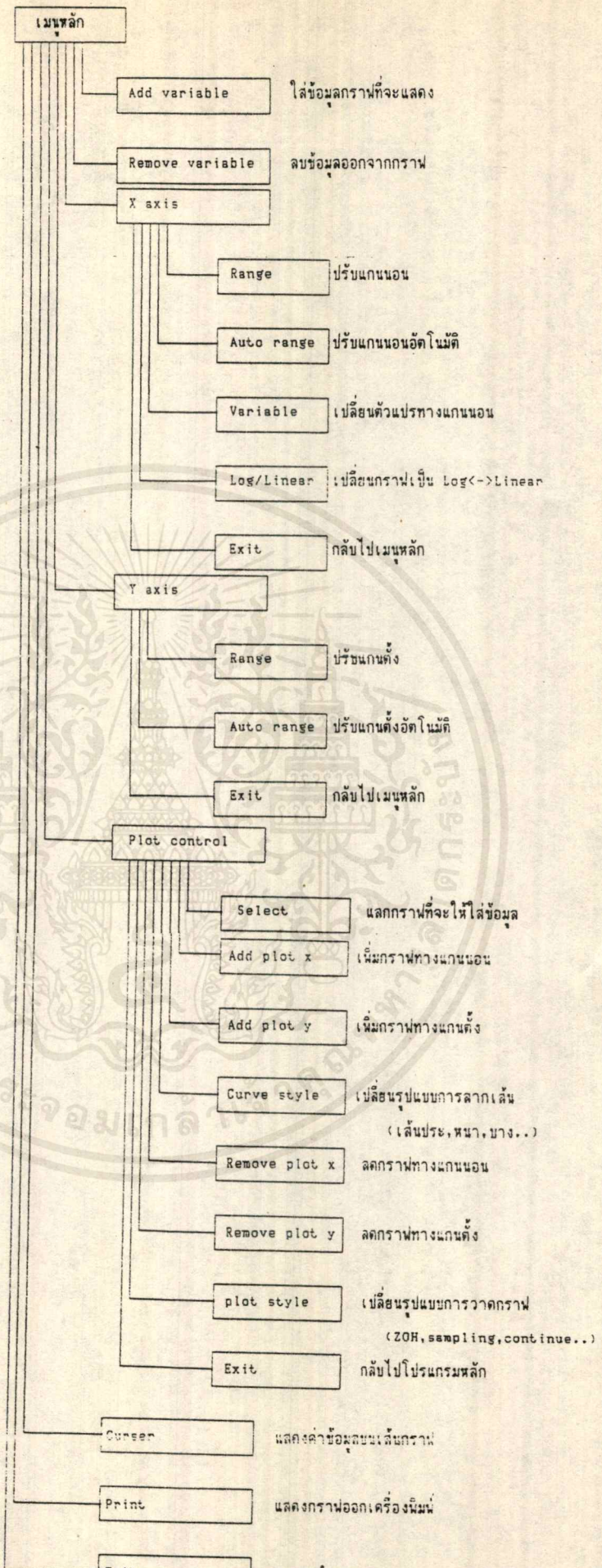
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 23. flow chart ของการวาดกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.3 รายละเอียด

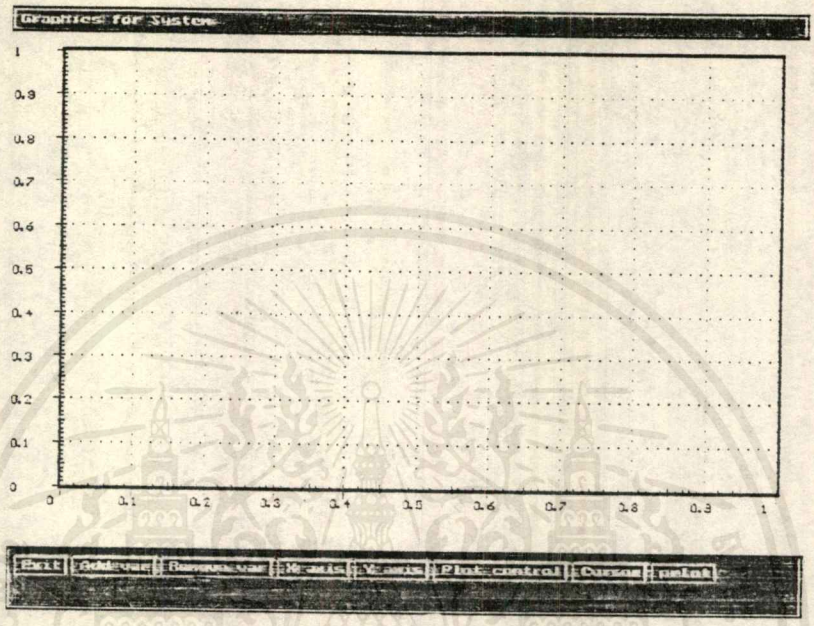


รูปที่ 24. แผนผังโปรแกรม GRAPH.EXE

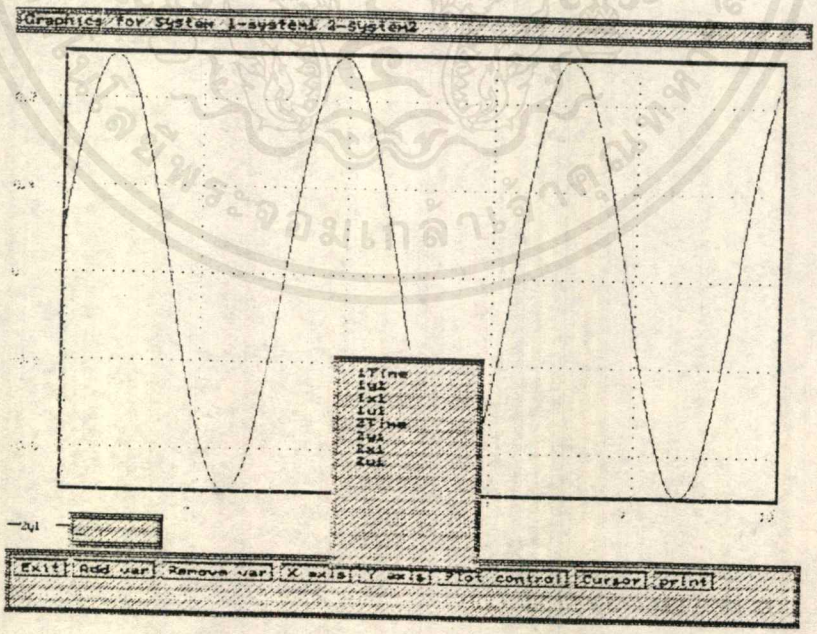
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.4 ผลการออกแบบ

มีรายละเอียดเมนูและการทำงานดังส่วนต่างๆดังนี้

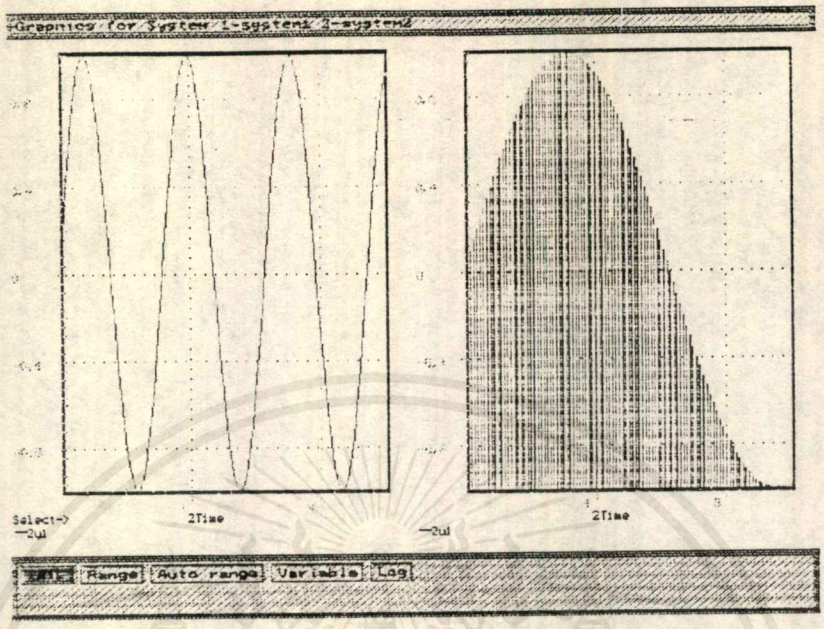


รูปที่ 25. แสดงเมนูหลักของ GRAPH.EXE

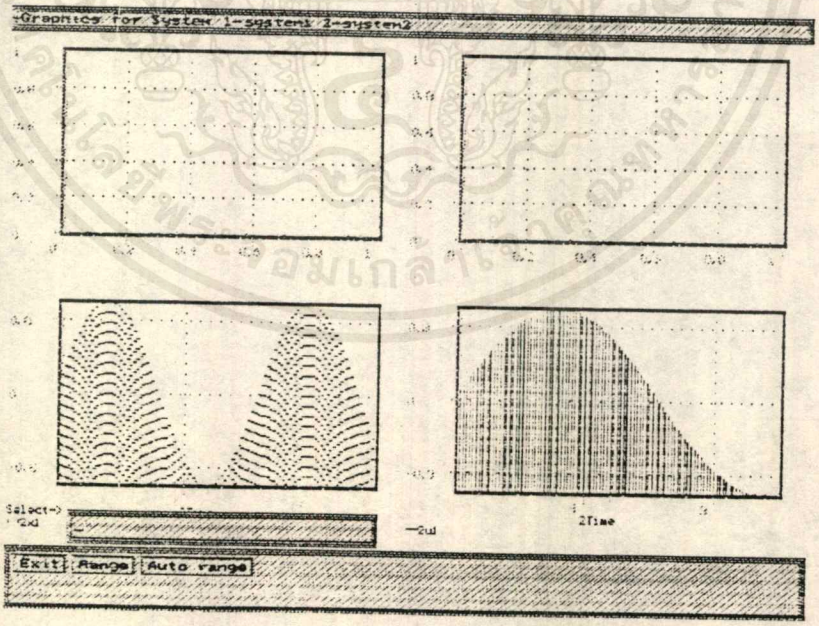


รูปที่ 26. แสดงการวาดรูปกราฟ และการ List รายชื่อข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

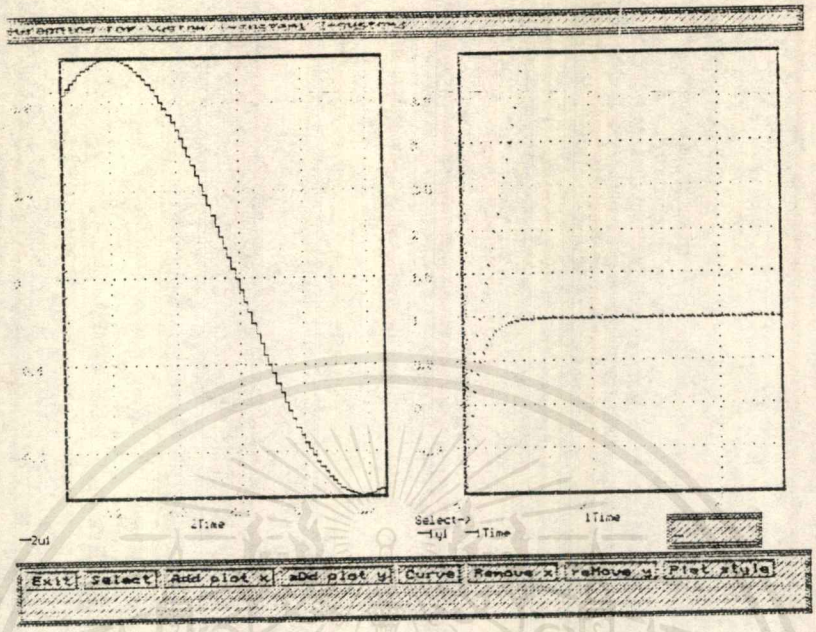


รูปที่ 27. แสดงการเปลี่ยนรูปแบบการ plot และ แสดงเมนู X_axis

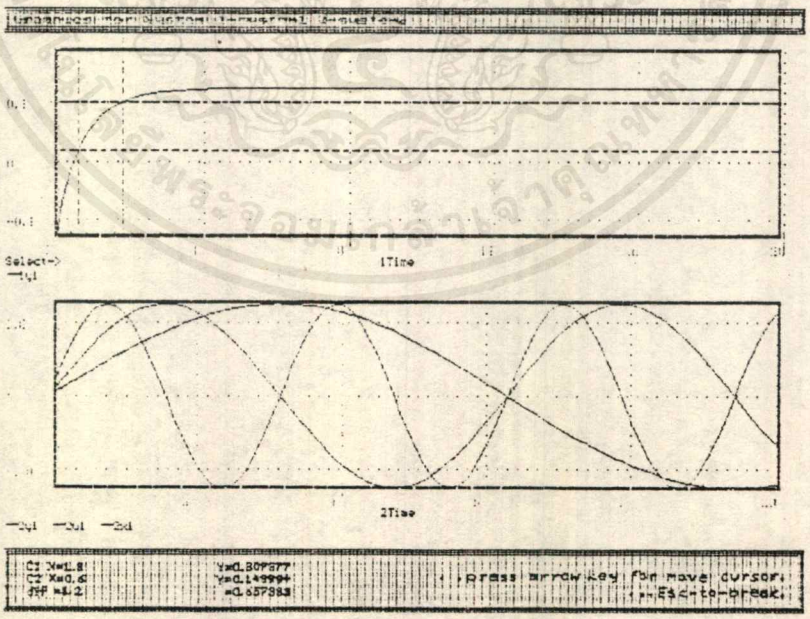


รูปที่ 28. แสดงการเปลี่ยนรูปแบบ carveและ แสดงเมนู Y_axis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 29. แสดงเมนู Plot_control



รูปที่ 30. แสดงการใช้เมนู Cursor ดูค่าข้อมูลที่จุดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11

โปรแกรม ANALYSIS

11.1 โปรแกรมกำหนดตำแหน่งโพล [POLE.EXE]

11.1.1 จุดประสงค์และหน้าที่

-รับค่าโพลใหม่สำหรับระบบอินพุตเดียว และตรวจสอบค่าโพลนั้น ส่วนหลายอินพุตจะรับสเตจเมตริกซ์ใหม่(A')เข้ามา

-กรณีอินพุตเดียวจะคำนวณหา Q (Controllable Matrix) แล้วหาเมตริกซ์ K เมื่อระบบควบคุมได้

-กรณีหลายอินพุตจะใช้วิธีแก้สมการ A-BK = A' หาค่า K

11.1.2 แนวคิด

กรณีอินพุตเดียวจะใช้ วิธี Ackermann's formula

$$K = [\,0\ 0\ \ldots\ 0\ 1\,][\,B \mid AB \mid \ldots \mid A^{n-1}B\,]^{-1}\phi(A)$$

โดย $\phi(A) = A^n + \alpha_1 A^{n-1} + \ldots + \alpha_{n-1} A + \alpha_n I$

$$\phi(s) = (s-\mu_1)(s-\mu_2)\ldots(s-\mu_n)$$

$$= s^n + \alpha_1 s^{n-1} + \ldots + \alpha_{n-1} s + \alpha_n$$

เมื่อ $\mu_1, \mu_2, \ldots, \mu_n$ คือตำแหน่งโพลที่ต้องการ

กรณีหลายอินพุตจะแก้สมการ A - BK = A' ได้ ในลักษณะ

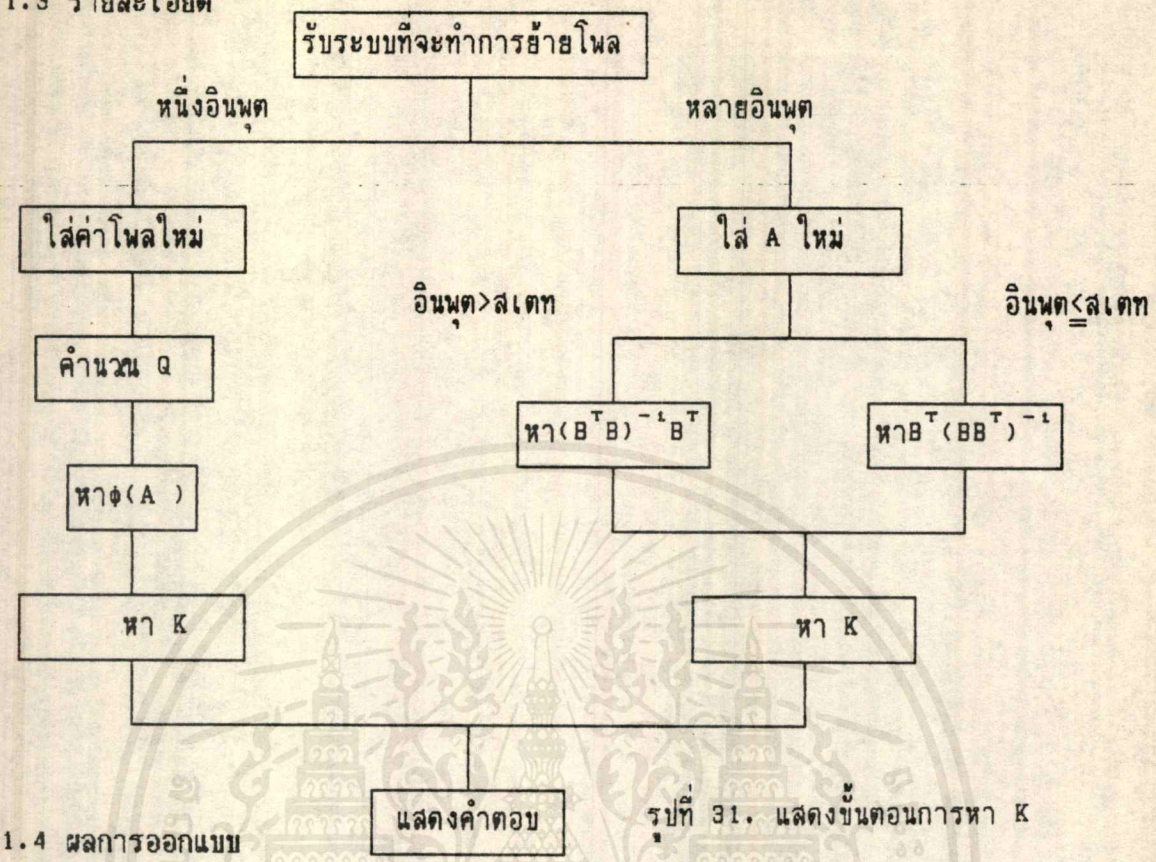
$K = (B^TB)^{-1}B^T(A - A')$ เมื่อจำนวนอินพุตมากกว่าจำนวนสเตท

$K = B^T(BB^T)^{-1}(A - A')$ เมื่อจำนวนสเตทมากกว่าจำนวนอินพุต

เนื่องจากหลายอินพุตค่า K อาจจะมีได้หลายคำตอบวิธีที่ใช้จะได้คำตอบที่เป็นค่า Minimum Norm

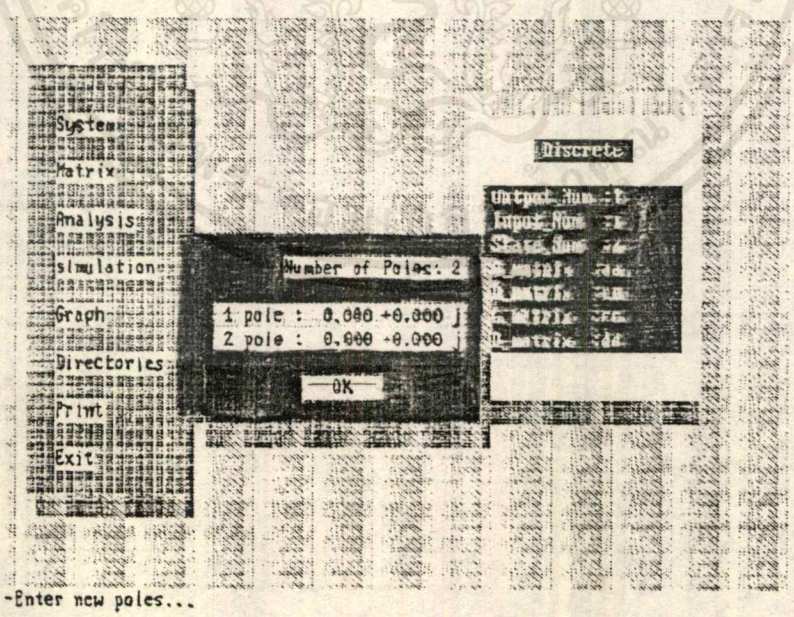
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.1.3 รายละเอียด



รูปที่ 31. แสดงขั้นตอนการหา K

11.1.4 ผลการออกแบบ



รูปที่ 32. แสดงการรับโพลใหม่ของ POLE.EXE (กรณี1อินพุต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.2 โปรแกรมคำนวณตัวสังเกตสเตท [OBSERVER.EXE]

11.2.1 จุดประสงค์และหน้าที่

-รับค่าโพลใหม่สำหรับระบบอินพุตเดียว และตรวจสอบค่าโพลนั้น ส่วนหลายอินพุตจะรับสเตจเมตริกซ์ใหม่(A')เข้ามา

-กรณีอินพุตเดียวจะคำนวนหา O (Observable Matrix) แล้วหาเมตริกซ์ K_e เมื่อระบบควบคุมได้

-กรณีหลายอินพุตจะใช้วิธีแก้สมการ $A-K_eC = A'$ หาค่า K_e

11.2.2 แนวคิด

เนื่องจากมีการคำนวณแบบเดียวกับการกำหนดตำแหน่งโพลโดยใช้สเตทป้อนกลับจึงใช้คุณสมบัติ Duality จากระบบ

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

จะสามารถทำการออกแบบ Full Order State Observor ของระบบได้โดยแก้ปัญหาการวางตำแหน่งโพล (Pole Placement) ของระบบคู่ Dual ของระบบเดิมคือ

$$\dot{z} = A^Tz + C^Tv$$

$$n = B^Tz$$

และสมมติให้สัญญาณควบคุม คือ $v = -Kz$

หรือ

$$\dot{x}" = A"x" + B"u"$$

$$y" = C"x"$$

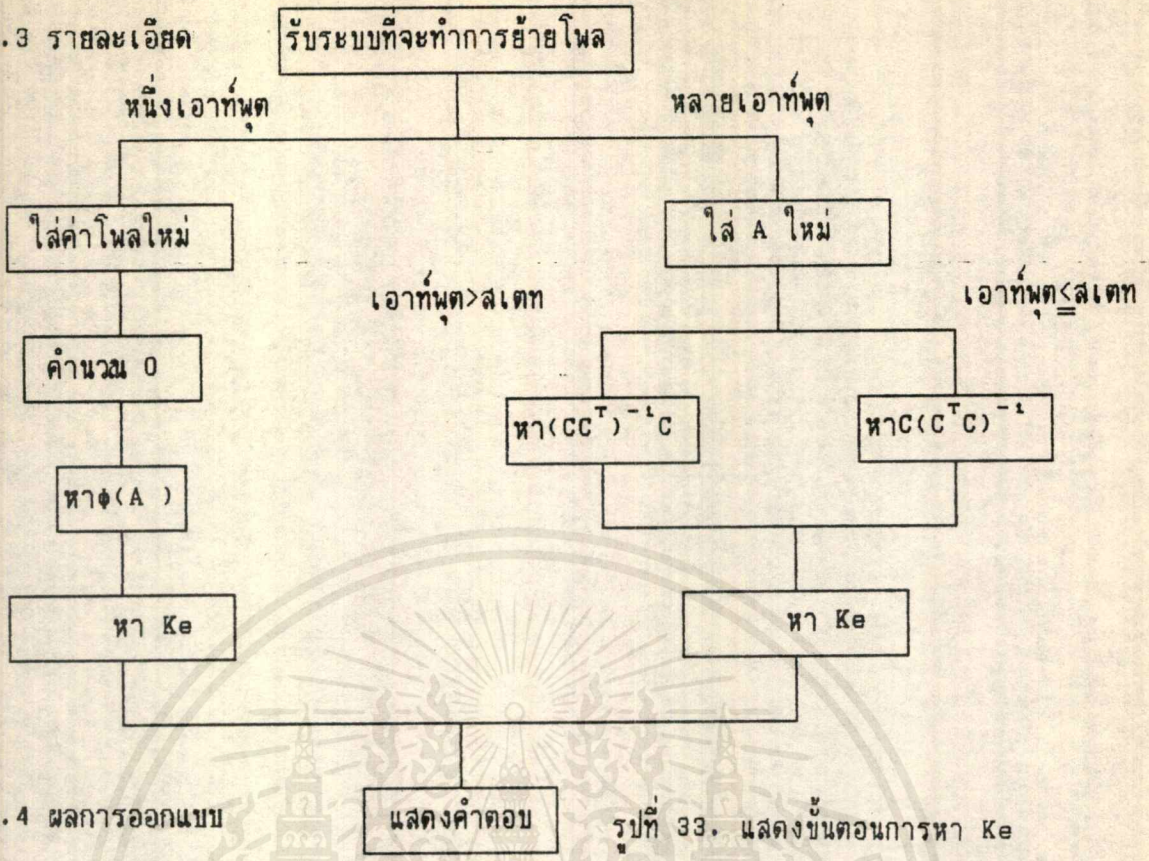
สามารถหา K ได้จาก

$$\left| sI - (A"-C"K) \right| = (s-\mu_1)(s-\mu_2)\ldots(s-\mu_n)$$

จากนั้นหา K_e ได้จาก $K_e = K^T$

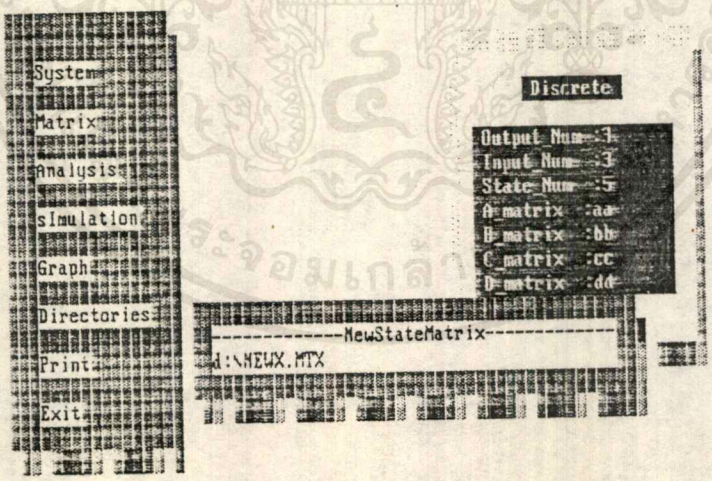
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.2.3 รายละเอียด



รูปที่ 33. แสดงขั้นตอนการหา Ke

11.2.4 ผลการออกแบบ



รูปที่ 34. แสดงการรับ A ใหม่ของ OBSERVER.EXE (กรณีหลายเอาท์พุต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 12
ตัวอย่าง

ตัวอย่างที่ 1

กำหนดระบบ

$$\dot{X} = AX + Bu$$

$$Y = CX + Du$$

โดยที่

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 20.601 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0.4905 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0.5 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

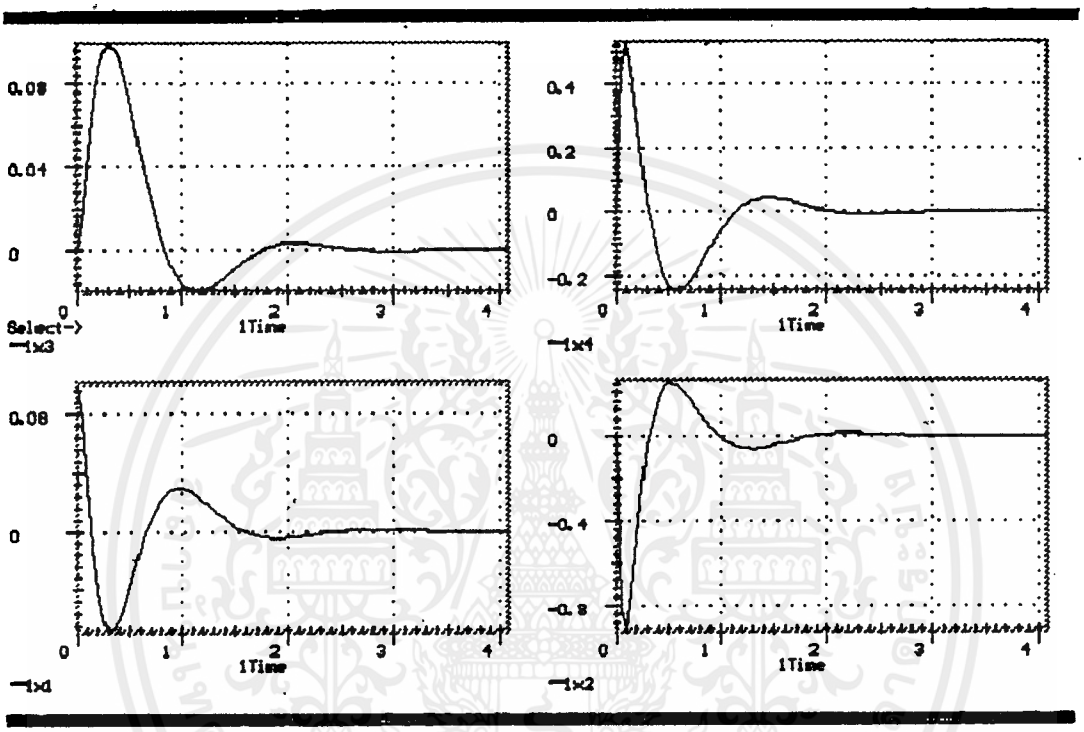
ต้องการกำหนดโพลที่ $\mu_1 = -2 + j3.464$, $\mu_2 = -2 - j3.464$, $\mu_3 = -10$, $\mu_4 = -10$

*ใช้โปรแกรม POLE.EXE หาเมตริกซ์ K ได้ดังนี้

$$K = [\ -298.1461 \quad -60.69653 \quad -163.0917 \quad -73.39307\ ]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Simulate ระบบที่ได้รับการย้าย และไม่ได้รับการย้ายตำแหน่งโพล จะพบว่า ระบบที่ได้รับการย้ายตำแหน่งโพลแล้ว จะมีผลตอบสนองของตัวแปรสเตททุกตัวลู่เข้าจุดสมดุลย์เสมอ



รูปที่ 35. แสดงรูปกราฟของสเตทต่างๆ ของระบบที่ย้ายโพลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ตัวอย่างที่ 2</u>

กำหนดระบบ

$$\dot{X} = AX + Bu$$

$$Y = CX + Du$$

โดยที่

$$A = \begin{bmatrix} 0 & 1 \\ 20.6 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

จะทำการกำหนดตำแหน่งโพล 3 กรณีด้วยกันดังนี้

(1) $\mu_1 = -1$,$\mu_2 = -3$

(2) $\mu_1 = -1.8 + j2.4$,$\mu_2 = -1.8 - j2.4$

(3) $\mu_1 = j2$,$\mu_2 = -j2$

ป้อนข้อมูลให้ MATRIX.EXE,SYSTEM.EXE และ POLE.EXE แล้วจะได้ค่า K จากทั้ง 3 กรณีดังนี้

กรณีที่ 1 K = [23.6 4]

กรณีที่ 1 K = [29.6 3.6]

กรณีที่ 1 K = [24.6 0]

จากค่า K ทั้ง 3 กรณี ทำการ Simulate ระบบใหม่ของแต่ละกรณี ซึ่งจะอยู่ในรูป

$$\dot{X} = (A - BK)X + u$$

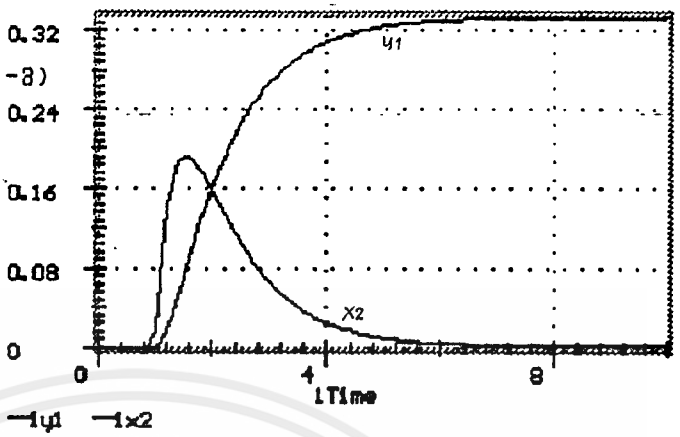
$$Y = CX + Du$$

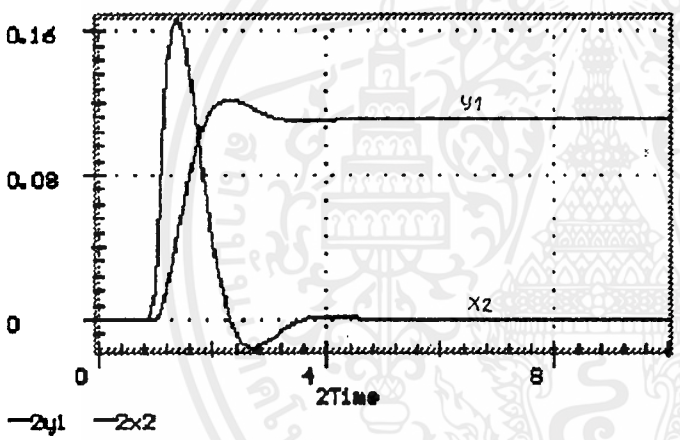
โดยป้อน u = Unit Step จะได้กราฟแสดงผลดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 36.

แสดงกราฟของ เอาท์พุตและ สเตท
เมื่อย้ายโพลไปอยู่บนแกนจริง(μ=-1,-8)



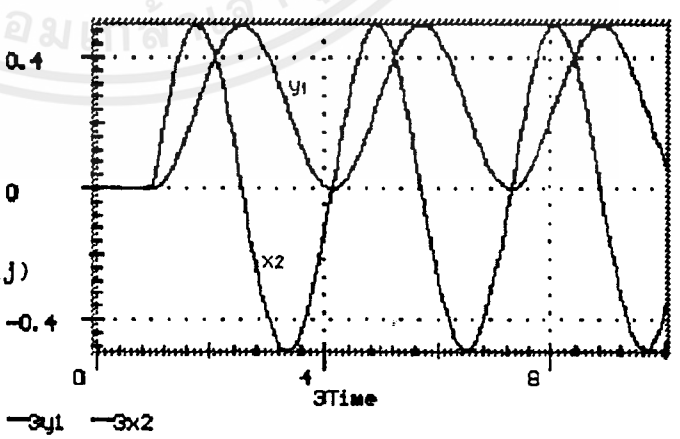


รูปที่ 37.

แสดงกราฟของ เอาท์พุตและ สเตท
เมื่อย้ายโพลไปที่ตำแหน่ง μ=-1.8$\pm$2.4j

รูปที่ 38.

แสดงกราฟของ เอาท์พุตและ สเตท
เมื่อย้ายโพลไปอยู่บนแกนจินตภาพ (μ=$\pm$2j)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 13
สรุปและวิจารณ์

โครงงานที่ได้ทำขึ้นมานี้ เป็นการเขียนโปรแกรมคอมพิวเตอร์เพื่อนำมาใช้ในการอำนวยความสะดวกแก่ผู้ใช้โปรแกรม นักศึกษา และผู้สนใจทั่วไป ในการออกแบบระบบควบคุมในแนว Modern Control โครงงานจึงประกอบด้วย 2 ส่วน คือ ส่วนทฤษฎี และส่วนการออกแบบเขียนโปรแกรม

-ส่วนทฤษฎี: การออกแบบระบบควบคุมเชิงเส้นไม่แปรตามเวลาในแนว Modern Control จะเขียนสมการแสดงระบบในรูป State Space สำหรับวิธีการออกแบบที่ใช้ในโครงงานนี้คือการกำหนดตำแหน่งโพล ระบบที่จะได้รับการออกแบบจะต้องควบคุมได้โดยสมบูรณ์ เป้าหมายการออกแบบคือการหาค่าเมตริกซ์ป้อนกลับ K ส่วนในการสร้างตัวสังเกตสเตท ระบบต้องสามารถสังเกตได้โดยสมบูรณ์ เป้าหมายการออกแบบคือการหาค่าเมตริกซ์ K_e

-ส่วนโปรแกรม: ได้ออกแบบโปรแกรมซึ่งสามารถคำนวนค่าเมตริกซ์ K และ K_e ของระบบได้ตามเป้าหมาย(ในโครงงาน ตัวสังเกตเป็นแบบ Full Order) พร้อมทั้งได้ทำส่วนอำนวยความสะดวกอื่น ๆ อีกมากมาย ได้แก่ ส่วนการคำนวนทางเมตริกซ์ทุกรูปแบบ ส่วนการทำ Simulationให้ระบบ ส่วนแสดงกราฟ และส่วนแสดงผลออกทางเครื่องพิมพ์ ซึ่งส่วนต่าง ๆ ที่เขียนขึ้นมานี้ สามารถรองรับการเพิ่มเติมรูปแบบการวิเคราะห์และออกแบบระบบควบคุมวิธีอื่น ๆ ได้

จึงหวังอย่างยิ่งว่า ในอนาคตต โครงงานนี้จะได้รับการสานต่อ กล่าวคือ มีการเขียนโปรแกรมการวิเคราะห์และออกแบบระบบควบคุมวิธีอื่น ๆ แล้วนำมาเชื่อมต่อเข้ากับโปรแกรมหลักที่เขียนขึ้นนี้ ซึ่งพร้อมที่จะรองรับการเขียนโปรแกรมเพิ่มเติมเข้ามาอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- MODERN CONTROL ENGINEERING Second Edition, KATSUHIKO OGATA, Prentice-Hall Inc.,1990

- DISCRETE-TIME CONTROL SYSTEMS, KATSUHIKO OGATA, Prentice-Hall Inc., 1987

- TURBO C USER'S GUIDE REFERENCE GUIDE ADDITION & ENHANCEMENTS Version 2.0, Borland Inc.,1988

-COMPUTATIONAL METHODS FOR LINEAR CONTROL SYSTEM, PETKOV AND KONSTANTINOV ,Prentice Hall International Series in System and Control Engineering

-LINEAR CONTROL SYSTEM ANALYSIS AND DESIGN : CONVENTIONAL AND MODERN , Third Edition,JOHN J. D'AZZO,CONSTANTINE H. HOUPIS ,New York ,Mc Graw Hill, 1988

- หลักการโปรแกรม Object-Oriented และภาษา C++ , ซีเอ็ดยูเคชั่น,พ.ศ. 2535

- เอกสารวิชาการ ABACUS Version 10, ชุมนุมคอมพิวเตอร์, คณะวิศวกรรมศาสตร์ พระจอมเกล้าลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

1. ข้อมูลเมตริกซ์

มีส่วนประกอบดังต่อไปนี้

```
struct  MATRIX{
            char  NAME[9];
            int  size[2];
            float  scale;
            float   *data;
            char   comment[50];
             };
```

-ชื่อ MATRIX

เนื่องจากจะนำไปกำหนดเป็นชื่อ FILE ของข้อมูล MATRIX ตัวนั้น จึงจะต้องเป็นไปตามที่กำหนดของการตั้งชื่อ FILE ถึงความยาวสูงสุด 8 ตัวอักษร และจะต้องไม่เป็นตัวอักษรหรือสัญญลักษณ์พิเศษ เช่น '\' , ':' , '*' , '.' , ... เป็นต้น ซึ่งไม่ต้องกำหนดนามสกุลให้กับชื่อ FILE (ชื่อ MATRIX) นั้น โดยโปรแกรมจะกำหนดให้นามสกุลเป็น .MTX เองโดยอัตโนมัติเมื่อมีการเก็บข้อมูลลงแผ่นดิสก์

-ขนาด MATRIX

เป็นตัวเลข (INTEGER 16 บิท) ใด ๆ แต่ในที่นี้ได้กำหนดขนาดของ MATRIX ให้มีค่าสูงสุด 10x10 เพราะการแสดงค่าในแต่ละหลัก แถว จะแสดงได้ไม่หมด ทำให้ไม่สะดวกแก่การใช้งาน ประกอบกับจุดประสงค์ที่จะนำไปเป็น MATRIX ในระบบ MODREN CONTROL ซึ่งส่วนใหญ่ที่ใช้วิเคราะห์จะเป็นที่ไม่ค่อยวิเคราะห์ระบบที่มีขนาดใหญ่ MATRIX มากนัก จึงกำหนดสูงสุดไว้เพียง 10X10

-ข้อมูลแต่ละแถวและหลักต่าง ๆ

เป็นจำนวนจริง (FLOAT) ซึ่งสามารถเขียนในรูปแบบต่าง ๆ เช่น 1, 3.40, 4E2, 4.0E+2, 4.E-2 ซึ่งสามารถใส่ตัวเลขหลังทศนิยม ตามขนาดของ FLOAT แต่ในโปรแกรมแสดงผลที่หน้าจออาจจะแสดงไม่ครบ ตามที่ใส่ เพราะอาจยาวเกินไปทำให้ตัวเลขไปทับกับหลักอื่น ๆ จึงจะเลือกแสดงตามเหมาะสมแต่ค่าใน MEMORY หรือ FILE จะเป็นค่าตามที่ใส่ เช่น ใส่ค่า 1.12345678 จะแสดงเป็น 1.123+ ซึ่งแล้ว แต่ขนาดของเมตริกซ์ที่ทำให้สามารถแสดงให้เห็นได้สูงสุดกี่หลักเพื่อไม่ให้ไปทับค่าของหลักอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-SCALE

จะเห็นว่าการใส่ค่า ถ้าทุกค่ามีขนาดที่ค่อนข้างใหญ่ หรือเล็ก ในการแสดงผลออกหน้าจอไม่พอดี และเพื่อให้การใส่ค่าข้อมูลเป็นไปได้สะดวกขึ้น อีกทั้งยังเพิ่มขีดจำกัดของขนาดข้อมูล (FLOAT) ให้รับค่าที่ใหญ่ขึ้นได้ จึงเพิ่ม SCALE สำหรับคูณทุกค่าในแถวและหลักต่าง ๆ ซึ่งสามารถใส่ค่าตาม FORMAT ดังนี้

<จำนวนจริง>, <จำนวนจริง>,<SCALE SUFFIXES>

SCALE SUFFIXES	K=1E3	m=1E-3
	M=1E6	u=1E-6
	G=1E9	n=1E-9
		P=1E-12

-MATRIX COMMENT

เพื่อบอกรายละเอียดที่มาหรือข้อมูลที่น่าสนใจใน MATRIX ตัวนั้น ซึ่งจะไม่มีผลต่อการคำนวณ

-รูปแบบการเก็บลงแผ่นดิกส์

```
[name]=file name
[scale],[size row],[size colunm],[data]..[data],[comment]
```

2. ข้อมูลระบบ

มีส่วนประกอบดังต่อไปนี้

```
struct   system{
      char  NAME[9],mode;
      int   size[3];
      char  MNAME[4][9];
      char  comment[50];
       };
```

-ชื่อ SYSTEM

เนื่องจากจะนำไปกำหนดเป็นชื่อ FILE ของข้อมูล SYSTEM ตัวนั้น จึงจะต้องเป็นไปตามที่กำหนดของการตั้งชื่อ FILE ถึงความยาวสูงสุด 8 ตัวอักษร และจะต้องไม่เป็นตัวอักษรหรือสัญญลักษณ์พิเศษ เช่น '\' , ':' , '*' , '.' , ... เป็นต้น ซึ่งไม่ต้องกำหนดนามสกุลให้กับชื่อ FILE (ชื่อ SYSTEM) นั้น โดยโปรแกรมจะกำหนดให้นามสกุลเป็น .STM เองโดยอัตโนมัติเมื่อมีการเก็บข้อมูลลงแผ่นดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ขนาด INPUT OUTPUT STATE

เป็นตัวเลข (INTEGER 16 บิท) ใด ๆ แต่ในที่นี้ได้กำหนดขนาดของ MATRIX ให้มีค่าสูงสุด 10x10 เพราะฉนั้นก็จะมีค่าไม่เกิน 10

-ชื่อเมตริกซ์

-COMMENT

เพื่อบอกรายละเอียดที่มาหรือข้อมูลที่น่าสนใจใน SYSTEM ตัวนั้น ซึ่งจะไม่มีผลต่อการคำนวณ

-รูปแบบการเก็บลงแผ่นดิกส์

[mode],[size of input],[size of output],[size of state],

[matrix A],[matrix B],[matrix C],[matrix D],[comment]

3. ข้อมูลกราฟ

ประกอบด้วย [sampling period],[data num],[data-name]..[data-name],

[data],...[data]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้