

บทที่ 3 ผลการทดลอง

3.1. ผลการทดลองจากการออกแบบตัวควบคุมแบบเรียนรู้ที่มีการย้อนกลับไปสู่ภาวะเริ่มต้นโดยวิธีการปรับค่าเกณฑ์ให้เกิดการเรียนรู้เร็วที่สุด

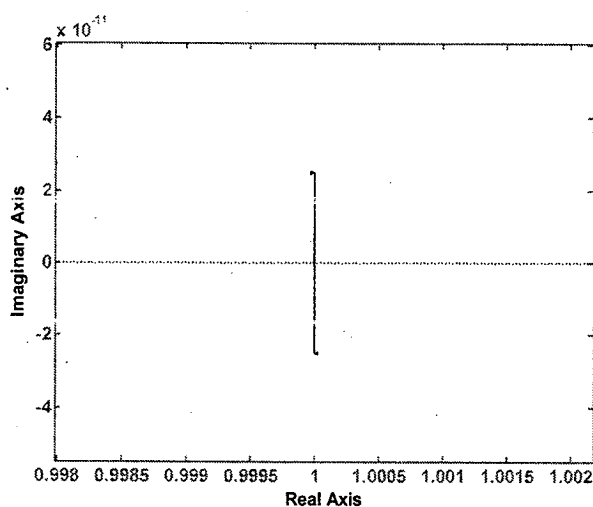
ในบทนี้จะเป็นการกล่าวถึงผลการทดลองที่ได้จากการออกแบบตัวควบคุมแบบเรียนรู้ที่มีการย้อนกลับไปสู่ภาวะเริ่มต้นโดยวิธีการปรับค่าเกณฑ์ให้เกิดการเรียนรู้เร็วที่สุด ดังที่กล่าวมาในหัวข้อ 2.5 โดยตัวควบคุม ILC ที่ออกแบบนั้นจะออกแบบโดยใช้ optimization technique เข้ามาช่วยดังอธิบายในหัวข้อ 2.4 และจะมีการศึกษาถึงค่า sensitivity ที่มีผลต่อการปรับเกณฑ์ของตัวควบคุมและพารามิเตอร์ต่างๆเช่นค่า gain size weight v และ cutoff number (c)

3.1.1. ผลการออกแบบ Repetitive Control Law แบบ Optimization

ก่อนที่จะออกแบบตัวควบคุมแบบเรียนรู้ที่มีการย้อนกลับไปสู่ภาวะเริ่มต้นโดยใช้ optimization technique นั้น เราจำเป็นต้องออกแบบตัวควบคุมแบบเรียนรู้ที่ทำแบบต่อเนื่องโดย optimization ก่อนดังอธิบายในหัวข้อ 2.4.1 โดยในหัวข้อนี้จะกล่าวถึงผลของการออกแบบตัวควบคุม RC ที่มีการเลือกใช้ $n=51$ และ $m=27$ และมีการตั้งค่า Sampling time เท่ากับ $1/100$ วินาที และ $1/50$ วินาที ตามลำดับ โดยมีขนาดของ gain size weight v ค่าต่างๆ เราสามารถทำการพิจารณา stability ของระบบได้จาก Polar plot ดังแสดง

3.1.1.1. กรณี $v=0$ และ sampling time เท่ากับ $1/100$ วินาที

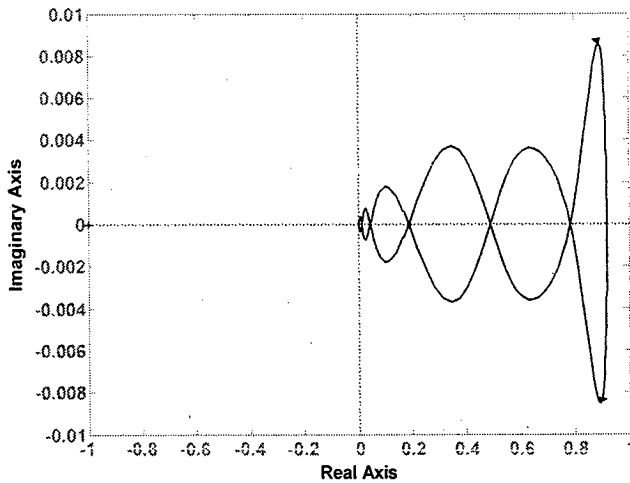
จากรูปจะเห็นว่าขนาดของพล็อตมีขนาดเล็กมากคล้ายจุดจุดหนึ่งที่ตำแหน่ง $(1,0)$ ซึ่งจากการพิจารณา stability ของระบบพบว่าระบบมีความเสถียรภาพมากเนื่องจากมีขนาดเล็กมากนั่นเอง



รูปที่ 3.1 Polar plot ของ $G(e^{i\omega T})F(e^{i\omega T})$ เมื่อ $v=0$ และ sampling time เท่ากับ $1/100$ วินาที

3.1.1.2. กรณี $v = 0.1$ และ sampling time เท่ากับ $1/100$ วินาที

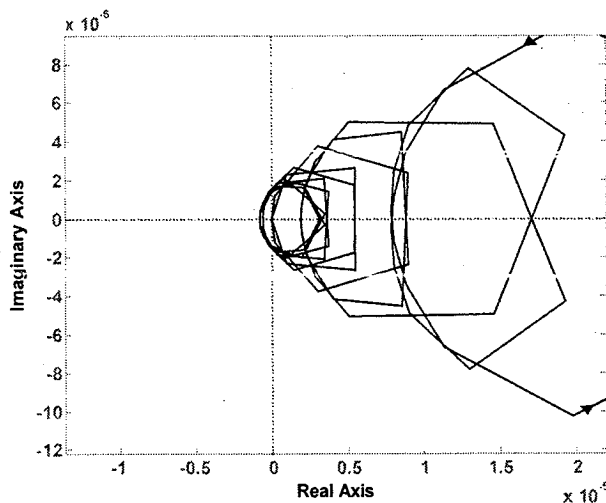
เมื่อพิจารณาพล็อตที่เกิดขึ้นจะเห็นได้ว่าขนาดของพล็อตมีค่าไม่เกิน 1 และลักษณะของพล็อตยังคงอยู่ทางด้านขวามือของแกน Real axis ดังนั้นเราจะสรุปได้ว่าที่ค่าความถี่ต่างๆระบบจะมีการเรียนรู้ได้รวดเร็วมากเนื่องจากขนาดของพล็อตจากจุด $(1,0)$ มีค่าน้อย แต่เมื่อความถี่สูงขึ้นเรื่อยๆ ขนาดของพล็อตจะมีค่ามากขึ้นเข้าใกล้ 1 ซึ่งทำให้ระบบเรียนรู้ได้ช้าลง ทั้งนี้ระบบ Repetitive control ก็ยังคง stable อยู่เนื่องจากพล็อตไม่มีส่วนของ negative real part อยู่เลย และยังอยู่ภายในวงกลม unit circle ที่มีจุดศูนย์กลางอยู่ที่จุด $(1,0)$ อีกด้วย



รูปที่ 3.2 Polar plot ของ $G(e^{i\omega T})F(e^{i\omega T})$ เมื่อ $v = 0.1$ และ sampling time เท่ากับ $1/100$ วินาที

3.1.1.3. กรณี $v = 0.2$ และ sampling time เท่ากับ $1/100$ วินาที

เมื่อเราทำการเพิ่มขนาด gain size weight v ขึ้น จากพล็อตจะเห็นได้ว่าพล็อตมีค่า negative real part อยู่ และอยู่ภายนอกวงกลมหนึ่งหน่วยที่มีจุดศูนย์กลางอยู่ที่จุด $(1,0)$ ทำให้เราทราบทันทีว่าระบบ Repetitive control จะ unstable ดังนั้นการที่เราเพิ่มขนาด v ขึ้นเรื่อยๆ เราควรต้องระมัดระวังเรื่องของ stability ของระบบด้วย

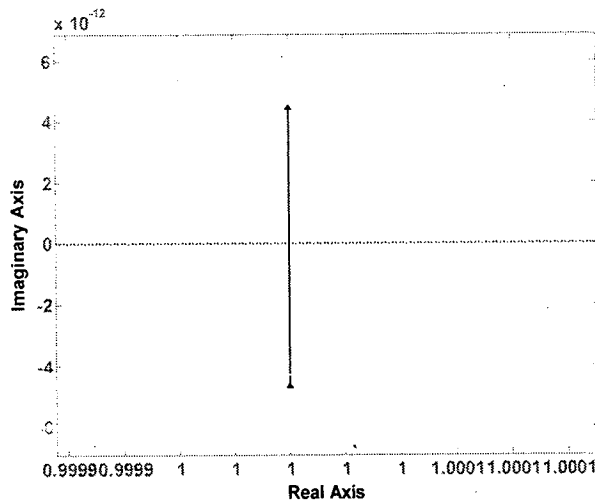


รูปที่ 3.3 Polar plot ของ $G(e^{i\omega T})F(e^{i\omega T})$ เมื่อ $v = 0.2$ และ sampling time เท่ากับ $1/100$ วินาที

เมื่อเราทำการทดลองใหม่ โดยตั้งค่า sampling time ให้มีค่ามากขึ้น เป็น 1/50 วินาที และมีค่าเกณฑ์ต่าง ๆ เหมือนเดิมคือ $n = 51$ และ $m = 27$ โดยเราจะศึกษาผลของการเพิ่มค่า v ขึ้นเหมือนเดิม โดยแยกเป็นกรณีต่าง ๆ ได้เป็น

3.1.1.4. กรณี $v = 0$ และ sampling time เท่ากับ 1/50 วินาที

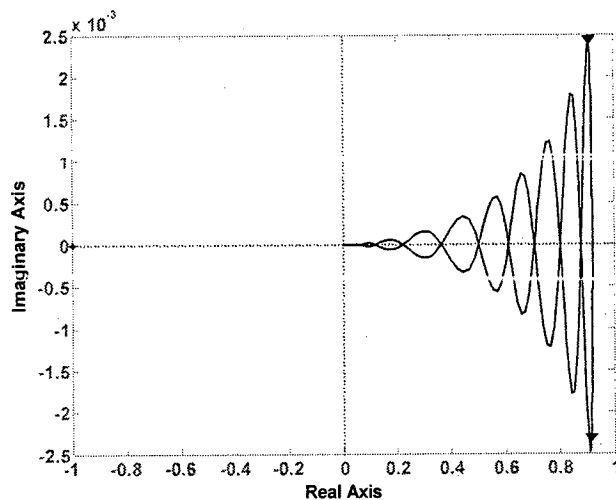
จากรูปจะเห็นได้ว่าขนาดของพล็อตมีขนาดเล็กมากคล้ายกับจุดที่ตำแหน่ง (1,0) เหมือนในกรณีที่ 1 โดยเราสามารถสรุปได้ว่าเมื่อเพิ่ม sampling time ให้สูงขึ้นในกรณีนี้ จะมีผลอะไรต่อการเรียนรู้หรือ stability ของ Repetitive control system น้อยมากๆ หรือกล่าวคือ ระบบยังสามารถเรียนรู้ได้อย่างรวดเร็วในทุกๆ ความถี่



รูปที่ 3.4 Polar plot ของ $G(e^{i\omega T})F(e^{i\omega T})$ เมื่อ $v = 0$ และ sampling time เท่ากับ 1/50 วินาที

3.1.1.5. กรณี $v = 0.1$ และ sampling time เท่ากับ 1/50 วินาที

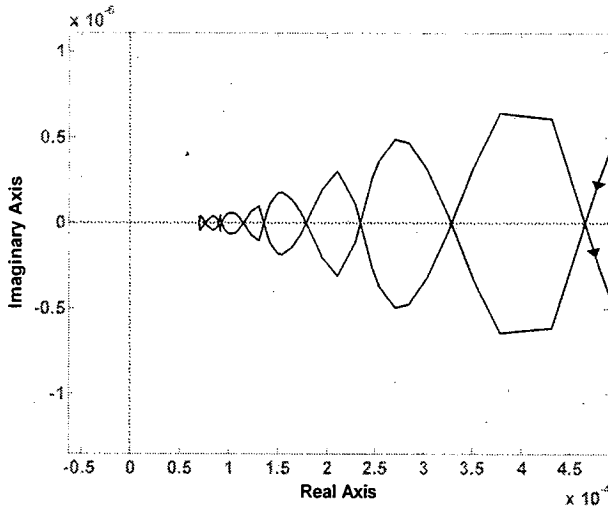
จากรูปจะเห็นได้ว่าที่ความถี่ต่ำๆ พล็อตจะอยู่บริเวณ (1,0) และเมื่อความถี่สูงขึ้น พล็อตก็จะเคลื่อนที่เข้าใกล้จุด (0,0) มากขึ้นด้วย แต่จะไม่ข้ามไปฝั่ง negative real part ซึ่งลักษณะก็จะคล้ายกับในกรณีที่ 2 ดังกล่าวมา



รูปที่ 3.5 Polar plot ของ $G(e^{i\omega T})F(e^{i\omega T})$ เมื่อ $v = 0.1$ และ sampling time เท่ากับ 1/50 วินาที

3.1.1.6. กรณี $\nu = 0.2$ และ sampling time เท่ากับ $1/50$ วินาที

พล็อตที่ได้จากกรณีนี้จะคล้ายกับกรณีที่ 5 แต่เมื่อทำการขยายบริเวณ $(0,0)$ แล้วจะเห็นได้ว่าที่ความถี่สูงๆ พล็อตก็ยังอยู่ภายในวงกลมรัศมีหนึ่งหน่วยที่มีจุดศูนย์กลางอยู่ที่จุด $(1,0)$ ซึ่งแสดงว่าระบบ stable และเรียนรู้ได้ช้าที่ความถี่สูงๆ ซึ่งจะแตกต่างจากผลในกรณีที่ 3 ดังนั้นเรายังสามารถนำ controller ที่ออกแบบมาในกรณีที่ 6 ไปใช้งานได้

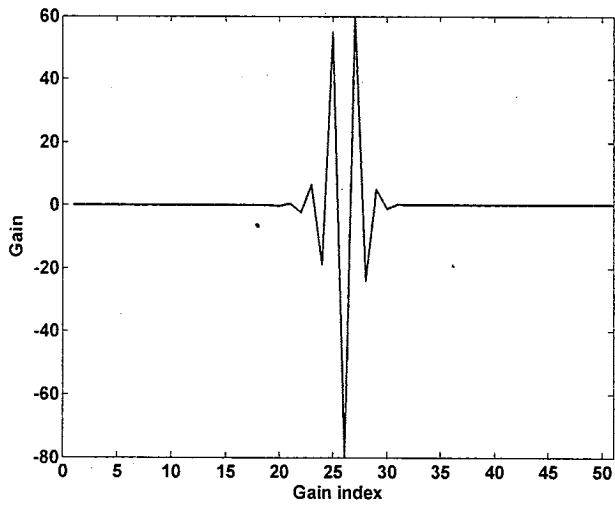


รูปที่ 3.6 Polar plot ของ $G(e^{i\omega T})F(e^{i\omega T})$ เมื่อ $\nu = 0.2$ และ sampling time เท่ากับ $1/50$ วินาที

3.1.2. ความสัมพันธ์ระหว่าง Learning cutoff frequency กับขนาดของ Gain size weight ν

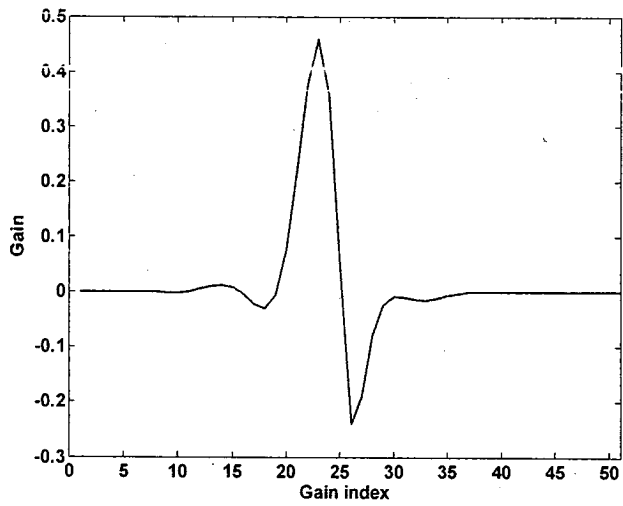
ดังได้กล่าวมาข้างต้นแล้วว่าเมื่อเราทำการเพิ่ม gain size weight หรือ ν ขึ้นเรื่อยๆแล้ว ขนาดของ gain จะมีค่าลดลงและความสามารถในการเรียนรู้ที่ความถี่ที่สูงขึ้นนั้นจะต่ำลง และในบางระบบของเราอาจจะ unstable ได้ ดังนั้น ผลการทดลองในหัวข้อนี้แสดงความสัมพันธ์ระหว่างค่า ν ที่เพิ่มขึ้นกับขนาดของเกนที่ใช้ และความถี่ที่ 95% ของความถี่สูงสุดก่อนที่ระบบจะไม่เรียนรู้ โดยมีค่า settling time = $1/50$ วินาที ซึ่งมีผลการทดลองเป็น

3.1.2.1.กรณี $\nu = 0$



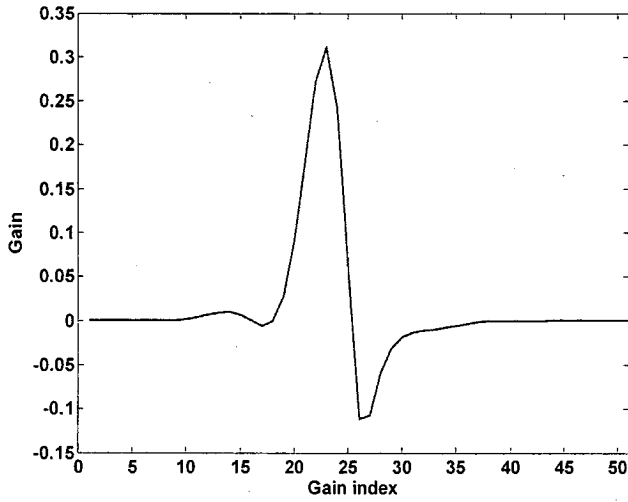
รูปที่ 3.7 ขนาดของแกน เมื่อ $\nu = 0$ และ sampling time เท่ากับ 1/50 วินาที

3.1.2.2.กรณี $\nu = 0.1$



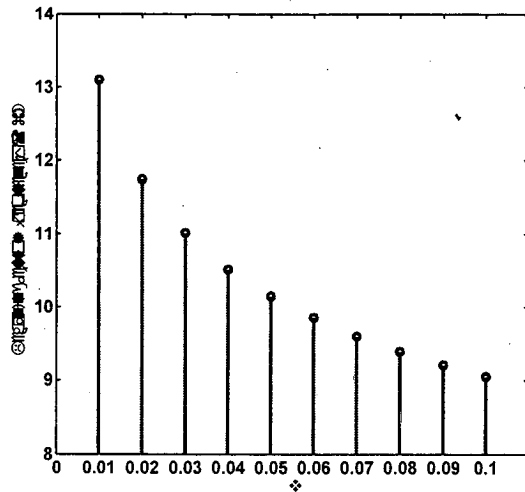
รูปที่ 3.8 ขนาดของแกน เมื่อ $\nu = 0.1$ และ sampling time เท่ากับ 1/50 วินาที

3.1.2.3. กรณี $\nu = 0.2$



รูปที่ 3.9 ขนาดของเกน เมื่อ $\nu = 0.2$ และ sampling time เท่ากับ $1/50$ วินาที

จะเห็นได้ว่าพล็อตจากทั้งสามกรณีมีขนาดเกนที่ลดลงเมื่อค่า ν มีค่าสูงขึ้นเรื่อยๆ และเมื่อทำการพิจารณาค่า ν กับความถี่ที่ 95% ของความถี่สูงสุดก่อนที่ระบบจะไม่เรียนรู้เราจะได้ผลการทดลองดังแสดง



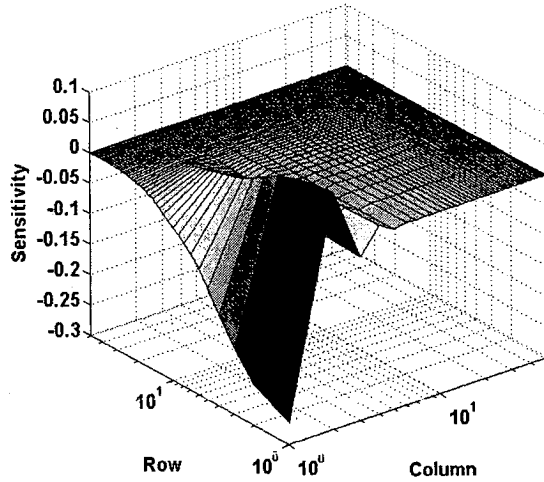
รูปที่ 3.10 ความสัมพันธ์ระหว่าง ν และ learning cutoff frequency

จากรูปจะเห็นได้ว่าเมื่อ ν มีค่าสูงขึ้น ความถี่ที่เราสามารถเรียนรู้ได้จะมีค่าลดลง ดังนั้นจะสรุปได้ว่าการที่เราเพิ่มค่า ν ขึ้นมีข้อดีคือทำให้ขนาดของ gain ที่ใช้ใน controller มีขนาดลดลงแต่จะทำให้ความสามารถในการเรียนรู้ที่ความถี่สูงๆลดลงด้วย

3.1.3. ค่า Sensitivity ของการออกแบบตัวควบคุม ILC แบบต่างๆ

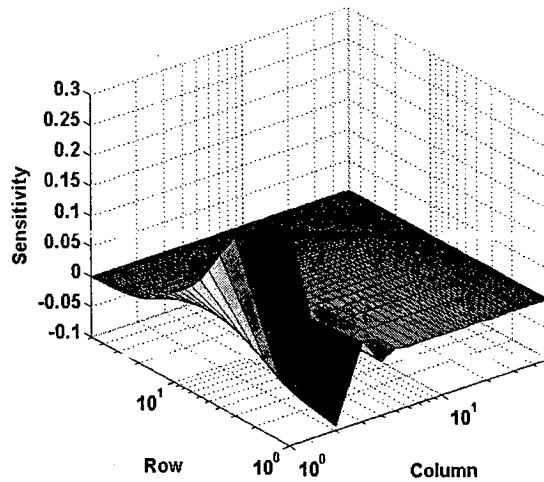
จากนิยามของ sensitivity ดังกล่าวข้างต้น เราพิจารณาถึงเมตริกซ์ $\frac{\partial \sigma_{\max}^2}{\partial \varepsilon_{ij}}$ ซึ่งดูผลของการเปลี่ยนแปลงค่า maximum singular value ต่อการเปลี่ยนแปลงค่าเกินในตำแหน่งต่างๆของเมตริกซ์การเรียนรู้ ผลการทดลองที่ได้เป็นดังนี้

3.1.3.1.กรณี $v=0$ $c=0$ และ sampling time เท่ากับ 1/50 วินาที



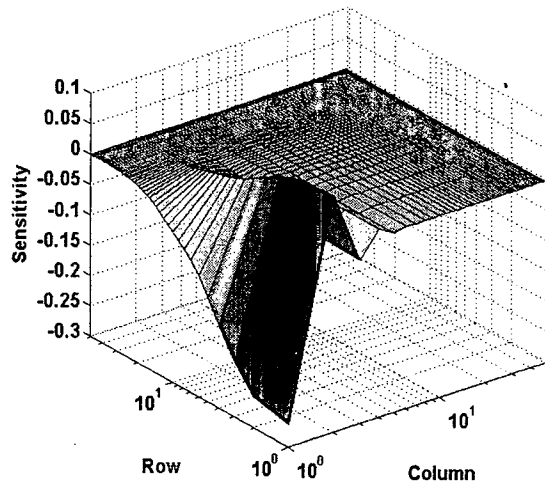
รูปที่ 3.11 Sensitivity เมื่อ $v=0$ $c=0$ และ sampling time เท่ากับ 1/50 วินาที

3.1.3.2.กรณี $v=0$ $c=1$ และ sampling time เท่ากับ 1/50 วินาที



รูปที่ 3.12 Sensitivity เมื่อ $v=0$ $c=1$ และ sampling time เท่ากับ 1/50 วินาที

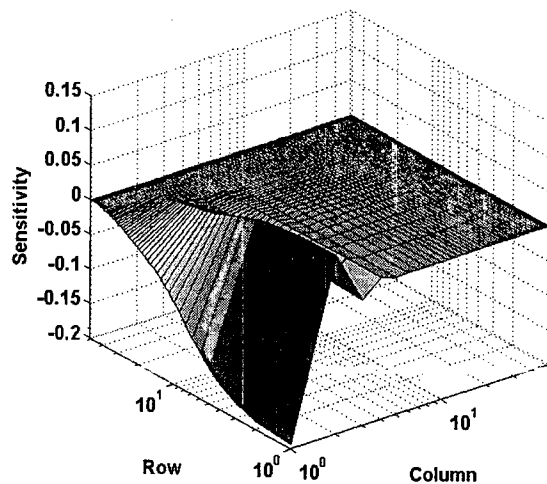
3.1.3.3.กรณี $\nu = 0$ $c = 2$ และ sampling time เท่ากับ 1/50 วินาที



รูปที่ 3.13 Sensitivity เมื่อ $\nu = 0$ $c = 2$ และ sampling time เท่ากับ 1/50 วินาที

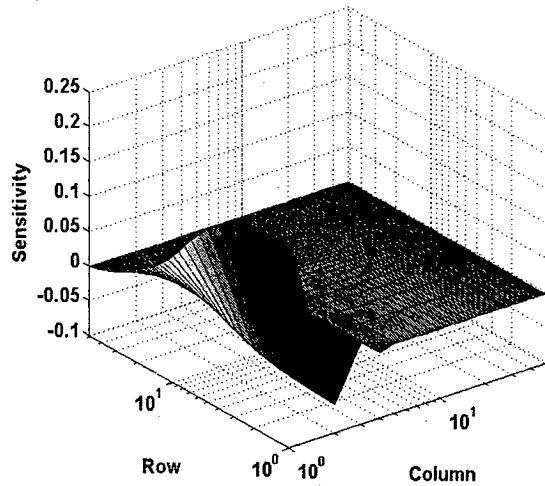
รูปที่ 3.11-3.13 แสดงค่า sensitivity ในกรณีที่มีค่า cutoff number (c) ที่แตกต่างกันโดยมีค่า sampling time เท่ากับ 50 วินาที จากรูปจะเห็นได้ว่า sensitivity มีค่าสูงบริเวณบนด้านซ้ายมือของเมตริกซ์การเรียนรู้ นั่นคือผลจากการเปลี่ยนแปลงค่าเกินบริเวณด้านบนทางซ้ายมือของเมตริกซ์การเรียนรู้จะทำให้เกิดการเปลี่ยนแปลงค่า maximum singular value ไปมากกว่าการเปลี่ยนแปลงค่าเกินในบริเวณอื่นๆ

3.1.3.4.กรณี $\nu = 0$ $c = 0$ และ sampling time เท่ากับ 1/100 วินาที



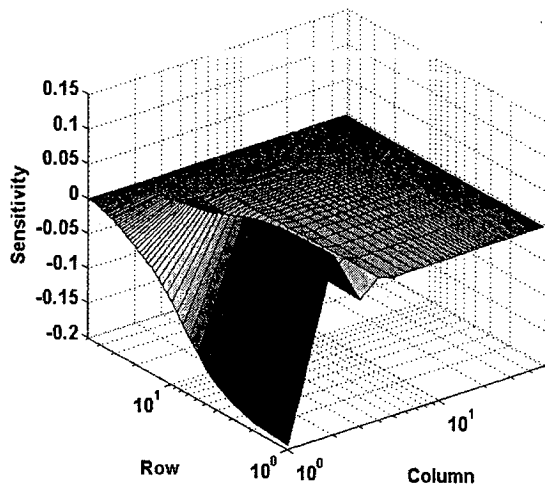
รูปที่ 3.14 Sensitivity เมื่อ $\nu = 0$ $c = 0$ และ sampling time เท่ากับ 1/100 วินาที

3.1.3.5.กรณี $v=0$ $c=1$ และ sampling time เท่ากับ 1/100 วินาที



รูปที่ 3.15 Sensitivity เมื่อ $v=0$ $c=1$ และ sampling time เท่ากับ 1/100 วินาที

3.1.3.6.กรณี $v=0$ $c=2$ และ sampling time เท่ากับ 1/100 วินาที



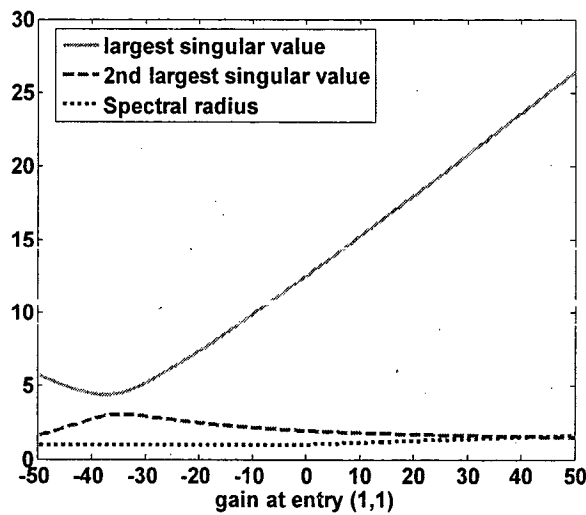
รูปที่ 3.16 Sensitivity เมื่อ $v=0$ $c=2$ และ sampling time เท่ากับ 1/100 วินาที

รูปที่ 3.14-3.16 แสดงค่า sensitivity ในกรณีที่มีค่า cutoff number (c) ที่แตกต่างกันโดยมีค่า sampling time เท่ากับ 100 วินาที ลักษณะของกราฟทั้งสามรูปจะคล้ายกับในรูปที่ 3.11-3.13 โดยค่าของ sensitivity จะแตกต่างกันเพียงเล็กน้อย ซึ่งการเปลี่ยนค่า sampling time ไม่ได้ส่งผลต่อการเปลี่ยนแปลงค่า sensitivity มากนัก

3.1.4. Stability ของตัวควบคุม ILC แบบมีการปรับค่าเกนเพียง 1 ค่า ที่ cutoff number (c) ค่าต่างๆ

ในหัวข้อนี้จะอธิบายถึงผลของการใช้ตัวควบคุม ILC โดยมีการเปลี่ยนแปลงค่า gain เพียงหนึ่งตัวไปเรื่อยๆ เพื่อที่จะหาค่า gain ค่าใดที่จะทำให้เกิดผลดีที่สุด (หรือค่าของ maximum singular value ของ matrix $(I - PL)$ ต่ำที่สุด) โดยในที่นี้เราจะพิจารณาที่ค่า cutoff number ที่แตกต่างกัน โดยกำหนดค่า $v = 0$ และ sampling time เท่ากับ $1/50$ sec ค่า gain ที่เราจะทำการเปลี่ยนแปลงค่านี้จะเป็น gain ของ Learning matrix ตัวที่อยู่ทางด้านซ้ายบนสุด ซึ่งพิจารณาจากค่า sensitivity ดังแสดงในหัวข้อ 3.1.3 โดยจะพิจารณาในกรณีที่ $c = 0, 1, 2$ จะเป็นเกน l_{11}, l_{12}, l_{13} ตามลำดับ

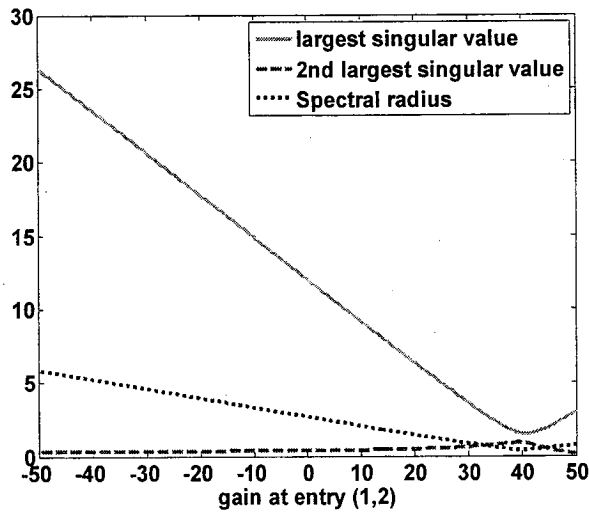
3.1.4.1. กรณี $c = 0$



รูปที่ 3.17 ค่า Singular value ที่สูงสุดและสูงสุดอันดับสอง และค่า spectral radius เมื่อ $c = 0$

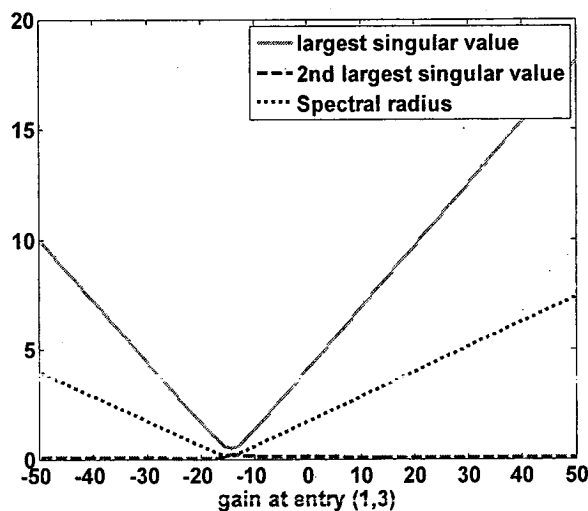
ในกรณีนี้จะเป็นการปรับค่าเกน l_{11} ไปเรื่อยๆตั้งแต่ -50 จนกระทั่ง +50 เพื่อดูว่าจะมีผลอย่างไรต่อ stability ของระบบ จากรูปจะเห็นได้ว่าคุณค่า spectral radius (ขนาดของ eigenvalue) มีค่าเป็น 1 เสมอไม่ว่าเราจะปรับค่า gain l_{11} ไปเป็นเท่าใด ในขณะที่ค่า singular value ที่สูงสุดมีค่าต่ำที่สุดประมาณ 4 นั่นคือเราเข้าใกล้ stability condition ในสมการที่ (10) แต่ไม่สามารถทำให้สมการที่ (11) หรือ monotonic decay condition เป็นจริงได้ไม่ว่าเราจะปรับค่า l_{11} ไปเป็นเท่าใดก็ตาม

3.1.4.2. กรณี $c = 1$



รูปที่ 3.18 ค่า Singular value ที่สูงสุดและสูงสุดอันดับสอง และค่า spectral radius เมื่อ $c = 1$ ต่อมาเป็นการปรับค่าของเกน I_{12} ตั้งแต่ -50 ไปจนถึง +50 เพื่อพิจารณาผลของ stability ของระบบ ในกรณีนี้เราสามารถทำให้ stability condition ในสมการที่ (10) เป็นจริงได้ เนื่องจากค่า spectral radius มีค่าต่ำกว่า 1 ที่ค่า gain I_{12} ประมาณ 40 และเรายังสามารถเข้าใกล้สมการที่ (11) หรือ monotonic decay condition ได้ที่ค่า I_{12} เดียวกันนี้ เนื่องจากค่า largest singular value มีค่าเข้าใกล้ 1 มากๆ

3.1.4.3. กรณี $c = 2$



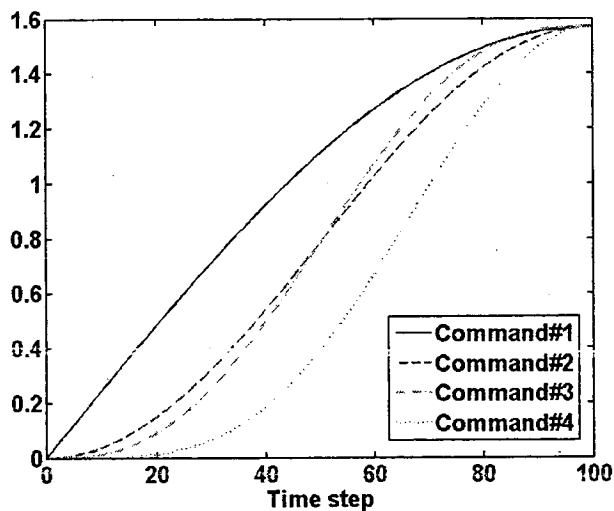
รูปที่ 3.19 ค่า Singular value ที่สูงสุดและสูงสุดอันดับสอง และค่า spectral radius เมื่อ $c = 2$ ในกรณีสุดท้ายนี้เป็นการปรับค่าของเกน I_{13} ตั้งแต่ -50 ไปจนถึง +50 เพื่อพิจารณาผลของ stability ของระบบ จะเห็นได้ว่าการปรับค่า I_{13} สามารถทำให้ทั้ง asymptotic stability condition และ monotonic decay condition เป็นไปตามสมการที่ (10) และ (11) ตามลำดับ กล่าวคือการตัดค่า gain ใน 2 column แรกออกไปนั้นทำให้เราสามารถ guarantee stability conditions ของ ILC ได้นั่นเอง

3.1.5. ผล simulation ของตัวควบคุม ILC แบบมีการปรับค่าเกนเพียง 1 ค่า ที่ cutoff number (c)
ค่าต่างๆ

จากการศึกษา stability ของระบบในหัวข้อที่แล้วนั้น เราจะนำ ILC controller ที่ได้ออกแบบมาในกรณีต่างๆ มาศึกษาผลการทำ simulation ในหัวข้อนี้ โดย Learning control matrix ที่จะใช้นั้นมีการปรับค่า gain เพียง 1 ค่า ที่ทำให้เกิดค่า largest singular value ของ matrix $I - PL$ น้อยที่สุดที่ c ค่าต่างๆ กำหนดให้ สัญญาณ input trajectory ที่จะทำการทดลองนั้นกำหนดให้เป็นสัญญาณ smooth trajectory ที่มีทั้งหมด 4 ประเภทดังต่อไปนี้

$$\begin{aligned}
 y_{d1} &= (\pi/2)\sin(\omega t) & ; \omega = 2\pi/8 & ; t = kT \\
 y_{d2} &= (\pi/4)[1 - \cos(\omega t)] & ; \omega = 2\pi/4 & \\
 y_{d3} &= \pi \left[5t^3/(100T)^3 - 7.5t^4/(100T)^4 + 3t^5/(100T)^5 \right] & & (32) \\
 y_{d4} &= (\pi/8)[1 - \cos(\omega t)]^2 & ; \omega = 2\pi/4 &
 \end{aligned}$$

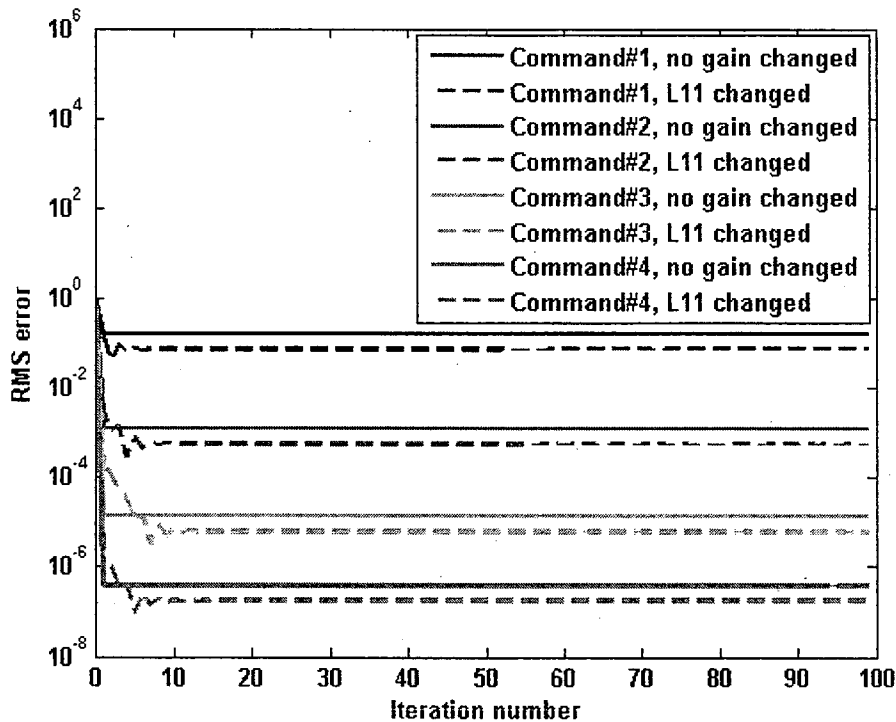
โดยมีค่า time step $k = 0, 1, 2, \dots, 100$ trajectory input ทั้ง 4 ประเภทนั้นถูกแสดงได้ดังรูปที่ 3.20 จากลักษณะของกราฟ trajectory y_{d1} เป็นฟังก์ชันต่อเนื่องโดยมีฟังก์ชันของ first derivative เป็นฟังก์ชันที่ไม่ต่อเนื่อง ในขณะที่ y_{d2} เป็นฟังก์ชันที่ต่อเนื่องและ first derivative ก็ยังเป็นฟังก์ชันที่ต่อเนื่องเช่นเดียวกัน แต่มี second derivative เป็นฟังก์ชันไม่ต่อเนื่อง y_{d3} เป็นฟังก์ชันที่ต่อเนื่องรวมไปถึง first และ second derivative ด้วย และสุดท้ายฟังก์ชัน y_{d4} เป็นฟังก์ชันที่ต่อเนื่องตลอดไปจนถึง third derivative



รูปที่ 3.20 Trajectory input ทั้ง 4 ประเภทที่ใช้ในการทำ simulation

เมื่อนำค่า trajectory input ป้อนเข้าไปยังระบบแขนกลหุ่นยนต์ ค่าของ root mean square (RMS) error สามารถแสดงได้จากกรณีต่างๆดังต่อไปนี้

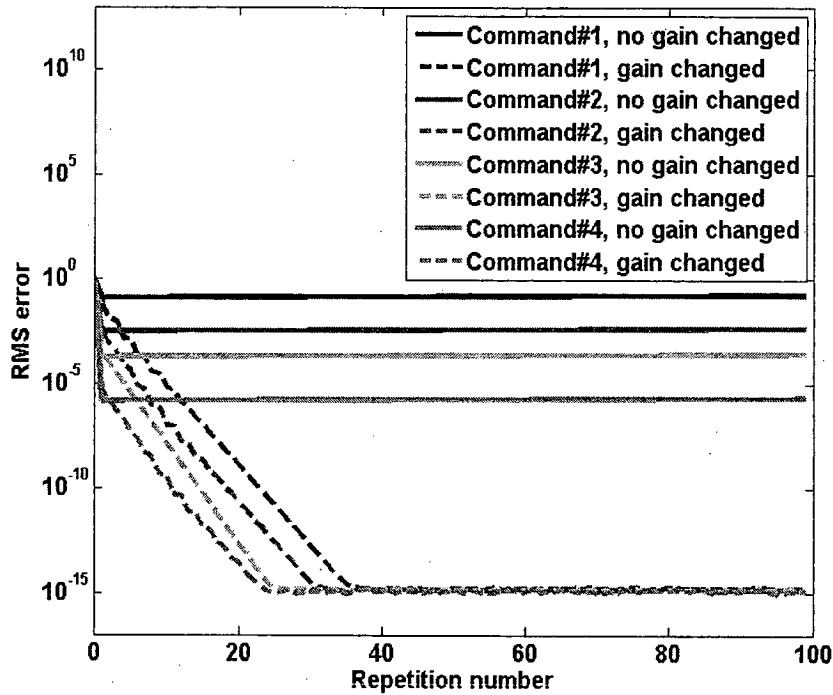
3.1.5.1. กรณี $c = 0$



รูปที่ 3.21 RMS error จาก Trajectory input ทั้ง 4 ประเภท

กราฟที่ได้แสดงถึงการเปรียบเทียบค่าของสัญญาณ RMS error ที่ได้จาก trajectory input ประเภทต่างๆ ด้วยการใช้ learning control matrix แบบที่ไม่มีการเปลี่ยนแปลงค่า gain และ learning control matrix ที่มีการเปลี่ยนแปลงค่า gain ตัวแรก หรือ L_{11} โดยเลือกใช้ L_{11} ที่มีค่าเป็น -37.5 กราฟได้แสดงให้เห็นว่าระบบ stable ไม่ว่าจะมีการปรับเปลี่ยนค่าของ learning gain L_{11} หรือไม่ก็ตาม ในกรณีที่ไม่มีค่า L_{11} ระบบมี spectral radius เท่ากับ 1.000 และ maximum singular value เท่ากับ 12.7055 ดังแสดงได้จากรูปที่ 3.17 และในกรณีที่มีการปรับค่า L_{11} เราจะเลือกใช้ค่า L_{11} ที่ทำให้ matrix $I - PL$ มีค่า maximum singular value ที่น้อยที่สุด โดยค่า L_{11} ที่ใช้มีค่าเท่ากับ -37.5 และทำให้เกิดค่า spectral radius ของ matrix $I - PL$ มีค่าเป็น 1.000 และ maximum singular value เท่ากับ 4.3769 ดังแสดงได้จากรูปที่ 3.17 ซึ่งทำให้ค่า RMS error ไม่ได้ลดค่าลงอย่างต่อเนื่อง (not monotonic decay) และระบบมีค่า steady state error ที่แตกต่างกันในแต่ละ input โดยยิ่งสัญญาณ trajectory input มีความ smooth มากขึ้นเท่าใด ค่า steady state error ก็จะมีค่าต่ำลงมากเท่านั้น

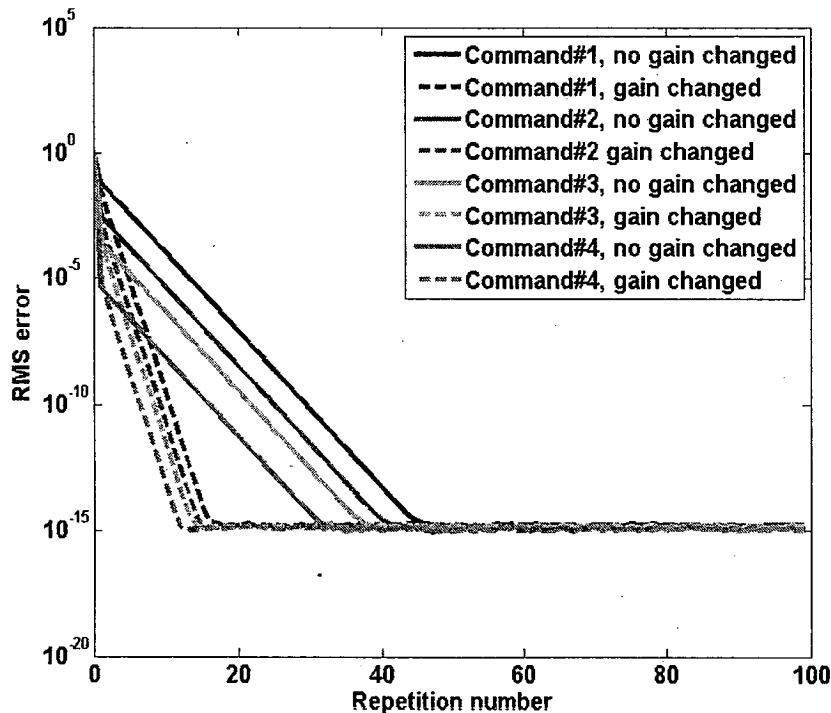
3.1.5.2. กรณี $c = 1$



รูปที่ 3.22 RMS error จาก trajectory input ทั้ง 4 ประเภท โดยไม่นำค่า error ใน time step แรกมาคิด

กราฟแสดงค่า RMS error ที่คำนวณได้ในแต่ละ repetition โดยไม่นำค่า error ใน time step แรกมาคิด เนื่องจากกรณีนี้ $c = 1$ ซึ่งจะเห็นได้ว่าเมื่อทำการตัดแถวแรกของ learning control matrix ออกไป หรือเมื่อเราไม่ต้องการสนใจค่าของ error ที่ time step แรกแล้ว ค่าของ RMS error จะลดลงอย่างต่อเนื่อง (monotonic decay) โดยมีค่า spectral radius ของ matrix $I - PL$ ในตอนนี้มีค่าเป็น 0.4132 และค่า maximum singular value ของเมตริกซ์ดังกล่าวมีค่าเป็น 1.4866 ในกรณีนี้ค่าของ error ที่ time step ที่ 1 ของทุกๆ repetition จะมีค่าเท่าเดิมเสมอ โดยเราไม่ได้นำมาคิดในการคำนวณของค่า RMS error ซึ่งค่า steady state error ของแต่ละ trajectory input มีค่าประมาณเท่ากันที่ 10^{-15}

3.1.5.3. กรณี $c = 2$

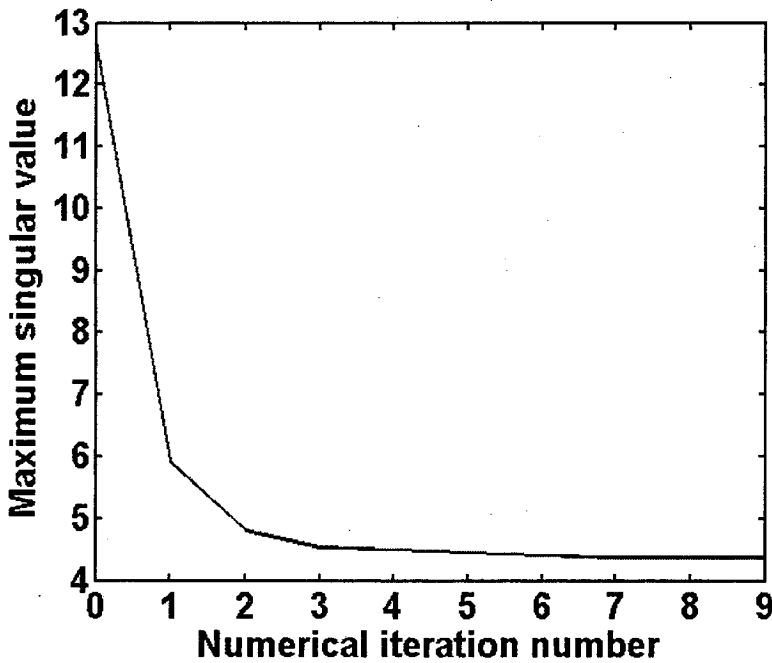


รูปที่ 3.23 RMS error จาก Trajectory input ทั้ง 4 ประเภทโดยไม่นำค่า error ใน 2 time step แรกมาคิด กราฟแสดงค่า RMS error ที่คำนวณได้ในแต่ละ repetition โดยไม่นำค่า error ใน time step ที่ 1 และ 2 มาคิดเนื่องจากกรณีนี้ $c = 2$ ซึ่งจะเห็นได้ว่าเมื่อทำการตัดสองแถวแรกของ learning control matrix ออกไป หรือเมื่อเราไม่ต้องการสนใจค่าของ error ที่ time step ที่ 1 และ 2 แล้ว ค่าของ RMS error จะลดลงอย่างต่อเนื่อง (monotonic decay) โดยค่า RMS error นั้นจะลดลงได้ไวกว่าและ smooth กว่าในกรณีข้างต้นที่ $c = 1$ ค่า spectral radius ของ matrix $I - PL$ ในตอนนี้มีค่าเป็น 0.1287 และค่า maximum singular value ของเมทริกซ์ดังกล่าวมีค่าเป็น 0.4808 ในกรณีนี้ค่าของ error ที่ time step ที่ 1 และ 2 ของทุกๆ repetition จะมีค่าเท่าเดิมเสมอ โดยเราไม่ได้นำมาคิดในการคำนวณของค่า RMS error ซึ่งค่า steady state error ของแต่ละ trajectory input มีค่าประมาณเท่ากันที่ 10^{-15}

3.1.6. Stability ของตัวควบคุม ILC แบบมีการปรับค่าเกินมากกว่า 1 ค่า

ในหัวข้อนี้จะกล่าวถึงการปรับค่าเกินใน Learning matrix จำนวน 3 ค่า (โดยมีค่า cutoff number เท่ากับ ศูนย์) ที่ตำแหน่ง 3 แถวแรกในหลักที่ 1 ของเมทริกซ์เพื่อดูว่า stability แตกต่างอย่างไรกับการปรับค่าเกินเพียง 1 ค่า ดังที่ได้กล่าวมาในหัวข้อที่แล้ว โดยในที่นี้จะใช้การปรับค่าที่ทำให้เกิดการเรียนรู้ได้เร็วที่สุดโดยการใช้ steepest descent method จากหัวข้อ 2.5.2 ซึ่งกำหนดค่า direction ของการ update ค่าเป็นค่าของ sensitivity function ที่คำนวณได้จากหัวข้อ 2.5.1 ซึ่งจะมีการ update direction ทุกๆครั้งในแต่ละ iteration และค่า step size เริ่มต้นมีค่าเป็น 0.1 ซึ่งจะมีการเพิ่มหรือลดในแต่ละ iteration ตามกฎการคำนวณของวิธี steepest descent ผลของการปรับค่าเกินทั้ง 3 ค่าทำให้เกิดการลดค่า Maximum singular value ในแต่ละ iteration ตามรูปที่ 3.24 โดยเมื่อทำการเปรียบเทียบกับผลของการปรับค่าเกินเพียง 1 ค่าในรูปที่ 3.17 แล้ว จะเห็นได้ว่าการเพิ่มจำนวนเกินในการปรับค่าขั้นนั้น ไม่ได้ช่วยทำให้ maximum singular

value มีค่าน้อยลงกว่าการเลือกปรับเกณฑ์เพียงแค่ 1 ค่าแต่อย่างไรก็ตาม การปรับค่าเกณฑ์เพิ่มขึ้นนั้นไม่มีประโยชน์ที่จะทำให้ประสิทธิภาพการเรียนรู้ของระบบดีขึ้นดังเหตุผลที่กล่าวมา



รูปที่ 3.24 ค่า Singular value ที่สูงสุดจากการปรับค่าเกณฑ์ 3 ค่าใน Learning matrix

3.2. ผลการทดลองจากการออกแบบตัวควบคุมแบบเรียนรู้ที่มีการย้อนกลับไปสู่ภาวะเริ่มต้นโดยวิธีการหาค่าเฉลี่ย

เพื่อเป็นการแสดงประสิทธิภาพของตัวควบคุมแบบเรียนรู้ที่มีการย้อนกลับไปสู่ภาวะเริ่มต้นโดยวิธีการหาค่าเฉลี่ย ในหัวข้อนี้จะเพิ่มความซับซ้อนของระบบหุ่นยนต์แขนกลเพื่อทำให้การควบคุมเป็นไปได้ยากมากขึ้น โดยจะเพิ่มจำนวน order ของระบบเป็น 5 order ซึ่งจะมี 4 complex poles และ 1 real pole ดังสมการ

$$G(s) = \left(\frac{a}{s+a} \right) \left(\frac{\omega_1^2}{s^2 + 2\zeta_1\omega_1s + \omega_1^2} \right) \left(\frac{\omega_2^2}{s^2 + 2\zeta_2\omega_2s + \omega_2^2} \right) \quad (33)$$

ในการทดลองนี้จะใช้ sampling frequency อยู่ที่ 100 Hz และกำหนดค่าตัวแปรของระบบจริงคือ $a = 8.8$, $\zeta_1 = \zeta_2 = 0.1$, $\omega_1 = 6(2\pi)$ rad/sec, $\omega_2 = 18(2\pi)$ rad/sec จากนั้นจะมีการสร้างระบบใหม่ขึ้นมาอีก 200 ระบบโดยมีการเปลี่ยนค่าตัวแปรต่างๆเพียงเล็กน้อย เพื่อใช้สร้างตัวควบคุมโดยวิธีการหาค่าเฉลี่ย ซึ่งค่าตัวแปรที่เปลี่ยนแปลงไปนั้นมีค่าต่างๆดังนี้

$$\begin{aligned} 6.613 < a < 10.977 \\ 0.0751 < \zeta_1 < 0.1243, 0.0751 < \zeta_2 < 0.1243 \\ 27.762 < \omega_1 < 46.234, 85.022 < \omega_2 < 141.114 \end{aligned} \quad (34)$$

โมเดลที่ได้สร้างขึ้นใหม่ 200 โมเดลนั้นมีการเปลี่ยนแปลงค่าตัวแปรเพียง 1 ตัว จากระบบจริง โดยเราได้ทำการออกแบบตัวควบคุมโดยวิธีการหาค่าเฉลี่ยที่เหมาะสมกับทั้ง 200 โมเดลที่สร้างขึ้นมา โดยตัวควบคุมที่ออกแบบมานั้นจะมี 3 แบบด้วยกันคือ contraction mapping, partial isometry, และ quadratic cost จากสมการที่ (29)-(31) ตามลำดับ เมื่อออกแบบตัวควบคุมเสร็จเรียบร้อยแล้ว เราได้นำมาทดสอบ stability ของระบบควบคุมแบบเรียนรู้ที่มีการย้อนกลับไปสู่ภาวะเริ่มต้น ว่าตัวควบคุมที่ได้ออกแบบมานั้นมีประสิทธิภาพมากน้อยเพียงใด โดยดูจาก condition ในสมการที่ (10) และ (11) ตารางที่ 1-3 แสดงผลของการทดสอบ controller แบบ contraction mapping, partial isometry, และ quadratic cost ตามลำดับ

ตารางที่ 1 ผลการทดสอบเสถียรภาพของระบบโดยใช้ตัวควบคุมแบบ contraction mapping

Contraction mapping	Max. Eigenvalues			Max. Singular values		
	>1	>1.001	>1.01	>1	>1.001	>1.01
Nominal	178	81	80	199	92	92
Average	155	0	0	200	13	12
Average with $\phi = 0.5$	139	0	0	200	5	2
Average with $\phi = 0.1$	148	0	0	200	0	0
200 different models: 1st run						
Nominal	173	89	87	200	99	98
Average	148	0	0	199	22	21
200 different models: 2nd run						
Nominal	169	80	79	200	93	88
Average	157	0	0	200	10	9
200 different models: 3rd run						
Nominal	163	73	71	200	87	86
Average	152	0	0	200	19	17

ตารางที่ 2 ผลการทดสอบเสถียรภาพของระบบโดยใช้ตัวควบคุมแบบ partial isometry

Partial Isometry	Max. Eigenvalues			Max. Singular values		
	>1	>1.001	>1.01	>1	>1.001	>1.01
Nominal	173	79	64	200	99	83
Average	158	1	1	200	42	33
Average with $\phi = 0.5$	159	0	0	200	15	9
Average with $\phi = 0.1$	155	0	0	200	2	0
200 different models: 1st run						
Nominal	171	78	66	200	99	87
Average	147	7	7	200	55	50
200 different models: 2nd run						
Nominal	171	76	62	200	94	80
Average	155	5	4	200	44	36
200 different models: 3rd run						
Nominal	177	70	57	200	87	75
Average	146	4	4	200	39	37

ตารางที่ 3 ผลการทดสอบเสถียรภาพของระบบโดยใช้ตัวควบคุมแบบ quadratic cost

Quadratic Cost	Max. Eigenvalues			Max. Singular values		
	>1	>1.001	>1.01	>1	>1.001	>1.01
Nominal	159	21	19	199	42	40
Average	136	0	0	200	9	5
Average with $\phi = 0.5$	154	0	0	200	2	0
Average with $\phi = 0.1$	142	0	0	199	0	0
200 different models: 1st run						
Nominal	160	35	31	200	53	49
Average	156	0	0	200	9	5
200 different models: 2nd run						
Nominal	156	16	16	200	44	43
Average	156	0	0	200	5	4
200 different models: 3rd run						
Nominal	152	27	-22	200	41	38
Average	150	0	0	200	8	5

จากข้อมูลในตาราง เราได้ทำการศึกษาค่าของ maximum eigenvalue และ maximum singular value ของเมทริกซ์ $I - PL$ โดย L คือ ตัวควบคุมที่ได้ออกแบบมา และ P คือระบบทั้ง 200 โมเดลที่สร้างขึ้นมา เพื่อที่ว่าถ้าระบบมีค่าพารามิเตอร์ที่เปลี่ยนแปลงไปแล้ว ตัวควบคุมที่ออกแบบมายังสามารถทำให้ระบบมีเสถียรภาพได้อยู่หรือไม่ ทั้งนี้ได้มีการแบ่งระดับค่าของ maximum eigenvalue และ maximum singular value ไว้ 3 ระดับด้วยกันคือ มากกว่า 1, มากกว่า 1.001, และ มากกว่า 1.01 ทั้งนี้เนื่องจากค่าที่คำนวณได้นั้นอาจเกิดจากการผิดพลาดจากการคำนวณ (numerical round off in computation) โดยจากสมการที่ (10)-(11) ถ้าค่า maximum eigenvalue มีค่าน้อยกว่า 1 แสดงว่าระบบนั้นมีเสถียรภาพ และถ้าค่า maximum singular value มีค่าน้อยกว่า 1 แสดงว่าค่า RMS error ของระบบจะค่อยๆ ลดค่าลงจนเข้าใกล้ศูนย์ เมื่อพิจารณาจากตัวควบคุมแบบ contraction mapping ในตารางที่ 1 แล้ว จะเห็นได้ว่าเมื่อมีการออกแบบตัวควบคุมสำหรับระบบจริงขึ้นมา (nominal) และนำไปทดสอบกับระบบจำลองที่สร้างขึ้นมา 200 โมเดล พบว่า มีระบบ 178 ระบบที่มีค่า maximum eigenvalue มากกว่า 1 นั่นคือ มีระบบมากถึง 178 ระบบที่ไม่มีเสถียรภาพถ้ามีการใช้ตัวควบคุมที่ออกแบบมาจากระบบจริงเพียงระบบเดียว แต่เมื่อมีการใช้ตัวควบคุมแบบวิธีการหาค่าเฉลี่ย (average) แล้ว พบว่ามีระบบเพียง 155 ระบบที่ไม่มีเสถียรภาพซึ่งมีจำนวนลดลง และเมื่อทำการปรับค่าเกนในตัวควบคุมให้มีค่า $\phi = 0.1$ และ $\phi = 0.5$ แล้วยิ่งทำให้จำนวนระบบที่ไม่มีเสถียรภาพลดลงตามลำดับ

ถัดมาเป็นการออกแบบตัวควบคุมจากระบบจริงเพียงระบบเดียว (nominal) มาทดสอบกับระบบที่สร้างขึ้น โดยมีค่าพารามิเตอร์เปลี่ยนแปลงไปแบบ random 200 โมเดลแต่ยังอยู่ในช่วงเดียวกับสมการที่ (34) ซึ่งผลการทดสอบระบุว่าจำนวนโมเดลที่ไม่มีเสถียรภาพ และเมื่อใช้ตัวควบคุมที่ออกแบบโดยวิธีการหาค่าเฉลี่ยพบว่ามีจำนวนโมเดลที่ไม่มีเสถียรภาพลดลงเหลือ 148 โมเดล ซึ่งในการทดสอบครั้งที่สองและครั้งที่สาม ได้ทำการเปลี่ยนพารามิเตอร์แบบ random 200 โมเดล ซึ่งผลการทดสอบสรุปได้ว่าตัวควบคุมที่ออกแบบจากระบบจริงเพียงอย่างเดียวทำให้จำนวนระบบไม่มีเสถียรภาพมีมากกว่าการใช้ตัวควบคุมที่ออกแบบโดยวิธีการหาค่าเฉลี่ย ซึ่งผลการทดลองมีแนวโน้มที่เหมือนกันไม่ว่าจะเป็นการใช้ตัวควบคุมแบบ contraction mapping หรือ partial isometry หรือ quadratic cost ก็ตาม