



บทที่ 3

คลังโปรแกรมแขนหุ่นยนต์

เนื้อหาในบทนี้เกี่ยวกับคลังโปรแกรมแขนกล ในส่วนของการคำนวณจลนศาสตร์และพลวัต ซึ่งประกอบด้วย 2 ส่วน ในส่วนแรก กล่าวถึงรายละเอียดของคลังโปรแกรมที่สร้างขึ้น ซึ่งประกอบด้วยโครงสร้างตัวแปรของแขนกล และฟังก์ชันภาษา C ที่มีอยู่ภายในคลังโปรแกรมสำหรับใช้ในการคำนวณ สำหรับในส่วนที่สองเป็นตัวอย่างการใช้งาน และการเปรียบเทียบผลการจำลองระบบกับ Robotic Toolbox [5]

3.1 รายละเอียดของคลังโปรแกรม

คลังโปรแกรมแขนหุ่นยนต์ที่สร้างขึ้นสามารถใช้กับแขนกลแบบท่อนเขียนเรียงลำดับ ซึ่งผู้ใช้สามารถกำหนดจำนวนและชนิดของข้อต่อ รวมถึงลักษณะของแขนกลแต่ละท่อนได้เอง โดยเก็บค่าพารามิเตอร์ของแขนกลเหล่านั้นไว้ในไฟล์ข้อมูล ซึ่งรูปแบบของการเก็บข้อมูลในไฟล์มีลักษณะใกล้เคียงกับโปรแกรมอื่นๆ ได้แก่ Robotic Toolbox และ ROBOOP

สำหรับฟังก์ชันภาษา C ภายในคลังโปรแกรมสำหรับการจำลองระบบแขนกล นอกจากจะมีฟังก์ชันการคำนวณหาจลนศาสตร์ และพลวัตของแขนกล ซึ่งเป็นส่วนสำคัญแล้ว ยังมีฟังก์ชันสำหรับทำหน้าที่อื่นๆ ที่จำเป็น อาทิเช่น การจัดการและคำนวณที่เกี่ยวข้องกับตัวแปรเมตริกซ์ การแก้สมการอนุพันธ์ การเก็บข้อมูลจากการคำนวณในไฟล์ โดยผู้ใช้มีหน้าที่เฉพาะเขียนโปรแกรมหลักสำหรับเรียกฟังก์ชันต่าง ๆ เหล่านี้มาคำนวณ

สาเหตุที่คลังโปรแกรมเขียนโดยใช้ภาษา C เนื่องจากมีข้อดีคือ สามารถคำนวณได้รวดเร็ว เหมาะสำหรับการนำไปใช้ควบคุมแขนกลแบบเวลาจริง (real time), ภาษา C มีการใช้อย่างแพร่หลายมาเป็นเวลานาน ทำให้มีชุดคำสั่งและคลังโปรแกรมอื่น ๆ ซึ่งช่วยลดภาระในการเขียนโปรแกรม เช่น ช่วยในการสร้างส่วนติดต่อกับผู้ใช้, การติดต่อกับพอร์ตอนุกรม (serial port)

ไฟล์ต้นฉบับของคลังโปรแกรมภาษา C ที่เขียนขึ้น สามารถนำไปแปล (compile) และเชื่อมโยง (link) เป็นไฟล์ปฏิบัติการ สำหรับใช้ทำงานบนระบบปฏิบัติการวินโดวส์และลินุกซ์ สำหรับขั้นตอนการติดตั้งและการใช้งานซึ่งอธิบายอยู่ในภาคผนวก ก นั้น ได้กล่าวถึงเฉพาะสำหรับกรณีที่ใช้งานลินุกซ์เท่านั้น เนื่องจากระบบปฏิบัติการนี้สามารถนำมาใช้อย่างถูกต้องโดยไม่ต้องเสียค่าใช้จ่าย และหาได้ง่ายจากอินเทอร์เน็ต

3.1.1 โครงสร้างของตัวแปรแขนหุ่นยนต์

ก่อนการใช้งานคลังโปรแกรม ผู้ใช้ต้องกำหนดโครงสร้างแขนหุ่นยนต์ในรูปแบบของไฟล์ข้อมูล (text file) สำหรับให้โปรแกรมอ่านค่าพารามิเตอร์ของแขนกล ลักษณะของไฟล์ข้อมูลดังกล่าวประกอบด้วย จำนวนข้อต่อของแขนกล, ตำแหน่ง q ความเร็ว \dot{q} ความเร่ง \ddot{q} ของแต่ละข้อ

ต่อ สำหรับเป็นค่าเริ่มต้นในการคำนวณ, พารามิเตอร์ของแขนแต่ละท่อน, เมตริกซ์การแปลงจากแกนอ้างอิงชุดสุดท้าย ไปยัง แกนอ้างอิงของ end-effector, เวกเตอร์ของโมเมนต์และแรงภายนอกที่กระทำต่อแขนท่อนสุดท้าย, และเวกเตอร์ของความเร่งเนื่องจากแรงโน้มถ่วง ในส่วนของพารามิเตอร์สำหรับแขนกลแต่ละท่อน ประกอบด้วย พารามิเตอร์ที่ได้จากการกำหนดโครงสร้างของแขน ตามวิธีของ Denavit-Hartenberg, ชนิดของข้อต่อ, มวล, ตำแหน่งของจุดศูนย์กลางมวล, ค่าพารามิเตอร์ในเมตริกซ์เทนเซอร์ความเฉื่อย, ค่าพารามิเตอร์ของมอเตอร์ ได้แก่ ความเฉื่อยของโรเตอร์, อัตราส่วนลดของเกียร์, ทิศทางของโรเตอร์, ค่าสัมประสิทธิ์ความเสียดทานจากความหนืด และความเสียดทานคูลอมบ์ ซึ่งรายละเอียดของพารามิเตอร์สามารถดูได้ในภาคผนวก ก ตัวอย่างของไฟล์ข้อมูลสำหรับแขนแบบ 5 ข้อต่อแบบหมุน แสดงในตาราง 3.1

3.1.2 ฟังก์ชันภายในคลังโปรแกรม

หลังจากกำหนดโครงสร้างและค่าพารามิเตอร์ของแขนกลแล้ว ผู้ใช้มีหน้าที่เขียนฟังก์ชันหลักสำหรับการจำลองระบบ ซึ่งฟังก์ชันหลักนี้ จะทำหน้าที่เรียกฟังก์ชันอื่นภายในคลังโปรแกรมมาใช้งาน โดยฟังก์ชันในคลังโปรแกรมนี้อาจแบ่งได้เป็น 3 กลุ่ม คือ

1. ฟังก์ชันการคำนวณในส่วนของแขนกล

- DirectKinematic ใช้หาเมตริกซ์การแปลงระหว่างเฟรมต่าง ๆ ของแขนกล
- JacobianKinematic ใช้หาจาโคเบียนเมตริกซ์
- InverseDynamic ใช้หาค่าแรงบิดที่กระทำต่อแขนกล ณ ตำแหน่ง q และ ความเร็ว \dot{q}
- InertiaMatrix ใช้หาเมตริกซ์ความเฉื่อย (B) ของแขนที่ตำแหน่ง q
- selectTorque ให้เลือกชนิดของแรงที่กระทำต่อแขนกลในขณะนั้น ว่าเป็นแรงที่เกิดเนื่องจากผลของความเฉื่อยของแขน หรือผลของแรงโคริโอลิสกับแรงเข้าสู่ศูนย์กลาง หรือผลของแรงโน้มถ่วง
- DirectDynamic ใช้คำนวณหาค่า \ddot{q} เมื่อกำหนดค่าแรงบิดให้กับแขนกล

2. ฟังก์ชันจัดการเกี่ยวกับเมตริกซ์

ฟังก์ชันในส่วนนี้เกี่ยวข้องกับคำนวณของเมตริกซ์ ได้แก่ การจองและคืนหน่วยความจำสำหรับตัวแปรเมตริกซ์, การบวกลบคูณเมตริกซ์ และการหาอินเวอร์สเมตริกซ์ สำหรับวิธีการหาอินเวอร์สเมตริกซ์ ใช้วิธี LU Decomposition [16]

3. ฟังก์ชันอื่น ๆ ได้แก่

- rk4 ใช้ในการแก้สมการอนุพันธ์ โดยวิธี Runge-Kutta อันดับ 4 [16] เพื่อหาค่าของ q และ \dot{q} จาก แรงบิดที่ป้อนให้กับแขนกล
- trajectory ใช้ในการหาค่า t_1 และ t_2 สำหรับสร้างแนวทางการเคลื่อนที่ที่ต้องการ (desired trajectory) โดยกำหนดให้ แนวทางการเคลื่อนที่ เริ่มจากตำแหน่ง q_i ไปยัง q_f ภายในเวลา t_f และมีลักษณะกราฟความเร็วเชิงมุมของข้อต่อ เป็นดังรูป

จำนวนข้อต่อ	⇒	5							
$q \quad \dot{q} \quad \ddot{q}$	⇒	0.0	0.0	0.0					
		0.0	0.0	0.0					
		0.0	0.0	0.0					
		0.0	0.0	0.0					
		0.0	0.0	0.0					
พารามิเตอร์	⇒	0.1	0.36	90.0	0.0	0	11.0	-0.0682	-0.0818
ของแขนกล	→	0.0	0.1075	0.1052	0.0175	-0.0071	-0.001	-0.001	
แต่ละท่อน	→	1e-6	100.0	0	0	1	0	0	
		0.27	0.1	0.0	0.0	0	6.0	-0.1125	0.0
	→	0.0	0.0051	0.1153	0.1152	-0.001	-0.001	-0.001	
	→	1e-6	100.0	0	0	1	0	0	
		0.23	-0.06	0.0	0.0	0	6.0	-0.0958	0.0
	→	0.0	0.0051	0.1111	0.1111	-0.001	-0.001	-0.001	
	→	1e-6	100.0	0	0	1	0	0	
		0.0	-0.04	90.0	0.0	0	6.0	0.0	0.0
	→	0.0583	0.1022	0.1022	0.0050	-0.001	-0.001	-0.001	
	→	1e-6	100.0	0	0	1	0	0	
		0.0	0.25	0.0	0.0	0	5.0	0.0	0.0
	→	-0.075	0.1	0.1	0.005	-0.001	-0.001	-0.001	
	→	1e-6	100.0	0	0	1	0	0	
เมตริกซ์การแปลง	⇒	1.0	0.0	0.0	0.0				
		0.0	1.0	0.0	0.0				
		0.0	0.0	1.0	0.0				
		0.0	0.0	0.0	1.0				
แรงภายนอก	⇒	0.0	0.0	0.0					
โมเมนต์ภายนอก	⇒	0.0	0.0	0.0					
แรงโน้มถ่วง	⇒	0.0	0.0	-9.81					

ตารางที่ 3.1: ตัวอย่างไฟล์ข้อมูลของแขนกลชนิด 5 ข้อต่อแบบหมุน

สำหรับรายชื่อฟังก์ชันอื่น ๆ นอกเหนือจากนี้ และวิธีการใช้งานของทุกฟังก์ชัน สามารถดูได้ในส่วนของภาคผนวก ก

3.2 การเปรียบเทียบผลการคำนวณระหว่าง คลังโปรแกรมกับ Robotic Toolbox [5]

เนื้อหาในส่วนนี้เป็น ตัวอย่างของการเปรียบเทียบผลการคำนวณที่ได้จากคลังโปรแกรม กับ โปรแกรม Robotic Toolbox ซึ่งทำงานบน Matlab 5.3 (R11) สาเหตุที่ต้องมีเนื้อหาในส่วนนี้ คือ เพื่อให้ผลการคำนวณในส่วนต่าง ๆ ของคลังโปรแกรมนี้มีความน่าเชื่อถือ ตั้งแต่ การหาจลนศาสตร์และสมการพลวัตของแขน ไปจนถึง การหาค่าพารามิเตอร์และวิธีในการแก้สมการอนุพันธ์ ลักษณะของแขนกลที่ใช้เป็นแขนกลชนิด 5 ข้อต่อแบบหมุน ซึ่งมีลักษณะเดียวกับแขนกลในห้องปฏิบัติการวิจัยระบบควบคุม โดยมีค่าพารามิเตอร์ตามตารางที่ 3.1 ซึ่งค่าพารามิเตอร์บางตัว เช่น ความยาวของแขนท่อนต่าง ๆ ได้จาก ค่าพารามิเตอร์จริงซึ่งอยู่ในตัวควบคุม

3.2.1 ผลของการหาจลนศาสตร์และสมการพลวัต

เนื่องจากเราสามารถอ่านโปรแกรมไฟล์ต้นฉบับของ Robotic Toolbox ได้ จึงสามารถทราบวิธีการ เขียนโปรแกรม ซึ่งเมื่อเปรียบเทียบขั้นตอนการคำนวณของโปรแกรมทั้งสอง พบว่ามีขั้นตอนเหมือนกัน จึงแน่ใจได้ว่าการคำนวณในส่วนนี้ของคลังโปรแกรมมีความถูกต้อง นอกจากนี้ได้ลองเปรียบเทียบจาก ค่าตัวเลขที่ได้จากการคำนวณ ปรากฏว่าตัวเลขมีค่าตรงกันถึงทศนิยมตำแหน่ง 15 ซึ่งเป็นความละเอียดสูงสุดในการแสดงค่าของ Matlab ดังในตัวอย่างข้างท้าย แต่เพื่อความสะดวกในการนำเสนอ จึงไม่แสดงตัวเลขทศนิยมจนครบทุกตำแหน่ง

1. จลนศาสตร์โดยตรง ได้กำหนดให้แขนกลอยู่ที่ตำแหน่ง $q = [\pi, \pi/2, \pi/3, \pi/4, \pi/5]$ จะได้เมตริกซ์การแปลงระหว่างเฟรมที่ 0 กับ 5 เป็น

$$A_0^5(q) = \begin{bmatrix} 0.781450 & -0.567756 & 0.258819 & 0.163890 \\ 0.587785 & 0.809016 & 0.0 & 0.0 \\ -0.209389 & 0.152130 & 0.965925 & 0.986481 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (3.1)$$

2. จลนศาสตร์เชิงอนุพันธ์ กำหนดให้แขนกลอยู่ที่ตำแหน่ง $q = [\pi, \pi/2, \pi/3, \pi/4, \pi/5]$ จะได้จาโคเบียนเมตริกซ์ เป็น

$$J(q) = \begin{bmatrix} 0.0 & 0.626481 & 0.356481 & 0.241481 & 0.0 \\ 0.163890 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -0.263890 & -0.263890 & -0.064704 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.258819 \\ 0.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.965925 \end{bmatrix} \quad (3.2)$$

3. แรงบิดที่กระทำต่อแขนกล กำหนดให้แขนกลมีค่าของตำแหน่ง ความเร็ว และความเร่งเป็น $q = \dot{q} = \ddot{q} = [\pi, \pi/2, \pi/3, \pi/4, \pi/5]$ จะได้ค่าของแรงบิดซึ่งหาจากการทำ Inverse Dynamic เป็น

$$\tau = [12.646935 - 32.046272 - 26.776050 - 3.985348 - 0.001293]$$

4. เมตริกซ์ความเฉื่อยของแขนกลที่ตำแหน่ง $q = [\pi, \pi/2, \pi/3, \pi/4, \pi/5]$ มีค่าเป็น

$$B(q) = \begin{bmatrix} 0.551260 & -0.172793 & -0.014493 & 0.001976 & 0.004868 \\ -0.172793 & 4.628813 & 2.345896 & 0.895392 & 0.001396 \\ -0.014493 & 2.345896 & 1.586116 & 0.575964 & 0.001396 \\ 0.001976 & 0.895392 & 0.575964 & 0.386669 & 0.001396 \\ 0.004868 & 0.001396 & 0.001396 & 0.001396 & 0.0150 \end{bmatrix} \quad (3.3)$$

3.2.2 ผลการหาเมตริกซ์ผกผัน

เนื่องจากอัลกอริทึมในการหาเมตริกซ์ผกผันของ Matlab มีอยู่หลายตัว ซึ่งการเลือกใช้ อัลกอริทึมใดนั้น ตัว Matlab จะทำการตัดสินใจเองจาก ผลของความถูกต้องในการคำนวณ รวมทั้ง การที่เราไม่ทราบว่าแต่ละอัลกอริทึมมีขั้นตอนการทำงานอย่างไร ในขณะที่คลังโปรแกรมนี้ใช้วิธี เดียวจากการทำ LU Decomposition [16] ทำให้ผลของค่าเมตริกซ์ผกผันที่ได้จึงแตกต่างกันบ้าง แต่เพื่อ เป็นการเปรียบเทียบว่าวิธีใดให้ผลการคำนวณที่ถูกต้องมากกว่า และแตกต่างกันเล็กน้อยแค่ไหน จึง ได้ทำการสุ่มเมตริกซ์จากเครื่องคอมพิวเตอร์มา 10 ตัว ซึ่งมีขนาดเป็น 10×10 แล้วหาค่าความคลาด เคลื่อนจากสมการ

$$e = \sqrt{\text{trace}((I - A * B)^T (I - A * B))} \quad (3.4)$$

โดยที่ B คือ เมตริกซ์ผกผันของ A ที่ได้จากการคำนวณ ผลของค่าความคลาดเคลื่อนดังแสดง ในตารางที่ 3.2 พบว่าผลที่ได้จาก Matlab มีความถูกต้องมากกว่าเล็กน้อย โดยมีค่าเฉลี่ยของความ คลาดเคลื่อนเป็น 9.850×10^{-15} ในขณะที่ ค่าความคลาดเคลื่อนโดยเฉลี่ยของคลังโปรแกรมเป็น 2.071×10^{-14}

3.2.3 ผลการแก้สมการอนุพันธ์

ในโปรแกรม Matlab วิธีการแก้สมการอนุพันธ์มีให้เลือกใช้หลายวิธีตามแต่ละชนิดของระบบ หลักเกณฑ์หนึ่งในการเลือกใช้วิธีเหล่านั้น คือ การเปรียบเทียบผลการคำนวณที่ได้จากหลายๆ วิธี ถ้าผลการคำนวณเหล่านั้นมีค่าต่างกันน้อยมาก ภายในขอบเขตที่ยอมรับได้ แสดงว่าผลที่ได้มีความ น่าเชื่อถือ แต่ถ้าวิธีใดให้ผลแตกต่างจากวิธีอื่นไปมากก็ควรที่จะหลีกเลี่ยงการใช้วิธีนั้นกับระบบที่ทำการ จำลองอยู่

สำหรับการเลือกใช้วิธีแก้สมการอนุพันธ์ของแขนหุ่นยนต์นั้น ควรพิจารณาจากหลักการข้าง ดัน โดยในที่นี้ได้ลองเปรียบเทียบระหว่างวิธี ode3 (Bogcki-Shampine), ode4 (Runge-Kutta) และ ode5 (Dormand-Prince) ของ Matlab ซึ่งแตกต่างกันตามจำนวนอันดับที่ใช้แก้สมการ กำหนดให้แขนกลที่ใช้

เมตริกซ์ตัวที่	ค่า e จากคลังโปรแกรม ($\times 10^{-12}$)	ค่า e จาก Matlab ($\times 10^{-12}$)
1	0.00450801720069	0.002432744808268
2	0.01748183712541	0.008597611563832
3	0.14211506355398	0.052598612920776
4	0.00210537114564	0.001859604782881
5	0.00445418897049	0.004335675724154
6	0.00950221652834	0.008404805238430
7	0.00741150483758	0.006103842419051
8	0.00656610254874	0.003975824093649
9	0.00468561631726	0.003300265708123
10	0.00833367747230	0.006892735511618

ตารางที่ 3.2: ค่าความคลาดเคลื่อนจากการหาเมตริกซ์ผกผัน

เป็นแบบ 5 ข้อต่อ ควบคุมโดยวิธีพลวัตผกผัน เคลื่อนที่จาก 0 rad ไปยัง $\pi/2$ rad ทุกข้อต่อ และใช้ค่าเวลาสุ่มตัวอย่างเป็น 0.01 วินาที

ผลการเปรียบเทียบในรูปที่ 3.1 และ 3.2 สังเกตได้ว่าทั้งสามวิธีให้ผลแตกต่างกันน้อยมาก เมื่อเทียบกับค่าของตำแหน่งของแกนกลที่อยู่ระหว่าง 0 ถึง $\pi/2$ จึงสามารถใช้วิธีใดก็ได้ ในการแก้สมการ สำหรับในส่วนของคลังโปรแกรมจะใช้อัลกอริทึมของ Runge-Kutta อันดับ 4 [16] เป็นหลัก เนื่องจากให้ผลความแตกต่างจากกรณีอันดับ 5 กับอันดับ 4 น้อยกว่าความแตกต่างของกรณีอันดับ 5 กับอันดับ 3 ในขณะเดียวกันจะประหยัดเวลาในการคำนวณ เนื่องจากมีขั้นตอนในการแก้สมการน้อยกว่า ซึ่งจะมีผลมากเมื่อต้องการจำลองระบบเป็นเวลานาน หรือใช้เวลาสุ่มตัวอย่างมีค่าน้อย ๆ

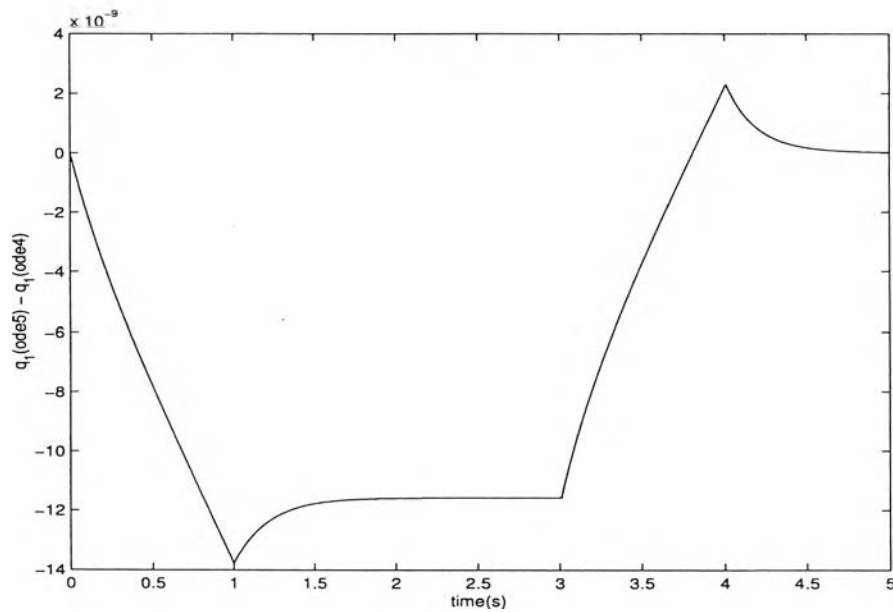
นอกจากนี้ ในการแก้สมการอนุพันธ์จากโปรแกรม Matlab ที่ต่างรุ่นกัน อาจจะได้ผลที่แตกต่างกัน ดังตัวอย่างในรูปที่ 3.3 ซึ่งเป็นความแตกต่างของมุมข้อต่อที่ 1 ที่ได้จากวิธี ode4 ของ Matlab รุ่นที่ 5.3 กับ 5.1 ความแตกต่างนี้เกิดจากอัลกอริทึมสำหรับหาเมตริกซ์ผกผัน และการแก้สมการอาจต่างกัน ในส่วนของการแก้สมการอนุพันธ์ แม้ว่าทั้งสองรุ่นใช้ Runge-Kutta อันดับ 4 เหมือนกัน แต่ในการทำให้เกิดผล (implementation) อาจแตกต่างกัน

ขั้นต่อมาเป็นการทดสอบความแตกต่างของอัลกอริทึม Runge-Kutta อันดับ 4 ของคลังโปรแกรมกับ ode3, ode4 และ ode5 ของ Matlab โดยใช้ระบบง่าย ๆ เป็นตัวอย่างในการเปรียบเทียบ ระบบที่ใช้มีสมการสถานะ $\dot{x} = -2x + 1$

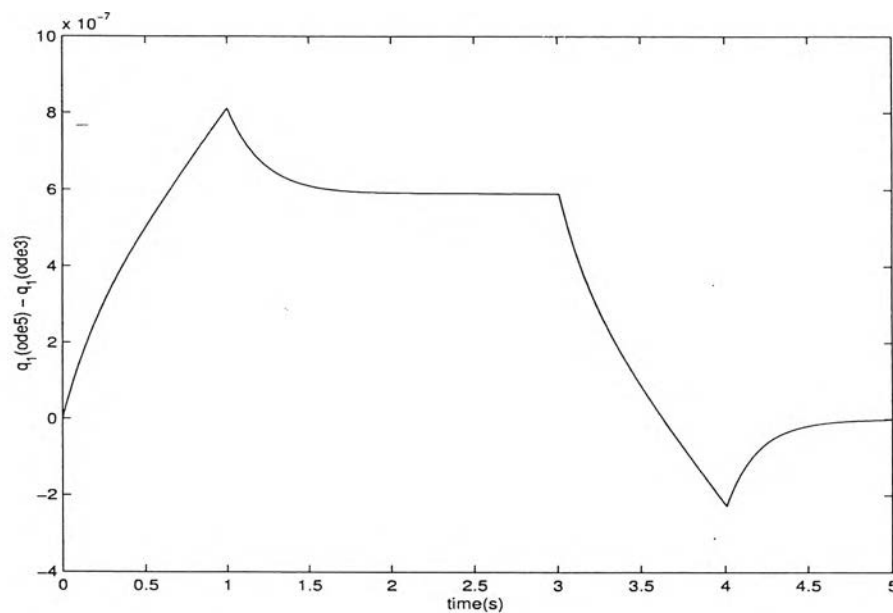
ผลการเปรียบเทียบระหว่าง Runge-Kutta อันดับ 4 ของคลังโปรแกรมกับของ Matlab ในรูปที่ 3.4 นั้น ความแตกต่างที่เกิดมีค่าน้อยมากเมื่อเทียบกับ กรณีที่เปรียบเทียบกับวิธี ode3 และ ode5 ในรูปที่ 3.5 และ 3.6

นอกจากนั้น ได้ทำการเปรียบเทียบผลของ Runge-Kutta อันดับ 4 ของคลังโปรแกรมกับของ Matlab ในกรณีที่ระบบขาดเสถียรภาพ โดยใช้ระบบที่มีสมการสถานะเป็น $\dot{x} = 2x + 1$

จากรูปที่ 3.7 จะเห็นว่าความแตกต่างของทั้งสองวิธีมีค่ามากขึ้น เนื่องจากวิธีการเก็บค่าตัว

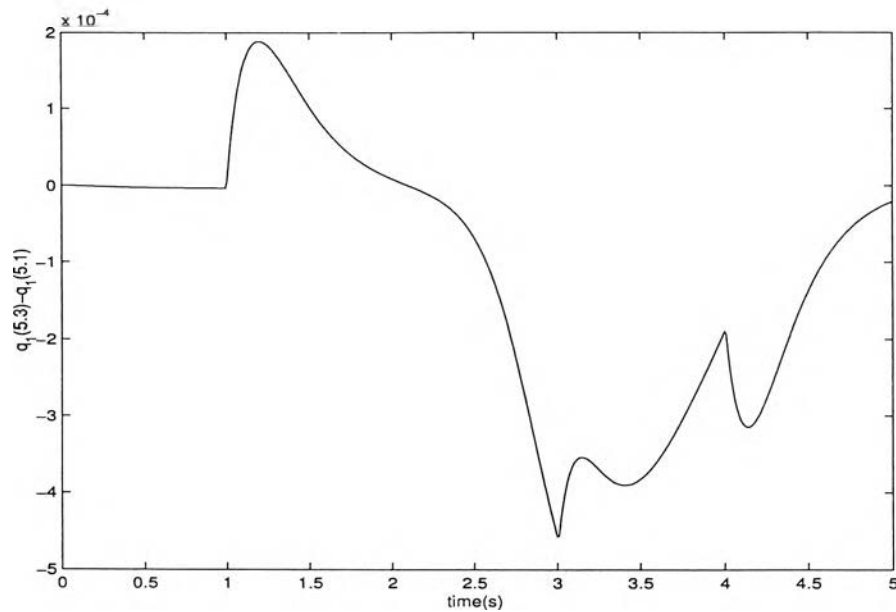


รูปที่ 3.1: ความแตกต่างของมุมของข้อต่อที่ 1 ซึ่งได้จาก ode5 และ ode4 ของ Matlab

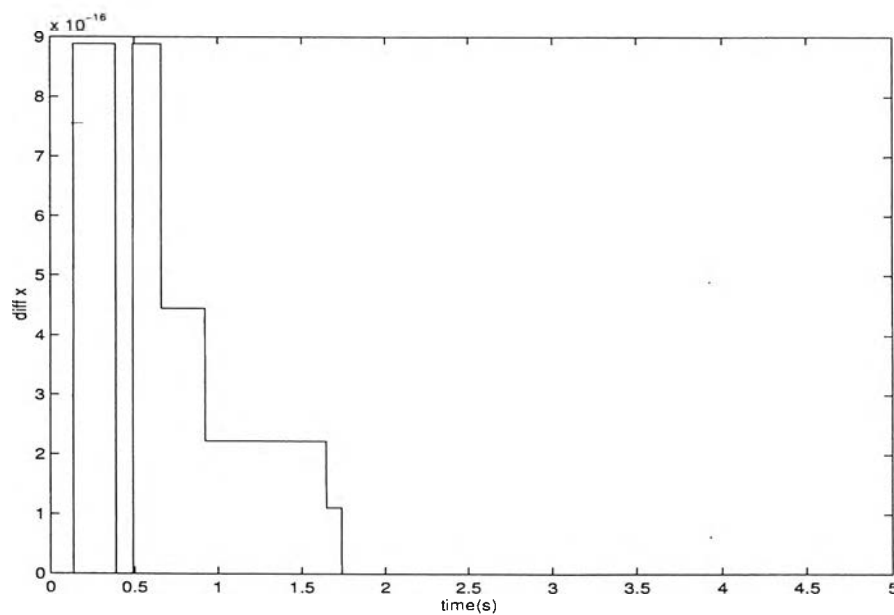


รูปที่ 3.2: ความแตกต่างของมุมของข้อต่อที่ 1 ซึ่งได้จาก ode5 และ ode3 ของ Matlab

เลขทศนิยมใด ๆ ของทั้ง Matlab และ ภาษาซี จะเก็บเฉพาะ ตัวเลขแมนทิสซา (mantissa) ซึ่งมีค่าระหว่าง 1 ถึง 10 กับส่วนของเลขยกกำลัง ดังนั้นเมื่อระบบขาดเสถียรภาพ ค่าตัวแปรสถานะจะมีค่าสูงชันเรื่อย ๆ ทำให้เมื่อเลขยกกำลังมีค่ามาก ค่าตัวเลขทศนิยมตำแหน่งท้ายๆ ของแมนทิสซา ซึ่งอาจต่างกันเล็กน้อย มีผลต่อความแตกต่างที่เกิดมากขึ้น ซึ่งถ้าพิจารณาความแตกต่างที่เกิด ณ เวลาต่าง ๆ เป็นร้อยละของค่าตัวแปรสถานะในขณะนั้น ดังรูปที่ 3.8 จะเห็นว่าเมื่อคิดค่าเป็นร้อยละ ผลความแตกต่าง ที่เกิดจะอยู่ภายในขอบเขตหนึ่งซึ่งมีค่าน้อยมาก จากตัวอย่างนี้ทำให้การเปรียบเทียบ



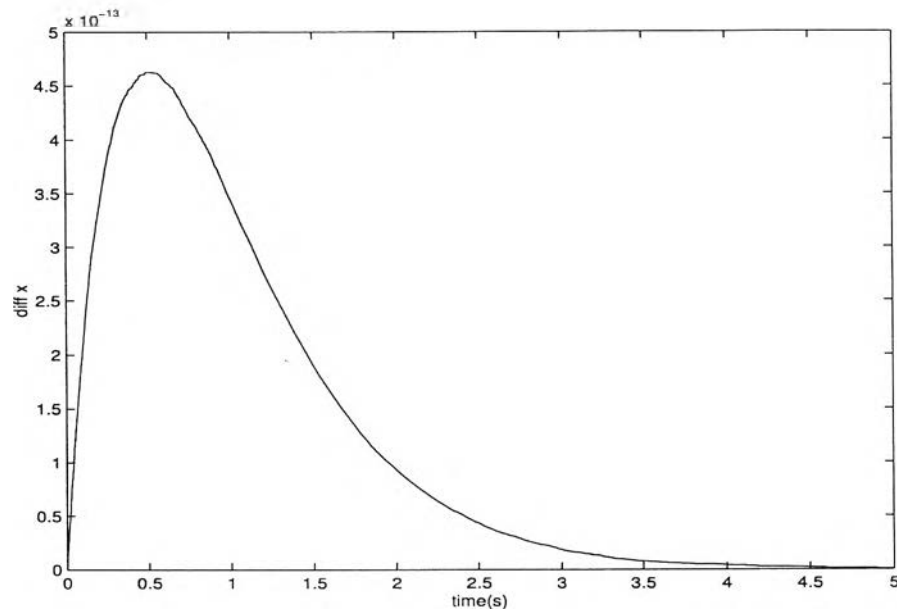
รูปที่ 3.3: ความแตกต่างของมุมของข้อต่อที่ 1 ซึ่งได้จาก ode4 ของ Matlab รุ่นที่ 5.3 และ 5.1



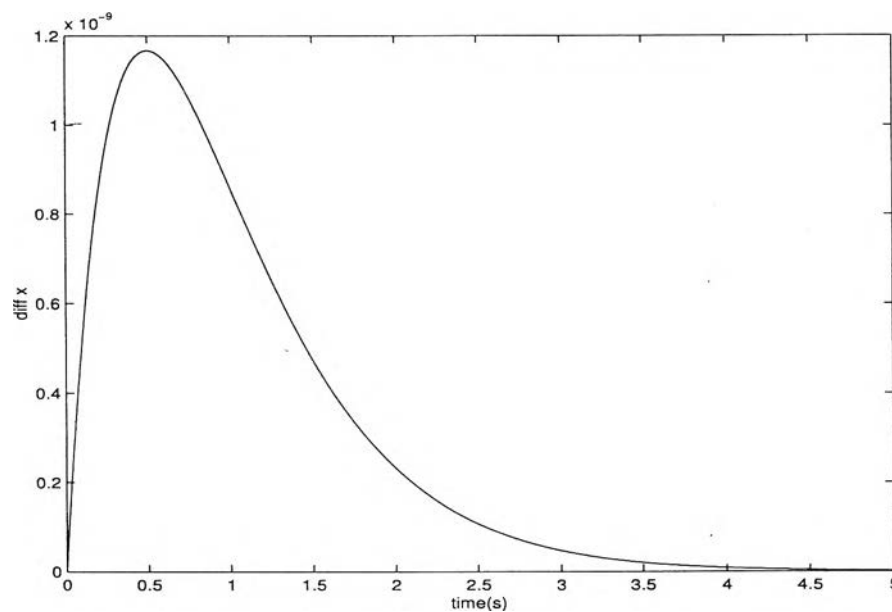
รูปที่ 3.4: ความแตกต่างของตัวแปรสถานะที่ได้จาก Runge-Kutta อันดับ 4 ของคลังโปรแกรมกับของ Matlab

ผลการจำลองเมื่อระบบเป็นแขนหุ่นยนต์ ไม่สามารถที่จะใช้สัญญาณเข้าแรงบิดเป็นค่าคงที่ได้ เนื่องจากจะทำให้ระบบขาดเสถียรภาพ จำเป็นที่ต้องมีการป้อนกลับ เพื่อควบคุมการเคลื่อนที่ของแขน

จากนั้นจะทำการเปรียบเทียบผลการคำนวณ Runge-Kutta อันดับ 4 ของคลังโปรแกรมกับ Matlab เมื่อใช้ในการจำลองระบบแขนหุ่นยนต์ชนิด 5 ข้อต่อแบบหมุน เป็นจำนวน 12 กรณี ซึ่งแต่ละกรณีได้ทำการเปลี่ยนตำแหน่งเริ่มต้นไปยังจุดต่าง ๆ ในปริภูมิ เปลี่ยนช่วงเวลาสุ่ม และเปลี่ยนวิธีการ



รูปที่ 3.5: ความแตกต่างของตัวแปรสถานะที่ได้จาก ode4 และ ode5 ของ Matlab

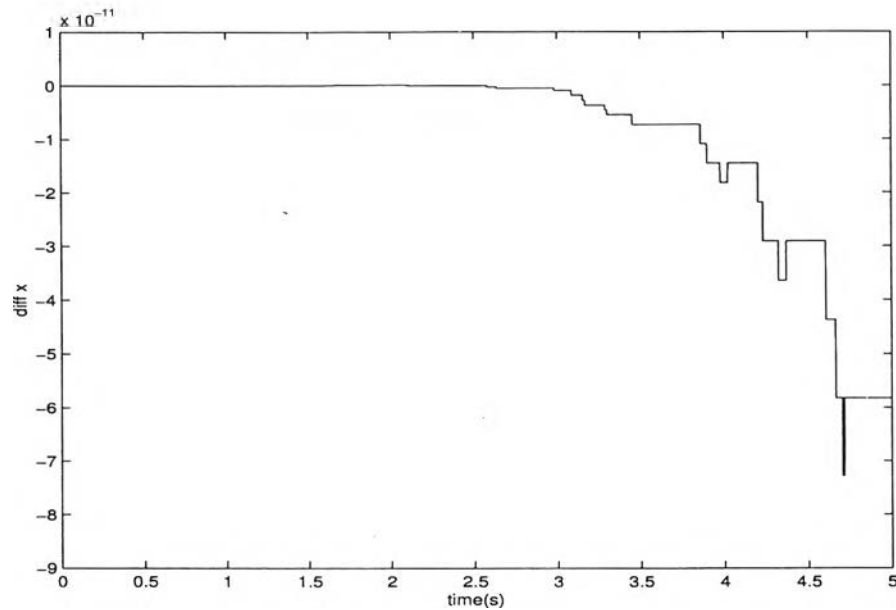


รูปที่ 3.6: ความแตกต่างของตัวแปรสถานะที่ได้จาก ode4 และ ode3 ของ Matlab

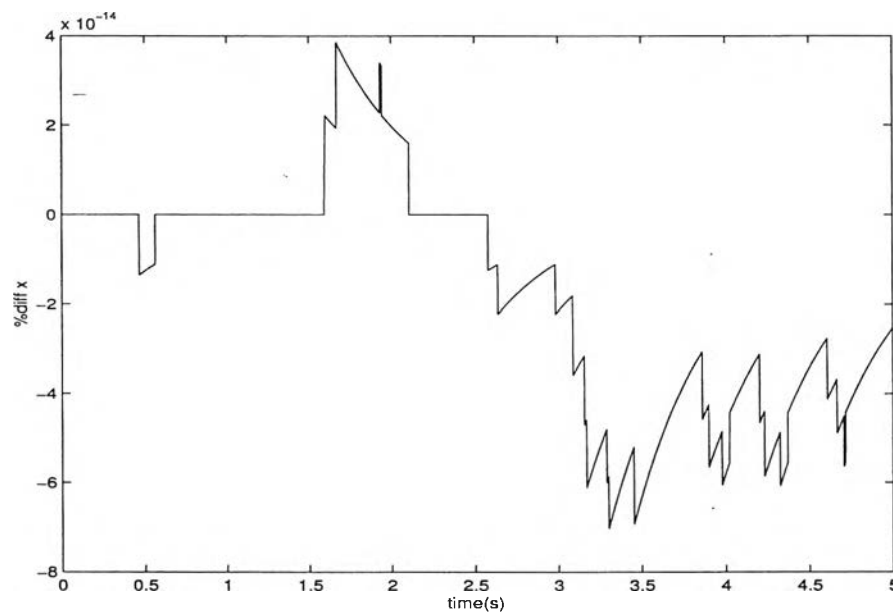
ควบคุม แต่ใช้ระยะเวลาในการจำลองนาน 5 วินาที เท่ากันดังในตารางที่ 3.3 แล้วเปรียบเทียบความแตกต่างที่เกิดขึ้นของมุมในแต่ละข้อต่อ ซึ่งคำนวณจาก

$$\text{err} = \sqrt{\frac{1}{k} \sum_{i=1}^k (q_{\text{matlab}}[i] - q_{\text{library}}[i])^2}$$

จากตารางที่ 3.4 แสดงให้เห็นว่าค่าความแตกต่างโดยเฉลี่ยของทั้ง 12 กรณี สำหรับข้อต่อที่



รูปที่ 3.7: ความแตกต่างของตัวแปรสถานะโดยวิธี Runge-Kutta อันดับ 4 กรณีที่ระบบขาดเสถียรภาพ



รูปที่ 3.8: ค่าร้อยละของความแตกต่างของตัวแปรสถานะโดยวิธี Runge-Kutta อันดับ 4 กรณีที่ระบบขาดเสถียรภาพ

ต่อที่ 1 ถึง 5 มีค่าเป็น 4.012×10^{-3} , 4.166×10^{-3} , 4.039×10^{-3} , 3.769×10^{-3} และ 3.182×10^{-3} ตามลำดับ ซึ่งมีค่าน้อยมาก เมื่อเทียบกับค่าตำแหน่งของข้อต่อ ทำให้ผลการคำนวณจากคลังโปรแกรมมีความเชื่อถือได้ในระดับหนึ่ง สาเหตุหนึ่งที่ทำให้ได้ผลการคำนวณต่างกัน มาจากขั้นตอนการแก้สมการอนุพันธ์ของแขนกล ต้องอาศัย การหาเมตริกซ์ผกผัน ผลของความแตกต่างนี้จะสะสมมากขึ้น จากรูปที่ 3.9-3.11 แสดงให้เห็นว่า ความแตกต่างของโปรแกรมทั้งสองมีค่าเพิ่มขึ้น

กรณีที	q_i (rad)	q_f (rad)	Ts (ms)	วิธีควบคุม
1	[0, 0, 0, 0, 0]	$[\pi/2, \pi/2, \pi/2, \pi/2, \pi/2]$	1	INV
2	$[\pi/2, \pi/2, \pi/2, \pi/2, \pi/2]$	$[\pi, \pi, \pi, \pi, \pi]$	1	INV
3	$[\pi, \pi, \pi, \pi, \pi]$	$[3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2]$	1	INV
4	$[3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2]$	$[2\pi, 2\pi, 2\pi, 2\pi, 2\pi]$	1	INV
5	$[\pi/2, \pi/2, \pi/4, \pi/4, \pi/4]$	$[\pi, \pi, 3\pi/4, 3\pi/4, 3\pi/4]$	1	INV
6	[0, 0, 0, 0, 0]	$[\pi/2, \pi/2, \pi/2, \pi/2, \pi/2]$	10	INV
7	$[\pi/2, \pi/2, \pi/2, \pi/2, \pi/2]$	$[\pi, \pi, \pi, \pi, \pi]$	10	INV
8	$[\pi, \pi, \pi, \pi, \pi]$	$[3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2]$	10	INV
9	$[3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2, 3\pi/2]$	$[2\pi, 2\pi, 2\pi, 2\pi, 2\pi]$	10	INV
10	$[\pi/2, \pi/2, \pi/4, \pi/4, \pi/4]$	$[\pi, \pi, 3\pi/4, 3\pi/4, 3\pi/4]$	10	INV
11	[0, 0, 0, 0, 0]	$[\pi/2, \pi/2, \pi/2, \pi/2, \pi/2]$	1	VSC
12	[0, 0, 0, 0, 0]	$[\pi/2, \pi/2, \pi/2, \pi/2, \pi/2]$	1	PID

ตารางที่ 3.3: กรณีต่างๆ ที่ใช้ในการเปรียบเทียบการคำนวณระหว่างจากคลังโปรแกรมกับ Matlab

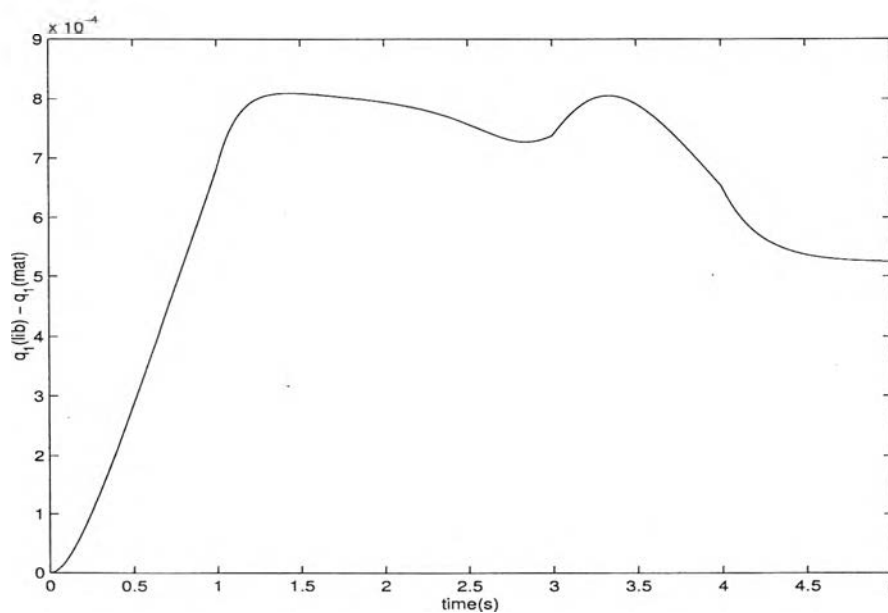
ในช่วงที่ข้อต่อหมุนด้วยความเร่ง (ช่วง 0-1 วินาที) และเมื่อตำแหน่งของข้อต่อเข้าสู่ตำแหน่งสุดท้ายแล้ว ค่าความแตกต่างจะค่อย ๆ มีค่าน้อยลง (ช่วง 4-5 วินาที) ขอบเขตของความแตกต่างจะถูกกำหนด โดยขนาดของความเร่ง และช่วงระยะเวลา ที่ข้อต่อหมุนด้วยอัตราเร่ง

ในกรณีของการควบคุมโดยตัว PID ในรูปที่ 3.11 ลักษณะของกราฟจะแตกต่างจากกรณีอื่น เนื่องจากการเคลื่อนที่ของ PID กำหนดให้มีลักษณะจากตำแหน่งเริ่มไปยังตำแหน่งปลาย โดยไม่ได้กำหนดแนวทางการเคลื่อนที่เป็นจุดที่เวลาต่างๆ ถ้าสังเกตกราฟในรูปที่ 3.12 จะเห็นว่า ในช่วงแรก ตำแหน่งข้อต่อที่ 1 ของตัวควบคุม PID มีค่าอัตราเร่งสูงมาก ส่งผลให้ในช่วงนี้ ความแตกต่าง ของคลังโปรแกรมกับ Matlab สูงกว่าของตัวควบคุมแบบพลวัตผกผัน เมื่อถึงช่วงปลายตำแหน่งของ ข้อต่อจากตัวควบคุม PID ลู่เข้าก่อน ทำให้ค่าความแตกต่างในช่วงท้ายมีค่าต่ำกว่า

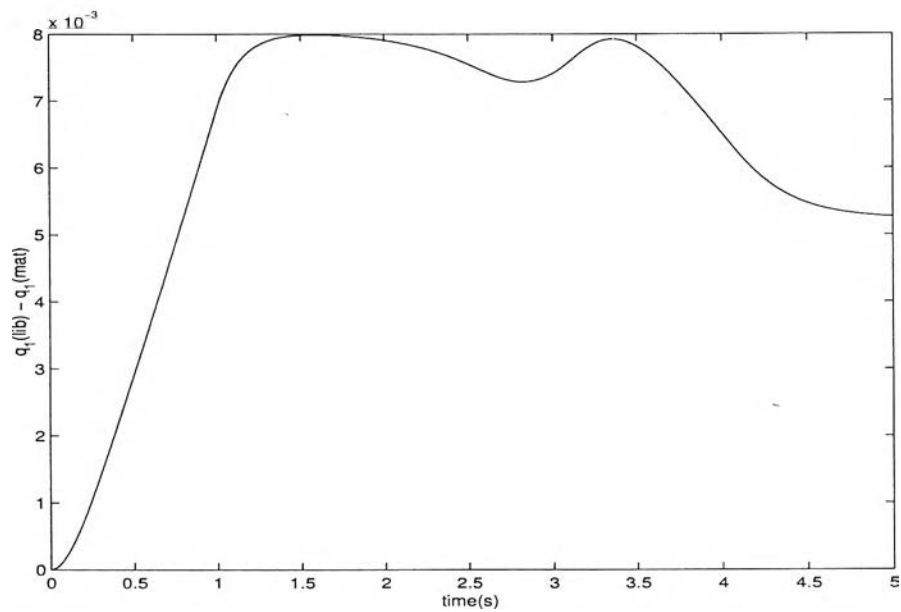
นอกจากนี้ เมื่อทำการเพิ่มช่วงเวลาลู่ตัวอย่าง จะทำให้ค่าความแตกต่างมีค่าเพิ่มขึ้น โดยพิจารณาจาก การหาค่าเฉลี่ยค่าความแตกต่างของตารางที่ 3.4 สำหรับกรณีที 1 ถึง 5 ซึ่งมีช่วงเวลาลู่ตัวอย่างเป็น 1 ms ของข้อต่อที่ 1 ถึง 5 มีค่าเป็น 7.329×10^{-4} , 8.083×10^{-4} , 8.137×10^{-4} , 8.401×10^{-4} และ 5.763×10^{-4} ตามลำดับ เมื่อพิจารณาค่าเฉลี่ยของความแตกต่างสำหรับกรณีที 6 ถึง 10 ซึ่งมีช่วงเวลาลู่ตัวอย่างเป็น 10 ms ของข้อต่อที่ 1 ถึง 5 มีค่าเป็น 7.502×10^{-3} , 6.554×10^{-3} , 8.140×10^{-3} , 8.398×10^{-3} และ 6.955×10^{-3} ตามลำดับ

กรณีที	ค่าความแตกต่างของมุม (err)				
	q_1	q_2	q_3	q_4	q_5
1	6.66474E-04	6.57079E-04	1.14765E-03	5.20136E-04	6.62999E-04
2	8.45827E-04	8.13102E-04	4.84607E-04	1.28679E-04	8.51391E-04
3	8.57020E-04	9.05153E-04	1.23288E-03	5.59019E-04	8.40898E-04
4	8.51597E-04	8.83657E-04	5.72070E-04	1.35208E-03	8.49058E-04
5	8.47970E-04	9.40770E-04	4.77470E-04	1.10447E-03	8.38805E-04
6	6.64139E-03	6.55591E-03	1.13895E-02	5.23727E-03	6.60508E-03
7	8.46732E-03	8.15212E-03	4.90229E-04	1.28022E-03	8.52230E-04
8	8.57990E-03	9.05525E-03	1.22813E-02	6.55597E-03	8.41520E-03
9	8.52069E-03	8.81970E-03	5.81675E-03	1.34207E-02	8.50059E-03
10	8.49111E-03	9.40754E-03	4.80193E-03	1.10207E-02	8.39800E-03
11	9.64550E-04	4.16745E-04	3.57623E-04	6.61592E-04	7.86048E-04
12	2.40718E-03	3.38704E-03	8.56985E-03	3.38166E-03	5.82213E-04

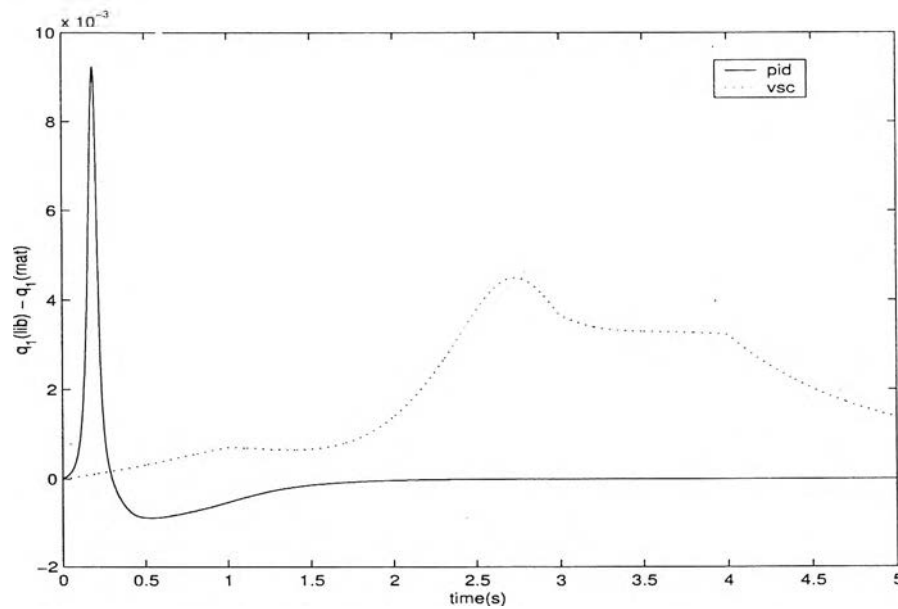
ตารางที่ 3.4: ค่าความแตกต่างของค่ามุม ซึ่งได้จากคลังโปรแกรมกับ Matlab



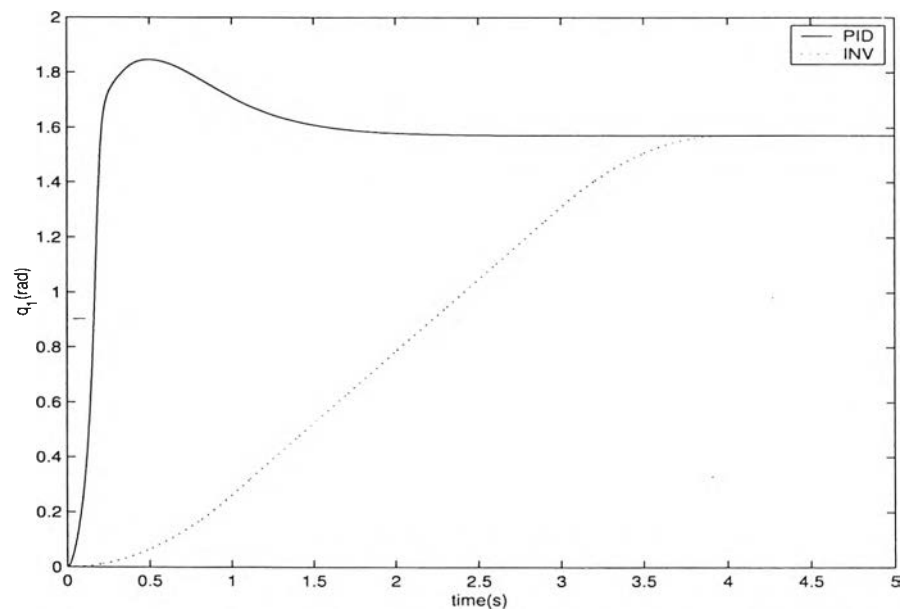
รูปที่ 3.9: ความแตกต่างของตำแหน่งข้อต่อที่ 1 ซึ่งได้จากคลังโปรแกรมกับ Matlab ในกรณีที 1



รูปที่ 3.10: ความแตกต่างของตำแหน่งข้อต่อที่ 1 ซึ่งได้จากคลังโปรแกรมกับ Matlab ในกรณีนี้ที่ 6



รูปที่ 3.11: ความแตกต่างของตำแหน่งข้อต่อที่ 1 ซึ่งได้จากคลังโปรแกรมกับ Matlab ในกรณีนี้ที่ 11 และ 12



รูปที่ 3.12: ตำแหน่งข้อต่อที่ 1 ซึ่งได้จากกรณีที่ 1 และ 11