



การศึกษาการควบคุมแบบโรบัสต์
ROBUST CONTROL



โดย
นายเกียรติพร เลิศสถิตย์พงษ์
นางสาวชนันสถา ตันจันทร์พงศ์

วัน เดือน ปี 30 ก.ค. 2540
เลขทะเบียน 03665
เลขเรียกหนังสือ T38058 ก 851 ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

036965

การศึกษาการควบคุมแบบโรบัสต์
ROBUST CONTROL



โดย
นายเกียรติพร เลิศสถิตย์พงษ์
นางสาวชนัสตา ตันจันทร์พงศ์
อาจารย์ที่ปรึกษา
รองศาสตราจารย์ ดร. จงกล งามวิวิทย์
รองศาสตราจารย์ วิพันธ์ ปรีชาพานิช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2538

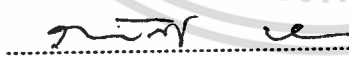
ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาการควบคุมแบบโรบัสต์
ROBUST CONTROL

ผู้จัดทำ นายเกียรติพร เลิศสถิตย์พงษ์
นางสาวชนันสถา ดันจันทร์พงษ์


.....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร. จงกล งามวิวิทย์)


.....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ วิพันธ์ ปรีชาพานิช)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาการควบคุมแบบโรบัสต์ ROBUST CONTROL

โดย

นายเกียรติพร เลิศสถิตย์พงษ์

นางสาวชนัสตา ดันจันทรพงศ์

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร. จงกล งามวิวิทย์

รองศาสตราจารย์ วิพันธุ์ ปรีชาพานิช

บทคัดย่อ

ในการออกแบบระบบควบคุมหนึ่ง ๆ จุดมุ่งหมายของผู้ออกแบบระบบนั้น คือสร้างระบบให้สามารถใช้งานได้ในสถานะจริง ซึ่งสถานะแวดล้อมอาจเกิดการเปลี่ยนแปลงหรือเงื่อนไขของระบบเกิดการเปลี่ยนแปลง จุดมุ่งหมายของผู้ออกแบบระบบนั้นยังต้องการให้ระบบที่สร้างขึ้นมานั้นสามารถใช้งานได้ต่อไปโดยมีเสถียรภาพและมีค่าผิดพลาดของระบบที่ยอมรับได้ ในปฏิญานพนธ์ฉบับนี้จะกล่าวถึงวิธีการรักษาเสถียรภาพที่เรียกว่า การควบคุมแบบโรบัสต์ โดยจะเน้นในส่วนของการออกแบบแบบเอชอินฟินิตี้ในส่วนที่เป็นระบบแน่นอน และพัฒนาการเขียนโปรแกรมบนพื้นฐานฟังก์ชันของโปรแกรม MATLAB เพื่อช่วยในการคำนวณหาค่าเกนมาและตัวควบคุมแบบเอชอินฟินิตี้

ABSTRACT

The ultimate goal of a control-system designer is to build a system that will work in the real environment. Since the real environment may change with time or operating conditions may vary, the control system must be able to withstand these variations. The purpose of designer that the designed-system is stable and has error in the boundary that we can accepted. This thesis mentions about the robust control theory to guarantee stability using H infinity method for certain system. The programs which can solve gamma and H-infinity compensator based on MATLAB function are developed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีก็เพราะได้รับความเมตตาจากรองศาสตราจารย์ ดร. จงกล งามวิวิทย์ และรองศาสตราจารย์ วิพันธ์ ปรีชาพานิช อาจารย์ที่ปรึกษาที่ได้ให้ความกรุณาแนะนำแก่ผู้จัดทำตลอดมา ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

ขอขอบพระคุณคณาจารย์ต่าง ๆ ที่ได้ประสิทธิประสาทวิชาความรู้แก่ผู้จัดทำ
ขอกราบขอบพระคุณคุณพ่อคุณแม่ของผู้จัดทำที่ได้อุปการะผู้จัดทำและยังเป็นผู้ให้กำลังใจแก่ผู้จัดทำ

ขอขอบคุณเพื่อน ๆ ที่ให้ความเป็นเพื่อนและให้กำลังใจแก่ผู้จัดทำ โดยเฉพาะ
นายถาวร เบญจนาสุทธีที่ให้ความช่วยเหลือในการแก้ไขโปรแกรม

ผู้จัดทำ

นายเกียรติพร เลิศสถิตย์พงษ์

นางสาวชนันสถา ตันจันทรพงศ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	i
กิตติกรรมประกาศ	ii
สารบัญ	iii
สารบัญรูปภาพ	v
บทที่ 1 บทนำ	1
บทที่ 2 Robust Stabilization	
2.1 บทเกริ่นนำ	2
2.2 Modeling Uncertainty	2
2.3 Frequency Domain	3
2.4 Perturbed Model	5
2.5 Robust Stability Analysis	7
2.6 Frequency Domain Uncertainty	7
2.7 การพิจารณาหาค่า K ในการป้อนกลับของระบบ	8
บทที่ 3 พื้นฐานที่นำไปพิจารณาแก้ปัญหา H infinity	
3.1 บทเกริ่นนำ	9
3.2 Quadratic Forms	9
3.3 สมการรีคัตติ (Riccati Equations)	11
3.4 Linear Quadratic Regulator	12
3.5 ตัวกรองแบบคาร์มาน (Kalman Filter)	14
3.6 Linear Quadratic Gaussian	15
บทที่ 4 การควบคุมแบบ H infinity	

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น

4.1 บทเกริ่นนำ 17

4.2 การควบคุมแบบ H_∞ 17

4.3 เงื่อนไขในการหาค่าเกมมา	19
4.4 การพิจารณาหาตัวเกมมาที่เหมาะสม	20
4.5 โปรแกรมการหาค่าเกมมาและตัวควบคุมแบบ H infinity	22
4.6 ตัวอย่างการพิจารณาและผลการพิจารณา	26
4.7 สรุปผลและวิจารณ์	31

ภาคผนวก

ภาคผนวก A	
นิยามและทฤษฎีพื้นฐานทางคณิตศาสตร์ที่นำไปใช้	32
ภาคผนวก B	
โปรแกรม MATLAB เบื้องต้น	35
ภาคผนวก C	
อธิบายคำสั่งที่ใช้ในโปรแกรมหาค่าเกมมาและตัวควบคุม	55
เอกสารอ้างอิง	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้า
รูปที่ 2.1 รูปการพิจารณา Perturbed System แบบ Additive Uncertainty	4
รูปที่ 2.2 รูปการพิจารณา Perturbed System แบบ Multiplicative Uncertainty	4
รูปที่ 2.3 รูปการพิจารณา Perturbed System แบบ Coprime Factor Uncertainty	4
รูปที่ 2.4 Generalized Uncertainty Model (with Feedback Controller)	6
รูปที่ 4.1 แสดงบล็อกไดอะแกรมของระบบ H infinity	17
รูปที่ 4.2 โครงสร้างบล็อกไดอะแกรมแบบ H infinity	22
รูปที่ 4.3 Bode Diagram ของระบบที่ยังไม่มีตัวชดเชย	30
รูปที่ 4.4 Bode Diagram ของระบบที่มีตัวชดเชยแบบ H infinity	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในการพิจารณาระบบต่าง ๆ สิ่งที่เราต้องการก็คือเสถียรภาพ และในความเป็นจริง เมื่อมีการเปลี่ยนแปลงของพารามิเตอร์ในระบบนั้น สิ่งที่ต้องการมากกว่าเสถียรภาพคือความคงทน (robustness) ต่อการเปลี่ยนแปลงของพารามิเตอร์ภายในขอบเขตนั้นคือระบบจะต้องรักษาเสถียรภาพของระบบได้ แม้เกิดการเปลี่ยนแปลงขึ้น โดยการเปลี่ยนแปลงนั้นจะเกิดขึ้นภายในขอบเขตหนึ่ง

ปริญญานิพนธ์ฉบับนี้จะกล่าวถึงพื้นฐานของโรบัสต์และการหาตัวควบคุมเพื่อควบคุมระบบให้เสถียรภาพแบบโรบัสต์โดยวิธี H_∞ โดยมีขอบเขตการพิจารณาในระบบ Certainty

ในบทที่ 2 จะกล่าวถึงพื้นฐานของโรบัสต์ การเกิดค่าการเปลี่ยนแปลงเมื่อจำลองระบบจริงมาเป็นระบบทางคณิตศาสตร์ ค่าผิดพลาดที่เกิดจากการเปลี่ยนแปลงพารามิเตอร์ทั้งสองแบบคือแบบที่มีโครงสร้าง (structure uncertainty) และแบบที่ไม่มีโครงสร้าง (unstructure uncertainty)

ในบทที่ 3 จะกล่าวถึงรูปแบบพื้นฐานทางคณิตศาสตร์ที่ควรรู้และวิธีหาตัวควบคุมแบบ LQR และ LQG เนื่องจากเราจะพิจารณาหาตัวควบคุมของระบบที่มีหลายตัวแปร ซึ่งเขียนอยู่ในรูปแบบสมการสเตท

ในบทที่ 4 จะยังมิได้พิจารณาค่าผิดพลาดแบบ uncertainty โดยจะพิจารณาเพียงในส่วนของการหาตัวควบคุมของระบบที่ไม่มีค่าเปลี่ยนแปลง (certainty system) ด้วยวิธี H_∞ โดยแสดงถึงเงื่อนไขในการหาค่าเกมมา ที่เหมาะสมพร้อมทั้งมีตัวอย่างในการหาค่าเกมมาประกอบด้วย

ทฤษฎีพื้นฐานทางคณิตศาสตร์บางส่วน และส่วนของโปรแกรม MATLAB ที่ใช้ในการหาค่าและแสดงผลของการควบคุมนั้นได้รวบรวมไว้ในภาคผนวก

บทที่ 2

เสถียรภาพของโรบัสต์ (Robust Stabilization)

2.1 บทเกริ่นนำ

ในการออกแบบเสถียรภาพของระบบ เราจะต้องจำลองระบบจากระบบจริง (actual model) เพื่อให้มีรูปแบบที่ง่ายและสะดวกต่อการคำนวณ และด้วยเหตุนี้ระบบจำลอง (nominal model) จึงมีค่าที่แตกต่างจากระบบจริง ค่าแตกต่างนี้เรียกว่า ค่าผิดพลาดที่ไม่แน่นอน (modelling errors) หรือ ค่าความไม่แน่นอนของระบบ (system uncertainty)

ฉะนั้นการออกแบบตัวควบคุมจะต้องทำให้รูปแบบโนมินอล (nominal model) และระบบภายในขอบเขตรอบ ๆ รูปแบบโนมินอล (nominal model : uncertainty region) นั้น เสถียรด้วย (ซึ่ง actual model อยู่ในขอบเขตนี้) และความสามารถของระบบวงปิด ที่จะคงเสถียรภาพไว้ภายในขอบเขตค่าผิดพลาดที่ยอมรับได้ (uncertainty region) นี้ เรียกว่าเสถียรภาพแบบโรบัสต์ (robust stability)

การออกแบบตัวควบคุมที่แบบเสถียรภาพแบบโรบัสต์ของระบบวงปิด นั้นมีหลายวิธีเราจะพิจารณาในกรณีที่เป็นแบบระบบค่าผิดพลาดที่ไม่มีโครงสร้างแน่นอน (unstructured uncertainty) คือการเกิดค่าผิดพลาดที่ไม่มีข้อมูลอื่นใดนอกจากขนาดของขอบเขตที่เป็นไปได้ อยู่ในรูปของฟังก์ชันความถี่ (frequency function) เราจะใช้วิธีเอชอินฟินิตี้ออฟดิโมเซชัน (H_∞ Optimization ; เป็นการทดสอบ H_∞ norm ของระบบวงปิด) ในการออกแบบ ซึ่งจะได้ตัวควบคุมเสถียรภาพแบบโรบัสต์ตามต้องการ

2.2 ค่าผิดพลาดของโมเดล (Modeling Uncertainty)

ในการเสนอค่าผิดพลาดของโมเดล เพื่อพิจารณาในการออกแบบตัวควบคุมนั้นจะให้อยู่ในรูปแบบทางคณิตศาสตร์ซึ่งสามารถมองได้ 2 รูปแบบ

1. โดเมนเชิงความถี่ (frequency domain)

ค่าผิดพลาด หรือเรียกว่า ค่ารบกวน (perturbation) นั้นจะพิจารณาในรูปของฟังก์ชันถ่ายโอนแยกจากตัวระบบ

2. สมการสเทตของกระบวนการ (state space of process)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G = (A, B, C, D)$$

ซึ่งเราพิจารณาตัวแปรในเมตริกซ์ (A,B,C,D)

ในบทนี้จะกล่าวถึงระบบที่พิจารณาในเชิงความถี่ซึ่งการพิจารณาในเชิงสมการสเตจจะกล่าวถึงในบทต่อ ๆ ไป

2.3 โดเมนเชิงความถี่

เราจะให้ค่าผิดพลาดขึ้นกับความถี่และพิจารณารูปแบบที่มีโครงสร้าง (structured) และรูปแบบที่ไม่มีโครงสร้าง (unstructured) แยกกัน

ดังที่กล่าวมาแล้วว่าระบบค่าผิดพลาดที่ไม่มีโครงสร้างแน่นอน (unstructured uncertainty) เป็นการพิจารณาที่ไม่มีข้อมูลอื่นใด นอกจากขอบเขตของค่าผิดพลาดหรือค่าขนาด (magnitude) ของฟังก์ชันความถี่ ส่วนระบบค่าผิดพลาดที่มี (structured uncertainty) จะมีข้อมูลที่เป็นรูปแบบมาให้ และจากข้อมูลนี้ สามารถแยกเป็นคลาส (class) ให้ออกแบบตัวควบคุมแต่ละคลาสได้

เนื่องจากระบบที่มีค่าผิดพลาด ส่วนใหญ่มักจะเป็นแบบระบบที่ไม่มีโครงสร้างแน่นอน เราจึงพิจารณาที่ระบบมีค่าผิดพลาดและไม่มีโครงสร้างแน่นอน ดังนี้

ให้ G เป็น ฟังก์ชันถ่ายโอนของโมเดลโนมินอล (nominal model)

G_Δ เป็น ฟังก์ชันถ่ายโอนของโมเดลที่มีการรบกวน (perturbed model)

(i) สัญญาณรบกวนแบบค่าผิดพลาดแบบบวก (perturbation additive uncertainty : Δ_A)

$$G_\Delta = G + \Delta_A$$

(ii) สัญญาณรบกวนแบบค่าผิดพลาดแบบคูณ (perturbation multiplicative uncertainty : Δ_P)

$$G_\Delta = (I + \Delta_P)G$$

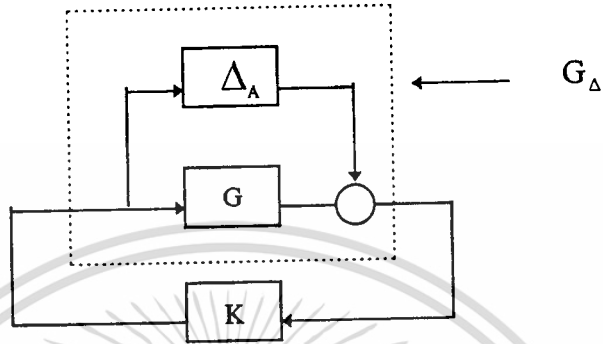
(iii) สัญญาณรบกวนแบบค่าผิดพลาดแบบแฟกเตอร์โคพราิม (perturbation coprime factor uncertainty : $[\Delta_N, \Delta_M]$)

$$G_\Delta = (\tilde{M} + \Delta_M)^{-1}(\tilde{N} + \Delta_N)$$

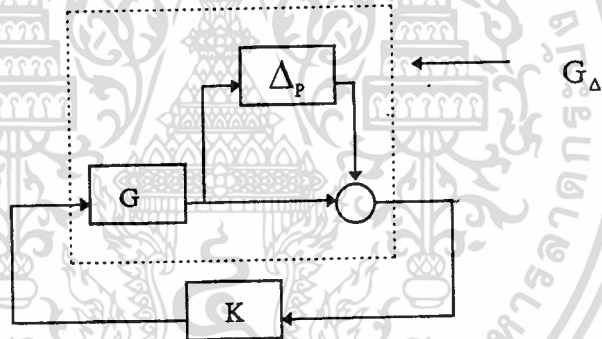
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ให้ $\tilde{N}_\Delta = \tilde{N} + \Delta_N$
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{และ } \tilde{M}_\Delta = \tilde{M} + \Delta_M$$

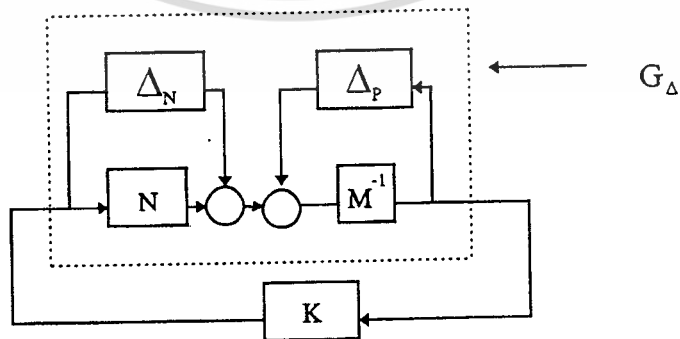
ซึ่งเป็น left coprime factorization ของ G_Δ



รูปที่ 2.1 รูปการพิจารณาระบบการรบกวนแบบค่าผิดพลาดแบบบวก
(perturbed system แบบ additive uncertainty)



รูปที่ 2.2 รูปการพิจารณาระบบการรบกวนแบบค่าผิดพลาดแบบคูณ
(perturbed system แบบ multiplicative uncertainty)



รูปที่ 2.3 รูปการพิจารณาระบบการรบกวนแบบค่าผิดพลาดแบบโคพริมาแฟกเตอร์ไรเซชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการพิจารณาการรบกวนในรูปแบบต่าง ๆ

$$G = \frac{1}{s+1}$$

$$G_{\Delta} = \frac{s-2}{(s+1)(s-3)}$$

(I) การรบกวนแบบบวก (additive perturbations)

$$\Delta_A = G_{\Delta} - G$$

$$= \frac{1}{(s+1)(s-3)}$$

(ii) การรบกวนแบบคูณ (multiplicative perturbations)

$$\Delta_P = G_{\Delta} G^{-1} - I = \frac{1}{(s-3)}$$

(iii) การรบกวนแบบแฟคเตอร์โคพราิม (coprime factor perturbations)

$$\Delta_N = \tilde{N}_{\Delta} - \tilde{N}$$

$$= \frac{(s-2)}{(s+3)(s+5)} - \frac{1}{(s+4)}$$

$$= -\frac{(6s+23)}{(s+3)(s+4)(s+5)}$$

$$\Delta_M = \tilde{M}_{\Delta} - \tilde{M}$$

$$= \frac{(s+1)(s-3)}{(s+3)(s+5)} - \frac{(s+1)}{(s+4)}$$

$$= -\frac{(7s+27)(s+1)}{(s+3)(s+4)(s+5)}$$

Δ_N , Δ_M จะเสถียรอยู่เสมอเนื่องจาก $\tilde{N}$, $\tilde{M}$, $\tilde{N}_{\Delta}$, $\tilde{M}_{\Delta}$ เสถียรตามนิยามของแฟคเตอร์โคพราิม (coprime factor)

2.4 โมเดลการรบกวน (Perturbed Model)

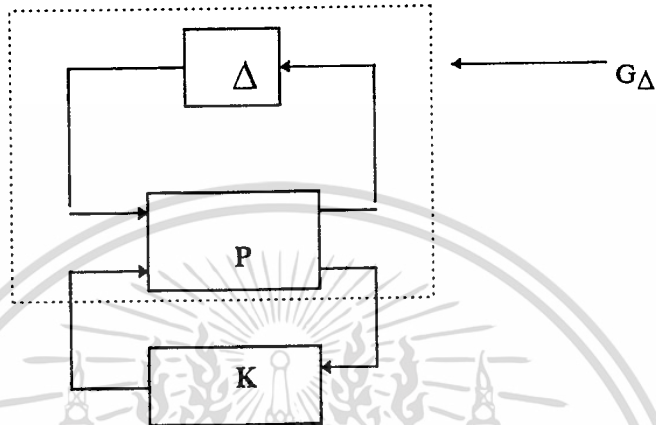
โมเดลการรบกวน G_{Δ} สามารถเขียนอยู่ในรูปของการแปลงเชิงเส้นบางส่วนแบบอัฟเฟอร์ (Upper Linear Fractional Transformation: ULFT) ได้ดังนี้

$$G_{\Delta} = F_U(P, \Delta)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P คือ แพลนท์มาตรฐาน ของ แพลนท์โนมินอล (Nominal Plant G)

Δ คือ โมเดลค่าผิดพลาดที่เหมาะสม



รูปที่ 2.4 generalized uncertainty model (with feedback controller)

(i) additive uncertainty

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & I \\ I & G \end{bmatrix} ; (\Delta = \Delta_A)$$

(ii) multiplicative uncertainty

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & G \\ I & G \end{bmatrix} ; (\Delta = \Delta_P)$$

(iii) coprime factor uncertainty

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & \vdots & I \\ \tilde{M}^{-1} & \vdots & G \\ \tilde{M}^{-1} & \vdots & G \end{bmatrix} ; (\Delta = [\Delta_N, -\Delta_M])$$

ที่ nominal model ไม่มี perturbation ดังนั้น

$$G = F_U(P, 0)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การวิเคราะห์เสถียรภาพแบบโรบัสต์ (Robust Stability Analysis)

เสถียรภาพแบบโรบัสต์ คือ ความสามารถที่จะควบคุมกระบวนการ (process) ให้เสถียรภาพอยู่ได้ จากหัวข้อข้างต้น โมเดลที่มีการรบกวน G_Δ เป็นส่วนประกอบของรูปแบบโนมินอล G และรูปแบบค่าผิดพลาด Δ ($\Delta \in D$; D คือ ระดับค่าผิดพลาดที่จะเป็นไปได้ รวมทั้ง $\Delta = 0$ ด้วย) เพราะฉะนั้น G_Δ จึงใช้แสดงแทนรูปแบบใด ๆ รอบรูปแบบโนมินอล

ในหัวข้อนี้เราจะหาเงื่อนไขที่เหมาะสม สำหรับตัวควบคุมป้อนกลับ K สำหรับเสถียรภาพแบบโรบัสต์ของ G โดยการหาการรบกวน Δ ที่ใหญ่ที่สุดที่ระบบวงปิดจะรักษาเสถียรภาพไว้ได้

2.6 การวิเคราะห์รูปแบบค่าผิดพลาดเชิงความถี่ (Frequency Domain Uncertainty Analysis)

Permissible Perturbation คือ $\Delta \in D_\epsilon$ ซึ่ง

$$D_\epsilon = D_{s\epsilon} \cup D_{u\epsilon}$$

โดย $D_{s\epsilon} = \{\Delta : \Delta \in RH_\infty ; \|\Delta\|_\infty < \epsilon\}$

$D_{u\epsilon} = \{\Delta : \Delta \in RL_\infty ; \eta(F_u(P,0)) = \eta(F_u(P,\Delta)) ; \|\Delta\|_\infty < \epsilon\}$

ซึ่ง $\eta(*)$ แทนจำนวนของ closed RHP poles ของฟังก์ชันถ่ายโอน
(RHP คือ right hand plane)

จะเห็นว่า $D_{s\epsilon}$ คือ เซตของ perturbation ขอบเขตเสถียร

$D_{u\epsilon}$ คือ เซตของ perturbation ที่ G และ G_Δ มีจำนวน closed RHP poles เท่ากัน

จะเห็นว่าในกรณี coprime factor uncertainty, perturbation นั้นเสถียร เพราะฉะนั้นทุก ๆ coprime factor perturbation จะอยู่ในเซต $D_{s\epsilon}$ จะเห็นว่า perturbed system ชนิดนี้ไม่เป็นไปตามสมการ $D_\epsilon = D_{s\epsilon} \cup D_{u\epsilon}$ เสมอไป

2.7 การพิจารณาหาค่า K ในการป้อนกลับของระบบ

K จะทำให้ $F_u(P,\Delta)$ เสถียรภาพที่ทุก ๆ Δ และ P ก็ต่อเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
(a) K ทำให้ $F_u(P,0)$ เสถียรภาพ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$(b) \quad \|F_L(P, K)\|_\infty \leq \varepsilon^{-1}$$

K ทำให้ $G_\Delta = G + \Delta_A$ เสถียรที่ทุก ๆ Δ_A ก็ต่อเมื่อ

(a) K ทำให้ G เสถียร

$$(b) \quad \|K(I - GK)^{-1}\|_\infty \leq \varepsilon^{-1}$$

K ทำให้ $G_\Delta = (I + \Delta_P)G$ เสถียรที่ทุก ๆ Δ_P ก็ต่อเมื่อ

(a) K ทำให้ G เสถียร

$$(b) \quad \|GK(I - GK)^{-1}\|_\infty \leq \varepsilon^{-1}$$

K ทำให้ $G_\Delta = (\tilde{M} + \Delta_M)^{-1}(\tilde{N} + \Delta_N)$ เสถียรที่ทุก ๆ $[\Delta_N, \Delta_M]$

ก็ต่อเมื่อ

(a) K ทำให้ G เสถียร

$$(b) \quad \left\| \frac{K(I - GK)^{-1}\tilde{M}^{-1}}{(I - GK)^{-1}\tilde{M}^{-1}} \right\|_\infty \leq \varepsilon^{-1}$$

ในการพิจารณาระบบแน่นอนนั้น จะเป็นการพิจารณาในลักษณะเดียวกันกับระบบไม่แน่นอน หากแต่เราจะไม่พิจารณาค่าเปลี่ยนแปลงของพารามิเตอร์ คือค่า Δ และการพิจารณาในเชิงความถี่นั้นจะใช้การคิดโดยใช้ coprime factorization แต่ในการพิจารณาในเชิงสมการสเทจจะใช้การคิดโดยใช้ H infinity ซึ่งจะกล่าวถึงในบทที่ 4 ซึ่งเป็นการศึกษาในขอบเขตของระบบแน่นอน

บทที่ 3

พื้นฐานที่นำไปพิจารณาแก้ปัญหา H_∞

3.1 บทเกริ่นนำ

ในการแก้ปัญหา H_∞ นั้นจำเป็นจะต้องมีพื้นฐานความรู้ทางคณิตศาสตร์ และทางการควบคุมส่วนหนึ่งนำไปพิจารณา เพื่อช่วยในการหาตัวควบคุม ตัวป้อนกลับ ซึ่งจะกล่าวถึงในบทนี้ ซึ่งได้แก่

- รูปแบบควอดราติก (quadratic form)
- สมการพีชคณิตรีคัตติ (Algebraic Riccati Equation)
- การพิจารณาหาตัวป้อนกลับแบบ LQR
- การหาตัวสังเกตได้โดยใช้ kalman filter
- การพิจารณาออฟติมัลโดยใช้ LQG

3.2 รูปแบบควอดราติก (Quadratic Forms)

รูปแบบควอดราติก เป็นรูปแบบฟังก์ชันสเกลาร์แบบหนึ่งของเวกเตอร์ x โดย

$$f(x) = x^T Q x$$

ซึ่งเราจะพิจารณา $f(x)$ โดย x เป็นเวกเตอร์ขนาด $n \times 1$

และ Q เป็นเมทริกซ์ขนาด $n \times n$ และจะต้องมีคุณสมบัติการสมมาตรของเมทริกซ์นั่นคือ สมาชิกของ Q ในลำดับที่ ij จะต้องมามีค่าเท่ากับสมาชิกในลำดับที่ ji นั่นคือ

$$Q^T = Q$$

ตัวอย่างเช่น

$$Q = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$$

นั่นคือ Q มีคุณสมบัติ

$$Q^T = Q$$

สมมติเมทริกซ์ Q_s โดยที่

$$Q_s = \frac{1}{2}(Q + Q^T)$$

และจากคุณสมบัติการสมมาตรของเมทริกซ์จะได้ว่า

เอกสารนี้เป็นเอกสารที่ส่ง Q_s^T สำรับ Q_s ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ และจากคุณสมบัติของเมทริกซ์ Q ใด ๆ จะได้ว่า

$$Q = \frac{1}{2}(Q + Q^T) + \frac{1}{2}(Q - Q^T)$$

$$= Q_s + Q_{sk}$$

เมื่อ $Q_s = \frac{1}{2}(Q + Q^T)$

$$Q_{sk} = \frac{1}{2}(Q - Q^T)$$

เราจะเรียก Q_s ว่าเมตริกซ์สมมาตร (symmetrical matrix)

และเรียก Q_{sk} ว่าเมตริกซ์สมมาตรแบบ skew (skew symmetrical matrix)

เนื่องจากพิจารณาในเชิงสมาชิกในเมตริกซ์จะได้ว่า

$$q_{skij} = \frac{1}{2}(q_{ij} - q_{ji})$$

$$= \frac{1}{2}(q_{ji} - q_{ij})$$

$$= -q_{skji}$$

และ $q_{skii} = 0$

นั่นคือสมาชิกของเมตริกซ์ Q_{sk} บนเส้นทแยงมุมจะเป็นศูนย์ทั้งหมด และสมาชิกที่มีลำดับที่ทรานสโพสกันจะมีขนาดเท่ากันแต่มีเครื่องหมายตรงกันข้ามกัน เช่น

$$Q = Q_s + Q_{sk}$$

$$\begin{bmatrix} 1 & -2 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} + \begin{bmatrix} 0 & -3 \\ 3 & 0 \end{bmatrix}$$

ดังนั้นเราสามารถพิจารณารูปแบบฟังก์ชันควอดราติกได้ว่า

$$x^T Q x = x^T [Q_s + Q_{sk}] x$$

$$= x^T Q_s x$$

ซึ่งเมตริกซ์สมมาตรแบบ skew จะเป็นศูนย์เสมอ เช่น

$$\begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} 0 & -3 \\ 3 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 0$$

ดังนั้นในการพิจารณา $f(x)$ ที่เป็นรูปแบบควอดราติกจะได้ Q ที่เป็นเมตริกซ์ที่สมมาตร

เสมอ

ต่อไปการพิจารณารูปแบบควอดราติกในเชิงอนุพันธ์จะได้ว่า

$$\frac{df(x)}{dx} = \frac{d(y^T x)}{dx} + \frac{d(x^T y_1)}{dx}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเราให้ $y = Q^T x$

ดังนั้น $y^T = x^T Q$

และ $y_1 = Qx$

และทำการอนุพันธ์ทีละส่วน จะได้ว่า

$$\begin{aligned} \frac{df(x)}{dx} &= y + y_1 \\ &= (Q^T + Q)x \end{aligned}$$

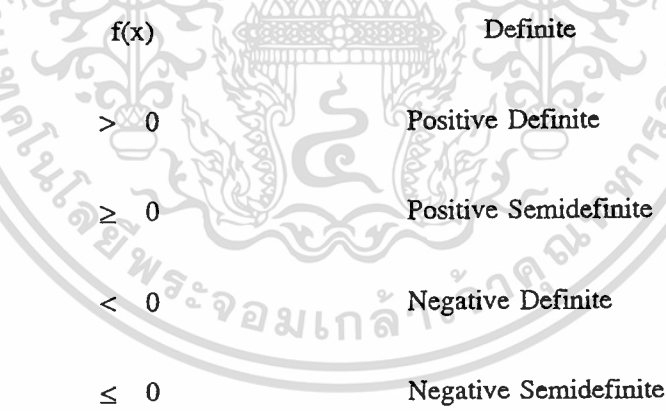
หากเป็นการพิจารณา Q ที่เป็นเมตริกซ์สมมาตรจะได้ว่า

$$\frac{df(x)}{dx} = 2Qx$$

ต่อไปเป็นการพิจารณานิยามของรูปแบบทางควอดราติก (definiteness)

ซึ่งมีอยู่ 5 รูป

แบบดังนี้



ไม่อยู่ในรูปแบบนี้

ไม่มีการนิยาม

3.3 สมการพีชคณิตรีกัตติ (Algebraic Riccati Equation)

ในการแก้ปัญหาทาง optimal control ส่วนสำคัญที่ช่วยในการหาผลเฉลย ซึ่งเป็นระบบทั้งอินพุตเดียวเอาต์พุตเดียว หรือ หลายอินพุตหลายเอาต์พุต มีหลายวิธี วิธีที่หนึ่งที่น่าสนใจมาพิจารณา คือ สมการรีกัตติ หรือ สมการพีชคณิตรีกัตติ (Algebraic Riccati Equation ; ARE) โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ปัญหามหาสมการที่อยู่ในรูปเมตริกซ์ซึ่งลักษณะของสมการที่พิจารณาจะขึ้นอยู่กับระบบว่าพิจารณาอะไรอยู่

ในการแก้ปัญหโดยใช้สมการรีคัตติวี่ จะเป็นการแก้ปัญหที่อยู่ในรูปแบบควอดราติก เช่นในการแก้ปัญหาค่าออกพิติมัล ซึ่งจะมีการสมมติเมตริกซ์ขึ้นมา 1-2 เมตริกซ์ซึ่งช่วยในการแก้ปัญห

ตัวอย่างเช่นการแก้ปัญหาคost function : J

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt$$

จะมีสมการรีคัตติวี่เขียนอยู่ในรูป

$$A^T P + P A + Q - P B R^{-1} B^T P = 0$$

ซึ่งจะมีการกำหนดเมตริกซ์ Q และ R เป็นไปตามเงื่อนไข Definiteness ในการพิจารณาแล้วแต่กรณีไป

3.4 การหาตัวป้อนกลับแบบ LQR

ในการพิจารณาระบบเชิงเส้นนั้น มีการพิจารณาอย่างหนึ่งซึ่งพิจารณาฟังก์ชันที่คงสถานะที่เรียกว่ารูปแบบทางควอดราติก และรูปแบบการควบคุม เราจะเรียกการพิจารณาฟังก์ชันนั้นว่า การพิจารณา cost function ซึ่งหากเป็นการพิจารณาระบบแบบออกพิติมัลเราจะเรียกมันว่า Linear Quadratic Regulator ซึ่งเป็นการพิจารณาในรูปแบบของสมการสเตททั่วไป

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

$$y = C x$$

เราจะหาฟังก์ชันควบคุม $u(t)$ ซึ่งจะทำให้ค่า cost function : J มีค่าต่ำสุดโดย cost function เป็นไปตามสมการ

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt$$

โดยฟังก์ชันจะอยู่ในรูปแบบควอดราติก และเมตริกซ์ Q และ R จะอยู่ในรูปเมตริกซ์สมมาตร และสมมติให้ R นิยามเป็น Positive Definite และ Q นิยามเป็น Positive Semidefinite ซึ่งจะทำให้ได้ค่า cost มีค่าที่ไม่เป็นลบ และมีค่าต่ำสุดอยู่ที่ศูนย์ และในการพิจารณาค่า cost function ต่ำสุดเมื่อ x และ u เป็นศูนย์ ซึ่งในการพิจารณาดังกล่าวจะเรียกว่า

การพิจารณา regulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการหาค่าควบคุมซึ่งจะทำให้ค่า LQR cost function ต่ำสุดจะหาได้จากเงื่อนไขของผลเฉลยแบบ Positive Definite ซึ่งสามารถหาได้จากสมการพีชคณิตรีกัตติ ซึ่งเป็นไปตามสมการ

$$A^T P + PA + Q - PBR^{-1}B^T P = 0$$

ซึ่ง R จะต้องถูกนิยามเป็น Positive Definite

$$\text{และ } Q = C_q^T C_q$$

$$\text{โดย } C_q = [I : 0]$$

และเงื่อนไข (C_q, A) จะต้องอยู่ในรูปที่สังเกตได้

จากสมการพีชคณิตรีกัตติ จะได้ว่า

$$u = -Kx$$

โดย K เป็นเกนการป้อนกลับเป็นไปตามสมการ $K = R^{-1}B^T P$

ตัวอย่างเช่น

$$\text{กำหนด } A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_q = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

และเราสมมติค่า Q และ R โดย

$$\begin{aligned} Q &= C_q^T C_q \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned}$$

$$\text{และ } R = 1$$

พิสูจน์เงื่อนไขการสังเกตได้ โดย $(C_q \ C_q A)^T$ มีเรงค์เท่ากับ 2

จากสมการพีชคณิตรีกัตติจะได้ว่า

$$A^T P + PA + Q - PBR^{-1}B^T P = 0$$

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} + \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\text{จะได้ว่า } p_2^2 = 1$$

$$p_1 = p_2 p_3$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 $2p_2 p_3 = 0$
 ไม่ว่าจะเป็นใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ว่า $P = \begin{bmatrix} \sqrt{2} & 1 \\ 1 & \sqrt{2} \end{bmatrix}$

$$K = B^{-1}R^TP$$

และ $= \begin{bmatrix} 1 & \sqrt{2} \end{bmatrix}$

ได้สมการคุณลักษณะเป็น

$$|\lambda I - A + BK| = \begin{vmatrix} \lambda & -1 \\ 1 & \lambda + \sqrt{2} \end{vmatrix}$$

$$\lambda = -\frac{\sqrt{2}}{2}(1 \pm j)$$

ซึ่งมีเสถียรภาพ และมี Damping Ratio เท่ากับ 0.707

3.5 ตัวกรองแบบคาร์มาน (Kalman Filter)

ในการพิจารณาในระบบทั่ว ๆ ไปว่ามีสัญญาณรบกวนจากภายนอกและสัญญาณรบกวนจากการวัดหรือไม่ เราจะสร้างตัวสังเกตขึ้นมาตัวหนึ่งซึ่งเราจะเรียกมันว่า kalman filter ซึ่งพิจารณาสมการสเตตดังนี้

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

$$y = Cx + v$$

โดย ω เป็นสัญญาณรบกวนทางอินพุต และ v เป็นสัญญาณรบกวนจากการวัด (sensor) ในการพิจารณา filter แบบ kalman จะทำการสร้างตัวสังเกตออฟติมัล (optimal observer) ซึ่งเป็นไปตามสมการสเตตตัวสังเกต

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

โดย $\hat{x}$ เป็นตัวสังเกตของ x และเกนการสังเกต L พิจารณาจาก

$$L = \sum C^TR^{-1}$$

และ $\sum$ เป็นไปตามสมการ

$$A\sum + \sum A^T + Q_0 + \sum C^TR_0^{-1}C\sum = 0$$

ซึ่งมีลักษณะคล้ายสมการคล้ายการหาผลเฉลยของ LQR ซึ่งเป็นไปตามสมการ

พีชคณิตรีกัตติ (Algebraic Riccati Equation)

เอกสารนี้เป็นเอกสารโดย Q_0 จะต้องเป็น Positive Definite เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ R_0 จะต้องเป็น Positive Definite เช่นกัน และยังต้องพิจารณาให้ระบบจะต้องสังเกตได้

3.6 การพิจารณาออฟติมัลโดยใช้ Linear Quadratic Gaussian (LQG)

ในการพิจารณาสมการสเตท สิ่งที่เราพิจารณาคือระบบป้อนกลับซึ่งต้องคงสถานะให้เกิดเสถียรภาพและให้ออฟติมัลที่สุด การพิจารณาแบบ LQG ก็เช่นกันเป็นการพิจารณาหาตัวควบคุมแบบ LQR รวมกับการหาตัวสังเกตแบบคาร์มาน และทำให้ระบบออฟติมัลที่สุด ซึ่งจะพิจารณาระบบ

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

$$y = Cx + v$$

และส่วนของตัวควบคุม

$$u = -K \hat{x}(t)$$

$$K = R^{-1}B^T P$$

โดยหาจากสมการรีคัตติ

$$A^T P + PA + Q - PBR^{-1}B^T P = 0$$

และตัวสังเกตหาได้จากสมการ

$$\dot{\hat{x}} = A \hat{x} + Bu + L(y - C \hat{x})$$

$$L = \sum C^T R_0^{-1}$$

จากสมการ

$$A \sum + \sum A^T + Q_0 + \sum C^T R_0^{-1} C \sum = 0$$

และเราจะหาฟังก์ชันถ่ายโอนของตัวชดเชยได้จาก

$$H(s) = K(sI - A + BK + LC)^{-1}L$$

ตัวอย่าง LQG เช่น

$$Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{และ} \quad R_0 = 1$$

จากการหาค่า $\sum$ ซึ่งให้เท่ากับ $\begin{bmatrix} a & b \\ b & c \end{bmatrix}$

จะได้ว่า $a^2 = 2b + 1$ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$b^2 = 1$$

จะได้ $\Sigma = \begin{bmatrix} \sqrt{3} & 1 \\ 1 & \sqrt{3} \end{bmatrix}$

และ $L = \Sigma C^T R_0^{-1} = \begin{bmatrix} \sqrt{3} \\ 1 \end{bmatrix}$

จะได้ฟังก์ชันถ่ายโอน

$$H(s) = \frac{3.14(s + 0.31)}{(s + 1.57 + j1.4)(s + 1.57 + j1.4)}$$

ซึ่งมีโพลวงปิดเท่ากับ $-\frac{\sqrt{2}}{2}(1 \pm j)$, $\frac{-\sqrt{3} \pm j}{2}$

เราสามารถเปรียบเทียบ LQR และ LQG ได้ดังนี้

1. การพิจารณา LQR จะมีมารจิ้นเสถียรภาพที่สูงกว่า
2. ในเกณฑ์การขยายในช่วงความถี่ต่ำ LQR จะมีคุณสมบัติในสภาวะคงตัวที่ดีกว่า
3. gain crossover ของ LQR มีค่าสูงกว่า นั่นคือ LQR มี bandwidth ที่สูงกว่า
4. ในการพิจารณาผลตอบสนองต่อสัญญาณรบกวน LQG มีคุณสมบัติตอบสนองที่ดีกว่า

บทที่ 4

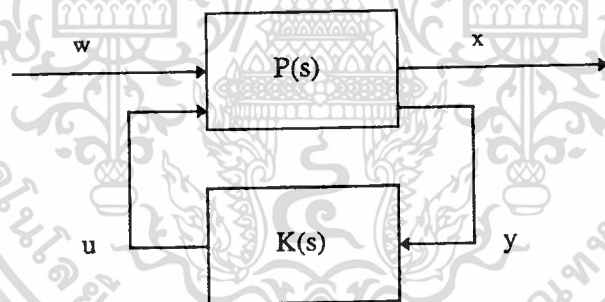
การควบคุมแบบ H_∞ infinity

4.1 บทเกริ่นนำ

✓ การออกแบบตัวควบคุมแบบ H_∞ ได้ถูกคิดค้นโดย G.Zames ในปีค.ศ. 1980 ซึ่งนำไปพัฒนาเป็นการหาตัวควบคุมแบบโรบัสต์ และในปีค.ศ. 1984 J.C.Doyle ได้คิดค้นวิธีในการหาตัวควบคุมที่เรียกว่า μ -Synthesis เป็นการพิจารณาระบบเชิงเส้นที่มีการรบกวนแบบคงที่จากภายนอก และสัญญาณที่เกิดจากอุปกรณ์การตรวจวัด (measurement noise) ซึ่งจะมีการหาค่าเกมมาเพื่อนำไปหาตัวควบคุมต่อไป ในบทนี้จะเป็นการพิจารณาการหาตัวควบคุมแบบ H_∞ ของระบบที่เป็นแบบ Certainty ซึ่งจะไม่พิจารณาการเปลี่ยนแปลงของพารามิเตอร์

4.2 การควบคุมแบบ H_∞

การควบคุมแบบ H_∞ จะทำการพิจารณาระบบแพลนที่ดังรูปที่ 3.1



รูปที่ 4.1 แสดงบล็อกไดอะแกรมของระบบ H_∞

โดย $P(s)$ เป็นส่วนของแพลนที่
 $K(s)$ เป็นส่วนของตัวควบคุม

ระบบมี 2 อินพุตอันได้แก่

- u เป็นอินพุตสัญญาณควบคุม (control input)
- w เป็นอินพุตที่ป้อนจากภายนอก (exogenous inputs)

และมี 2 เอาท์พุตได้แก่

- y เป็นสัญญาณที่ป้อนกลับไปยังตัวควบคุม (เป็นสัญญาณจากการวัด)
- z เป็นเอาท์พุตที่ถูกควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเรายังไม่ทำการต่อตัวควบคุม $K(s)$ เข้ากับระบบจะได้ว่า

$$z = P_{zw}w + P_{zu}u$$

$$y = P_{yw}w + P_{yu}u$$

และเมื่อทำการต่อตัวควบคุมพิจารณาได้ว่า

$$u = ky$$

ซึ่งจะมีฟังก์ชันถ่ายโอนวงปิดระหว่างเอาต์พุต y ที่เกิดจากการวัด กับสัญญาณเอาต์พุต ภายนอกซึ่งแสดงตามสมการ

$$y = P_{yw}w + P_{yu}u$$

$$u = ky$$

$$y = P_{yw}w + P_{yu}ky$$

$$(I - P_{yu}k)y = P_{yw}w$$

$$y = (I - P_{yu}k)^{-1} P_{yw}w$$

และจากการต่อตัวควบคุมจะได้

$$u = ky$$

$$= k(I - P_{yu}k)^{-1} P_{yw}w$$

พิจารณาเอาต์พุตที่ถูกควบคุม z จะได้ว่า

$$z = P_{zw}w + P_{zu}u$$

$$= P_{zw}w + P_{zu}k(I - P_{yu}k)^{-1} P_{yw}w$$

$$= \left(P_{zw} + P_{zu}k(I - P_{yu}k)^{-1} P_{yw} \right) w$$

หรือเขียนใหม่ได้ว่า

$$z = T_{zw}w$$

$$\text{เมื่อ } T_{zw}w = \left(P_{zw} + P_{zu}k(I - P_{yu}k)^{-1} P_{yw} \right)$$

และ $T_{zw}w$ เรียกว่า Linear Fraction Transformation (LFT)

LFT เป็นฟังก์ชันถ่ายโอนจาก w ไปยัง z คือในส่วนของแพลนท์ $P(s)$ ซึ่งคำนึงถึงสัญญาณรบกวนจากภายนอกและสัญญาณรบกวนที่เกิดจากการวัด ซึ่งเขียนเป็นรูปสมการสเตทได้ดังนี้

$$\dot{x} = Ax + B_1w + B_2u$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ $C_1x + D_{11}w + D_{12}u$ เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ย้ำทั้งห้ามีแอลกอฮอล์และแอลกอฮอล์ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



และเขียนในรูปเมตริกซ์ได้เป็น

$$P(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right]$$

และหากเราไม่พิจารณาสัญญาณรบกวนจากภายนอกและสัญญาณรบกวนจากอุปกรณ์การตรวจวัด ($w = 0$) จะเขียน $P(s)$ ได้เป็น $G(s)$ ซึ่งเป็นฟังก์ชันเมตริกซ์ระบบทั่ว ๆ ไป

4.3 เงื่อนไขในการหาค่าแกมมา

ในการพิจารณาตัวควบคุมแบบ H_∞ จะเป็นการหาค่าแกมมา หรือเรียกว่าวิธี γ -iteration เป็นการหาค่าแกมมาที่น้อยที่สุด นั่นคือเราต้องการให้สถานะของ $G(s)$ และ $K(s)$ เข้าสู่ศูนย์ในทุก ๆ ค่าเริ่มต้น และการหาตัวควบคุมนั้นจะต้องทำให้

$$\|T_{zw}\|_\infty < \gamma$$

โดยมีจะต้องทำการกำหนดเงื่อนไขดังต่อไปนี้

- 1) $H_\infty \in \text{dom}(\text{Ric})$
 $X_\infty = \text{Ric}(H_\infty) \geq 0$
- 2) $J_\infty \in \text{dom}(\text{Ric})$
 $Y_\infty = \text{Ric}(J_\infty) \geq 0$
- 3) $\rho(X_\infty Y_\infty) < \gamma^2$

โดยจะทำการหาค่า γ ที่มีค่าน้อยที่สุดที่เป็นไปตามเงื่อนไขทั้ง 3 ข้อดังกล่าว และอัลกอริทึมที่ใช้ในการหาค่า γ นี้เป็นการหาค่า γ บนฟังก์ชันความโค้งของฟังก์ชัน γ^2 ซึ่งหาผลเฉลยได้อย่างรวดเร็ว โดยจะมีการตั้งข้อสมมติฐานเพื่อใช้ในการหาดังนี้

ข้อสมมติฐาน

1. (A, B_2) มีเสถียรภาพ (stabilizable)

พิจารณาจากการควบคุมได้ของ Q

(C_2, A) ตรวจสอบได้ (detectable)

พิจารณาจากการสังเกตได้ของ Q

2. แรงค์ของ $D_{12} = m_2$ และ แรงค์ของ $D_{21} = p_2$ โดยขนาดของเวกเตอร์ต่าง ๆ กำหนดไว้

ดังนี้ ขนาดของเวกเตอร์ x เท่ากับ n , ขนาดของเวกเตอร์ w เท่ากับ m_1 , ขนาดของเวกเตอร์ u เท่ากับ m_2 , ขนาดของเวกเตอร์ z เท่ากับ p_1 และ ขนาดของเวกเตอร์ y เท่ากับ p_2

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{โดย } \hat{A} = A - B_2 D_{12}^T C_1$$

$$\hat{C} = D_{12} D_{12}^T C_1$$

$$\tilde{A} = A - B_1 D_{21}^T C_2$$

$$\tilde{B}_1 = A - B_1 D_{21}^T D_{21}$$

ทำการหาค่า X_∞ และ Y_∞ ออกมาแล้วพิจารณาเงื่อนไขของ $\rho(X_\infty Y_\infty)$ โดยนิยามค่า $\rho(X_\infty Y_\infty)$ ว่าเป็นค่าไอเก้น (eigenvalues) ที่มีค่ามากที่สุดของ $X_\infty Y_\infty$

3. พิจารณาเพิ่ม ลดค่าของ γ เพื่อให้เป็นไปตามเงื่อนไข

$$\rho(X_\infty Y_\infty) < \gamma^2$$

โดยหาค่า γ ที่ทำให้ได้ค่า $\rho(X_\infty Y_\infty)$ ใกล้เคียงกับ γ^2 ให้มากที่สุด

หากเป็นการพิจารณาหาตัวป้อนกลับและตัวสังเกตในโครงสร้างของการควบคุมแบบ H infinity จะทำการหาค่า เกณฑ์การป้อนกลับและเกณฑ์การขยาย เพื่อนำไปพิจารณาตามรูปโครงสร้างของ H infinity โดยที่

$$\begin{aligned} K_C &= \tilde{D}_{12} (B_2^T X_\infty + D_{12}^T C_1) \\ \text{เมื่อ } K_e &= (Y_\infty C_2^T + B_1 D_{21}^T) \tilde{D}_{21} \\ \tilde{D}_{12} &= (D_{12}^T D_{12})^{-1}, \quad \tilde{D}_{21} = (D_{21} D_{21}^T)^{-1} \end{aligned}$$

รูปที่ 4.2 เป็นการพิจารณาโครงสร้างของตัวควบคุมแบบ H Infinity ซึ่งเราสามารถยุบส่วนของตัวควบคุมให้เหลือเพียงตัวเดียวที่เรียกว่า K_{sup}

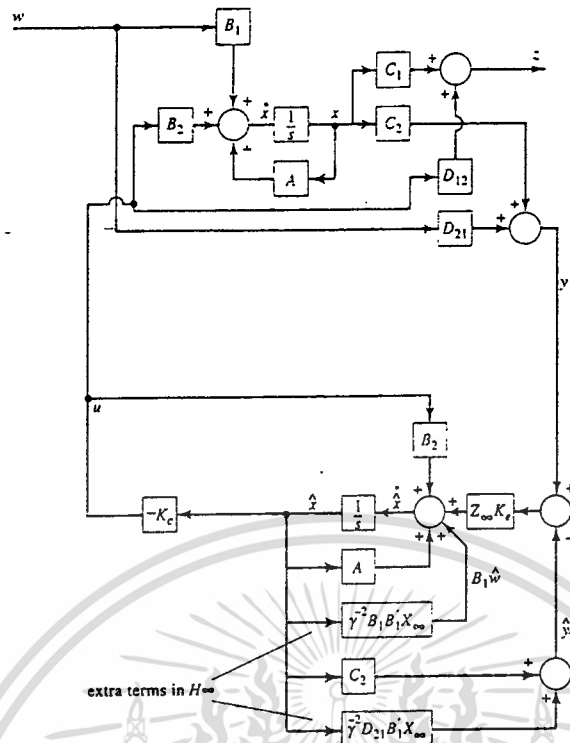
หาโดยแทนค่า X_∞ และ Y_∞ ลงในสมการ

$$\hat{A}_\infty = A + \gamma^{-2} B_1 B_1^T X_\infty + B_2 F_\infty + Z_\infty L_\infty C_2$$

$$\text{เมื่อ } F_\infty = -B_2^T X_\infty \quad L_\infty = -Y_\infty C_2^T$$

$$Z_\infty = (I - \gamma^{-2} Y_\infty X_\infty)^{-1}$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงโครงสร้างบล็อกไดอะแกรมแบบ H Infinity

4.5 โปรแกรมการหาค่าเกมมาและตัวควบคุมแบบ H infinity

4.5.1 โปรแกรม MATLAB เพื่อหาค่าเกมมาในระบบ Certain (Certainty System)

```

clc;
disp(' ');
disp(' initial gamma ');
disp(' ');
format long
gam = input('initial value of gamma : ');
ae = input('ammissible error : ');
error = gam;
while error > ae;
gam2 = 1/(gam*gam);
a1 = a-b2*d12'*c1;
c1_ = c1-d12*d12'*c1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

b1_ = b1-b1*d21'*d21;
ahx = a1;
bhx = gam2*b1*b1'-b2*b2';
chx = -c1_'*c1_;
dhx = -a1';
ahy = a2';
bhy = gam2*c1'*c1-c2'*c2;
chy = -b1_*b1_';
dhy = -a2;
%ทำการป้อนค่าสั่งเพื่อคำนวณหาค่า  $X_\infty, Y_\infty$ 
xi = are(ahx,-bhx,-chx);
yi = are(ahy,-bhy,-chy);
%คำนวณหาค่า  $\rho$ 
rho = max(real(eig(xi*yi)));
minus = abs(sqrt(rho)-gam);
gam3 = 1/(gam*gam*gam);
ax = a-b2*d12'*c1+(gam2*b1*b1'-b2*b2')*xi;
ay = a-b1*d21'*c2+yi*(gam2*c1'*c1-c2'*c2);
ahxd = ax;
bhxd = zeros(l(1),l(2));
chxd = 2*gam3*xi*b1*b1'*yi;
dhxd = -ax';
ahyd = ay';
bhyd = zeros(l(1),l(2));
chyd = 2*gam3*yi*c1'*c1*yi;
dhyd = -ay;
%คำนวณหาค่า  $\dot{X}_\infty, \dot{Y}_\infty$ 
xid = are(ahxd,-bhxd,-chxd);
yid = are(ahyd,-bhyd,-chyd);
add = (xid*yi + xi*yid);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

%คำนวณหาค่า  $\rho$ 
[s,t] = eig(xi*yi);
i=1 ;
while t(i,i) ~= rho;
i = i+1 ;
end
w = s(:,i);
[g,h] = eig((xi*yi)');
i = 1;
while h(i,i) ~= rho;
i = i+1;
end
v = g(:,i);
%คำนวณค่าผิดพลาดเพื่อเปรียบเทียบกับค่าผิดพลาดที่ตั้งไว้
q = (v*add*w)/(v*w);
q1 = q*gam-rho;
m=[1 -q q1 ];
gnew = max(roots(m));
error = abs(gam-gnew);
gam = gnew;
disp('gamma : '); disp(gam); disp('error : '); disp(error);
end;
%ทำการหาค่า  $X_\infty, Y_\infty$  จากค่าเกมมาที่โปรแกรมเลือกออกมา
gam2 = 1/(gam*gam);
a1 = a-b2*d12'*c1;
c1_ = c1-d12*d12'*c1;
a2 = a-b1*d21'*c2;
b1_ = b1-b1*d21'*d21;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ควรเผยแพร่ หักล้าง หรือแก้ไขเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chx = -c1_ '*c1_';
dhx = -a1';
ahy = a2';
bhy = gam2*c1'*c1-c2'*c2;
chy = -b1_*b1_';
dhy = -a2;
xi = are(ahx,-bhx,-chx);
yi = are(ahy,-bhy,-chy);
rho = max(real(eig(xi*yi)));
disp(' rho   ');
disp(rho);
gamma2 = gam*gam;
disp(' gamma2   ');
disp(gamma2);
disp(' ');
disp(' infimal gamma ');
disp(' ');
disp(gam)

```

4.5.2 โปรแกรม MATLAB เมื่อนำค่าเกมมาที่ได้ไปหาตัวควบคุมที่เหมาะสม

```

n = size(a);
n = n(1);
m1 = size(b1);
m1 = m1(2);
m2 = size(b2);
m2 = m2(2);
p1 = size(c1);
p1 = p1(1);p2 = size(c2);
p2 = p2(1);

```

```
gam2 = 1/(gam*gam);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if rank1 ~= n+m2,
    disp('rank1 fails: no solution'),
end
rank2 = rank([a b1;c2 d21]);
if rank2 ~= n+p2,
    disp('rank2 fails : no solution'),
end
d21_ = inv(d21*d21');
d12_ = inv(d12'*d12);
%คำนวณหาค่า gain และ compensator
kc = d12_*(b2'*xi+d12'*c1)
ke = (yi*c2'+b1*d21)*d21_
zi = inv(eye(n)-gam2*yi*xi);
ak = a-b2*kc-zi*ke*c2+gam2*(b1*b1'-zi*ke*d21*b1')*xi;
bk = zi*ke;
ck = -kc;
dk = 0*ones(m2,p2);
nsk = n;
[z_k,p_k,g_k] = ss2zp(ak,bk,ck,dk)
%พิจารณาบบวงปิด
acl = [a , -b2*kc ;
        zi*ke*c2 , a-b2*kc+b1*b1'*xi*gam2-zi*ke*(c2+d21*b1'*xi*gam2)];
bcl = [b1 ; zi*ke*d21];
ccl = [c1 -d12*kc];
clpoles = eig(acl)

```

4.6 ตัวอย่างการพิจารณาและผลการพิจารณา

จะทำการพิจารณาระบบวงปิดที่มีฟังก์ชันถ่ายโอนวงเปิดเป็น $\frac{s-2}{s^2-s-2}$ หรือมีสมการ

สเตตซึ่งเขียนในรูปของ Packed Matrix ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G(s) = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

โดย $A = \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix}$

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$D_{11} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$D_{12} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$D_{22} = \begin{bmatrix} 0 \end{bmatrix}$$

โดยป้อนค่าของ $A, B_1, B_2, C_1, C_2, D_{11}, D_{12}, D_{21}, D_{22}$ ผ่านคำสั่งทางโปรแกรม MATLAB แล้ว
สั่ง Run โปรแกรมเพื่อหาค่าแกมมา เมื่อได้ค่าแกมมาจึงสั่ง Run โปรแกรมหาตัวควบคุม

ผลการ Run โปรแกรม

initial value of gamma : 100

ammissible error : 0.01

gamma :

4.12118443491676

error :

95.87881556508324

gamma :

4.40686188143674

error :

0.28567744651999

เอกสารนี้เป็นเอกสารที่เผยแพร่ขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gamma :
4.40823435117229
```

```
error :
0.00137246973555
```

```
infimal gamma
4.40823435117229
```

```
rho =
13.50600564988085
```

```
gamma2 =
19.43253009485538
```

```
xi
```

```
xi =
0.47969697477657 -1.38964429653349
-1.38964429653349 4.47595113404214
```

```
yi
```

```
yi =
0.49749988232719 -0.70575064158265
-0.70575064158265 3.85349014301757
```

```
kc =
-0.38964429653349 4.47595113404214
```

```
ke =
-0.20825075925546
4.14773950143492
```

```
z_k =
-0.93834436508020
```

```
p_k =
1.0e+005 *
```

```
-0.00000589957426
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
-5.88404901616953
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$g_k =$

$-2.593807267235024e+006$

$clpoles =$

$1.0e+005 *$

-5.88400493400196

$-0.00001499087092 + 0.00000842173415i$

$-0.00001499087092 - 0.00000842173415i$

-0.000010000000000

จากผลการ Run โปรแกรม

เราใส่ค่าเริ่มต้นเกมมาเท่ากับ 100 และให้ค่าของความผิดพลาดที่ยอมรับได้เท่ากับ 0.01
จะได้ค่าเกมมาและค่าผิดพลาดดังนี้

ครั้งที่	gamma	error
1.	4.12118443491676	95.87881556508324
2.	4.40686188143674	0.28567744651999
3.	4.40823435117229	0.00137246973555

จะเห็นว่าเป็นไปตามเงื่อนไขที่ 3 คือ $\rho(X_\infty Y_\infty) < \gamma^2$

$13.50600564988085 < 19.43253009485538$

ทำการเลือกค่าเกมมาเท่ากับ 4.40823435117229 และได้ $K(s)$ ซึ่งเป็นตัวชดเชย

$$K(s) = \frac{-2.593807267235024e + 0.06(s + 0.93834436508020)}{(s + 0.589957426)(s + 588404.901616953)}$$

และได้ค่าของโพลวงปิดเท่ากับ

$1.0e+005 *$

-5.88400493400196

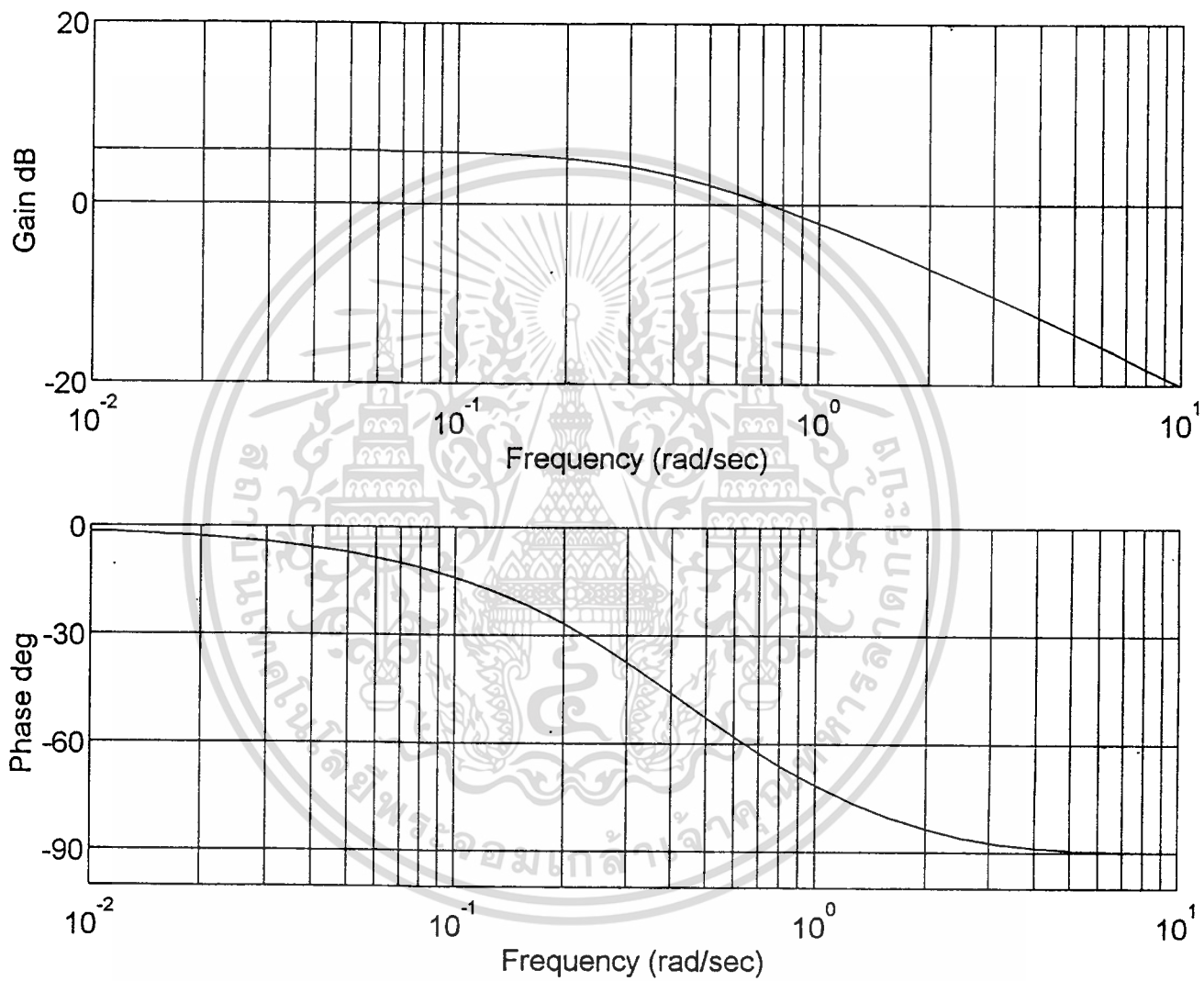
$-0.00001499087092 + 0.00000842173415i$

$-0.00001499087092 - 0.00000842173415i$

-0.000010000000000

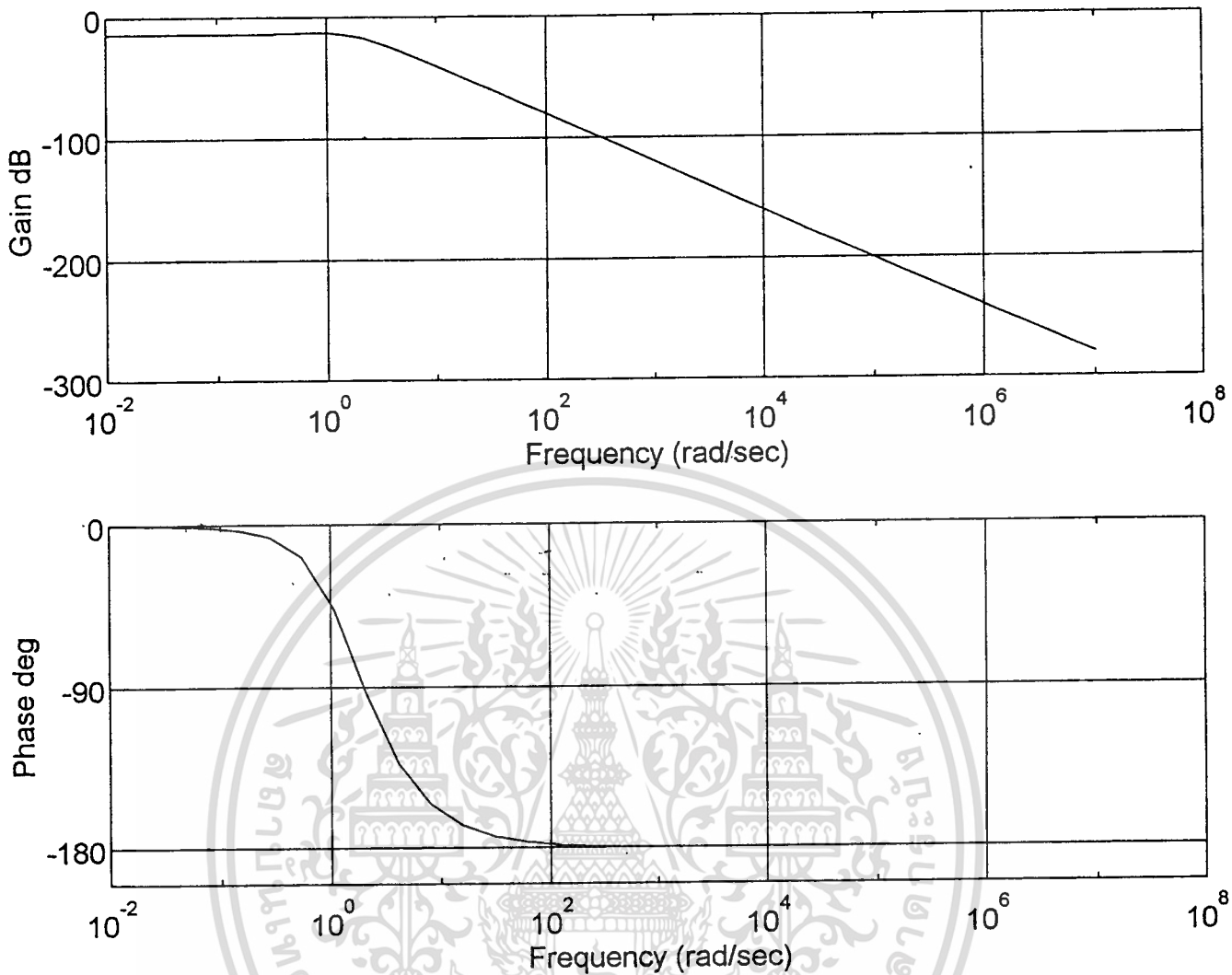
จากค่าตัวควบคุมที่ได้เราจึงพิจารณา Bode Diagram ของระบบที่ไม่มีตัวชดเชยดังในรูปที่ 4.3 และ Bode Diagram ของระบบที่มีตัวชดเชย $K(s)$ ดังในรูปที่ 4.4 จะเห็นว่าในรูปที่ 4.3 นั้น

ระบบไม่เสถียรโดยสังเกตจาก phase margin ที่เป็นลบ แต่เมื่อมีตัวชดเชยระบบจะเสถียรดังรูปที่ 4.4 สังเกตจาก phase margin ที่เป็นค่าบวก



รูปที่ 4.3 Bode Diagram ของระบบที่ยังไม่มีตัวขดเซช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Bode Diagram ของระบบที่มีตัวชดเชยแบบเอชอินฟินิตี้

4.7 สรุปผลและวิจารณ์

จากขอบเขตของโครงการเป็นการศึกษาหาตัวควบคุมป้อนกลับด้วยวิธีเอชอินฟินิตี้สำหรับระบบแน่นอน โดยการหาค่าเกมมาตามเงื่อนไข และนำค่าเกมมาที่ได้ไปพิจารณาหาตัวควบคุมป้อนกลับ $K(s)$ ที่ทำให้ระบบเสถียรแบบโรบัสต์ และได้ทำการพัฒนาโปรแกรมบนพื้นฐานซอฟต์แวร์ MATLAB เพื่อช่วยในการคำนวณ

จากโปรแกรมทั้ง 2 โปรแกรมที่ทำการหาค่าเกมมาและนำค่าเกมมาไปหาตัวควบคุมแบบป้อนกลับ ซึ่งเป็นไปตามวิธีเอชอินฟินิตี้ที่ใช้กับระบบแน่นอนนั้นสามารถยืนยันเสถียรภาพของระบบได้ และจะเป็นพื้นฐานในการหาตัวควบคุมเอชอินฟินิตี้สำหรับระบบอื่นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก A

นิยามและทฤษฎีพื้นฐานทางคณิตศาสตร์ที่นำไปใช้

A 1 นิยามของสุมัรุ่มและอินฟิมัรุ่มสำหรับเซต

(Definition of Supremum and Infimum for sets)

A 1-1 สุมัรุ่ม (Supremum)

ค่าจำนวนจริง U จะถูกเรียกว่าเป็นค่าสุมัรุ่ม (Supremum) ของเซตจำนวนจริง S ก็ต่อเมื่อ U เป็นค่าต่ำสุดของ Set ของค่าทั้งหมดในขอบเขตที่เรากำหนด

เราจะเขียนสัญลักษณ์แทนได้ว่า

$$U = \sup S$$

ซึ่งเป็นไปตามสมการ

$$\sup S = \min\{u : x \in S \Rightarrow x \leq u\}$$

หากเราพิจารณาในช่วง a, b ค่า $\sup$ ของช่วงที่พิจารณาก็คือค่าสูงสุดในช่วงก็คือค่า b หรือเขียนได้ว่า

$$\sup (a, b) = b = \max(a, b]$$

A 1-2 อินฟิมัรุ่ม (Infimum)

ค่าจำนวนจริง L จะถูกเรียกว่าเป็นค่าอินฟิมัรุ่ม (Infimum) ของเซตจำนวนจริง S ก็ต่อเมื่อ L เป็นค่าสูงสุดของ Set ของค่าทั้งหมดในขอบเขตที่เรากำหนด

เราจะเขียนสัญลักษณ์แทนได้ว่า

$$L = \inf S$$

ซึ่งเป็นไปตามสมการ

$$\inf S = \max\{l : x \in S \Rightarrow l \leq x\}$$

หากเราพิจารณาในช่วง a, b ค่า $\inf$ ของช่วงที่พิจารณาก็คือค่าต่ำสุดในช่วงก็คือค่า a หรือเขียนได้ว่า

$$\inf (a, b) = a = \min[a, b)$$

$$\sup(\infty^-, 0) = 0 = \inf(0, \infty^+)$$

A 2 Frequency Domain Spaces

เอกสารนี้เป็นเอกสารที่สําคัญในหัวข้อนี้เราจะพิจารณา frequency domain spaces ของ real, rational, matrix ค่าไม่ว่า value และฟังก์ชันถ่ายโอน อย่างย่อลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิยาม

R คือ พื้นที่ของทุกเมตริกซ์ฟังก์ชันถ่ายโอน real rational

นิยาม

L_2/H_2 norm ของ G เขียนได้เป็น

$$\|G\|_2 = \Delta = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Trace}(G^*(j\omega)G(j\omega))d\omega}$$

โดย (a) RL_2 คือพื้นที่ของทุกเมตริกซ์ฟังก์ชันถ่ายโอน real rational ที่ไม่มีโพลอยู่บนแกนจินตภาพและนอร์มของ L_2 ไม่นันต์

(b) RH_2 ทุกเมตริกซ์ฟังก์ชันถ่ายโอนใน RL_2 ที่ไม่มีโพลเป็นจำนวนจริงบวก

นิยาม

L_∞/H_∞ norm ของ G เขียนได้เป็น

$$\|G\|_\infty = \Delta = \sup_{\omega \in \mathcal{R}} \sigma[G(j\omega)]$$

โดย (a) RL_∞ คือพื้นที่ของทุกเมตริกซ์ฟังก์ชันถ่ายโอน real rational ที่ไม่มีโพลอยู่บนแกนจินตภาพและนอร์มของ L_∞ ไม่นันต์

(b) RH_∞ ทุกเมตริกซ์ฟังก์ชันถ่ายโอนใน RL_∞ ที่ไม่มีโพลเป็นจำนวนจริงบวก

เรามักจะใช้พื้นที่ RL_2 และ RH_2 ในกรณีฟังก์ชันถ่ายโอนของค่าเวกเตอร์ (vector valued) และในกรณีของสัญญาณระบบควบคุมมาตรฐาน

ทฤษฎี

ถ้า $G \in RH_\infty$ แล้ว $GRH_2 \subset RH_2$ และ

$$\|G\|_\infty = \sup\{\|G_U\|_2 : u \in RH_2, \|u\|_2 = 1\}$$

ถ้า $y = Gu$ แล้วทฤษฎีนี้จะกล่าวถึงอินพุต u ใด ๆ สำหรับเวลาที่ $t \geq 0$ ถูกกำหนดขอบเขตโดย $\|G\|_\infty$ และเรียก $\|G\|_\infty$ ว่า induced norm of the mapping

A 3 การถ่ายโอนเชิงเส้น (Linear Fractional Transformations)

มักจะใช้ใน H_∞ Optimization และในทฤษฎีการควบคุมอื่น ๆ

นิยาม

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

อีก (a) lower linear fractional transformation (LLFT) เขียนในรูปสมการได้เป็น

$$F_L \left(\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, K \right) \stackrel{\Delta}{=} P_{11} + P_{12} K (I - P_{22} K)^{-1} P_{21}$$

เมื่อ $\det(I - P_{22} K) \neq 0$

(b) upper linear fractional transformation (ULFT) เขียนในรูปสมการได้เป็น

$$F_U \left(\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \Delta \right) \stackrel{\Delta}{=} P_{22} + P_{21} \Delta (I - P_{11} \Delta)^{-1} P_{12}$$

เมื่อ $\det(I - P_{11} \Delta) \neq 0$

เราจะใช้ LLFT ในการพิจารณา H_∞ และ ULFT ในกรณีระบบ Uncertain ทั่ว ๆ

ไป

ทฤษฎีบทย่อย

LLFT ใด ๆ ที่ $\det(P_{21}) \neq 0$ และสามารถเขียนได้เป็น

$$T_U[K] = (U_{11}K + U_{12})(U_{21}K + U_{22})^{-1}$$

ที่

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} = \begin{bmatrix} P_{12} - P_{11}P_{21}^{-1}P_{22} & P_{11}P_{21}^{-1} \\ -P_{21}P_{22}^{-1} & P_{21}^{-1} \end{bmatrix}$$

ซึ่ง $\det(U_{21}K + U_{22}) \neq 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก B

โปรแกรม MATLAB เบื้องต้น

B-1 บทเกริ่นนำ

โปรแกรม MATLAB (MATrix LABoratory) เป็นโปรแกรมที่ใช้ในทางวิศวกรรมระบบควบคุมเพื่อทำการแสดงผลการ Simulate และยังทำการพล็อตกราฟในเชิงระบบควบคุม และยังมีอุปกรณ์สำหรับตัวมันเอง (ToolBox) เพื่อใช้ในการประมวลผล หาดั้วควบคุม เช่นการใช้ อุปกรณ์ ToolBox สำหรับ Robust , ToolBox สำหรับการหาดั้วควบคุมแบบ μ -Synthesis เป็นต้น

ข้อดีของโปรแกรม MATLAB คือความง่ายต่อการใช้งาน การคำนวณบางอย่าง ซึ่งหากใช้โปรแกรมภาษาสูงอื่น ๆ อาจใช้เวลานานและยุ่งยากซับซ้อน ขณะที่โปรแกรม MATLAB สามารถประมวลได้โดยคำสั่งไม่กี่บรรทัด

ในบทโปรแกรม MATLAB เบื้องต้นนี้จะเป็นการบอกถึงชุดคำสั่งพื้นฐาน, คำสั่งสายงานควบคุม (Control Flow), การแสดงผลทางกราฟ และการเขียนโปรแกรมพื้นฐาน และจะมีตัวอย่างโปรแกรมที่ใช้จริงอยู่ในภาคผนวกพร้อมทั้งคำอธิบายคำสั่งต่าง ๆ ที่ได้ใช้ในภาคผนวก ซึ่งหากต้องการค้นหาคำสั่งอื่น ๆ ก็สามารถหาอ่านได้จากหนังสือ MATLAB ทั่ว ๆ ไป

B-2 พื้นฐานโปรแกรม MATLAB

B-2.1 การพิมพ์คำสั่งบนโปรแกรม MATLAB

1. เมื่อเข้าโปรแกรม MATLAB เป็นครั้งแรกหากต้องการทราบลักษณะการทำงานของโปรแกรม MATLAB ให้ใช้คำสั่ง demo หลัง prompt >> ,โปรแกรม MATLAB จะแสดงคำสั่งต่าง ๆ พร้อมทั้งตัวอย่างที่คุณสามารถเรียนรู้ได้ด้วยตนเอง
2. ถ้าต้องการออกจากโปรแกรม MATLAB ให้พิมพ์คำสั่ง Quit หรือ Exit
3. ถ้าต้องการที่จะ save ข้อมูลก่อนจะเลิกการทำงาน โดยพิมพ์คำสั่ง save คอมพิวเตอร์จะทำการเก็บบันทึกข้อมูลต่าง ๆ โดยอัตโนมัติ ซึ่งจะเก็บไว้ในไฟล์ชื่อ MATLAB.mat
4. เมื่อออกจากเครื่องและต้องการใช้โปรแกรมอีกครั้ง สามารถเรียกไฟล์ที่ save แล้วมาใช้ได้ใหม่โดยใช้คำสั่ง load

B-2.2 หน้าต่างที่ใช้บนโปรแกรม MATLAB

โปรแกรม MATLAB จะมีหน้าต่าง 2 หน้าต่าง ได้แก่

1. หน้าต่างคำสั่ง (Command Window) ถูกใช้เพื่อป้อนคำสั่งและข้อมูลเพื่อพิมพ์ผลข้อมูล (ด้านการคำนวณ)
- ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้าคำสั่งกราฟฟิค (Graphics Window) ถูกใช้ในการพล็อตข้อมูลบนระบบแกน

B-2.3 คำสั่งพื้นฐานบางคำสั่ง

1. ถ้าต้องการที่จะเคลียร์ Command Window ในระหว่างการทำงานให้ใช้คำสั่ง “clc”
2. ถ้าต้องการที่จะเคลียร์หน้าต่างกราฟฟิค ใช้คำสั่ง “clf”
3. ถ้าต้องการที่จะเคลียร์พื้นที่ทำงาน (Workspace) ซึ่งบรรจุค่าตัวแปรต่าง ๆ ทั้งหมดใช้คำสั่ง “clear”
4. ถ้าต้องการยกเลิกคำสั่งในขณะที่คอมพิวเตอร์กำลังทำงานอยู่ ให้กด Control Key และกดอักษร “c” เพื่อยกเลิกการทำงานในโปรแกรม MATLAB
5. โปรแกรม MATLAB เป็นรูปแบบภาษาที่ไวต่อชนิดของอักขระ หมายความว่าเราสามารถที่จะกำหนดค่าหนึ่งค่าให้กับตัวแปรที่ชื่อ “HELLO” และอีกตัวแปรชื่อ “hello” ถ้าต้องการที่จะยกเลิกคุณสมบัติข้อนี้ก็สามารถใช้คำสั่ง “casesen off” และใช้ “casesen” เพื่อให้กลับไปมีคุณสมบัติเหมือนเดิม
6. คำสั่ง “who” จะใช้ในการแสดงรายการของตัวแปรที่คุณได้กำหนด
7. คำสั่ง “whos” จะใช้ในการแสดงตัวแปรที่ได้กำหนด ตามด้วยขนาดของมัน
8. คำสั่ง “size” จะถูกใช้เพื่อถามขนาดของเมตริกซ์ เช่น size(A)
9. คำสั่งของโปรแกรม MATLAB โดยปกติจะถูกป้อนแยกบรรทัด ถึงแม้ว่าหลาย ๆ Statements จะสามารถป้อนเข้าไปในบรรทัดเดียวกันได้ โดยที่ Statements เหล่านั้นถูกแยกโดย “;”
10. เครื่องหมาย “%” จะถูกใช้เพื่ออธิบายให้ผู้ที่ดูโปรแกรมอ่าน ซึ่งโปรแกรม MATLAB จะไม่ดำเนินการข้อความที่ตามด้วยเครื่องหมายนี้
11. คำสั่ง “help” เป็น facility command ที่มีความจำเป็นในโปรแกรม MATLAB เมื่อใช้คำสั่ง “help” list ของหัวข้อ help จะปรากฏขึ้น ทำให้สามารถที่จะเลือก topic ที่ต้องการรายละเอียดเพิ่มเติมได้
12. นอกจากการป้อนคำสั่งผ่านแป้นพิมพ์โดยตรง ยังสามารถเก็บชุดคำสั่งในไฟล์ที่มีนามสกุล “.m” เช่น “program1.m” หรือ “labtest.m”
13. คำสั่ง “echo” จะทำให้ M-Files แสดงคำสั่งที่ได้ทำไป
14. คำสั่ง “what” จะแสดงรายชื่อ (Directory) ที่ list ของ M-Files ที่มีอยู่ใน Directory ในแผ่นดิสก์ของคุณ
15. คำสั่ง “type” จะแสดง content ของ file name ถ้า file name ที่ถูกใช้ด้วยคำสั่งนี้จะไม่มีนามสกุล คำสั่งสมมติว่าไฟล์นั้นเป็น M-Files

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B-2.4 เมตริกซ์

โปรแกรม MATLAB ใช้พื้นฐานของเมตริกซ์ ซึ่งพิจารณาตามแถวและหลัก ดังนั้นเมื่อต้องการพิจารณาค่าใด ๆ ก็ใช้ลักษณะของเมตริกซ์มาใช้ ซึ่งมีการใช้โคออดิเนตของเมตริกซ์

ข้อสังเกต

1. ค่าภายในเมตริกซ์จะเขียนในวงเล็บใหญ่
2. เมตริกซ์โดยทั่วไปจะถูกตั้งชื่อโดยใช้อักษรตัวใหญ่
3. เมตริกซ์ขนาด 1×1 เราหมายถึงตัวเลขนั้นเป็นปริมาณสเกลาร์ ในทำนองเดียวกันเมื่อเมตริกซ์ที่มีขนาด $1 \times n$ และ $n \times 1$ หมายถึงเมตริกซ์ตามแถว (row vector) และ เวกเตอร์ตามหลัก (column vector) ตามลำดับ
4. วิธีการง่าย ๆ ในการบอกตำแหน่งสมาชิกก็คือการใช้เลขแถวและเลขหลักนั่นเอง
5. ใช้เลขแถวและเลขหลักเป็น subscript
6. เพื่อที่จะเป็นตัวแทนของเมตริกซ์ทั้งเมตริกซ์ทำได้โดยไม่มี subscript
7. อาจใช้ "[]" วงเล็บใหญ่ล้อมรอบสัญลักษณ์ที่ใช้แทนสมาชิกแต่ละตัว ซึ่งจะใช้อักษรแทนที่ตัวเลข
8. ในรูปแบบเป็นทางการ ชื่อเมตริกซ์จะเป็นอักษรตัวใหญ่ และ subscript จะใช้อักษรตัวเลข เช่น ij
9. ถ้าเมตริกซ์มี m แถว , n หลัก จะมีจำนวนสมาชิก $m \times n$ สมาชิก
10. ถ้าเมตริกซ์มีตัวเลขแถวและหลักเท่ากันจะเรียกว่าเมตริกซ์จัตุรัส (square matrix)

การกำหนดเมตริกซ์

การกำหนดเมตริกซ์สามารถแสดงได้ 4 วิธีคือ

1. Explicit Lists
2. การอ่านข้อมูลจาก data file
3. การใช้เครื่องหมายคอมม่า (colon operator)
4. การป้อนทางคีย์บอร์ด

Explicit Lists : เป็นวิธีที่ง่ายที่สุดในการกำหนดเมตริกซ์โดยการเขียนตัวเลขในวงเล็บใหญ่

หลักการ

1. ชื่อของเมตริกซ์ต้องเริ่มด้วยตัวหนังสือเสมอ และความยาวได้ 19 ตัวอักษรซึ่งอาจเป็นตัวเลขหรือตัวหนังสือก็ได้
2. ทางขวาของเครื่องหมายเท่ากับจะบรรจุข้อมูลภายในวงเล็บใหญ่ ตามลำดับของแถว

3. เซมิโคลอน จะเป็นตัวแยกแถวแต่ละแถว และค่าในแถวจะแยกโดยใช้คอมม่า หรือช่องว่าง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เฉพาะเพื่อการเรียนการสอนเท่านั้น ไม่สามารถนำเอกสารนี้ไปใช้ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ค่าแต่ละค่าจะมีเครื่องหมายบวกหรือลบอยู่ด้านหน้าก็ได้ แต่จะไม่สามารถมีคอมม่าบรรจอยู่ (ค่า 10,000 ในโปรแกรม MATLAB ต้องเขียนเป็น 10000)
5. เมื่อกำหนดเมตริกซ์ โปรแกรม MATLAB จะพิมพ์ค่าของเมตริกซ์ในบรรทัดถัดมา
6. ถ้าไม่ต้องการให้โปรแกรม MATLAB พิมพ์ค่าของเมตริกซ์ให้พิมพ์เครื่องหมายเซมิโคลอนที่ท้ายบรรทัดนั้น
7. เราจะพิมพ์เครื่องหมายเซมิโคลอนด้วยโดยทั่วไป เพื่อให้โปรแกรม MATLAB ยังไม่พิมพ์ค่าออกมา
8. ถ้ามีตัวเลขมากเกินไปในแต่ละแถวของเมตริกซ์นั้น ซึ่งยากที่จะเขียนให้จบภายในบรรทัด สามารถที่จะเขียนต่อในบรรทัดถัดไป โดยการใช้จุดสามจุดต่อท้ายเรียกว่า ellipsis
9. โปรแกรม MATLAB ยังสามารถกำหนดเมตริกซ์ขึ้นมาโดยใช้เมตริกซ์ที่ได้กำหนดไปแล้ว เช่น

$$X = [1] ;$$

$$Y = [X,2];$$

$$Y = [1 \ 2]$$

10. สามารถที่จะเปลี่ยนค่าในเมตริกซ์หรือบวกค่าเพิ่มโดยการใช้ subscript บรรจุในวงเล็บเล็ก เช่น

$$Y(2) = 3;$$

ซึ่งมีความหมายเท่ากับเปลี่ยนค่าที่สองในเมตริกซ์ Y จาก 2 เป็น 3

11. นอกจากนี้ยังสามารถขยายเมตริกซ์โดยการเพิ่มสมาชิกใหม่ โดยคำสั่ง

$$Y(3) = 4;$$

จาก Y ซึ่งแต่เดิมมีสมาชิกเพียง 2 ตัว จะมีสมาชิกตัวที่ 3. โดยมีค่าเท่ากับ 4

B-2.5 เพิ่มข้อมูล (Data Files)

เมตริกซ์สามารถที่จะถูกกำหนดจากข้อมูลที่ได้เก็บในแฟ้มข้อมูลของโปรแกรม MATLAB ซึ่งแฟ้มข้อมูล มีสองแบบที่แตกต่างกันนั่นคือ MAT-files และ ASCII files

1. MAT-files จะบรรจุข้อมูลที่ถูกเก็บในหน่วยความจำในรูปของเลขฐานสอง
2. ASCII-files จะบรรจุข้อมูลที่ถูกเก็บในหน่วยความจำในรูปของอักขระ ASCII

MAT-files จะเก็บข้อมูลได้ดีกว่าเมื่อเราต้องการที่จะใช้โดยโปรแกรม MATLAB แต่ในบางครั้ง ASCII-file ก็มีความจำเป็น ถ้าข้อมูลถูกแบ่ง (โดยการนำเข้าหรือส่งออก) ด้วย โปรแกรมอื่นนอกเหนือจาก MATLAB MAT-file ถูกเขียนขึ้นโดยใช้โปรแกรม MATLAB โดยใช้คำสั่ง "save" ซึ่งบรรจุ filename และเมตริกซ์ซึ่งถูกบรรจุในไฟล์ซึ่งสกุลจะถูกใส่ต่อท้ายชื่อไฟล์โดย

อัตโนมัติ(.mat) ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 save infor A B;
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะ save เมตริกซ์ A และ B ในไฟล์ชื่อ infor.mat

เมื่อต้องการที่จะเรียกไฟล์นี้มาใช้ใหม่อีกสามารถทำได้โดยใช้คำสั่งดังนี้

```
load infor;
```

ASCII data file ที่จะถูกใช้ด้วยโปรแกรม MATLAB จะต้องบรรจุเฉพาะข้อมูลที่เป็นตัวเลขเท่านั้น และแต่ละแถวของไฟล์ต้องมีจำนวนตัวเลขที่เหมือนกัน ไฟล์สามารถสร้างขึ้นมาโดยใช้ word processing program และนอกจากนี้ยังสามารถสร้างโดยการ run โปรแกรมที่ถูกเขียนในภาษาคอมพิวเตอร์เช่น Editor ของภาษาต่าง ๆ (Basic , C ฯลฯ) สามารถสร้างขึ้นโดยใช้โปรแกรม MATLAB ในรูปคำสั่ง save ดังนี้

```
save infor.dat x/ascii;
```

ชื่อไฟล์ในภาษา ASCII จะมีชื่อสกุลเป็น “.dat” เพื่อเป็นการง่ายที่จะแยกมันจาก MAT files และ M-files

คำสั่ง load ตามด้วยชื่อของไฟล์จะอ่านข้อมูลเข้าไปเก็บไว้ในรูปของเมตริกซ์ด้วยชื่อเดียวกันกับ data.file ตัวอย่างเช่น

```
load infor.dat;
```

B-2.6 เครื่องหมายโคลอน (Colon Operator)

สามารถใช้เพื่อสร้างเวกเตอร์จากเมตริกซ์หนึ่ง ซึ่งมีประโยชน์มากในหลาย ๆ การประยุกต์ใช้

1. ตัวปฏิบัติการโคลอนจะสร้างจำนวนจริงทั้งหมดระหว่าง 2 จำนวนที่ถูกระบุด้วยตัวอย่างเช่น การกำหนดค่าของเวกเตอร์ชื่อ A ที่บรรจุตัวเลขจาก 0 ถึง 10

```
A = 0:10;
```

2. ถ้าโคลอนถูกใช้เพื่อแยกตัวเลขตามจำนวน ตัวปฏิบัติการโคลอนก็จะสร้างค่าระหว่างตัวเลขตัวแรก กับตัวเลขตัวที่สาม การใช้ตัวปฏิบัติการโคลอนจะใช้ตัวเลขตัวที่สองในฐานะที่เป็น increment ตัวอย่างเช่น ถ้าต้องการสร้าง row vector ชื่อ “STEP” ซึ่งบรรจุตัวเลขจาก 0.0 ถึง 5.0 โดยใช้ increment เท่ากับ 1

```
STEP = 0:1:10;
```

increment ยังสามารถเป็นลบได้ด้วย ดังตัวอย่างต่อไปนี้ซึ่งจะสร้างตัวเลขจาก 5,4,3,2,1,0 ในเวกเตอร์ตามแถวชื่อ “NEG”

```
NEG = 5:-1:0;
```

3. ตัวปฏิบัติการโคลอนยังสามารถใช้เลือกเมตริกซ์ย่อยจากเมตริกซ์อื่น เช่น สมมติว่า A เป็นเมตริกซ์ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

» A = [1 2 0
        3 4 0
        5 6 0];
» A_1 = A(:,1:2)
A_1 =
     1     2
     3     4
     5     6
» A_2 = A(1:2,1:3)
A_2 =
     1     2     0
     3     4     0

```

4. ถ้าโคลนกำหนดเมตริกซ์ด้วยการใช้ subscripts ที่ไม่ถูกต้องเช่น

`A(:,3:4)` ก็จะปรากฏตัวอักษรคำว่า “error message” ขึ้น (เนื่องจากไม่มีสมาชิกในหลักที่สี่)

5. ในโปรแกรม MATLAB ขอมรับเมตริกซ์ว่าง (empty matrix) ดังต่อไปนี้

```
x = [ ] ;
```

empty matrix แตกต่างจากเมตริกซ์ที่บรรจุค่า 0 เนื่องจากเมตริกซ์ที่บรรจุค่า 0 ยังมีสมาชิกอยู่ แต่ เมตริกซ์ว่างคือเมตริกซ์ที่ไม่มีสมาชิกอยู่เลย

B-2.7 อินพุตจากผู้ใช้ (User Input)

ค่าของเมตริกซ์สามารถป้อนโดยผ่านแป้นพิมพ์โดยใช้คำสั่ง “input” ซึ่งจะแสดง text string ให้พิจารณาต่อต่อไปนี้

```
x = input('Enter x=');
```

เมื่อคำสั่งนี้ได้ถูกกระทำ text string ‘Enter x=’ ก็จะปรากฏบนจอภาพ ผู้ใช้ก็สามารถป้อนข้อมูลผ่านแป้นพิมพ์ได้

รูปแบบการพิมพ์เมตริกซ์ (Printing Matrices)

วิธีง่าย ๆ ก็คือการใส่ชื่อของเมตริกซ์ ชื่อของเมตริกซ์ จะถูกทำซ้ำและค่าของเมตริกซ์จะถูกพิมพ์ในบรรทัดถัดมา มีคำสั่งที่สามารถใช้ในการพิมพ์เมตริกซ์ซึ่งสามารถควบคุมรูปแบบ (Format) ได้มากมายดังนี้

1. เมื่อสมาชิกของเมตริกซ์หนึ่งถูกพิมพ์ เลขจำนวนจริงก็จะถูกพิมพ์เป็นเลขจำนวนจริงเสมอ
2. ค่าของทศนิยมจะถูกพิมพ์โดยใช้ default format หรือเรียกว่า short format ซึ่งแสดงเลขระยะสำคัญห้าตำแหน่ง (five significant decimal digits) เช่นเราต้องการค่าที่จะแสดงในเลขทศนิยมสิบห้าตำแหน่ง เราจะใช้คำสั่ง “format long”

3. รูปแบบสามารถที่จะกลับไปแสดงเลขทศนิยมห้าตำแหน่งได้อีกโดยใช้คำสั่ง “format short”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในโปรแกรม MATLAB ค่าในเชิงวิทยาศาสตร์จะถูกพิมพ์ด้วยอักษร “e” แยกแมนทิสซา (mantissa) และ เอกซ์โปเนนท (exponent) เช่น $1.02e+5$
 5. ถ้าต้องการให้โปรแกรม MATLAB พิมพ์ค่าเชิงวิทยาศาสตร์ด้วยเลขนัยสำคัญห้าตำแหน่ง เราต้องใช้คำสั่ง “format short e”
 6. หากมีการเว้นช่องว่างโปรแกรม MATLAB จะตีความเป็นค่าตัวเลขมากกว่าหนึ่งตัว เช่น $1.02e+5$ กับ $1.02\ e+5$
 7. รูปแบบคำสั่ง “format +” เมื่อเมตริกซ์ถูกพิมพ์ด้วยคำสั่งนี้ อักขระที่ถูกพิมพ์ก็คือเครื่องหมาย “+” และ “-” เท่านั้น ถ้าค่าเป็นบวก เครื่องหมาย “+” ก็จะถูกพิมพ์ ถ้าค่าเป็นศูนย์ที่ว่างนั้นก็จะถูกกระโดดข้าม ถ้าค่าเป็นลบ เครื่องหมาย “-” ก็จะถูกพิมพ์
- การพิมพ์ตัวอักษรหรือค่าเมตริกซ์

คำสั่ง “disp” ใช้แสดงค่าพุดที่ถูกปิดล้อมด้วยเครื่องหมายคำพูด มันสามารถใช้พิมพ์รายละเอียดของเมตริกซ์หนึ่งโดยที่ไม่ต้องพิมพ์ชื่อของเมตริกซ์ได้ด้วย ดังนั้นถ้าสเกลาร์หนึ่งชื่อ “Voltage” บรรจุค่าของแรงดันไฟฟ้าในหน่วยของโวลต์ เราก็สามารถที่จะพิมพ์ค่าบนบรรทัดเดียวตามด้วยหน่วยบนบรรทัดถัดมา โดยใช้คำสั่งดังนี้

```
disp(voltage); disp('volts')
```

ถ้าค่าของ Voltage คือ 15 ผลที่ออกมาจะเป็น

15

volts

ส่วนรูปแบบของเอาต์พุต

คำสั่ง “fprintf” จะช่วยให้สามารถควบคุม output ที่จะออกมาได้ดีกว่าคำสั่ง “disp” ยิ่งกว่านั้นการพิมพ์ทั้งข้อความและค่าของเมตริกซ์ ยังสามารถจะระบุรูปแบบในการพิมพ์ค่าออกมาได้ด้วย รูปแบบของคำสั่งนี้โดยทั่วไปคือ

```
fprintf(format,matrices)
```

รูปแบบ (format) จะระบุรูปแบบของข้อความตามที่ต้องการ

โดยปกติข้อความจะต้องอยู่ในเครื่องหมายคำพูด เพื่อจะพิมพ์ตามด้วยเมตริกซ์ที่ถูก list ภายในเครื่องหมายคำพูด ตัวกำหนด %e , %f และ %g จะถูกใช้เพื่อแสดงค่าของเมตริกซ์ที่ถูกพิมพ์

ถ้า %e ถูกใช้ค่าจะถูกพิมพ์ในรูปของเอกซ์โปเนนเชียล

%f ถูกใช้ค่าจะถูกพิมพ์ในรูปของ fixed point หรือเลขทศนิยม

%g ขึ้นอยู่กับว่า %e หรือ %f รูปแบบไหนสั้นกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า string “n” ปรากฏใน format บรรทัดที่ถูกระบุไปจนถึงจุดนั้น จะถูกพิมพ์และส่วนที่เหลือของข้อมูลจะถูกพิมพ์บนบรรทัดถัดไป format ควรจบด้วย “\n” เสมอ เพื่อว่า output จะจบบนบรรทัดเดียวในตัวเอง

ตัวอย่างของ fprintf แสดงได้ดังนี้

```
fprintf('The voltage is %f volts \n',voltage)
```

output The voltage is 15.000000 volts

ถ้าเราเปลี่ยนแปลงคำสั่งให้อยู่ในรูป

```
fprintf('The voltage is \n %f volts \n',voltage)
```

ดังนั้น output คือ

The voltage is

15.000000 volts

ตัวกำหนด %f , %e และ %g

สามารถที่จะบรรจุค่าเจาะจงจำนวนของเลขทศนิยมที่เราต้องการแสดงเช่น

```
fprintf('The voltage is %5.1f volts \n',voltage)
```

output The voltage is 15.00 volts

B-3 คำสั่งสายงานควบคุม

B-3.1 การจัดการทางความสัมพันธ์และทางตรรก

(Relational and Logical Operator)

Realational Operator

น้อยกว่า	ใช้สัญลักษณ์	<
มากกว่า	ใช้สัญลักษณ์	>
มากกว่าหรือเท่ากับ	ใช้สัญลักษณ์	>=
เท่ากับ	ใช้สัญลักษณ์	==
ไม่เท่ากับ	ใช้สัญลักษณ์	~=

Logical Operator

และ (and)	ใช้สัญลักษณ์	&
หรือ (or)	ใช้สัญลักษณ์	
ไม่ (not)	ใช้สัญลักษณ์	~

ข้อควรระวัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
- วงเล็บจะมีผลต่อลำดับของการกระทำ
ไม่ว่ากรณีใดๆ หงสน์ อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จะต้องเป็นประโยคที่สมบูรณ์เท่านั้นจึงจะใช้ logical operator ได้
- จำนวนที่ไม่เท่ากับ 0 MATLAB จะถือว่าเป็นค่าจริง

B-3.2 คำสั่ง if

จากเงื่อนไขที่เป็นไปได้ทั้งจริง(1) หรือ เท็จ(2) เราสามารถนำมาใช้ได้กับคำสั่ง if และมีรูปแบบดังนี้

```
if    เงื่อนไข (จริงหรือเท็จ)
    คำสั่ง
end
```

ถ้าเงื่อนไขเป็นจริงก็จะกระทำคำสั่ง แต่ถ้าเป็นเท็จจะกระโดดไปที่ end

คำสั่ง if ที่ซับซ้อนมากขึ้น

ในบางกรณีที่ต้องการให้มี การตัดสินใจหลายทางและมีการทดสอบเงื่อนไข จากบนลงล่างจนพบเงื่อนไขที่เป็นจริงก็จะกระทำคำสั่งนั้น สามารถทำได้โดยมีรูปแบบดังนี้

```
if    เงื่อนไข 1
    คำสั่ง 1
    if    เงื่อนไข 2
        คำสั่ง 2
    end
    คำสั่งที่ 3
end
คำสั่งที่ 4
```

B-3.3 คำสั่ง if-else

เป็นคำสั่งที่ช่วยให้สามารถกระทำชุดคำสั่งอันหนึ่งถ้าพบว่าเงื่อนไขเป็นจริงและกระทำอีกชุดคำสั่งหนึ่งถ้าเงื่อนไขเป็นเท็จมีรูปแบบดังนี้

```
if    เงื่อนไข 1
    คำสั่ง 1
else
    คำสั่ง 2
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B-3.4 คำสั่งลูป (loop)

ลูปที่กระทำคำสั่งลักษณะเป็นการวนรอบในโปรแกรม MATLAB มีอยู่ 2 ประเภทคือ for และ while ซึ่งลักษณะคล้ายกับ loop ของภาษาคอมพิวเตอร์ทั่วไป

4.3.4.1 for ช่วยให้กระทำคำสั่งซ้ำเป็นจำนวนรอบที่คงที่มีรูปแบบดังนี้

```
for    index = expression
    คำสั่ง
end
```

ข้อควรระวัง

- index ใน expression จะต้องเป็นตัวแปร
- colon สามารถนำมาสร้างเงื่อนไขกับ expression ได้โดยมีรูปแบบดังนี้

for j = ค่าเริ่มต้น:ค่าที่เพิ่มขึ้น:ขอบเขต

4.3.4.2 while ช่วยให้กระทำคำสั่งซ้ำเป็นจำนวนที่ไม่จำกัดรอบ (infinite) โดยอยู่ภายใต้การควบคุมของเงื่อนไข มีรูปแบบดังนี้

```
while expression
    คำสั่ง
end
```

ข้อคำสั่งจะถูกกระทำไปเรื่อย ๆ จนกระทั่ง expression จะเป็นเท็จ (0)

B-4 การแสดงผลทางกราฟ

B-4.1 กราฟ X-Y

มีคำสั่งในการใช้งานดังนี้

title('text')	คำสั่งนี้ใช้เขียนหัวข้อของกราฟ
xlabel('text')	คำสั่งนี้ใช้เขียนข้อความไว้ใต้แกน x
ylabel('text')	คำสั่งนี้ใช้เขียนข้อความไว้ใต้แกน y
text(x,y,'text')	คำสั่งนี้ใช้เขียนข้อความไว้ ณ จุด (x,y) ที่ต้องการ
text(x,y,'text','sc')	คำสั่งนี้ใช้เขียนข้อความไว้ ณ จุด (x,y) ที่ต้องการโดยมีจุด (0,0) อยู่ด้านล่าง ซ้าย และจุด(1,1) อยู่ด้านขวาบน
gtext('text')	คำสั่งนี้ใช้เขียนข้อความไว้ ณ จุดที่ใช้เมาส์ชี้หรือใช้ลูกศรชี้
grid	คำสั่งนี้จะเพิ่มเส้นกริดให้กับกราฟ
plot(x,y)	คำสั่งนี้ใช้พลอตค่า x และ y ลงบนกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

semilogx(x,y)	คำสั่งนี้ใช้เพื่อพลอตค่า x และ y ลงบนกราฟโดยใช้สเกลแบบ ลอการิทึมในแกน x และสเกลแบบธรรมดาในแกน y
semilogy(x,y)	คำสั่งนี้ใช้เพื่อพลอตค่า x และ y ลงบนกราฟโดยใช้สเกลแบบ ลอการิทึมในแกน y และสเกลแบบธรรมดาในแกน x
loglog(x,y)	คำสั่งนี้ใช้เพื่อพลอตค่า x และ y ลงบนกราฟโดยใช้สเกลแบบ ลอการิทึมในแกน x และแกน y

B-4.2 กราฟเชิงขั้ว (polar)

จากที่ทราบว่าจุด x,y เป็นการแสดงตำแหน่งในลักษณะของ rectangular ซึ่งสามารถ
เปลี่ยนเป็นลักษณะเชิงขั้วได้ดังนี้

$$d = \sqrt{x^2 + y^2}$$

$$\alpha = \tan^{-1}\left(\frac{y}{x}\right)$$

โดยที่ d คือขนาดและ α คือมุม

สามารถเปลี่ยนกลับเป็น rectangular โดย

$$x = d \sin \alpha$$

$$y = d \cos \alpha$$

B-4.3 กราฟแท่งและกราฟขั้นบันได

โดยแท้จริงแล้ว ทั้งกราฟแท่งและขั้นบันไดจะเหมือนกันจะต่างก็ที่เส้นในแนวตั้งที่
ตกลงมา ในแกน x จะไม่มีในกรณีของกราฟขั้นบันได การสร้างกราฟนี้สามารถทำได้โดยใช้คำสั่ง
ดังนี้

bar(y) คำสั่งนี้จะวาดกราฟแท่งของสมาชิกในเวกเตอร์ y

bar(x,y) คำสั่งนี้จะวาดกราฟแท่งของสมาชิกในเวกเตอร์ y ณ จุดที่เวก
เตอร์ x นั้นซึ่งจะต้องมีค่าความกว้างของช่วงที่เท่ากัน

stairs(y) คำสั่งนี้จะวาดกราฟขั้นบันไดของสมาชิกในเวกเตอร์ y

stairs(x,y) คำสั่งนี้จะวาดกราฟขั้นบันไดของสมาชิกในเวกเตอร์ y ณ จุดที่
เวกเตอร์ x นั้นซึ่งจะต้องมีค่าความกว้างของช่วงที่เท่ากัน

B-4.4 รายละเอียดปลีกย่อยที่เพิ่มได้ในการพลอตกราฟ

เครื่องหมายจุด ลายเส้น และสี

ในโปรแกรม MATLAB สามารถที่จะพลอตกราฟด้วยลายเส้นหรือจุดที่มีรูปร่างต่าง ๆ

เอกสารและสื่อต่าง ๆ ตามต้องการ ได้ดังตัวอย่าง เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องหมายจุด-ใช้สัญลักษณ์ เช่น จุดธรรมดา (.), บวก (+)

ลายเส้น-ใช้สัญลักษณ์ เช่น เส้นประ(-----), เส้นจุด(::::)

สี-ใช้สัญลักษณ์ เช่น สี(i), เขียว(g), แดง(r), น้ำเงิน(b), ขาว(w)

การพลอตกราฟหลายเส้น เป็นไปได้หลายวิธีดังนี้

(1) ใช้ multiple arguments ในคำสั่ง plot เช่น

`plot(X,Y,U,V)`

โดยที่ X,Y,U,V เป็นเวกเตอร์ ดังนั้นโปรแกรม MATLAB ก็จะพลอตกราฟของ X และ Y ก่อนหลังจากนั้น ก็จะพลอตกราฟของ U และ V และโปรแกรม MATLAB จะจัดการแทนกราฟทั้งสองเส้นนี้ด้วย สัญลักษณ์ที่ต่างกัน

(2) ใช้คำสั่ง “hold” (“hold on” และ “hold off”)

คำสั่งนี้จะรักษารูปกราฟที่กำลังพลอตอยู่ไว้บนจอและจะพลอตกราฟอีกอันลงบนแกนที่มีอยู่แล้ว

ตัวอย่าง `plot(X,Y,X,Y,'+')`

คำสั่งนี้จะสร้างกราฟที่เป็นเส้นหนาและมีสัญลักษณ์ + แทนจุด

`plot(X,Y,'g',X,Y,'+r')`

คำสั่งนี้จะสร้างกราฟที่เป็นเส้นสีเขียวและมีสัญลักษณ์ + สีแดงแทนจุด

การแบ่งสเกล

ถึงแม้ว่าโปรแกรม MATLAB จะแบ่งสเกลให้กราฟที่พลอตอย่างอัตโนมัติแต่บางครั้ง ถ้าต้องการเปรียบเทียบกราฟหลายเส้นมันเป็นการยากที่จะเปรียบเทียบกันถ้าสเกลของแกนไม่เหมือนกัน MATLAB จึงมีคำสั่งช่วยดังนี้

`axis` คำสั่งนี้จะทำให้กราฟที่จะพลอตตามมาใช้ ค่าของสเกลที่ใช้อยู่ตอนนี้คงอยู่ แต่ถ้ากระทำคำสั่งนี้อีกครั้งจะกลับสู่สเกลอัตโนมัติ

`axis(v)` เป็นการกำหนดค่าของแกน โดย $v=[Xmin,Xmax,Ymin,Ymax]$

`axis('square')` `axis('normal')`

คำสั่งนี้ใช้เพื่อกำหนด aspect ratio ซึ่งสามารถให้เป็น 'square' หรือ 'normal'

การแบ่งหน้าต่าง

ใช้คำสั่ง 'subplot' สามารถแบ่ง window ให้ย่อยลงไปได้เป็นหลาย ๆ subwindow มีรูปแบบของคำสั่งคือ `subplot(mnp)` โดยที่ค่า m และ n จะเป็นตัวบอกถึง graph window จะถูกแบ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น subwindows โดยมีเส้นกริดขนาด $m \times n$ และ p จะเป็นตัวบอกถึงว่ากราฟปัจจุบัน ว่าอยู่ใน subwindow ไหน เช่น

`subplot(211),plot(X,Y)`

จะแบ่ง graph window เป็น 2 ส่วน (ล่างและบน) และกราฟปัจจุบันและอยู่ข้างบน

Graph Screen Input

จะช่วยให้สามารถทราบตำแหน่ง (coordinates) ของจุดต่าง ๆ ในกราฟโดยใช้เมาส์หรือ arrow key มีรูปแบบดังนี้

`[X,Y] = ginput`

คำสั่งนี้จะช่วยให้เลือกค่าตำแหน่งจุดที่ต้องการทราบโดยใช้ mouse หรือ arrow key โดยไม่จำกัดจำนวนและเก็บค่าไว้ในเวกเตอร์ X และ Y ถ้าต้องการหยุดก็กด "return"

`[X,Y] = ginput[n]`

คำสั่งนี้จะช่วยในการเลือกตำแหน่งจุดที่ต้องการโดยใช้ mouse หรือ arrow key เป็นจำนวน n จุดและเก็บค่าไว้ในเวกเตอร์ X และ Y ถ้าต้องการหยุดก็กด "return"

การควบคุมจอ

มีการควบคุม command window และ graph window ด้วยคำสั่งดังนี้

shg	คำสั่งนี้จะแสดง graph window กดยี่ห้อใด ๆ จะกลับเข้ามาสู่ command window
clg	ลบ graph window
clc	ลบ command window
home	ให้เคอร์เซอร์ (cursor) มาอยู่ตำแหน่ง home

Graphics Hard Copy

คำสั่งนี้จะช่วยในการให้ได้มาซึ่ง hard copy แต่ก็จะเป็นเปลี่ยนแปลงตามรุ่นของคอมพิวเตอร์ที่ใช้

print	คำสั่งนี้จะพิมพ์ plot ปัจจุบันนี้ด้วย high resolution ไปยัง printer
prtsc	คำสั่งนี้จะพิมพ์ จอคอมพิวเตอร์โดยมีขนาดความละเอียดเท่ากับที่เห็นในจอ
meta file	คำสั่งนี้จะเปิด high resolution graphics metafiles โดยจะเขียนกราฟที่มีอยู่ในปัจจุบันลงในไฟล์ที่เปิดไว้ ถ้าใช้ meta เลข ๆ โดยไม่มีชื่อ filename มันก็จะเก็บกราฟไว้ใน filename ชื่อที่ตั้งไว้ก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B-4.5 กราฟ 3 มิติ

แบ่งเป็น 2 ประเภท

- (1) พื้นผิวร่างแห (mesh surface)
- (2) เส้นโครงร่าง (contour surface)

B-4.5.1 พื้นผิวร่างแห

ชุดของเมตริกซ์สามารถสร้างกราฟ 3 มิติในลักษณะพื้นผิวร่างแห ก่อนอื่นลองมาพิจารณา

สมการ $X^2 + Y^2 + Z^2 = R$

จะเป็นสมการของทรงกลมที่มีจุดศูนย์กลางที่ (0,0) และมีรัศมีเท่ากับ R

เช่นถ้ากำหนด $-0.4 \leq x \leq 0.4$ และ $-0.4 \leq y \leq 0.4$ ให้มี R เท่ากับ 1 ได้

$$Z = \sqrt{(1 - X^2 - Y^2)}$$

รวมกับคำสั่ง mesh(z) ก็จะสามารถสร้างกราฟพื้นผิวร่างแหได้ ดังนั้นจึงเขียนคำสั่งให้
พลอตในโปรแกรม MATLAB ได้ดังนี้

```
for i=1:11
    x=(i-0.5)*0.1;
    for j=1:11
        y=(j-0.5)*0.1;
        z(i,j)=sqrt(abs(1-x.^2-y.^2));
    end
end
mesh(z)
```

จะได้กราฟ 3 มิติเป็นรูปครึ่งวงกลม

mesh(Z,G)

จะได้พื้นผิวร่างแหของ Z ที่มีมุมที่กำหนดโดย G และ G = [AZ,EL] โดยที่ AZ(Azimuth) คือมุมกวาด ส่วน EL(Elevation) คือมุมยก ค่า AZ ที่เป็นบวกจะหมายถึงมุมที่ทวนเข็มนาฬิกาและ AZ ที่เป็นลบจะหมายถึงมุมที่ตามเข็มนาฬิกา ค่า EL ที่เป็นบวกจะหมายถึงมุมที่ยกขึ้น ส่วน EL ที่เป็นลบหมายถึงมุมลง

ข้อสังเกต

mesh(Z,[-37.5,-30]) จะเป็นค่า default ที่ตั้งไว้

ในบางกรณีที่ต้องการควบคุม สเกลของแกน X,Y และ Z สามารถทำได้โดยใช้คำสั่ง
mesh(Z,S) หรือ mesh(Z,S,G) โดย S คือเวกเตอร์ที่มี 3 elements คือ [SX SY SZ]
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B-4.5.2 การพล็อตเส้นโครงร่าง (Contour Plots)

เป็นอีกรูปแบบหนึ่งของการพลอตกราฟ 3 มิติคือการใช้เส้นโครงร่าง (contour) ถ้าเขียนกราฟรูป 3 มิตินั้นจะได้เส้นซึ่งจะเป็นเส้นที่บอกความสูง (elevation) ของกราฟ 3 มิติ โปรแกรม MATLAB สามารถใช้พล็อตเส้นโครงร่างของเมตริกซ์ได้ดังนี้

<code>contour(Z)</code>	จะวาดเส้นโครงร่างของเมตริกซ์ Z จำนวนเส้นที่เกิดขึ้นจะเกิดขึ้นอย่างอัตโนมัติ มุมซ้ายสุดของพลอตจะเป็นตำแหน่ง Z(1,1)
<code>contour(Z,N)</code>	จะวาดเส้นโครงร่างของเมตริกซ์ Z ที่มีจำนวนเส้นโครงร่างเท่า และ โดยทั่วไปจะมีค่า default เท่ากับ 10
<code>contour(Z,V)</code>	จะวาดเส้นโครงร่างของเมตริกซ์ Z ที่มีจำนวนเส้นโครงร่างเท่าที่บอกไว้ในเวกเตอร์ V
<code>contour(Z,N,X,Y)</code> และ <code>contour(Z,V,X,Y)</code>	จะสร้างเส้นโครงร่างของเมตริกซ์ Z และใช้ข้อมูลในเวกเตอร์ X และ Y เพื่อควบคุมสเกลบนแกน X และ Y

B-5 การวิเคราะห์สมการโพลิโนเมียล

B-5.1 การคำนวณโดยใช้สมการโพลิโนเมียล

สมการโพลิโนเมียลเป็นสมการที่มีรูปสมการมาตรฐานดังนี้

$$f(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$$

ตัวอย่างเช่น สมการ $g(x)$

$$g(x) = a_0x^2 + a_1x + a_2$$

การเรียกชื่อของโพลิโนเมียลนั้นเราจะเรียกตามกำลังสูงสุดของตัวแปร เช่นเรียก $g(x)$ ว่า สมการโพลิโนเมียลดีกรีสอง

การคำนวณ

การคำนวณเพื่อที่จะวิเคราะห์ค่าของสมการโพลิโนเมียลนั้นจริงแล้วมีอยู่หลายวิธีด้วยกัน สำหรับการใส่โปรแกรม MATLAB ตัวอย่างเช่นถ้าต้องการที่จะหาค่าของสมการโพลิโนเมียลดังต่อไปนี้

$$f(x) = 5x^4 - 0.5x^3 + x - 5$$

สามารถจะใช้คำสั่งดังต่อไปนี้

$$f = 5*x^4 - 0.5*x^3 + x - 5;$$

ถ้าตัวแปร x เป็นค่าของเวกเตอร์หรือค่าของเมตริกซ์จะต้องระบุการดำเนินการสมาชิกต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า สมาชิก (Element by Element) หรือการกระทำเชิงอาร์เรย์ เช่น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f = 5*x.^4 - 0.5*x.^3 + x - 5;$$

ซึ่งผลลัพธ์ที่ได้คือค่าของ f จะออกมาในรูปของเมตริกซ์ซึ่งมีขนาด (size) เท่ากับขนาดของ x

คำสั่ง `polyval function`

ฟังก์ชันโพลิโนเมียลสามารถถูกหาค่าได้โดยการใช้คำสั่ง `polyval function` ซึ่งรูปแบบของคำสั่งจะมีการบรรจุค่าของตัวแปร 2 ตัว (two arguments) ซึ่งตัวแปรตัวแรกจะบรรจุค่าของสัมประสิทธิ์ของโพลิโนเมียล และตัวแปรที่สองจะบรรจุค่าของตัวแปรตัวที่เราสนใจจะหาค่า ตัวอย่างต่อไปนี้เป็นตัวอย่างการใช้คำสั่ง `polyval function` ในการหาค่าของฟังก์ชันโพลิโนเมียล

$$f(x) = 5x^4 - 0.5x^3 + x - 5$$

คำสั่งในโปรแกรม MATLAB

```
a = [5 , -0.5 , 0 , 1 , -5];
```

```
f = polyval(a,x);
```

เมื่อ a คือเมตริกซ์ของสัมประสิทธิ์ของฟังก์ชันโพลิโนเมียลเรียกตามกำลังของตัวแปร x
 f คือเมตริกซ์ของฟังก์ชันโพลิโนเมียลซึ่งมีขนาดเท่ากับเมตริกซ์ x
 x คือช่วงของค่าที่ต้องการจะหาค่าของฟังก์ชัน โพลิโนเมียลซึ่งตัวแปร x เป็นได้ทั้งค่าคงที่ เวกเตอร์ใด ๆ หรือค่าของเมตริกซ์

จากตัวอย่างข้างต้นสามารถรวมคำสั่งในโปรแกรม MATLAB ให้สั้นลงได้ดังนี้

```
f = polyval([5 , -0.5 , 0 , 1 , -5],x);
```

ถ้าต้องการหาค่าของฟังก์ชันโพลิโนเมียลในช่วง $-5 \leq x \leq 5$ สามารถเขียนคำสั่งในโปรแกรม MATLAB ได้ดังนี้

```
x = -5:0.5:5;
```

```
a = [5 , -0.5 , 0 , 1 , -5];
```

```
f = polyval(a,x);
```

ซึ่งจะเห็นว่าฟังก์ชัน `polyval` นั้นจะช่วยให้การหาค่าของฟังก์ชันโพลิโนเมียลทำได้ง่ายแต่พอจะสรุปได้ตรงนี้ว่า ถ้าค่าของ x ที่ต้องการเป็นค่าของเวกเตอร์ หรือค่าคงที่นั้นสามารถใช้คำสั่ง `polyvalm.function`

คำสั่ง `polyvalm`

ฟังก์ชันโพลิโนเมียลสามารถที่จะถูกหาค่าในฐานะที่เป็นเมตริกซ์ได้ด้วย โดยใช้คำสั่ง `polyvalm` ซึ่งการใช้นี้จะไม่มีความแตกต่างใดเลยกับคำสั่ง `polyval` คือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f = \text{polyvalm}(a,x);$$

สำหรับข้อจำกัดของคำสั่ง polyvalm คือ x จะต้องเป็นเมตริกซ์จัตุรัส

การดำเนินการทางพีชคณิต (Arithmetic Operations)

การดำเนินการทางพีชคณิตของฟังก์ชันโพลิโนเมียลนั้นการดำเนินการที่คุ้นเคยและใช้บ่อยมากคือการบวก การลบ การคูณ และการหารซึ่งการดำเนินการที่โหมคนั้นสามารถกระทำได้โดยใช้โปรแกรม MATLAB ที่โหมคดังต่อไปนี้

การบวกของโพลิโนเมียล

ถ้าต้องการผลบวกของฟังก์ชันโพลิโนเมียลดังต่อไปนี้

$$f(x) = x^4 - 5x^3 - x + 2$$

$$q(x) = 2x^4 - 2x^3 + 5x - 4$$

$$h(x) = f(x) + g(x)$$

คำสั่งในโปรแกรม MATLAB สามารถเขียนสั้น ๆ ได้ดังนี้

$$f = [1, -5, 0, -1, 2];$$

$$q = [2, -2, 0, 5, -4];$$

$$h = f+g;$$

ซึ่งจะได้คำตอบเป็น $[3, -7, 0, 4, -2]$ ซึ่งหมายถึง

$$h(x) = 3x^4 - 7x^3 + 4x - 2$$

สำหรับการลบ (subtraction) นั้นการดำเนินการก็กระทำเช่นเดียวกันกับการบวก

การคูณ (multiplying) โพลิโนเมียล

สำหรับการคูณฟังก์ชันโพลิโนเมียลนั้นจะมีความซับซ้อนมากกว่า การบวกและการลบโพลิโนเมียล สำหรับการคูณฟังก์ชันโพลิโนเมียลนั้น จะมีฟังก์ชันที่ใช้เฉพาะคือ ฟังก์ชัน conv function ซึ่งมีรูปแบบดังนี้

$$f(x) = x^4 - 5x^3 - x + 2$$

$$q(x) = 2x^4 - 2x^3 + 5x - 4$$

$$h(x) = f(x).g(x)$$

คำสั่งในโปรแกรม MATLAB สามารถเขียนสั้น ๆ ได้ดังนี้

$$f = [1, -5, 0, -1, 2];$$

$$q = [2, -2, 0, 5, -4];$$

$$h = \text{conv}(f,g);$$

ซึ่งได้ค่า h เท่ากับ $[2, -12, 10, 3, -23, 16, -5, 10, -10]$

ซึ่งหมายถึง

$$h(x) = 2x^8 - 12x^7 + 10x^6 + 3x^5 - 23x^4 + 16x^3 - 5x^2 + 10x - 10$$

การหาร (dividing) โพลิโนเมียล

การหารโพลิโนเมียลนั้นจะมีฟังก์ชันที่ใช้ในการหารโดยเฉพาะเช่นเดียวกับการคูณฟังก์ชันที่ใช้ในการหารคือ deconv function

ตัวอย่างเช่น ถ้าต้องการหาค่าของผลหารของโพลิโนเมียลดังต่อไปนี้

$$f(x) = \frac{3x^8 + 4x^7 - 9x^6 + 13x^5 - x^4 + 1.5x^3 - 10.5x^2x - 5}{x^5 + 13x^4 - x^2 + 2.5}$$

ซึ่งสามารถใช้คำสั่งในโปรแกรม MATLAB ได้ดังนี้

$$a = [3,4,-9,13,-1,1.5,-10.5,1,-5];$$

$$b = [1,13,0,-1,0,2.5];$$

$$[q,r] = \text{deconv}(a,b);$$

ซึ่งจะได้สัมประสิทธิ์เวกเตอร์ของผลหารคือ $[3,-5,6,-2]$ ซึ่งจะแทนโพลิโนเมียล $3x^3 - 5x^2 + 6x - 2$ ซึ่งการหารนี้ลงตัว มีเศษของการหารเท่ากับศูนย์

ข้อระวังของการใช้ function deconv

ถ้าเศษของสัมประสิทธิ์ของโพลิโนเมียลไม่เท่ากับศูนย์ต้องใช้คำสั่ง polyval เพื่อหาค่าโพลิโนเมียลเศษและโพลิโนเมียลส่วนก่อน หลังจากนั้นเอาผลลัพธ์มาหารกันแทนที่จะใช้ polyval function เพื่อที่จะแสดงในตัวอย่างข้างล่าง ans1 และ ans2 จะมีค่าเหมือนกับเศษของเวกเตอร์ r เท่ากับศูนย์ ถ้าเศษไม่เท่ากับศูนย์แล้ว ans1 และ ans2 จะไม่ใช่ polynomial เดียวกัน

$$[q,r] = \text{deconv}(a,b);$$

$$\text{ans1} = \text{polynomial}(q,x);$$

$$\text{ans2} = \text{polynomial}(a,x)/\text{polyval}(b,x);$$

B-5.2 การหารากของสมการ (Root of Polynomials)

คำตอบของปัญหาทางด้านวิศวกรรมเกี่ยวข้องกับการหารากของสมการโพลิโนเมียลซึ่งสามารถอยู่ในรูป

$$y = f(x)$$

ซึ่งรากของโพลิโนเมียลคือค่าของ x ที่ซึ่ง y เท่ากับศูนย์

ตัวอย่างการหารากของโพลิโนเมียลคือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= (x+3)(x+1)$$

ดังนั้นถ้า $f(x)$ เท่ากับศูนย์จะได้

$$(x+3)(x+1) = 0$$

ซึ่งจะได้ $x = -3$ และ $x = -1$ สำหรับในโปรแกรม MATLAB นั้นสามารถหารากของฟังก์ชันโพลิโนเมียลได้จาก Roots function ตัวอย่างการฟังก์ชัน Roots ในการหารากของฟังก์ชันโพลิโนเมียลคือ

$$f(x) = x^2 + 4x + 3$$

เขียนคำสั่งใน MATLAB ได้ดังนี้

$$f = [1,4,3];$$

$$r = \text{roots}(f);$$

หรือสามารถเขียนคำสั่งให้กระชับและสั้นได้โดย

$$r = \text{roots}([1,4,3]);$$

ซึ่งจะได้ค่ารากเท่ากับ -3 และ -1

นอกจากนี้ยังสามารถพิสูจน์ได้ว่าค่าเหล่านี้เป็นค่ารากของโพลิโนเมียลได้โดยการใช้คำสั่งต่อไปนี้

$$\text{polyval}([1,4,3],r);$$

ซึ่งค่าของโพลิโนเมียลที่ได้จะต้องเท่ากับศูนย์ ถ้ามีค่าของรากของโพลิโนเมียล การหาค่าของสัมประสิทธิ์ของโพลิโนเมียลสามารถทำได้โดยใช้คำสั่ง poly function เช่น สัมประสิทธิ์ของโพลิโนเมียลที่มีราก -3 และ -1 จะใช้คำสั่งใน MATLAB ดังนี้

$$a = \text{poly}([-3,-1]);$$

จะได้ค่าของ a เท่ากับ [1,4,3] ซึ่งหมายถึง

$$a = x^2 + 4x + 3$$

ภาคผนวก C

อธิบายคำสั่งที่ใช้ในโปรแกรมตัวอย่างที่ใช้ในการหาค่าเกมมา

lqr

จุดประสงค์ เพื่อใช้ในการออกแบบตัวควบคุมแบบ LQR (Linear Quadratic Regulator)

วิธีการเขียน $[K, S, E] = \text{lqr}(a, b, Q, R)$

$[K, S, E] = \text{lqr}(a, b, Q, R, N)$

คำอธิบาย lqr เป็นการแก้ปัญหา LQR ที่ต่อเนื่อง มีการใช้สมการรีคัตติ

$K = \text{lqr}(a, b, Q, R)$ เป็นการคำนวณหาเมตริกซ์เกนการป้อนกลับแบบออฟติมัล K ซึ่งใช้ในกฎการป้อนกลับ

$$u = -Kx$$

และ minimize ฟังก์ชันคอสต์ (cost function) ซึ่ง ฟังก์ชันคอสต์นิยามตามสมการ

$$J = \int (x^T Q x + u^T R u) dt$$

ซึ่งพิจารณาจากสมการสเตท

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

$[K, S, E] = \text{lqr}(a, b, Q, R)$ ค่าของ S เป็นค่า Unique Positive Definite ซึ่งหาได้จก

สมการรีคัตติ

$$0 = SA + A^T S - SBR^{-1}B^T S + Q$$

และค่าของ E เป็นค่าเจาะจง (Eigenvalue) ของระบบวงปิด

$$E = \text{eig}(A - BK);$$

$[K, S, E] = \text{lqr}(a, b, Q, R, N)$ เป็นการคำนวณค่าเมตริกซ์เกนการป้อนกลับแบบ

ออฟติมัลซึ่ง minimize ค่าของฟังก์ชันคอสต์ด้วยเมตริกซ์ N (cross weighting matrix) ดังสมการ

$$J = \int (x^T Q x + 2u^T N x + u^T R u) dt$$

ข้อจำกัด

การพิจารณานี้ตั้งบนสมมติฐานการหาค่าตอบแบบ Unique Positive Definite เพื่อแก้ปัญหา LQR มีเงื่อนไขดังนี้

1. เมตริกซ์ Q จะต้องเป็นเมตริกซ์สมมาตรและเป็นเมตริกซ์แบบ Positive Semidefinite

2. เมตริกซ์ R จะต้องเป็นเมตริกซ์สมมาตรและเป็นเมตริกซ์แบบ Positive Definite

3. ระบบจะต้องสามารถควบคุมได้ พิจารณาจาก (A, B)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นการนำเอกสารนี้ไปเผยแพร่โดยไม่แสวงหาผลประโยชน์

อัลกอริทึม lqr ใช้เวกเตอร์เจาะจง (Eigenvector) ในเมตริกซ์ Hamiltonian
คำสั่งที่เกี่ยวข้อง lqe, dlqr, dlqry

disp

จุดประสงค์ เพื่อแสดง text หรือ เมตริกซ์

วิธีการเขียน disp(x)

คำอธิบาย disp(x) จะแสดงเมตริกซ์ โดยไม่ต้องพิมพ์ชื่อของเมตริกซ์ที่ต้องการให้แสดง
ถ้า x รับค่าตัวอักษร ตัวอักษรเหล่านั้นจะถูกนำไปแสดง
อีกวิธีหนึ่งที่จะแสดงเมตริกซ์บนหน้าจอ โดยการพิมพ์ชื่อเมตริกซ์ แต่การพิมพ์นี้
จะต้องเริ่มด้วย "x=" , (ซึ่งค่าของ x นั้นไม่ได้ถูกเก็บไว้)

ตัวอย่าง

```
» disp('Corn Oats Hey');
```

```
Corn Oats Hey
```

```
» disp(rand(5,3))
```

```
0.4940 0.5007 0.4644
```

```
0.2661 0.3841 0.9410
```

```
0.0907 0.2771 0.0501
```

```
0.9478 0.9138 0.7615
```

```
0.0737 0.5297 0.7702
```

เป็นการแสดงผลของทั้งข้อมูลและตัวอักษร

คำสั่งที่เกี่ยวข้อง num2str, setstr, sprintf

input

จุดประสงค์ ต้องการให้ผู้ใช้งานป้อนอินพุต

วิธีการเขียน x = input('prompt')

```
x = input('prompt','string')
```

คำอธิบาย x = input('prompt') จะแสดงตัวอักษรที่มีอยู่ในเครื่องหมาย '...' บนหน้าจอ
โดยรออินพุตจากคีย์บอร์ด และนำค่ากลับไปใส่ใน x ซึ่งค่า x ประมวลผลทางคำ
สั่งของโปรแกรม MATLAB, รวมถึงฟังก์ชันที่เขียนขึ้นใหม่ในลักษณะ M-Files
ถ้าฟังก์ชันประมวลผลได้ผลลัพธ์หลายค่า ค่าคำตอบเหล่านี้ก็มาจากการรับค่าอิน
พุต เพื่อนำไปประมวลผล

`x = input('prompt','string')` รับค่าตัวอักษรไปเก็บในลักษณะของค่า (text variable) ถ้ากดปุ่ม Return โดยไม่มีการป้อนอินพุต อินพุตจะรับค่ากลับไปเป็นเมตริกซ์ว่าง (Empty Matrix)

ตัวอย่าง เป็นโปรแกรมทดสอบเมื่อกดปุ่ม Return แล้วจะเช็คว่าเป็นเมตริกซ์ว่างหรือไม่

```
I = input('Do you want more ? Y/N [Y] :','s');
```

```
if isempty(i)
```

```
    I = 'y';
```

```
end.
```

คำสั่งที่เกี่ยวข้อง `ginput,keyboard,menu,uicontrol`

are

จุดประสงค์ เพื่อแก้ปัญหาสถาปัตยกรรมพีชคณิตรีคัตติ

วิธีการเขียน `x = are(A,B,C)`

คำอธิบาย เป็นการหาผลเฉลย Stabilizing (หากสามารถได้) จากสมการรีคัตติ

$$A^T X + XA - XBX + C = 0$$

ซึ่งสมมติให้ B เป็นเมตริกซ์สมมาตรและเป็นเมตริกซ์ nonnegative และ C เป็นเมตริกซ์สมมาตร

คำสั่งที่เกี่ยวข้อง `tic`

max

จุดประสงค์ เพื่อหาค่าที่มากที่สุดในเมตริกซ์

วิธีการเขียน `y = max(x)`

```
[ y,i ] = max(x)
```

```
C = max(A,B)
```

คำอธิบาย สำหรับเวกเตอร์, ค่าของ `max(x)` ก็คือสมาชิกที่มีขนาดใหญ่ที่สุดในเวกเตอร์ x สำหรับเมตริกซ์, ค่าของ `max(x)` จะเป็นเวกเตอร์ ตามแถว ซึ่งรับค่าที่มากที่สุดในแต่ละหลัก และ `max(max(x))` จะเป็นสมาชิกที่มีค่าที่มากที่สุดในเมตริกซ์ `[ y,i ] = max(x)` จะเป็นการเก็บดัชนีของค่าที่มากที่สุดในเวกเตอร์ I ถ้าหากมีค่ามากที่สุดหลายค่า ดัชนีจะเก็บค่าแรกที่พบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$C = \max(A,B)$ จะเป็นเมตริกซ์ขนาดเดียวกันกับเมตริกซ์ A และ B โดยรับค่าที่มากที่สุดแล้วระหว่างเมตริกซ์ A และ B โดยพิจารณาสมาชิกทีละตัว สำหรับค่าอินพุตที่เป็นค่าเชิงซ้อน ค่าสูงสุดจะพิจารณาจากค่าเชิงซ้อนที่มีค่าโมดูลัสสูงสุด โดยคำนวณจาก $\max(\text{abs}(x))$ ถ้าสมาชิกของเมตริกซ์เป็นค่า NaN ค่า $\max$ จะรับค่า NaN

ตัวอย่าง

พิจารณา » $A = \text{magic}(3)$

$A =$

```
8 1 6
3 5 7
4 9 2
```

ค่า $\max(A)$ จะเป็นเวกเตอร์

» $\max(A)$

ans =

```
8 9 7
```

และค่า $\max(\max(A))$ จะเป็นค่าสเกลาร์

» $\max(\max(A))$

ans =

```
9
```

นอกจากนี้ $[m,i] = \max(A)$ จะได้

» $[m,i] = \max(A)$

m =

```
8 9 7
```

i =

```
1 3 2
```

คำสั่งที่เกี่ยวข้อง `isnan,min,sort`

real

จุดประสงค์ เพื่อพิจารณาหาส่วนที่เป็นจำนวนจริง

วิธีการเขียน $x = \text{real}(z)$

คำอธิบาย $\text{real}(z)$ เป็นค่าส่วนจริงของสมาชิกใน z

ตัวอย่าง $\text{real}(2 + 3*j)$

คำสั่งที่เกี่ยวข้อง `conj,imag`

eig

จุดประสงค์ เพื่อหาค่าเฉพาะ (Eigenvalue) และเวกเตอร์เฉพาะ (Eigenvector)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเขียน $d = \text{eig}(A)$

$$[V,D] = \text{eig}(A)$$

$$[V,D] = \text{eig}(A, 'nobalance')$$

$$d = \text{eig}(A,B)$$

$$[V,D] = \text{eig}(A,B)$$

คำจำกัดความ การหาค่าเฉพาะจากการหาผลเฉลยของสมการ

$$Ax = sx$$

โดยที่ A มีขนาดเมตริกซ์ $n \times n$, x เป็นเวกเตอร์ตามหลักขนาด n และ s เป็นค่าสเกลาร์

ค่าของ s จำนวน n ค่าซึ่งหาจากสมการจะเป็นค่าที่เรียกว่าค่าเฉพาะ

ในโปรแกรม MATLAB ฟังก์ชัน eig ใช้ในการหาค่าเฉพาะ s และเวกเตอร์เฉพาะ x โดยที่

$$- d = \text{eig}(A)$$

เป็นการรับค่าเฉพาะของ A ไว้ในเวกเตอร์ d

$$- [V,D] = \text{eig}(A)$$

จะสร้างเมตริกซ์เส้นทแยงมุม D ของค่าไอเกน และเมตริกซ์ V ซึ่งคอลัมน์

ประกอบด้วยเวกเตอร์เฉพาะ หาจากสมการ

$$A*V = V*D$$

ซึ่งค่านอร์มของเวกเตอร์เฉพาะมีค่าเท่ากับ 1.00

การหา left eigenvectors สามารถหาค่าสั่ง 2 คำสั่งต่อไปนี้

$$[W,D] = \text{eig}(A')$$

$$W = W'$$

เวกเตอร์เฉพาะใน D คำนวณจาก A และ A' เช่นเดียวกันซึ่งจะต่างจาก V ซึ่ง D จะพิจารณาจากสมการ

$$W*A = D*W$$

$$- [V,D] = \text{eig}(A, 'nobalance')$$

เป็นการหาค่าเฉพาะและเวกเตอร์เฉพาะโดยปราศจากขั้นตอนที่เรียกว่าการ

balance ซึ่งการทำการ balance ในเมตริกซ์ขนาดเล็กจะทำให้เกิดค่าผิดพลาดได้

ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

$$- d = \text{eig}(A,B)$$

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า A และ B เป็นเมทริกซ์จัตุรัสซึ่งจะได้เวกเตอร์เฉพาะที่เรียกว่า *generalized eigenvalues*

- $[V,D] = \text{eig}(A,B)$

จะได้เมทริกซ์เส้นทแยงมุม D ของ *generalized eigenvalues* และเมทริกซ์ V เป็นเมทริกซ์ซึ่งหาได้จากสมการ

$$A \cdot V = B \cdot V \cdot D$$

คำสั่งที่เกี่ยวข้อง `balance, hess, qz, schur`

abs

จุดประสงค์ ทำการแปลงค่าสัมบูรณ์และค่าสตริงในเชิงตัวเลข

วิธีการเขียน $y = \text{abs}(x)$

คำอธิบาย $\text{abs}(x)$ คือค่าสัมบูรณ์ของทุก ๆ สมาชิกของ x

ถ้า x เป็นจำนวนเชิงซ้อน, $\text{abs}(x)$ คือค่าโมดูลัสเชิงซ้อน

$$\text{abs}(x) = \sqrt{\text{real}(x)^2 + \text{img}(x)^2}$$

ถ้า x เป็นค่าสตริง (MATLAB String), $\text{abs}(x)$ คือค่าตัวเลขของรหัสตัวอักษรแอสกี (ASCII) ในค่าสตริง

ตัวอย่าง

» `abs(-5)`

`ans =`
`5`

» `abs(3+4i)`

`ans =`
`5`

» `abs('3 + 4i')`

`ans =`

`51 32 43 32 52 105`

(ค่าแอสกีของ '3' คือ 51 , ค่าแอสกีของ ' ' คือ 32 , ค่าแอสกีของ '+' คือ 43 , ค่าแอสกีของ '4' คือ 52 , ค่าแอสกีของ 'i' คือ 105)

คำสั่งที่เกี่ยวข้อง `angle, setstr, sign, strings, unwrap`

sqrt

จุดประสงค์ เพื่อหาค่ารากที่สอง

วิธีการเขียน $y = \text{sqrt}(x)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบาย `sqrt(x)` เป็นการหาค่ารากที่สองของสมาชิกของ `x` สำหรับสมาชิกของ `x` ที่เป็นค่าลบหรือค่าเชิงซ้อน, `sqrt(x)` จะพิจารณาผลในเชิงซ้อน
(ถ้าเป็นการหาค่ารากที่สองเมตริกซ์ทั้งเมตริกซ์ จะใช้คำสั่ง `sqrtm(A)`)

ตัวอย่าง » `sqrt((-2 : 2)')`
`ans =`
`0 + 1.4142i`
`0 + 1.0000i`
`0`
`1.0000`
`1.4142`

คำสั่งที่เกี่ยวข้อง `exp`, `log`, `sqrtm`

zeros

จุดประสงค์ เพื่อทำเมตริกซ์ศูนย์

วิธีการเขียน `y = zeros(m,n)`
`y = zeros(size(A))`

คำอธิบาย `y = zeros(n)`
 คือการสร้างเมตริกซ์ `y` ซึ่งเป็นเมตริกซ์ศูนย์ขนาด $n \times n$
`y = zeros(m,n)`
 คือการสร้างเมตริกซ์ `y` ซึ่งเป็นเมตริกซ์ศูนย์ขนาด $m \times n$
`y = zeros(size(A))`
 คือการสร้างเมตริกซ์ `y` ซึ่งเป็นเมตริกซ์ศูนย์ขนาดเท่ากับเมตริกซ์ `A`

ตัวอย่าง เมื่อ `n = 1000`, พิจารณา for loop
`for j = 1 : n,`
`x(j) = 0`
`end;`
 เป็นการสร้างเมตริกซ์ศูนย์ขนาด $1 \times n$ (ในที่นี้ใช้ `n = 1000`) อาจใช้แทนโดยคำสั่ง
`x = zeros(1,n)` ซึ่งจะใช้เวลาในการประมวลผลเร็วกว่า

คำสั่งที่เกี่ยวข้อง `eye`, `ones`, `rand`, `random`

roots

จุดประสงค์ เพื่อหาค่ารากของสมการโพลิโนเมียล

วิธีการเขียน $r = \text{roots}(p)$

สัมประสิทธิ์โพลิโนเมียลของ s กำลังต่าง ๆ หากเวกเตอร์ c ประกอบด้วย $n+1$ ลำดับ ซึ่งเขียนแทนด้วยโพลิโนเมียล

$$c_1 s^n + \dots + c_n s + c_{n+1}$$

ตัวอย่าง พิจารณาโพลิโนเมียล

$$s^3 - 6s^2 - 72s - 27$$

ซึ่งเขียนใน MATLAB ได้เป็น

$p = [1 \ -6 \ -72 \ -27];$

$r = \text{roots}(p)$

$r =$

12.1229

-5.7345

-0.3884

คำสั่งที่เกี่ยวข้อง conv, fzero, poly, polyval, residue

เอกสารอ้างอิง

1. จงกล งามวิวิทย์ รศ.ดร. , สุเมธ เขาวนัไว ., การออกแบบตัวควบคุมแบบ H_∞ สำหรับระบบเชิงเส้น , วิศวสาร ลาดกระบัง ., Vol.12 , No.1 , หน้า 49-52 , มิถุนายน 2538
2. Duan C. McFarlane, Keith Glover ., Robust Controller Design using NCF Plant Descriptions. Springer-Verlag : Berlin , Heidelberg , 1990.
3. Frank L. Lewis ., Optimal Control. John Wiley & Son : USA, 1986.
4. Bahram Shahian, Michael Hassul., Control System Design Using MATLAB. Prentice-Hall : USA, 1993
5. Raymond T. Stetani , Clement J. Savan Jr. , Bahram Shahian , Gene H. Hostetter ., Design of Feedback Control System. Saunders College Publishing : USA, 1982.
6. The Math Works Inc., The Student Edition of MATLAB. Prentice-Hall : USA, 1995
7. John C. Doyle, Keith Glover P. Khargonekar, Bruce A. Francis, "State Space Solutions to Standard and H_2 and H_∞ Control," IEEE Trans.Automat.Contr.; Vol.34, No.8, pp.838-847,Aug 1989
8. Xie, M. Fu, and C.E. de Souza, " H_∞ Control and Quadratic Stabilization of Systems with Parameter Uncertainty Via Output Feedback," IEEE Trans.Automat.Contr., Vol.37, No.8, pp. 1253-1526, Aug 1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้