

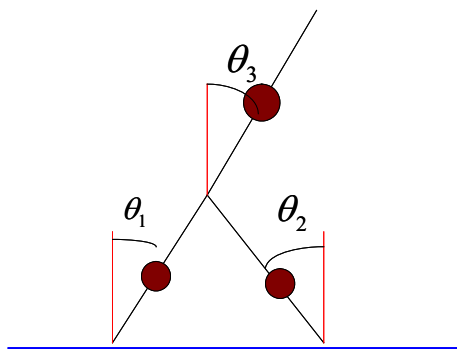


รูปที่ 4.4 แสดงระบบวัดแรงที่เท้า และชุดรับสัญญาณจัดสร้างขึ้น

4.2 หุ่นยนต์สองขาขนาดเล็ก

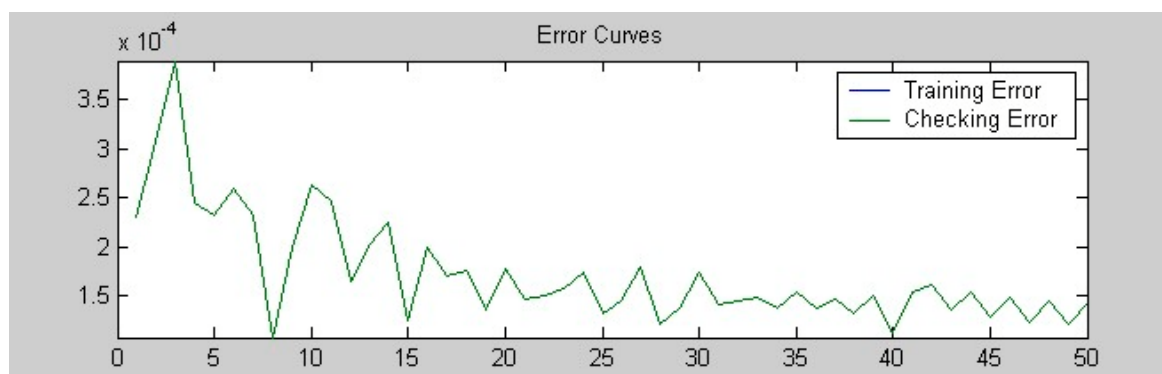
เนื่องจากต้องการศึกษาและวิเคราะห์หาระบบควบคุมแบบ Soft-computing ในการควบคุมลักษณะการเดินของหุ่นยนต์ให้หุ่นยนต์สองขาให้มีการเดินที่มีเสถียรภาพของหุ่นยนต์ 6 แขนก่อน เพื่อเป็นพื้นฐานในการออกแบบระบบควบคุมหุ่นยนต์ 12 แขนของ FIBO ต่อไป จึงได้ออกแบบและสร้างหุ่นยนต์ขึ้น สำหรับทดลองการเคลื่อนที่ทั้งแบบสถิตศาสตร์และพลวัตในแนวด้านข้าง (Sagittal) โดยให้สามารถเดินไปข้างหน้าเพียงอย่างเดียวก่อน ไม่มีการเลี้ยว เดินเป็นทางโค้งหรือเดินถอยหลัง ขึ้นลง บันไดหรือวิ่ง ลักษณะพื้นผิวที่ใช้ทดลองจะเป็นพื้นผิวราบ เรียบ แข็ง ไม่ใช้พื้นผิวขรุขระหรือเอียงขึ้นลง

ในขั้นต้นได้ทำการจำลองระบบหุ่นยนต์สองขา โดยเริ่มต้นที่ 3 แขน มีลักษณะดังรูปที่ 4.5 เราใช้ Adaptive - network - based fuzzy inference system (ANFIS) เป็นตัวควบคุม และใช้ในการหารูปแบบการเดิน(Walking planning level, gait generation level) และการควบคุมแต่ละข้อต่อ โดยไม่จำเป็นต้องมีรายละเอียดทาง kinematics หรือ dynamic ของหุ่นยนต์สองขา ANFIS Controller คือ Sugeon fuzzy controller ซึ่งใช้ความสามารถในการเรียนรู้ด้วยตัวเองจาก adaptive network ทำให้สามารถรวบรวมความรู้จาก fuzzy rule เข้ากับสิ่งที่ได้กับทำงานของหุ่นยนต์ (On-line learning)จากการจำลอง การเคลื่อนที่ของ 3 - link biped robot เราแสดงให้เห็นถึงการออกแบบระบบควบคุมโดยใช้ข้อมูล input-output จากการทดลองในการเรียนรู้และยังแสดงให้เห็นว่าการ online self-learning ของ ANFIS Controller สามารถเพิ่มประสิทธิภาพในการเดินของ Biped robot ได้



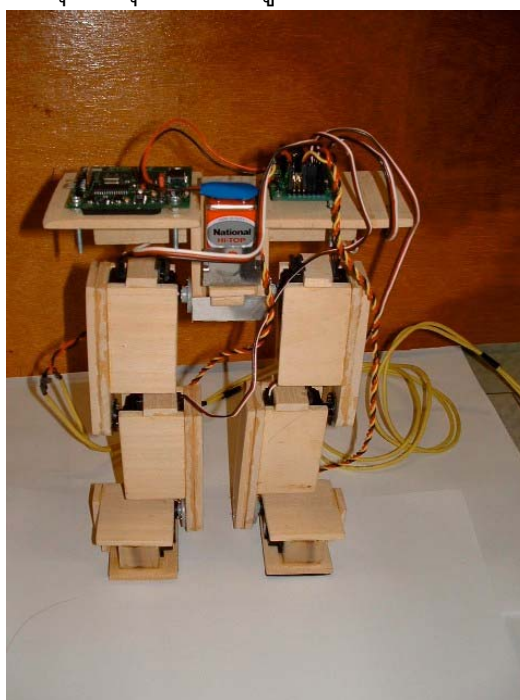
รูปที่ 4.5 แผนภาพหุ่นยนต์ที่นำมาทดลอง

และจากการเรียนรู้พบว่าค่าความผิดพลาดจากการควบคุมด้วย ANFIS ซึ่งมีการเรียนรู้แบบ off-line มีค่าน้อย ดังรูปที่ 4.6

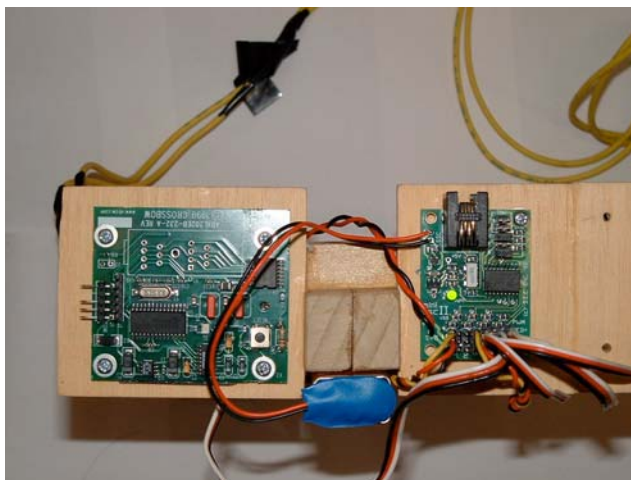


รูปที่ 4.6 ค่าความผิดพลาด เมื่อสอนแบบ Offline ในการทำ Inverse Learning

จากนั้นได้ขยายผลสู่การจัดสร้างหุ่นยนต์ 6-แกน ขึ้นโดยหุ่นยนต์ตัวนี้สร้างจากไม้และขาแต่ละขามีข้อต่อสามข้อต่อ โดยสะโพก หัวเข่าและข้อเท้าอย่างละหนึ่งข้อต่อ เพื่อทดลองการเคลื่อนที่แบบสถิติศาสตร์ ดังรูปที่ 4.7 และติดตั้งระบบควบคุมและอุปกรณ์ตรวจวัดดังรูปที่ 4.8 จากนั้นได้ทำการทดลองควบคุมตำแหน่งของแต่ละข้อต่อให้เดินแบบสถิติศาสตร์ ผลปรากฏว่ายังไม่สามารถเดินได้ (ล้ม) เนื่องจากไม่สามารถควบคุมหุ่นยนต์ให้จุดสมดุลมวลตกอยู่ภายใต้ฝ่าเท้าได้

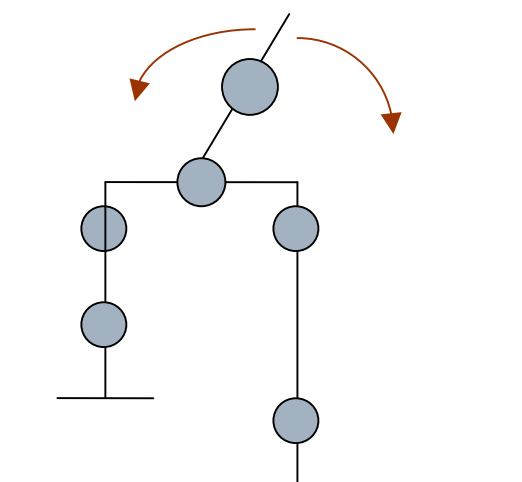


รูปที่ 4.7 รูปหุ่นยนต์ขนาดเล็กที่ทำการสร้างเพื่อทดลองการเดินในแนวด้านข้าง

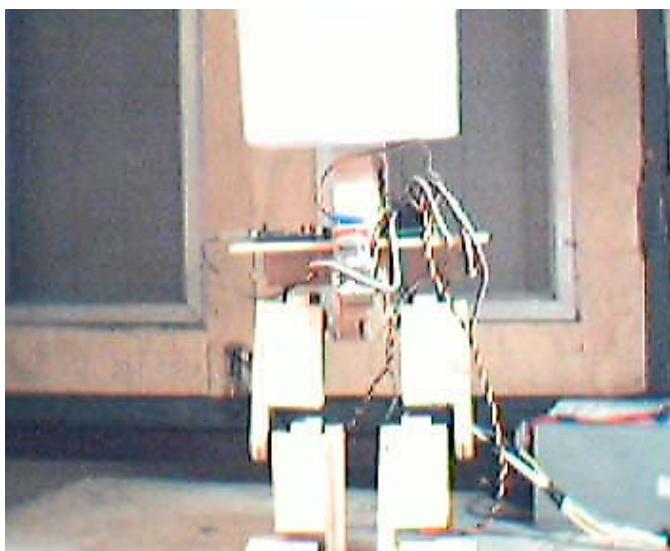


รูปที่ 4.8 ชุดวัดมุมเอียงของหุ่นยนต์ (ด้านขวามือ) และชุดควบคุมมอเตอร์ (ด้านซ้ายมือ)

จากนั้นได้ทำการออกแบบหุ่นยนต์ใหม่โดยเพิ่มแกนอีกหนึ่งแกนที่ลำตัวเพื่อช่วยในการปรับเปลี่ยนจุดศูนย์กลางมวลในแนวด้านหน้า (Frontal plane) ดังรูปที่ 4.9 ซึ่งแสดงแบบจำลองของหุ่นยนต์ 7 แกน และมีลักษณะดังรูปที่ 4.10 และ 4.11 ตามลำดับ



รูปที่ 4.9 แสดงรูปจำลองหุ่นยนต์สองขาแบบ 7 แกน ซึ่งมีการเคลื่อนที่ของมวลในแนวด้าน Lateral เพื่อรักษาสสมดุล



รูปที่ 4.10 แสดงรูปหุ่นยนต์สองขาแบบ 7 แกน ขณะยืน

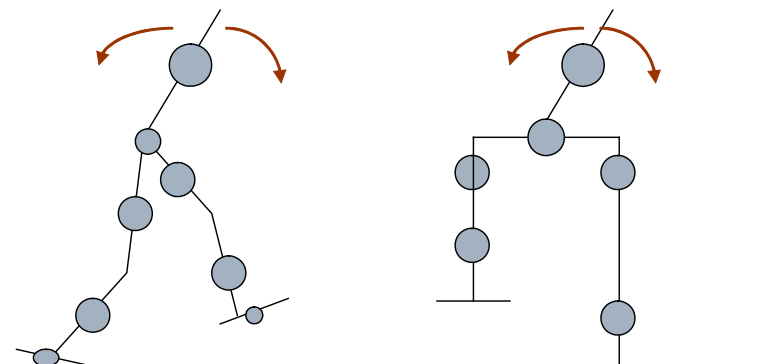


รูปที่ 4.11 แสดงรูปหุ่นยนต์สองขาแบบ 7 แกน และทำการรักษาสมดุลขณะยืนยกขาหนึ่งข้าง

อย่างไรก็ดีเมื่อได้เพิ่มมวลเข้าไปในส่วนลำตัว เพื่อรักษาสมดุลในแนวด้านข้างแล้วนั้นเมื่อทำการทดสอบหุ่นยนต์ให้เดินไปข้างหน้า เราพบว่าเกิดอาการสั่นที่ข้อหัวเข่าเนื่องจากแรงบิดที่ออกแบบไว้ตอนแรกโดยใช้มอเตอร์ของFUTABAรุ่น S3001 มีกำลังบิด 3.0 kg-cm.ที่แรงดัน4.8 V. ไม่พอ จึงได้ทำการปรับเปลี่ยนมอเตอร์เป็นใช้มอเตอร์ของ GWServo รุ่น S03TD มีกำลังบิด 7.0 kg-cm. ที่แรงดัน 4.8 V. แทนและได้ออกแบบเพิ่มเติมจาก 7 แกน เป็น 8 แกน โดยเพิ่มแกนหมุนในแนวด้านข้างเพิ่มอีกหนึ่งแกน ซึ่งรายละเอียดจะได้กล่าวถึงในหัวข้อต่อไป

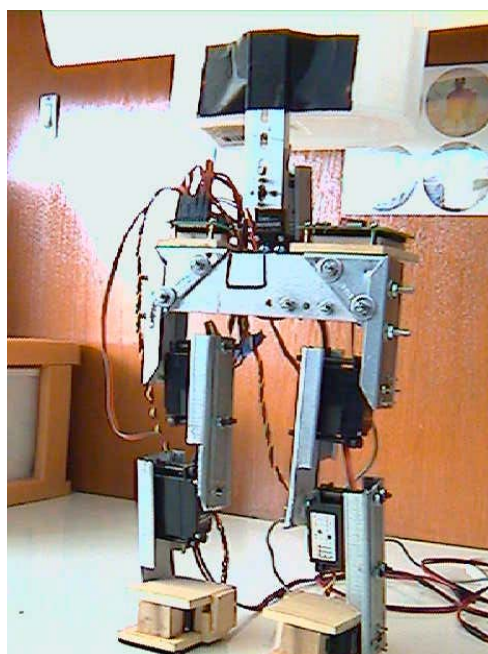
ลักษณะระบบทางกลของหุ่นยนต์ 8 แขน

หุ่นยนต์สองขาที่ใช้ในงานวิจัยนี้ ประกอบด้วยข้อต่อทั้งหมดแปดข้อต่อ ได้แก่ สะโพก(hip) 2 ข้อต่อ หัวเข่า(knee) 2 ข้อต่อ ข้อเท้า(ankle) 2 ข้อต่อ และแกนหมุนที่ลำตัว(Trunk) อีก 2 ข้อต่อ โดยข้อต่อสะโพก หัวเข่าและข้อเท้า มีลักษณะการหมุนแบบแนวระนาบด้านข้าง(Sagittal Plane) และส่วนลำตัว ซึ่งมี 2 แกนนั้นสามารถหมุนได้ในแนวแกนด้านข้าง(Sagittal) และด้านหน้า(Lateral Plane) ดังรูปที่ 4.12



รูปที่ 4.12 แสดงรูปจำลองหุ่นยนต์สองขาแบบ 8 แขน
โดยรูป ก. ด้านSagittal และรูป ข. ด้าน Lateral

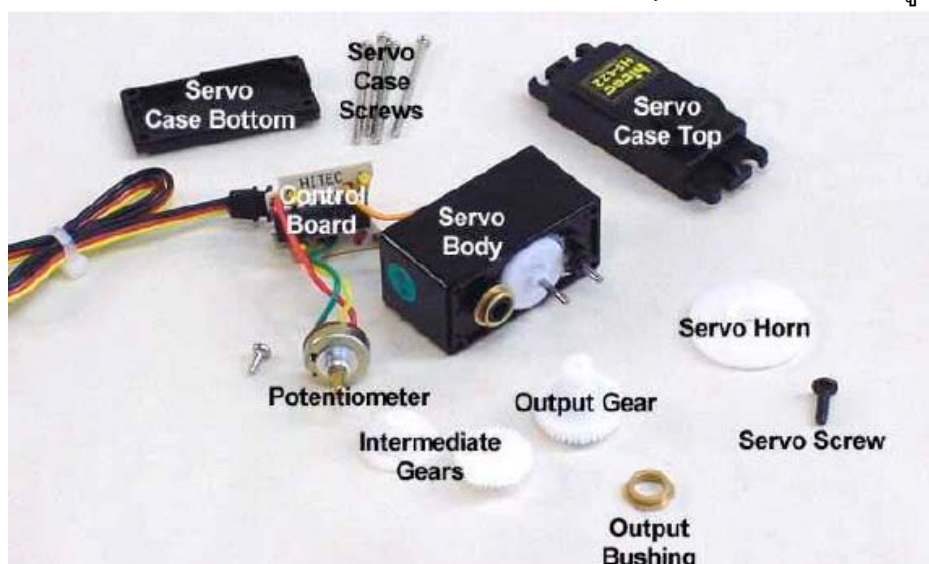
หุ่นยนต์ที่ใช้มีขนาดโดยประมาณ กว้าง 20 ซม. หนา 10 ซม. และขนาดยืนยี่ดขาสุดมีความสูง 45 ซม. ที่ลำตัวด้านขวาติดตั้งแผงวงจรควบคุมมอเตอร์ 8 แขนขนาดกว้าง 3.5 ซม.และยาว 4.5 ซม. และที่ลำตัวด้านซ้ายติดตั้งแผงวงจรวัดค่ามุมเอียงของตัว ขนาดกว้าง 6 ซม. และยาว 6.5 ซม. และที่ฝ่าเท้าติดตั้งตัววัดค่าแรงกด(Pressure Sensor–FSR Unit) ลักษณะของหุ่นยนต์สองขาเป็นดังรูปที่ 4.13



รูปที่ 4.13 แสดงภาพด้านหน้าของหุ่นยนต์สองขาแบบ 8 แขนขนาดยืน

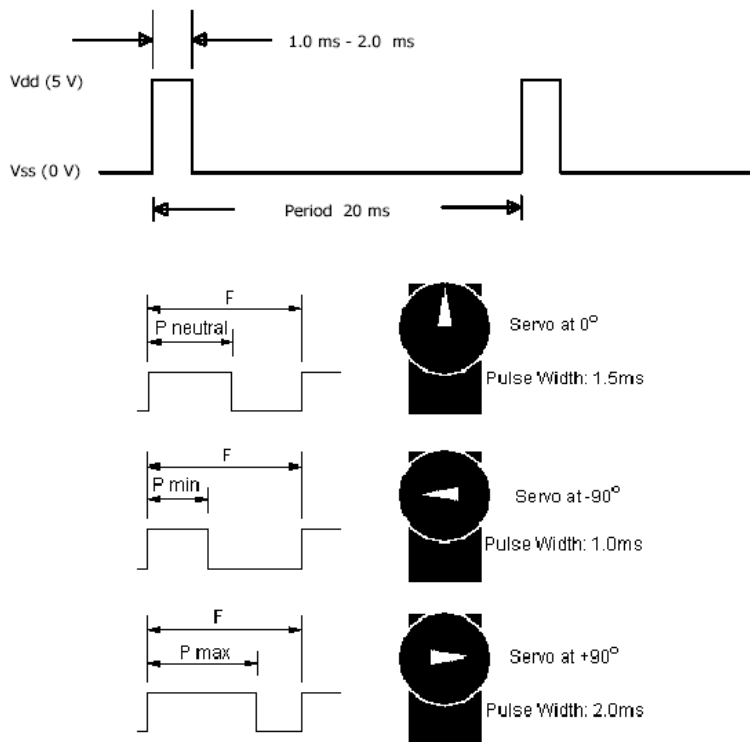
ระบบต้นกำลังของหุ่นยนต์สำหรับหุ่นยนต์แปดแกน

ดังที่กล่าวมาแล้ว หุ่นยนต์ตัวนี้มีทั้งหมด 8 แกน โดยแต่ละแกนใช้มอเตอร์แบบ DC Servo Motor ขนาดเล็ก ซึ่งเป็นมอเตอร์ที่ใช้ในเครื่องเล่นประเภทเครื่องบิน เรือ หรือรถบังคับวิทยุและในภายหลังเริ่มเป็นที่นิยมในการทำหุ่นยนต์ต้นแบบ ดังเช่นหุ่นยนต์ที่ญี่ปุ่นหลายตัว เช่น HOAP PINO เป็นต้น มอเตอร์ชนิดนี้เป็น มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับ ชุดเกียร์ และ ส่วนควบคุม ต่างๆ ไว้ ในโมดูลเดียวกัน หรือ ภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC,GND และ สายสัญญาณควบคุม(Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้าย หรือ ขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณ พัลส์วิดมอด (PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ใน ช่วงประมาณ 4 ถึง 6 โวลท์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ก็คือ จะมีขนาดเล็กกะทัดรัด, ให้แรงบิดสูง, กินพลังงานน้อย และ สามารถควบคุม ด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรง ไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่นๆ เพราะ มอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไป ใน ตำแหน่ง หรือ ทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเตอร์นี้จะหมุนได้ แค่เพียงในช่วงประมาณ 180° หรือ ครึ่งรอบเท่านั้น หรือ บางรุ่นอาจหมุนได้ถึง 210° แต่จะไม่สามารถหมุนเป็นวงรอบได้ เนื่องจากโครงสร้างภายในจะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของ มอเตอร์ และ ตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัวต้านทานปรับค่านี้ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้น เซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียงแค่ ประมาณ 180 องศา หรือ ครึ่งรอบเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดกับตัวต้านทานปรับค่าได้ แต่ถ้าหากเราต้องการให้มอเตอร์หมุนเป็นวงรอบ (360°) นั้นก็สามารถทำได้ โดยจะต้องทำการปรับแต่ง (Modify) ดัดแปลงชิ้นส่วนบางอย่างของมอเตอร์ ส่วนประกอบต่างๆของ Servo Motorดังรูปที่ 4.14



รูปที่ 4.14 แสดงรายละเอียดต่างๆของตัว Servo Motor

การควบคุมการทำงานของ เซอร์โวมอเตอร์ ทำได้โดย การป้อนสัญญาณความกว้างพัลส์ ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูปที่ 4.15 คือ



รูปที่ 4.15 แสดงการควบคุมการทำงานของ เซอร์โวมอเตอร์โดยการป้อนสัญญาณความกว้างพัลส์

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่ง มุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่ง มุม + 90 องศา หรือในทิศทางตามเข็มนาฬิกาในทิศทางตามเข็มนาฬิกา เป็นต้น

สำหรับการวิจัยครั้งนี้ใช้เซอร์โวมอเตอร์อยู่สามขนาด เนื่องจากในแต่ละข้อต่อของหุ่นยนต์ รับแรงได้ไม่เท่ากันโดยที่ข้อเท้ามากที่สุด และ ที่สะโพก ลำตัว หัวเข้าตามลำดับ

ที่ข้อเท้า ใช้มอเตอร์ของ Hitec รุ่น HS-645MG มีกำลังบิด 7.7 kg-cm. ที่แรงดัน 4.8 V.

ที่สะโพก ใช้มอเตอร์ของ GWS รุ่น S15 HPXMG มีกำลังบิด 7.0 kg-cm. ที่แรงดัน 4.8 V.

ที่เข้า ใช้มอเตอร์ของ GWServo รุ่น S03TD มีกำลังบิด 7.0 kg-cm. ที่แรงดัน 4.8 V.

และที่เอว ใช้มอเตอร์ของ GWServo รุ่น S03TD มีกำลังบิด 7.0 kg-cm. ที่แรงดัน 4.8 V.

ระบบควบคุมและอุปกรณ์ตรวจรู้ของหุ่นยนต์

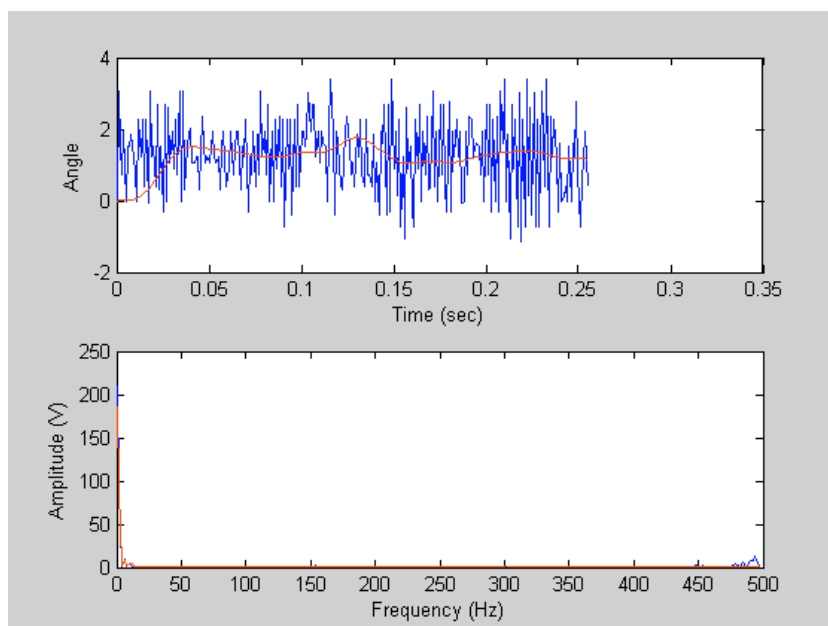
สำหรับหุ่นยนต์ตัวนี้ในขั้นต้นเราต้องการศึกษาการเดินแบบสถิติศาสตร์ ระบบของหุ่นยนต์ตัวนี้จึงประกอบด้วย บอร์ดประมวลผลขนาดเล็ก อุปกรณ์ตรวจรู้มุมของส่วนสะโพกซึ่งทั้งสองถูกติดตั้งที่บริเวณเหนือสะโพกดังรูปที่ 4.13 และอุปกรณ์ตรวจรู้แรงที่ฝ่าเท้าดังรูปที่ 4.16 โดยในขั้นต้นเราจะทำการทดลองเดินและควบคุมการเดินโดยใช้ อุปกรณ์ตรวจรู้มุมของส่วนสะโพกก่อน จากนั้นนำผลของแรงกระทำที่ฝ่าเท้ามาคิด



รูปที่ 4.16 แสดงการติดตั้งอุปกรณ์วัดแรงที่ฝ่าเท้าของหุ่นยนต์

บอร์ดประมวลผลที่ติดตั้งนี้ใช้ในการควบคุมคือ mini SSC II Serial Servo Controller ของ บ. Scott Edwards Electronics(S.E.E) เพื่อใช้ในการควบคุมการทำงานของมอเตอร์ทั้ง 8 ตัว โดยส่งสัญญาณพัลส์ไปยังมอเตอร์ และรับคำสั่งตำแหน่งการหมุนของเซอร์โวแต่ละตัวจากเครื่องคอมพิวเตอร์ผ่านพอร์ตอนุกรม ส่วนบอร์ดของอุปกรณ์ตรวจรู้ค่ามุมของ บ. Crossbow ADXL202EB-232-A โดยใช้ชิพสัญญาณค่ามุม ADM202JRN ของบ. Analog Device Inc และสามารถอ่านค่าผ่านพอร์ตอนุกรมไปยังคอมพิวเตอร์

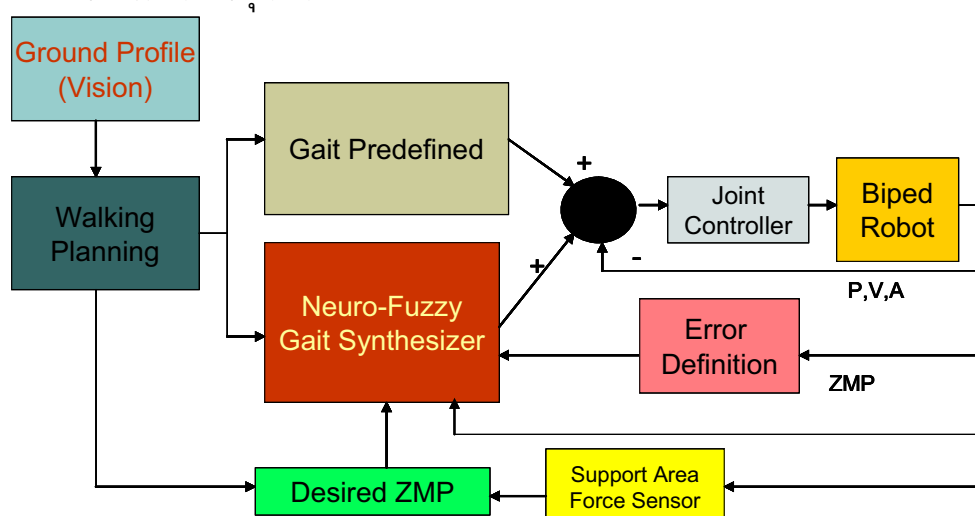
บอร์ดของอุปกรณ์ตรวจรู้ค่ามุม วัดค่ามุมสองแนวแกนคือแนวแกน X และ แนวแกน Y และสามารถอ่านค่าได้ทาง RS 232 ค่าที่อ่านได้มีค่า 8 บิต หรือ 256 ชั้น จากการอ่านค่าของทั้งสองแกนเมื่อหุ่นยนต์อยู่นิ่งๆพบว่า ค่ามุมที่วัดได้มีค่าแกว่งอยู่ในช่วง 2 – 4 องศา ดังรูปที่ 4.17 ซึ่งทำให้ก่อนที่จะนำค่าไปประมวลผลอย่างอื่นต่อไป จึงต้องมีการกรองสัญญาณรบกวนออกเสียก่อน ซึ่งมีด้วยกันสองวิธีการใหญ่ๆคือ แบบใช้วงจรอิเล็กทรอนิกส์(Hardware) และแบบใช้โปรแกรมคอมพิวเตอร์ (Software) ในที่นี้เลือกใช้วิธีการกรองสัญญาณรบกวนโดยใช้โปรแกรม และใช้วิธีการของ Butterworth ทำ Low pass Filter ที่ความถี่สัญญาณ 25 Hz ออกไป ดังรูปที่ 4.17



รูปที่ 4.17 รูปสัญญาณจากอุปกรณ์ตรวจวัดมุมจาก Low pass Filter ที่ความถี่สัญญาณ 25 Hz

4.3 การออกแบบท่าทางการเดินและโปรแกรมเคลื่อนที่ของหุ่นยนต์สองขา

ในส่วนนี้เป็นส่วนที่ทำหน้าที่ในการสร้างรูปแบบการเคลื่อนที่ของแต่ละข้อต่อ โดยอาศัยข้อมูลที่ได้จาก Walking Pattern Generator อันได้แก่ค่าคงที่ต่างๆ เช่น ระยะแต่ละก้าว (Step length) ความเร็วในการเดิน (Walking Speed) ระยะยกเท้าสูงสุด (maximum foot lift) และส่งคำสั่งอ้างอิงไปให้ข้อต่อของหุ่นยนต์สองขา ประกอบด้วยสองส่วนด้วยกันคือ Gait Predefined Generator และ Neuro-fuzzy Gait Synthesizer โดยในส่วนแรกเป็นหน้าที่ใช้สมการทางสถิติศาสตร์ของหุ่นยนต์ และส่วนที่สองเป็นส่วนที่อาศัยข้อมูลเชิงบรรยายและข้อมูลที่เกิดในขณะที่หุ่นยนต์เคลื่อนที่เพื่อมาทำการสอนระบบสังเคราะห์ท่าทางการเดินของหุ่นยนต์



รูปที่ 4.18 แผนภาพแสดงส่วนที่สังเคราะห์ท่าทางการเดินของหุ่นยนต์

ในการหาฟังก์ชันแบบคาบเวลาเพื่อพัฒนาท่าทางการเดินนั้น เรามีตัวแปรที่สำคัญสามตัวคือ ระยะการก้าว (Step length) ระยะความสูงในการก้าว (step height) และระยะเวลาในแต่ละก้าว (Step period) เพื่อใช้ในการควบคุมหุ่นยนต์เพื่อให้เกิดเสถียรภาพในการเดินแบบพลวัต ที่ทุกๆตำแหน่งและความเร็วของหุ่นยนต์

Step length ระยะในการก้าวจะชี้ให้เห็นถึงหุ่นยนต์สามารถเคลื่อนที่ได้ไกลแค่ไหนในแต่ละก้าว ซึ่งมีความสัมพันธ์โดยตรงกับการเคลื่อนที่ของหุ่นยนต์ เราสามารถหาความเร็วโดยเฉลี่ยในแต่ละก้าวได้ดังนี้

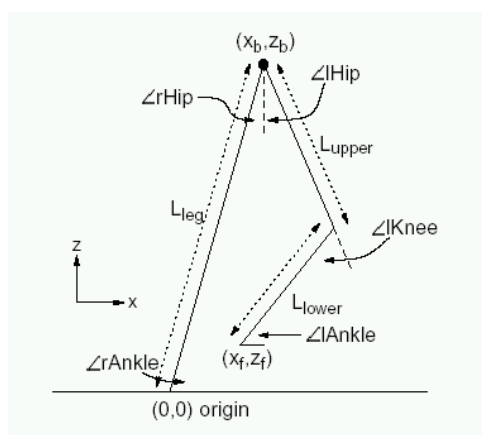
$$\text{ความเร็วเฉลี่ย} = \text{ความยาวแต่ละก้าว} / \text{เวลาในแต่ละก้าว}$$

ตัวแปรนี้ใช้เพื่อปรับค่าความไม่มีเสถียรภาพใน Sagittal plane ว่าแต่ละก้าวว่ามีระยะมากหรือน้อย ของขาให้สัมพันธ์กับส่วนบนของหุ่นยนต์

Step period ระยะเวลาในแต่ละก้าวเพื่อสามารถควบคุมหุ่นยนต์ให้เคลื่อนที่เร็วหรือช้า ซึ่งมีความสัมพันธ์กับค่าความเร็วเฉลี่ยในส่วนของการก้าว ตัวแปรนี้ช่วยในการปรับค่าเสถียรภาพในแนว Lateral Plane ที่มากเกินไป โดยให้แน่ใจว่าเวลาในแนวSagittal และ lateral ตรงใกล้เคียงกัน โดยวัดจากเวลาในการที่เท้าสัมผัสพื้น เพื่อรักษาเสถียรภาพทั้งสองแนว ตัวอย่างเช่นเมื่อส่วนบนของหุ่นยนต์กำลังเคลื่อนตัวไปเร็วกว่าการก้าวขา ระบบจะปรับส่วนขาเพื่อให้สัมพันธ์กับส่วนบนของหุ่นยนต์

Step length การควบคุมระยะความสูงในการก้าวคือการควบคุมความสูงที่มากที่สุดของเส้นทางของเท้า เพื่อช่วยให้สามารถปรับตัวในกรณีพื้นที่ไม่เรียบ ตัวอย่างเช่นถ้ามีสิ่งกีดขวางขวางหน้าหุ่นยนต์ มันจะสามารถเคลื่อนที่ข้ามสิ่งกีดขวางเหล่านี้ได้

จากตัวแปรต่างๆที่ได้กล่าวมาข้างต้น เราสามารถใช้ inverse kinematics เพื่อคำนวณหาตำแหน่งของเท้าข้างที่เคลื่อนที่ไปข้างหน้า(Swing Phase) โดยพิจารณาจากมุมของแต่ละข้อต่อ ดังรูปที่ 4.19



รูปที่ 4.19 แสดงรายละเอียดท่าทางการเดินของหุ่นยนต์

จากรูป ถ้าเรากำหนดจุดเริ่มต้นที่เท้าที่รองรับ (x,z) คือจุด (0,0) และสามารถหาตำแหน่งเท้าที่สวิงคือ (x_f, z_f) ดังนี้

$$x_f = \frac{4L_{step}t}{T} - L_{step} \quad (4.9)$$

$$z_f = H_{step} \sin\left(\frac{2\pi t}{T}\right) \quad (4.10)$$

เมื่อ L_{step} คือระยะก้าว และ H_{step} คือความสูงของเท้าที่สวิง.

เมื่อหุ่นยนต์เคลื่อนที่ไปข้างหน้า เราสามารถหาสมการของช่วงตัวที่เคลื่อนที่ไปข้างหน้า (x_b, z_b) ระหว่างอยู่ในสถานะ single support ดังนี้

$$x_b = -L_{leg} \sin(\angle rHip) \quad (4.11)$$

$$z_b = -L_{leg} \cos(\angle rHip) \quad (4.12)$$

$$\angle rHip = \arcsin\left(L_{step} \cos\left(\frac{2\pi t}{T}\right)\right) \quad (4.13)$$

$$L_{leg} = L_{upper} + L_{lower} \quad (4.14)$$

เมื่อ L_{leg} คือความยาวยืดยืดสุดของขา, L_{upper} และ L_{lower} คือความยาวขาบนและล่าง $\angle rHip$ คือมุมบิดของขา support ในแนวแกน z.

ต่อมาเราสามารถหามุมบิดของหัวเข่า โดยใช้กฎของ Cosines ดังนี้

$$\cos(\angle IKnee) = \frac{(x_b - x_f)^2 + (z_b - z_f)^2 - L_{upper}^2 - L_{lower}^2}{2L_{upper}L_{lower}} := K \quad (4.15)$$

$$\sin(\angle IKnee) = \pm\sqrt{1 - K^2} \quad (4.16)$$

$$\angle IKnee = \arctan(-\sqrt{1 - K^2}, K) \quad (4.17)$$

โดยที่

$$\cos(\theta) = \frac{x}{(x^2 + y^2)^{1/2}}, \quad \sin(\theta) = \frac{y}{(x^2 + y^2)^{1/2}} \quad (4.18)$$

ในทำนองเดียวกัน เราสามารถหามุม swing สะโพก $\angle lHip$ ดังนี้

$$\angle lHip = -\alpha - \beta \quad (4.19)$$

เมื่อ

$$\alpha = \arctan(z_b - z_f, x_f) \quad (4.20)$$

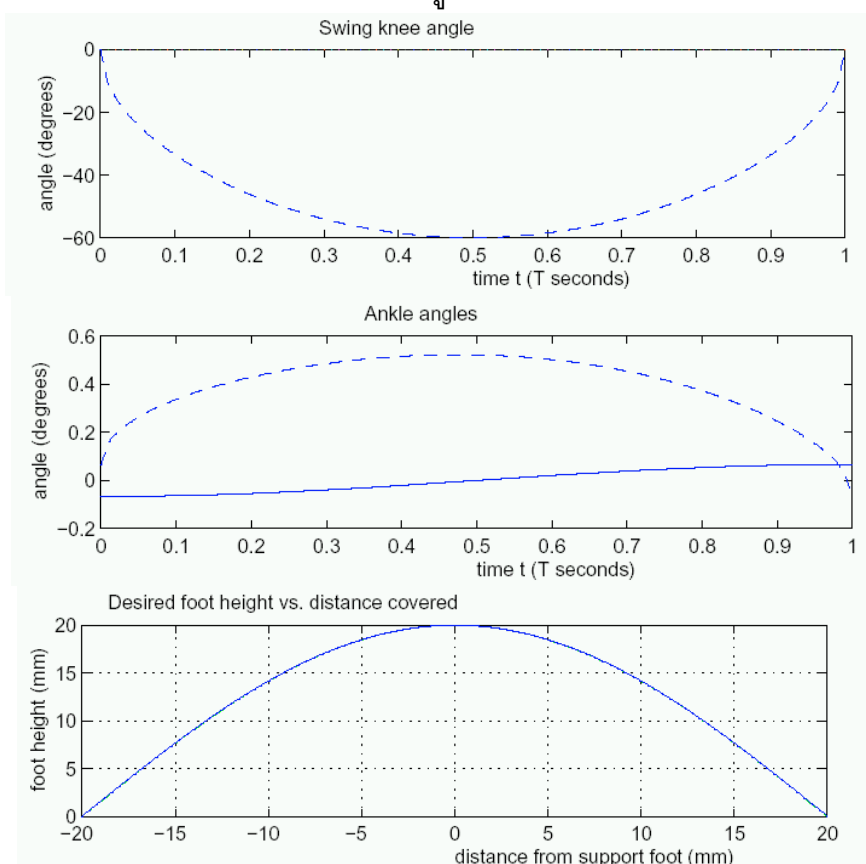
$$\beta = \arctan(L_{lower} \sin(\angle IKnee), L_{upper} + L_{lower} \cos(\angle IKnee)) \quad (4.21)$$

สุดท้าย เราสามารถหามุมที่หัวเข่า ดังนี้

$$\angle rAnkle = -\angle rKnee - \angle rHipB \quad (4.22)$$

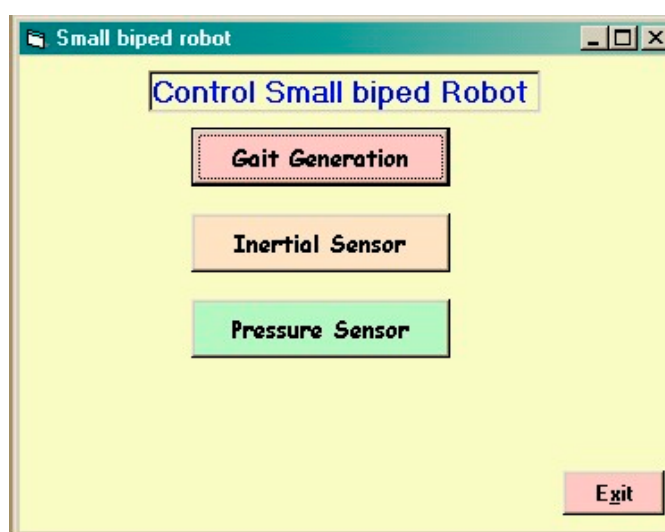
$$\angle lAnkle = -\angle lKnee - \angle lHipB \quad (4.23)$$

จากสมการต่างที่กล่าวมา เราสามารถสร้างรูปแบบการเดินตามคาบเวลาได้ดังนี้

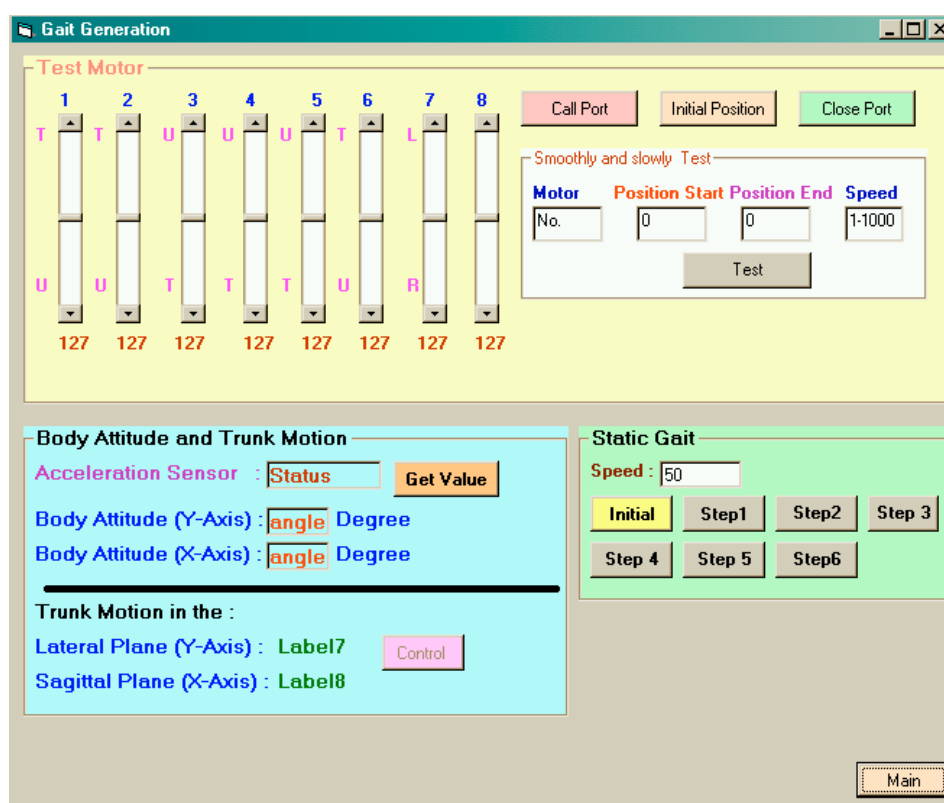


รูปที่ 4.20 รูปการเคลื่อนที่เมื่อเวลา 1 วินาที ความสูงการก้าว 20 มม.
และความยาวในการก้าว 20 มม.

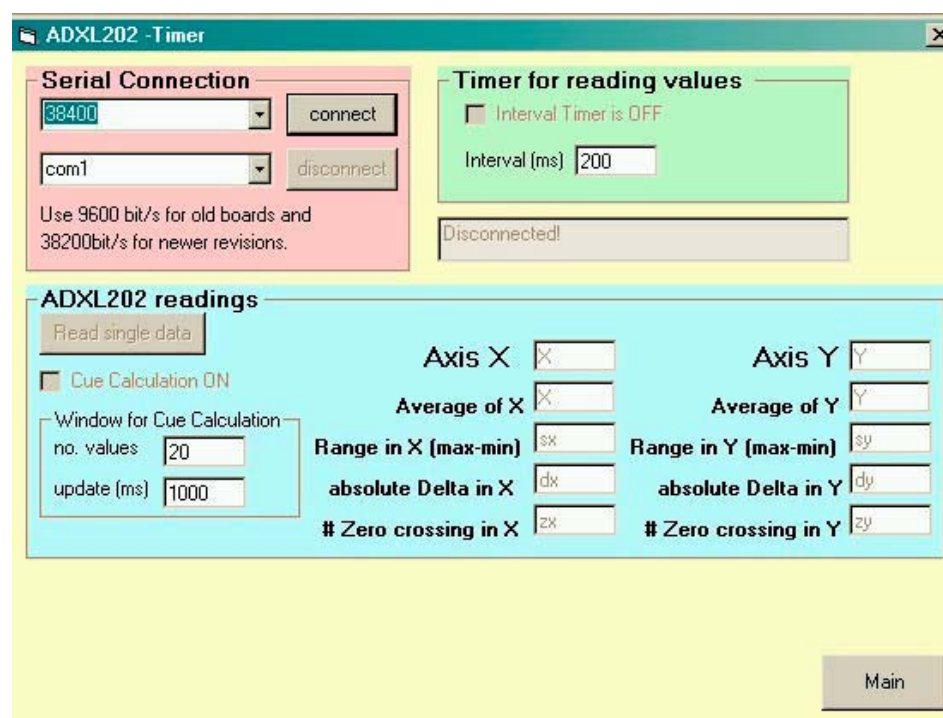
โปรแกรมการเคลื่อนที่ของหุ่นยนต์ประกอบด้วยสามส่วนหลักๆคือ ส่วนที่ใช้สร้างท่าทางการเดิน ส่วนที่ใช้วัดมุมของลำตัวและส่วนที่วัดแรงที่กระทำที่ฝ่าเท้า แสดงดังรูปที่ 4.21



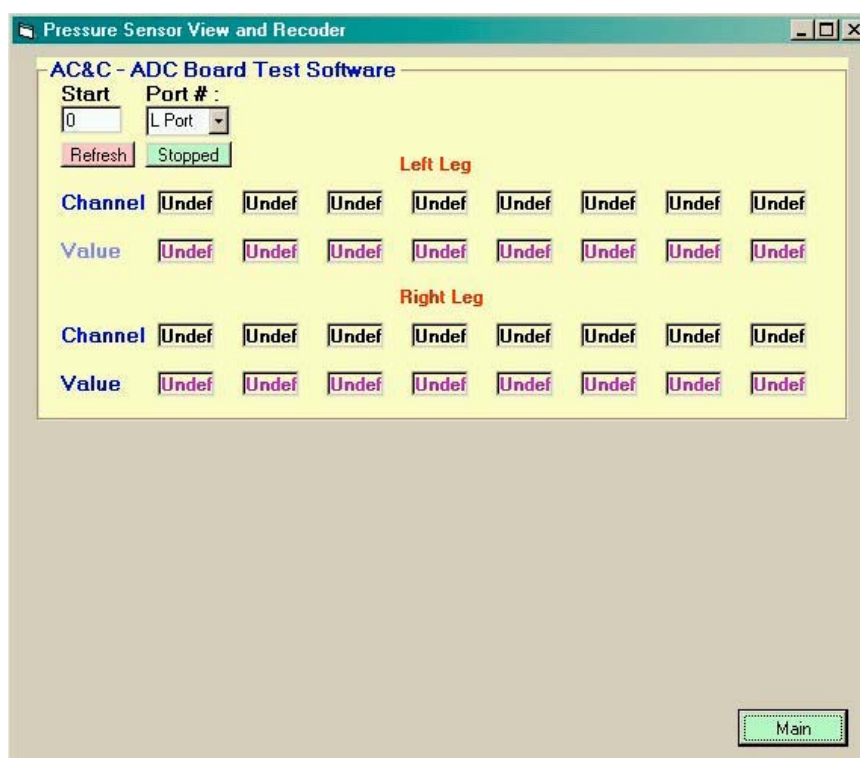
รูปที่ 4.21 แสดงหน้าโปรแกรมหลักของโปรแกรมควบคุมการเดินของหุ่นยนต์สองขา



รูปที่ 4.22 แสดงโปรแกรมในการออกแบบท่าทางการเดินของหุ่นยนต์สองขา



รูปที่ 4.23 แสดงโปรแกรมในส่วนการวัดค่ามุมเอียงของตัวหุ่นยนต์



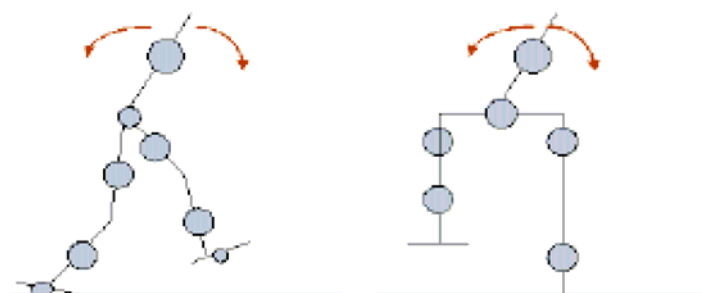
รูปที่ 4.24 แสดงโปรแกรมในส่วนการวัดค่าแรงที่กระทำที่ฝ่าเท้าของหุ่นยนต์

จากที่กล่าวมาแล้วนั้น หุ่นยนต์สองขาที่ใช้ในการทดลองนั้น มีทั้งขนาดเล็กไปจนถึงขนาดใหญ่ และมีตั้งแต่ สามแกน หกแกน แปดแกน จนกระทั่งสิบสองแกน เพื่อใช้ทดสอบระบบนิวโรฟิซชีที่ใช้ในการออกแบบท่าทางการเดินของหุ่นยนต์สองขา โดรนในบทต่อไปจะกล่าวถึงการทดลองกับหุ่นยนต์สองขาแบบต่างๆ ต่อไป

บทที่ 5 การทดลอง

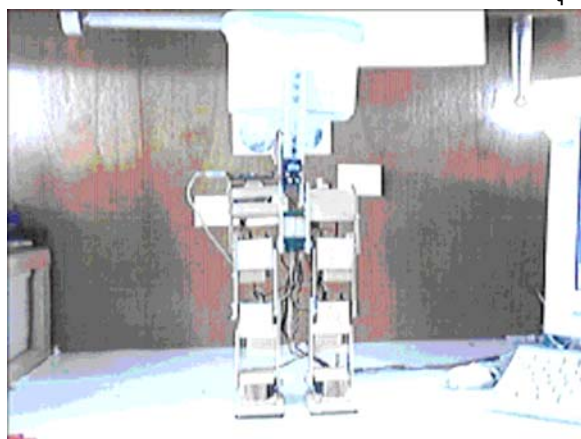
5.1 การทดลองควบคุมหุ่นยนต์ 8 แขน

หุ่นยนต์สองขาที่ใช้ในงานวิจัยนี้ ประกอบด้วยข้อต่อทั้งหมดแปดข้อต่อ ได้แก่ สะโพก(hip) 2 ข้อต่อ หัวเข่า(knee) 2 ข้อต่อ ข้อเท้า(ankle) 2 ข้อต่อ และแกนหมุนที่ลำตัว(Trunk) อีก 2 ข้อต่อ โดยข้อต่อสะโพก หัวเข่าและข้อเท้า มีลักษณะการหมุนแบบแนวระนาบด้านข้าง(Sagittal Plane) และส่วนลำตัว ซึ่งมี 2 แกนนั้นสามารถหมุนได้ในแนวแกนด้านข้าง(Sagittal) และด้านหน้า(Lateral Plane) ดังรูปที่ 5.1



รูปที่ 5.1 แสดงรูปจำลองหุ่นยนต์สองขาแบบ 8 แกน
โดยรูป ก. ด้านSagittal และรูป ข. ด้าน Lateral

หุ่นยนต์ที่ใช้มีขนาดโดยประมาณ กว้าง 20 ซม. หนา 10 ซม. และขนาดยืนยึดขาสุดมีความสูง 45 ซม. ที่ลำตัวด้านขวาติดตั้งแผงวงจรควบคุมมอเตอร์ 8 แกนขนาดกว้าง 3.5 ซม.และยาว 4.5 ซม. และที่ลำตัวด้านซ้ายติดตั้งแผงวงจรวัดค่ามุมเอียงของตัว ขนาดกว้าง 6 ซม. และยาว 6.5 ซม. และที่ฝ่าเท้าติดตั้งตัววัดค่าแรงกด(Pressure Sensor –FSR Unit) ลักษณะของหุ่นยนต์สองขาเป็นดังรูปที่ 5.2

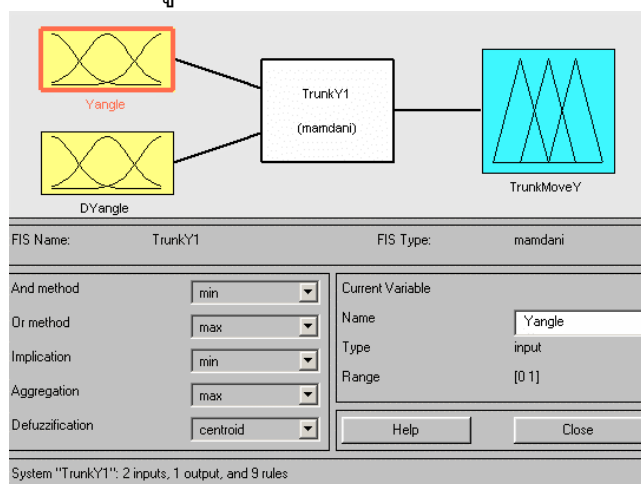


รูปที่ 5.2 แสดงภาพด้านหน้าของหุ่นยนต์สองขาแบบ 8 แกนขนาดยืน

ดังที่กล่าวมาแล้ว หุ่นยนต์ตัวนี้มีทั้งหมด 8 แกน โดยแต่ละแกนใช้มอเตอร์แบบ DC Servo Motor ขนาดเล็ก ซึ่งถูกควบคุมด้วยบอร์ดประมวลผลขนาดเล็ก และมีอุปกรณ์ตรวจจับมุมของส่วนสะโพก ซึ่งทั้งสองถูกติดตั้งที่บริเวณเหนือสะโพก และอุปกรณ์ตรวจจับแรงที่ฝ่าเท้า โดยในขั้นต้นเราจะทำการ

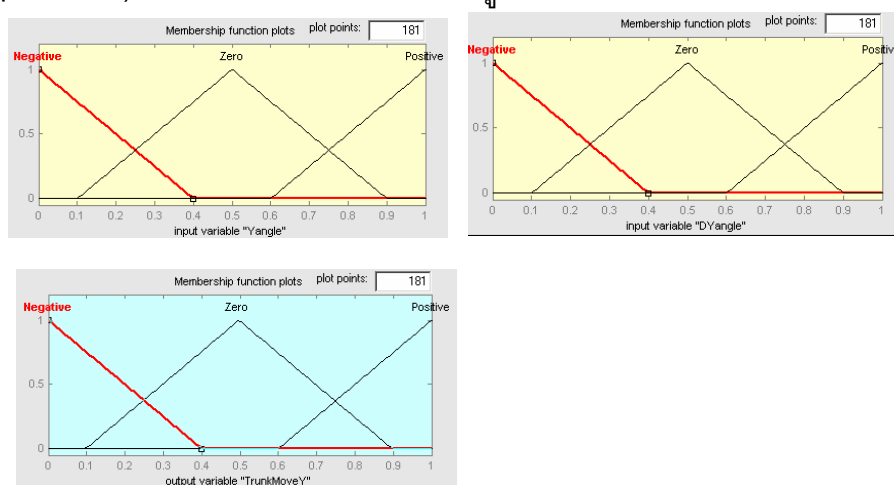
ทดลองเดินและควบคุมการเดินโดยใช้ อุปกรณ์ตรวจรู้มุมของส่วนสะโพกก่อน จากนั้นนำผลของแรงกระทำที่ฝ่าเท้ามาคิด

การทดลองโปรแกรมเคลื่อนที่ของหุ่นยนต์สองขา ในการทดลองขั้นต้น ได้ทำการออกแบบระบบควบคุมการทรงตัวของหุ่นยนต์สองขา โดยใช้ระบบควบคุมแบบฟัซซี่ (Fuzzy System) โดยอาศัยข้อมูลจากอุปกรณ์วัดมุมเอียง เพื่อวัดมุมเอียงของลำตัวหุ่นยนต์ ทั้งด้านหน้าและด้านข้างเพื่อนำไปปรับเปลี่ยนมุมเอียงของลำตัวนั้นคือ แกนที่ 7 และ 8 ของหุ่นยนต์ ซึ่งจะทำให้หุ่นยนต์สามารถทรงตัวอยู่ได้ในขณะ ยืน ยกขา และเดิน มีลักษณะดังรูปที่ 5.3



รูปที่ 5.3 แสดงรูปแบบระบบควบคุมแบบฟัซซี่

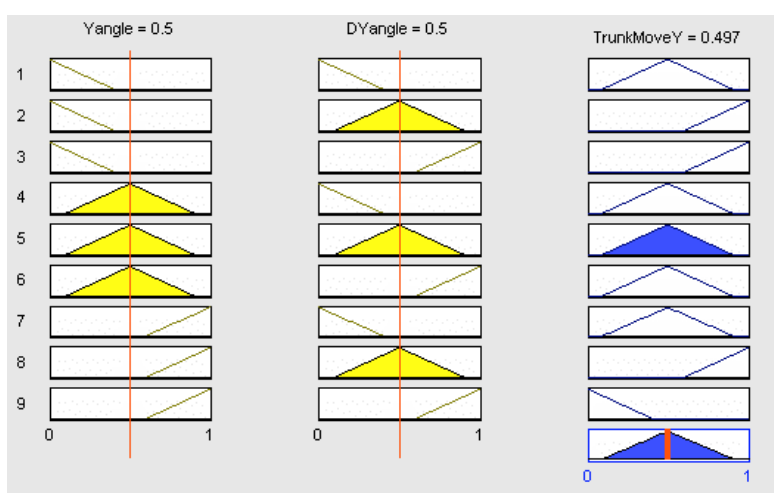
ในการออกแบบระบบควบคุมด้วยฟัซซี่เริ่มจากการกำหนดข้อมูลขาเข้าของระบบ สำหรับระบบนั้น เราใช้ข้อมูลการเอียงในแนวแกน และค่าอัตราการเปลี่ยนแปลงมุมเอียง โดยแยกระบบเป็นสองส่วนคือ ในแนวด้านหน้า (Lateral Plane) หรือแนวแกน y และในแนวด้านข้าง (Sagittal plane) หรือแนวแกน x ของระบบหุ่นยนต์สองขา ทั้งสองระบบควบคุมจะแยกการควบคุมอิสระต่อกัน ค่าฟังก์ชันสมาชิก (membership function) ของระบบฟัซซี่ทั้งสอง เป็นดังรูปที่ 5.4



รูปที่ 5.4 ค่าฟังก์ชันสมาชิก (membership function) ของระบบฟัซซี่

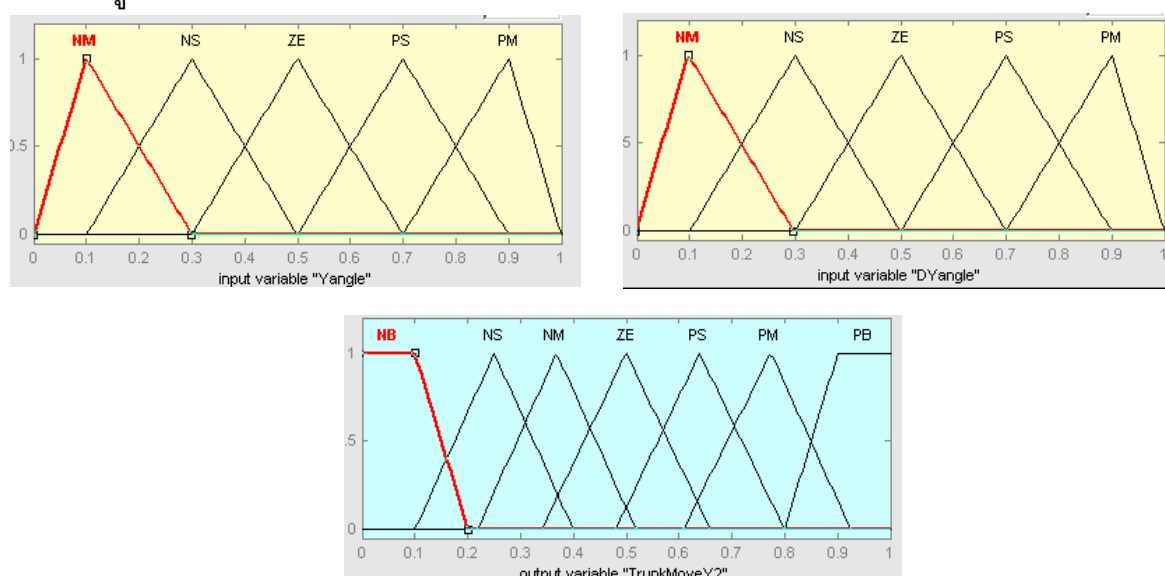
ทั้ง Sagittal plane และ Lateral plane

จากรูปที่ 5.4 ประกอบด้วยค่าฟังก์ชันสมาชิก (membership function) ของทั้งข้อมูลเข้าสองข้อมูล (พื้นสีเหลือง) และข้อมูลขาออกหนึ่งข้อมูล(พื้นฟ้า) โดยเริ่มต้นทดลอง กำหนดให้ค่าฟังก์ชันสมาชิกมีลักษณะเป็นฟังก์ชันสามเหลี่ยม ต่อกันสามช่วงคือ ค่าเป็นลบ(Negative) ค่าตรงกลาง(Zero) ในที่นี้ หมายถึงหุ่นยนต์ไม่มีการเอียงตัว และค่าเป็นบวก(Positive) และนำค่าข้อมูลขาเข้า และค่าข้อมูลขาออก มาหาความสัมพันธ์ทางตรรกะ ในรูปของกฎ (If-then rule) โดยมีลักษณะกฎเบื้องต้น ดังรูปที่ 5.5 จากรูปแสดงให้เห็นถึงความสัมพันธ์ระหว่างข้อมูลขาเข้า และออกในรูปเชิงตรรกะ ตัวอย่างเช่น เมื่อค่ามุมของลำตัว เอียง 0.5 และมีอัตราการเปลี่ยนแปลงค่ามุมเอียง 0.5 ระบบควบคุมแบบฟัซซี่จะสั่งให้มอเตอร์ที่แกนของลำตัวเปลี่ยนตำแหน่งไป 0.497 เป็นต้น



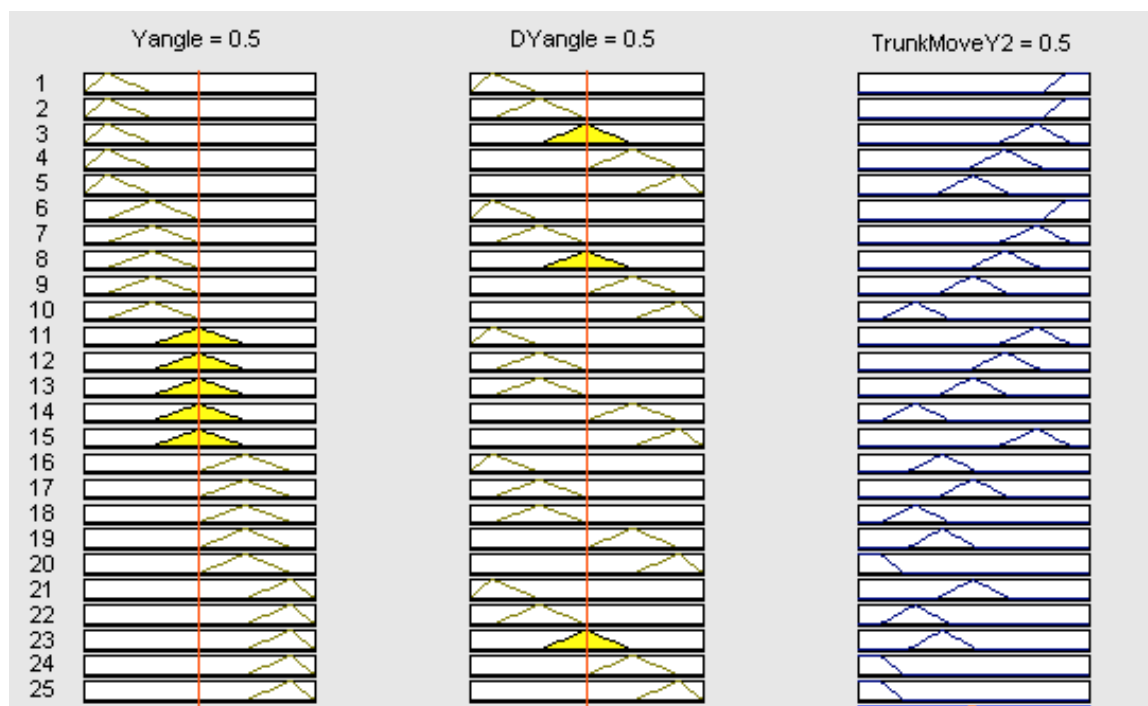
รูปที่ 5.5 แสดงค่าของกฎ If –then rule ในเชิงรูปภาพ

นอกจากนี้ ได้ออกแบบระบบฟัซซี่อีกแบบหนึ่ง เพื่อเปรียบเทียบกับระบบข้างต้น โดยระบบฟัซซี่อีกแบบหนึ่งนั้น มีค่าฟังก์ชันสมาชิกที่แตกต่างกันในแต่ละข้อมูลขาเข้า และข้อมูลขาออก โดยมีรายละเอียดดังรูปที่ 5.6



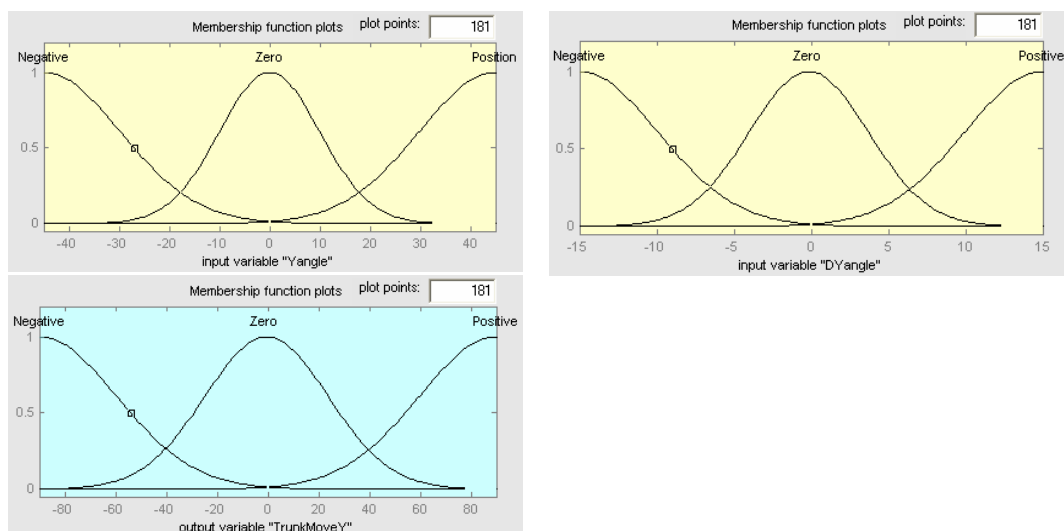
รูปที่ 5.6 แสดงค่าฟังก์ชันสมาชิกของข้อมูลขาเข้าและขาออก ของอีกระบบหนึ่ง

จากรูปที่ 5.7 จะเห็นได้ว่าระบบฟัซซี่นี้มีค่าฟังก์ชันสมาชิกที่กระจายมากขึ้นทั้งข้อมูลขาเข้าและออก ดังนั้นการออกแบบกฎความสัมพันธ์ระหว่างข้อมูลขาเข้า และขาออก นั้น จำนวนกฎจึงต้องมากขึ้นด้วย ดังรูปที่ 5.7

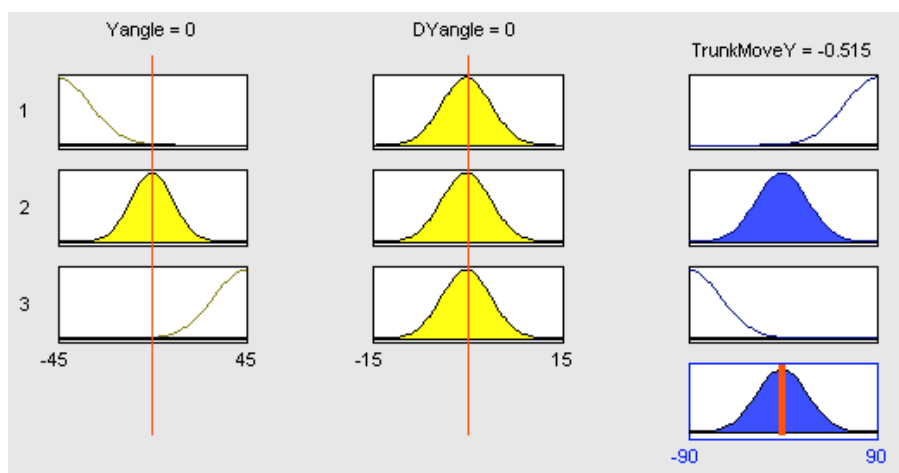


รูปที่ 5.7 แสดงค่าของกฎ If –then rule ในเชิงรูปภาพของอีกระบบหนึ่ง

จากระบบควบคุมแบบฟัซซี่ทั้งสองแบบตามที่ได้ออกแบบมาแล้วข้างต้น มาทดลองควบคุมระบบหุ่นยนต์สองขา โดยเริ่มต้นจากการยืนทรงตัว พบว่าระบบฟัซซี่แบบที่สองมีการตอบสนองต่อการเปลี่ยนแปลงมุมของลำตัวมากเกินไป ทำให้หุ่นยนต์สั่นและล้มในที่สุด ส่วนในระบบการควบคุมแบบแรกนั้น มีการตอบสนองที่ช้ากว่า พอเหมาะกับการควบคุมให้หุ่นยนต์สองขารักษาสถิตอยู่ได้ ต่อจากนั้นทำการปรับปรุงระบบฟัซซี่ให้สามารถตอบสนองการยืนของหุ่นยนต์ได้ดีมากยิ่งขึ้น ระบบที่ได้มีลักษณะค่าฟังก์ชันสมาชิกดังรูปที่ 5.8 และกฎ If-then ดังรูปที่ 5.9 ตามลำดับ จากการทดลองดังกล่าว สามารถใช้ระบบควบคุมแบบฟัซซี่ในการควบคุมการรักษาสมดุลของหุ่นยนต์สองขา แต่มีข้อเสียตรงที่เราต้องทำการออกแบบค่าฟังก์ชันสมาชิกและกฎความสัมพันธ์เอง เพื่อให้สอดคล้องกับระบบที่สนใจ ดังนั้นจึงควรมีวิธีการที่สามารถปรับเปลี่ยนระบบควบคุมแบบฟัซซี่ให้สอดคล้องกับระบบที่สนใจ ซึ่งจะได้ศึกษาในหัวข้อถัดไป



รูปที่ 5.8 แสดงค่าฟังก์ชันสมาชิกของข้อมูลขาเข้าและขาออก ของระบบที่สอดคล้องกับการรักษาสมดุลของหุ่นยนต์สองขา ในแนว Lateral plane



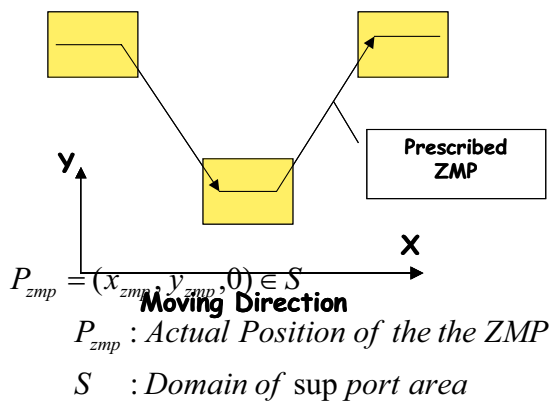
รูปที่ 5.9 แสดงค่าของกฎ If –then rule ในเชิงรูปภาพของระบบข้างต้น

5.2 การสังเคราะห์รูปแบบการเดินของหุ่นยนต์สองขาด้วยระบบนิวโรฟuzzy

ในการพิจารณาการเดินของหุ่นยนต์ให้มีเสถียรภาพนั้น ใช้วิธีการคำนวณทดแทนแบบพลวัตเพื่อรักษา ZMP ของหุ่นไว้ โดยอาศัยการปรับแต่งการเคลื่อนไหวของส่วนลำตัวเป็นหลัก การปรับเปลี่ยนนี้ อาศัยสัญญาณป้อนกลับของสัญญาณที่ใช้ประเมินหาเสถียรภาพในการเดินของหุ่นยนต์

เริ่มต้น เราสมมุติว่าหุ่นยนต์เดินบนพื้นเรียบ และสนใจการควบคุมลำตัวเพื่อรักษาสมดุลในการเดิน เราเลือกจุดประสงค์หลักในการควบคุม เพื่อสังเคราะห์เส้นทางการเดินของหุ่นยนต์ คือ

$$\text{Minimize} \int_{t_i}^{t_f} \|P_{zmp}(t) - P_{zmp}^d(t)\|^2 dt, \quad (5.1)$$



รูปที่ 5.10 แผนภาพแสดงส่วนที่สังเคราะห์ท่าทางการเดินของหุ่นยนต์

ในที่นี้สามารถหาสมการของ ZMP ดังนี้

$$x_{zmp} = \frac{\sum_{i=0}^{n-1} m_i (z_i + g) x_i - \sum_{i=0}^{n-1} m_i \ddot{x}_i z_i}{\sum_{i=0}^{n-1} (\ddot{Z}_i + g)}, \text{ and} \quad (5.2)$$

$$y_{zmp} = \frac{\sum_{i=0}^{n-1} m_i (z_i + g) y_i - \sum_{i=0}^{n-1} m_i \ddot{y}_i z_i}{\sum_{i=0}^{n-1} (\ddot{Z}_i + g)}.$$

เมื่อ m_i คือ มวลของแต่ละข้อต่อ

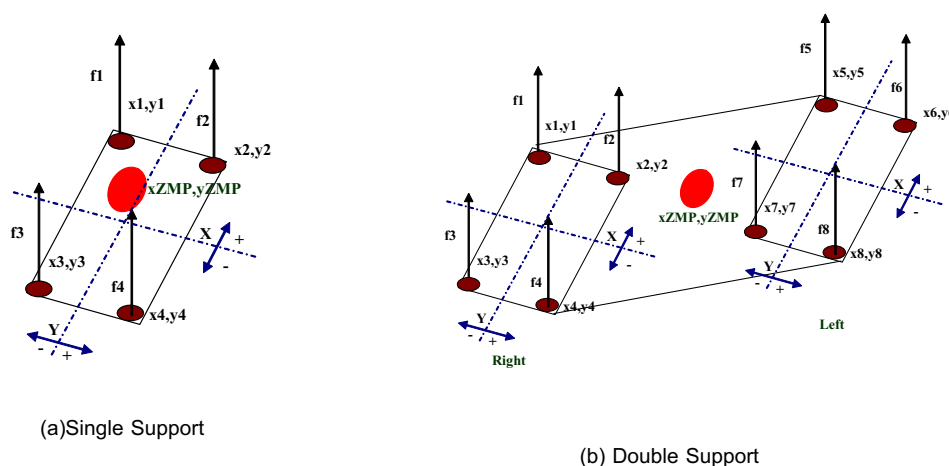
x_i, y_i และ z_i คือตำแหน่งในแต่ละแนวแกนของจุดศูนย์กลางมวลในแต่ละข้อต่อ

x_{zmp}, y_{zmp} คือ ตำแหน่งของจุด ZMP บนระนาบ xy

g คือความเร่งโน้มถ่วง มีค่าเท่ากับ 9.81 m/s^2

แต่ในการวัดที่หุ่นยนต์จริงนั้น อยู่บนสมมุติฐานที่ว่าตำแหน่งของ ZMP มีค่าเทียบเท่ากับค่าแรงลัพธ์ที่กระทำที่ฝ่าเท้า ในที่นี้ใช้ Force Sensing Resister (FSR) เป็นตัววัดค่าแรงที่กระทำ ซึ่งได้ทำการติดไว้ 4 จุด รอบฝ่าเท้า ดังรูปที่ 4 และสามารถคำนวณหาตำแหน่งของจุด ZMP ดังสมการ

$$x_{ZMP} = \frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i}, \quad y_{ZMP} = \frac{\sum_{i=1}^n f_i y_i}{\sum_{i=1}^n f_i} \quad (5.3)$$



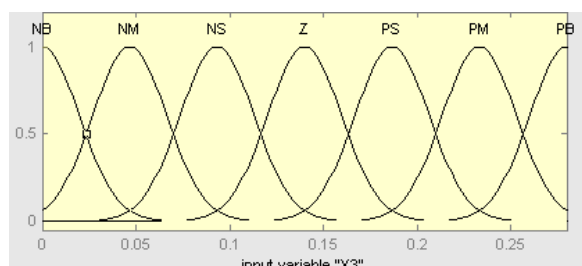
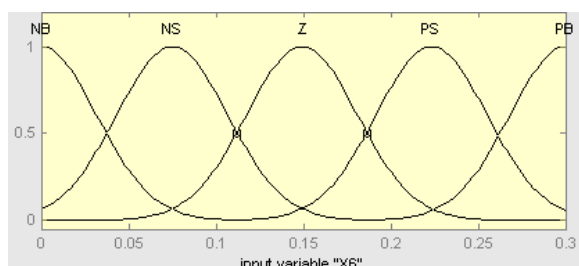
รูปที่ 5.11 แผนภาพแสดงการวัดแรงที่กระทำที่ฝ่าเท้าของหุ่นยนต์

5.2.1 ระบบการสังเคราะห์ ทำทางการเดินด้วยนิวโรฟัซซี่

เป็นส่วนที่ใช้วิธีการคำนวณทดแทนแบบพลวัตเพื่อรักษา ZMP ของหุ่นไว้ โดยอาศัยการปรับแต่งการเคลื่อนไหวของส่วนลำตัวเป็นหลัก การปรับเปลี่ยนนี้อาศัยสัญญาณป้อนกลับของสัญญาณที่ใช้ประเมินหาเสถียรภาพในการเดินของหุ่นยนต์

เริ่มต้น เราสมมุติว่าหุ่นยนต์เดินบนพื้นเรียบ และสนใจการควบคุมลำตัวเพื่อรักษาสมดุลในการเดิน เราเลือกจุดประสงค์หลักในการควบคุม เพื่อสังเคราะห์เส้นทางการเดินของหุ่นยนต์ ในขั้นตอนของการออกแบบกฎเพื่อสร้าง Gait Synthesizer เราสามารถสรุปได้ดังนี้

1. ยกเว้นส่วนของลำตัว แต่ละข้อต่อและเส้นทางของ ZMP เราจะกำหนดไว้ล่วงหน้า โดยอาศัยวิธีการคำนวณทางคณิตศาสตร์
2. สร้างกฎการสังเคราะห์ทำทางการเดินของหุ่นยนต์ โดยเราแยกเป็น ระนาบด้านข้าง และระนาบด้านหน้า โดยในขั้นต้นเราสร้างกฎนี้อาศัยการนำค่าข้อมูลขา-และออกของหุ่นยนต์ โดยข้อมูลขาเข้าคือ ตำแหน่งเส้นทางการเคลื่อนที่ของแต่ละข้อต่อ $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ และข้อมูลขาออกคือตำแหน่งของ ZMP (x_{ZMP}, y_{ZMP}) ตัวอย่างสมาชิกฟัซซี่ เป็นดังรูปที่ 5.12



รูปที่ 5.12 แสดงตัวอย่างค่าฟังก์ชันสมาชิกของตัวแปรข้อต่อต่างๆ

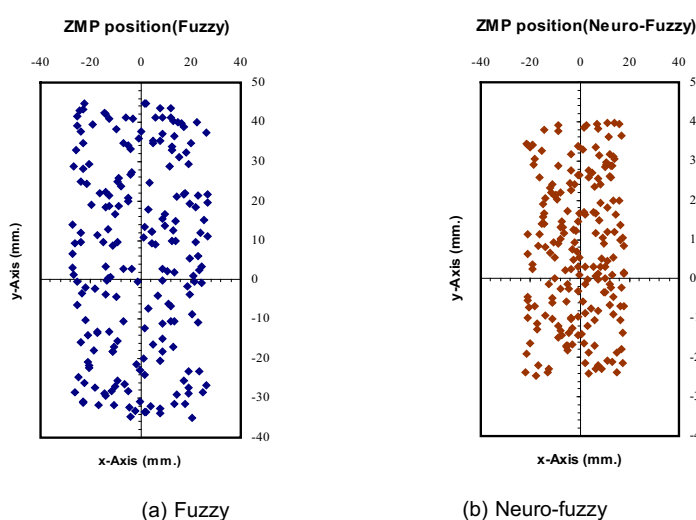
3. ออกแบบตัวสังเคราะห์ทำทางการเดิน ด้วย Neuro-Fuzzy เพื่อใช้กับกฎสมดุลย์การเดินของลำตัว กล่าวคือรับค่าความผิดพลาดของ ZMP และส่งสัญญาณไปปรับตัวแปรส่วน

Consequent ของระบบนิวโรฟัซซี่โดยวิธี gradient descent ทำให้ค่า ZMP เปลี่ยนแปลงไป

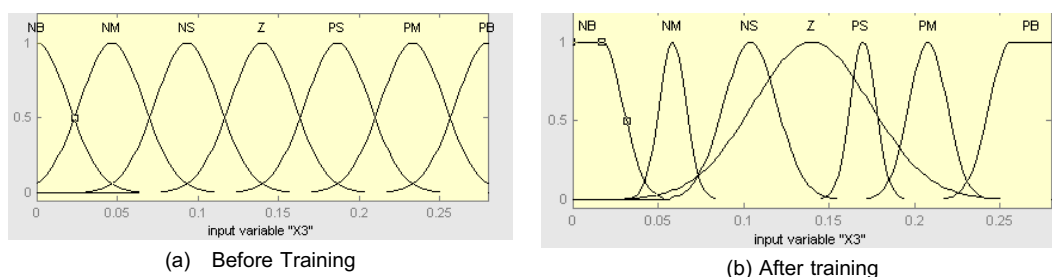
5.2.2 ผลการทดลอง

ในการทดลองหุ่นยนต์เคลื่อนที่ด้วยความเร็วคงที่ และระยะแต่ละก้าวมีค่าเท่ากับ 30 มม. และระยะยกก้าวเท่ากับ 10 มม. ทำการจำลองท่าทางการเดินของหุ่นยนต์และเก็บข้อมูลระหว่างเส้นทางการเคลื่อนที่ของแต่ละข้อต่อ และ ค่า ZMP เพื่อสร้างระบบนิวโรฟัซซี่เริ่มต้น จากนั้นใช้ค่าดัชนีสมรรถนะนั้นคือค่า ZMP ที่เหมาะสม ทำการปรับค่าตัวแปรต่างๆของระบบนิวโรฟัซซี่นี้

รูปที่ 5.13 เป็นการวัดผลข้อมูลประมาณ 300 จุด และแสดงให้เห็นถึงผลของระบบนิวโรฟัซซี่ ในรูปที่ 5.13 (b) จะให้ตำแหน่ง ZMP ใกล้กึ่งกลางฝ่าเท้ามากกว่า ระบบฟัซซี่ปกติ ในรูปที่ 5.13(b) ซึ่งแสดงให้เห็นถึงความมีเสถียรภาพในการเดินมีมากขึ้น และรูปที่ 5.14 แสดงถึงการเปลี่ยนแปลงของค่าฟังก์ชันสมาชิกของระบบนิวโรฟัซซี่หลังจากการเรียนรู้โดยใช้ค่า ZMP ที่เหมาะสมเป็นเกณฑ์



รูปที่ 5.13 แสดงตำแหน่ง ZMP ที่ถูกควบคุมโดยระบบนิวโรฟัซซี่(b) และระบบฟัซซี่ (a).



รูปที่ 5.14 แสดงค่าฟังก์ชันสมาชิกที่ตำแหน่งข้อต่อที่ 3 ก่อน(a). และ หลัง (b) การเรียนรู้

และได้กฎของฟัซซี่หลังจากการเรียนรู้ ดังตัวอย่างต่อไปนี้

if $x_3(k)$ is NB and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.37x_3(k) + 0.05x_6(k) + 2.14$,
 if $x_3(k)$ is NS and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.91x_3(k) + 0.06x_6(k) + 0.23$,
 if $x_3(k)$ is Z and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.40x_3(k) + 0.03x_6(k) + 5.63$,
 if $x_3(k)$ is PS and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.83x_3(k) + 0.09x_6(k) + 0.03$,
 if $x_3(k)$ is PB and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.46x_3(k) + 0.02x_6(k) + 1.11$,
 if $x_3(k)$ is NB and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.37x_3(k) + 0.05x_6(k) + 2.14$, and etc.

if $x_3(k)$ is NB and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.24x_3(k) + 0.05x_6(k) + 1.29$,
 if $x_3(k)$ is NS and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.31x_3(k) + 0.11x_6(k) + 0.45$,
 if $x_3(k)$ is Z and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.32x_3(k) + 0.07x_6(k) + 4.80$,
 if $x_3(k)$ is PS and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.11x_3(k) + 0.12x_6(k) + 5.01$,
 if $x_3(k)$ is PB and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.67x_3(k) + 0.09x_6(k) + 2.3$,
 if $x_3(k)$ is NB and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.45x_3(k) + 0.23x_6(k) + 6.67$, and etc.

จากที่กล่าวมาในงานวิจัยนี้ได้ทำการประยุกต์ใช้ระบบนิวโรฟัซซีแบบใหม่เพื่อนำไปใช้กับการสังเคราะห์ท่าทางการเดินของหุ่นยนต์สองขา พบว่าให้ผลการทดลองกว่าระบบฟัซซีแบบเดิม ระบบนิวโรฟัซซีนี้สามารถสร้างฟังก์ชันสมาชิกที่เหมาะสมและกฎของฟัซซีที่เหมาะสมให้สอดคล้องกับกลุ่มข้อมูลขาเข้าและออก และดัชนีสมรรถนะที่กำหนด จึงทำให้สามารถนำไปประยุกต์ใช้กับระบบ(Physic Model) แบบอื่นๆได้อีก

5.3 การสร้างแบบจำลองทางกายภาพสำหรับหุ่นยนต์ฮิวแมนนอยด์ ด้วยระบบนิวโรฟัซซี

ในขั้นต้น สำหรับงานวิจัยนี้ได้นำระบบนิวโรฟัซซีที่ได้ออกแบบขึ้นมาทำการจำลองระบบทางกายภาพของหุ่นยนต์ฮิวแมนนอยด์ โดยทำการเก็บข้อมูลอันได้แก่ ค่าตำแหน่ง ความเร็วและความเร่งของแต่ละข้อต่อของหุ่นยนต์ ตำแหน่งการเคลื่อนที่ของปลายเท้า รวมถึงตำแหน่งของแรงที่กระทำที่ฝ่าเท้า ในแต่ละรูปแบบการเคลื่อนที่ของหุ่นยนต์ เช่น การยืนอยู่กับที่ การเอียงตัวไปด้านข้างทั้งด้านซ้ายและขวา การเอียงตัวไปด้านหน้าทั้งด้านหน้าและหลัง การยกขาซ้ายและขาขวา เป็นต้น และได้ทำการหาความสัมพันธ์ระหว่างค่าที่กล่าวมา โดยแบ่งออกเป็นสี่รูปแบบความสัมพันธ์ได้แก่

- ความสัมพันธ์ระหว่างตำแหน่งการเคลื่อนที่ของฝ่าเท้าหุ่นยนต์และตำแหน่งแรงที่กระทำที่ฝ่าเท้า โดยข้อมูลขาเข้าคือ ตำแหน่งในแนวแกน x, y, z ของแต่ละขารวมทั้งหมด 6 ข้อมูล และข้อมูลขาออกคือตำแหน่งแรงที่กระทำที่ฝ่าเท้าในแนวแกน x, y ทั้งสองขารวม 4 ข้อมูล
- ความสัมพันธ์ระหว่างตำแหน่งของแรงที่กระทำที่ฝ่าเท้าและตำแหน่งการเคลื่อนที่ของฝ่าเท้าหุ่นยนต์ โดยข้อมูลขาเข้าคือ ตำแหน่งแรงที่กระทำที่ฝ่าเท้าในแนวแกน x, y ทั้งสองขารวม 4 ข้อมูล และข้อมูลขาออกคือตำแหน่งแรงที่กระทำที่ฝ่าเท้าในแนวแกน x, y ทั้งสองขารวม 4 ข้อมูล ตำแหน่งในแนวแกน x, y, z ของแต่ละขารวมทั้งหมด 6 ข้อมูล

- ความสัมพันธ์ระหว่างตำแหน่งการเคลื่อนที่ของแต่ละข้อต่อและตำแหน่งของแรงที่กระทำที่ฝ่าเท้า โดยข้อมูลขาเข้าคือตำแหน่งการเคลื่อนที่ของแต่ละข้อต่อ $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9, \theta_{10}, \theta_{11}, \theta_{12})$ รวมทั้งหมด 12 ข้อมูล และข้อมูลขาออกคือตำแหน่งแรงที่กระทำที่ฝ่าเท้าในแนวแกน x, y ทั้งสองขารวม 4 ข้อมูล
- ความสัมพันธ์ระหว่างตำแหน่งการเคลื่อนที่ของแต่ละข้อต่อและตำแหน่งของแรงที่กระทำที่ฝ่าเท้า โดยข้อมูลขาเข้าคือตำแหน่งการเคลื่อนที่ของแต่ละข้อต่อ $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9, \theta_{10}, \theta_{11}, \theta_{12})$ รวมทั้งหมด 12 ข้อมูล และข้อมูลขาออกคือตำแหน่งแรงที่กระทำที่ฝ่าเท้าในแนวแกน x, y ทั้งสองขารวม 4 ข้อมูล

ในการพิจารณาว่าสามารถใช้ระบบนิวโรฟัซซีหาความสัมพันธ์ระหว่างข้อมูลขาเข้าและข้อมูลขาออกนั้น สามารถใช้สมการที่ 5.4 ที่กล่าวต่อไปในการประเมินค่าความถูกต้องได้ ดังสมการ

$$VAF = 100\% * [1 - \frac{\text{var}(y1 - y2)}{\text{var}(y1)}] \quad (5.4)$$

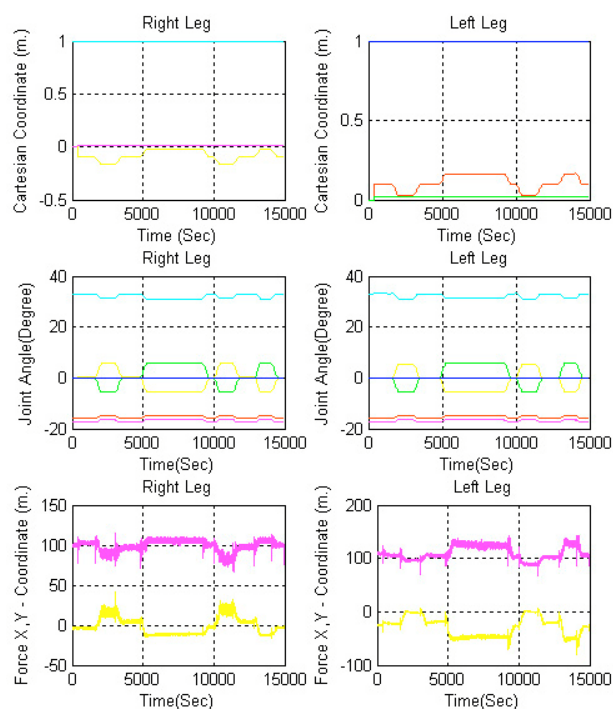
โดยค่า Variance accounted for (VAF) นี้ใช้ในการเปรียบเทียบสัญญาณสองสัญญาณ นั่นคือสัญญาณจากระบบจริงและที่สามารถหาค่าได้จากระบบนิวโรฟัซซีของความสัมพันธ์นั้นๆ ดังนั้นถ้าค่า VAF มีค่าต่ำ ก็แสดงว่าสัญญาณทั้งสองมีค่าไม่ตรงกันมากๆ

ดังนั้นจากที่กล่าวมาจึงสามารถหาความสัมพันธ์ทั้งสี่แบบสามารถใช้ระบบนิวโรฟัซซีในการสร้างแบบจำลองทางคณิตศาสตร์ของความสัมพันธ์เหล่านั้นได้ ทำให้สามารถประยุกต์ระบบนิวโรฟัซซีในการสร้างระบบควบคุมและการสังเคราะห์ท่าทางการเดินต่อไปได้

5.3.1 ผลการทดลอง

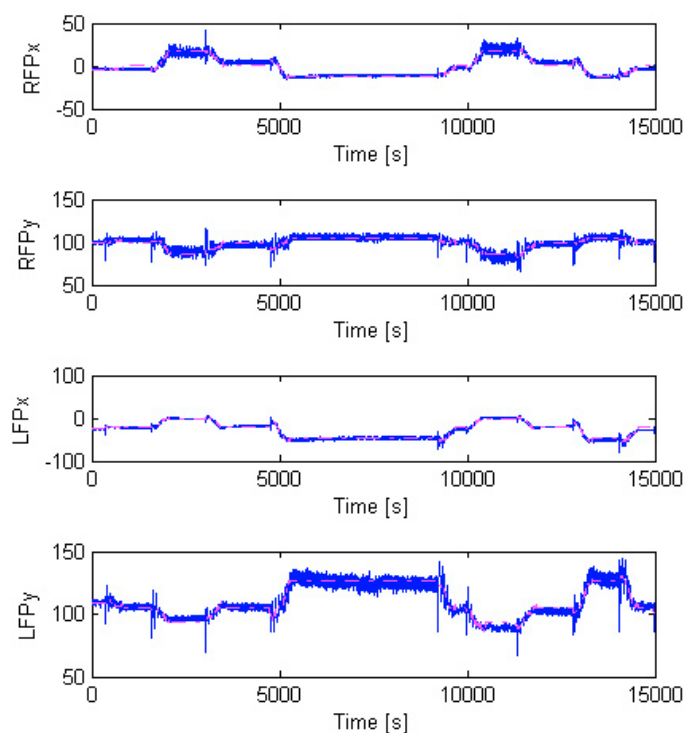
ในการทดลองนี้ ได้ทำการเคลื่อนที่หุ่นยนต์ในหลายรูปแบบ เช่น การยืนอยู่กับที่ การเอียงตัวไปด้านข้างทั้งด้านซ้ายและขวา การเอียงตัวไปด้านหน้าทั้งด้านหน้าและหลัง การยกขาซ้ายและขวามาเป็นต้น และเก็บค่าโดยทำการเก็บข้อมูลอันได้แก่ ตำแหน่งการเคลื่อนที่ของปลายเท้า (Cartesian coordinate of foot) ค่าตำแหน่ง ความเร็วและความเร่งของแต่ละข้อต่อของหุ่นยนต์ (Joint angle of leg) รวมถึงตำแหน่งของแรงที่กระทำที่ฝ่าเท้า (Force x-y Coordinate of foot) ดังรูปที่ 5.15 เป็นลักษณะการเคลื่อนที่แบบ ยืนสองขาและโยกตัวไปซ้ายและขวา)

จากนั้นนำข้อมูลที่เก็บมาไว้เพื่อสร้างความสัมพันธ์ ระหว่างข้อมูลขาเข้าและขาออกเป็นสี่แบบ ตามที่เสนอในหัวข้อที่ 5 เพื่อนำไปใช้ในการเรียนรู้ของระบบนิวโรฟัซซีทั้งสี่แบบต่อไป ตัวอย่างเช่นรูปแบบความสัมพันธ์ที่หนึ่ง โดยข้อมูลขาเข้าคือ ตำแหน่งเส้นทางการเคลื่อนที่ของแต่ละข้อต่อ $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8, \theta_9, \theta_{10}, \theta_{11}, \theta_{12})$ และข้อมูลขาออกคือตำแหน่งของ ZMP (x_{ZMP}, y_{ZMP}) ซึ่งในที่นี้คือตำแหน่งของแรงที่กระทำที่ฝ่าเท้านั่นเอง ตัวอย่างสมาชิกฟัซซี เป็นดังรูปที่ 8 แสดงถึงค่าฟังก์ชันสมาชิก (membership function) ก่อนและหลังการเรียนรู้โดยใช้กลุ่มข้อมูลข้างต้น



รูปที่ 5.15 ข้อมูลท่าทางการเดินของหุ่นยนต์ฮิวแมนนอยด์แยกเป็นขาขวาและซ้ายตามลำดับ

ส่วนรูปที่ 5.16 แสดงถึงค่าข้อมูลขาออกที่ได้เปรียบกันระหว่างระบบนิวโรฟัชชีและข้อมูลจริงโดย RFPx คือตำแหน่งแรงในแนวแกน x ที่ขาขวา RFPy คือตำแหน่งแรงในแนวแกน y ที่ขาขวา ส่วน LFPy และ LFPx แทนตำแหน่งในฝ่าเท้าซ้ายนั่นเอง



รูปที่ 5.16 แสดงค่าข้อมูลขาออกที่ได้เปรียบกันระหว่างระบบนิวโรฟัชชีและข้อมูลจริง

เราใช้สมการคำนวณค่าดัชนีสมรรถนะ(VAF) ของค่าที่ได้ระหว่างระบบนิวโรฟuzzyกับข้อมูลจริง สำหรับความสัมพันธ์ทั้งสี่แบบ สามารถสรุปได้ดังนี้

ก.ความสัมพันธ์ระหว่างตำแหน่งการเคลื่อนที่ของฝ่าเท้าหุ่นยนต์และตำแหน่งแรงที่กระทำที่ฝ่าเท้า

$$VAF_1 = \begin{bmatrix} 93.88 \\ 80.77 \\ 97.68 \\ 90.95 \end{bmatrix}$$

ข.ความสัมพันธ์ระหว่างตำแหน่งของแรงที่กระทำที่ฝ่าเท้าและตำแหน่งการเคลื่อนที่ของฝ่าเท้าหุ่นยนต์

$$VAF_2 = \begin{bmatrix} 93.21 \\ 77.97 \\ 50.90 \\ 90.35 \\ 78.02 \\ 50.47 \end{bmatrix}$$

ค.ความสัมพันธ์ระหว่างตำแหน่งการเคลื่อนที่ของแต่ละข้อต่อและตำแหน่งของแรงที่กระทำที่ฝ่าเท้า

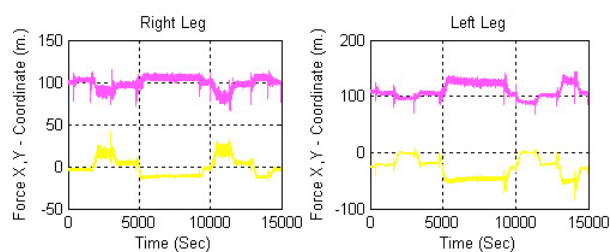
$$VAF_3 = \begin{bmatrix} 97.95 \\ 85.07 \\ 99.00 \\ 94.24 \end{bmatrix}$$

ง.ความสัมพันธ์ระหว่างตำแหน่งการเคลื่อนที่ของแต่ละข้อต่อและตำแหน่งของแรงที่กระทำที่ฝ่าเท้า

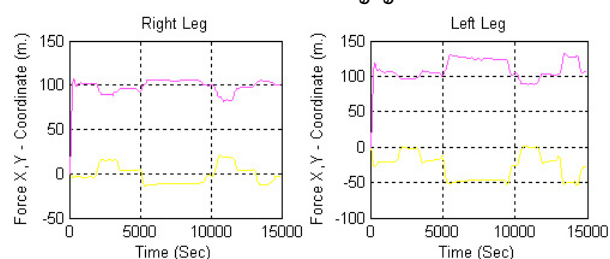
$$VAF_4 = \begin{bmatrix} 97.38 \\ 85.07 \\ 87.06 \\ 87.55 \\ 97.35 \\ 76.55 \\ 97.32 \\ 85.33 \\ 86.11 \\ 87.33 \\ 97.12 \\ 72.31 \end{bmatrix}$$

จากค่าดัชนีสมรรถนะทั้งสี่แบบพบว่ารูปแบบความสัมพันธ์ที่สองมีค่าต่ำ อันเนื่องมาจากผลของสัญญาณรบกวนที่เกิดที่อุปกรณ์วัดแรงที่ฝ่าเท้า ดังนั้นได้ทำการทดลองเพิ่มโดยใช้โปรแกรมตัวกรองสัญญาณเข้าช่วย ดังรูปที่ 5.17 และใช้ระบบนิวโรฟuzzyเรียนรู้ใหม่ พบว่าค่าดัชนีสมรรถนะมีค่าสูงขึ้น แสดงให้เห็นว่าสัญญาณรบกวนมีผลต่อการนำค่าแรงที่กระทำที่ฝ่าเท้าไปใช้งานต่อไป โดยค่าดัชนีสมรรถนะที่ได้ใหม่มีดังนี้

$$VAF_{2,Filter} = \begin{bmatrix} 94.73 \\ 88.88 \\ 70.75 \\ 94.92 \\ 88.49 \\ 70.07 \end{bmatrix}$$



ก. ก่อนทำการกรองสัญญาณรบกวน



ข. หลังทำการกรองสัญญาณรบกวน

รูปที่ 5.17 แสดงการกรองสัญญาณรบกวนที่เกิดในอุปกรณ์วัดแรงที่ฝ่าเท้า

จากการทดลองกับหลาย ๆ รูปแบบการเคลื่อนที่พบว่าได้ผลดัชนีสมรรถนะใกล้เคียงกัน ทำให้เป็นไปได้ที่สามารถนำระบบนิเวศน์ที่ออกแบบไว้มาใช้ในการสร้างระบบสร้างท่าทางการเดินของหุ่นยนต์ด้วยระบบนิเวศน์ต่อไป

บทที่ 6 สรุปงานวิจัยและข้อเสนอแนะ

การคำนวณเชิงอ่อน(Soft computing, SC) คือ ขบวนการในการออกแบบและพัฒนาระบบอัจฉริยะ เพื่อใช้ในการหาคำตอบของระบบที่เราสนใจภายใต้การอาศัยความสามารถของมนุษย์ในเรื่องของการใช้เหตุผล และการปรับตัวในสภาวะการที่ไม่แน่นอนและไม่ถูกต้อง การคำนวณเชิงอ่อนนี้ประกอบด้วยวิธีการหลายวิธี อาทิเช่น ข่ายงานระบบประสาท(Neural Network, NL) ตรรกศาสตร์คลุมเครือ(Fuzzy Logic, FL) การประมาณค่าเหตุผล(Approximate Reasoning, AR) การคำนวณเชิงสถิติ(Probabilistic Computing, PC) และวิธีการหาค่าเหมาะสม เช่นขบวนการเชิงพันธุกรรม(Genetic Algorithm, GA) หรือ Simulated Annealing

เป็นที่ทราบกันดีแล้วว่าตรรกศาสตร์แบบคลุมเครือและข่ายงานระบบประสาทต่างก็มีข้อดีและข้อเสียภายในตัวของมันเองและได้ถูกประยุกต์ใช้กับการควบคุมระบบต่างๆรวมทั้งในเรื่องของระบบการมองเห็น มาเป็นเวลาหลายสิบปี แต่ทั้งสองแบบยังไม่สมบูรณ์ในตัวเอง แต่ถ้านำระบบทั้งสองมารวมกันจะทำให้ได้ระบบที่สมบูรณ์ยิ่งขึ้น เราเรียกระบบใหม่นี้ว่า “นิวโรฟัซซี่ (Neuro-fuzzy system)” โดยรูปแบบการนำตรรกศาสตร์แบบคลุมเครือมารวมกับข่ายงานระบบประสาทนั้น มีผู้เสนอไว้ดังนี้ FALCON [16], ANFIS[15], GARIC[17], NEFCON[18], FUN[19], SONFIN[20], EFUNN[21], DMEFUN[22], Evaluation design of Neuro-Fuzzy [23]

จากที่ได้กล่าวมาแล้วนั้น สิ่งที่เราต้องการคือระบบนิวโรฟัซซี่ที่สามารถ เรียนรู้ด้วยตัวเองและสามารถจดจำรูปแบบความ ไม่เป็นเชิงเส้นของระบบพลวัตทางกลต่างๆได้ จากการศึกษาค้นคว้าผลงานที่ผ่านมาทำให้สามารถสรุปได้ว่า ระบบนิวโรฟัซซี่ที่ควรออกแบบควรเริ่มจากระบบนิวโรฟัซซี่ Takagi-Sugeno-Kang ซึ่งเราสามารถสร้างระบบนี้จากข้อมูลทางตัวเลข(Numerical data) หรือจากข้อมูลเชิงบรรยาย (Knowledge base) ได้ จากระบบนิวโรฟัซซี่ที่ออกแบบนั้นแสดงให้เห็นว่าระบบนิวโรฟัซซี่มีความสามารถในการประมาณค่าสมการไม่เป็นเชิงเส้นได้เป็นอย่างดี และมีความสามารถในการเรียนรู้ระบบหลายข้อมูลอินพุตและและข้อมูลเอาต์พุตได้ดี และที่สำคัญคือเวลาในการเรียนรู้ที่รวดเร็วทำให้เราสามารถนำระบบดังกล่าวนี้ ไปใช้ในการเรียนรู้แบบทันที (Online training) ได้ แต่ในการออกแบบระบบนิวโรฟัซซี่ ขณะนี้ได้พัฒนาระบบไปในสองส่วนคือส่วนการสร้างกฎขึ้นมาอย่างอัตโนมัติในตอนเริ่มต้น (Initial Structure Identification) และการหาค่าพารามิเตอร์ของระบบในส่วน consequent ของแต่ละกฎ ส่วนที่เหลือคือส่วนที่ใช้ตัดสินใจว่าจะเพิ่มหรือลดกฎลงเมื่อมีข้อมูลชุดใหม่เข้ามา ซึ่งจะพัฒนาต่อไปในอนาคต

ในงานวิจัยนี้ได้ทำการประยุกต์ใช้ระบบนิวโรฟัซซี่แบบใหม่เพื่อนำไปใช้กับการสังเคราะห์ท่าทางการเดินของหุ่นยนต์สองขา พบว่าให้ผลการทดลองกว่าระบบฟัซซี่แบบเดิม ระบบนิวโรฟัซซี่นี้สามารถสร้างฟังก์ชันสมาชิกที่เหมาะสมและกฎของฟัซซี่ที่เหมาะสมให้สอดคล้องกับกลุ่มข้อมูลขาเข้าและออก และดัชนีสมรรถนะที่กำหนด จึงทำให้สามารถนำไปประยุกต์ใช้กับระบบ(Physic Model) แบบอื่นๆได้อีก

แต่ในเรื่องของการกำหนดค่าดัชนีสมรรถนะ ซึ่งในที่นี้คือแนวเส้นทาง ZMP นั้น ยังคงเป็นปัญหาที่น่าสนใจอยู่ว่า แนวเส้นทางของ ZMP ควรอยู่กึ่งกลางของฝ่าเท้าหรือไม่ หรือว่าควรอยู่ที่ขอบฝ่าเท้า เนื่องมาจากการเดินแบบพลวัติของหุ่นยนต์ เป็นเรื่องของการเดินแบบไม่รักษาสมดุลในช่วงเวลาหนึ่ง เพื่อให้การเดินมีความเร็วมากขึ้น นอกจากนี้ในการเดินของหุ่นยนต์นั้นเป็นแบบความเร็วคงที่ ระยะก้าวคงที่และระยะยกคงที่ ดังนั้นถ้าตัวแปรเหล่านี้เปลี่ยนแปลงไป จะมีผลต่อการเดินของหุ่นยนต์อย่างไร และมีผลต่อระบบนิวโรฟuzzyเพื่อการสังเคราะห์ท่าทางการเดินอย่างไร อย่างไรก็ตามระบบหุ่นยนต์ในงานวิจัยนี้เป็นหุ่นยนต์แบบ 8 แกนข้อหมุนซึ่งยังไม่คล้ายคลึงกับมนุษย์ จึงควรนำระบบนิวโรฟuzzyนี้เป็นประยุกต์ใช้กับหุ่นยนต์ขนาด 14 แกน ต่อไป

หนังสืออ้างอิง

1. M. Vukobratovic, A. A. Frank, and D. Juricic, "On the Stability of Biped Locomotion," *Proc. IEEE Transactions on Biomedical Engineering*, Vol. BME-17, No.1, pp. 25-36, 1970.
2. M. Vukobratovic, "Mathematical Models of Legged Locomotion Robots," Mihailo Pulpin Institute Beograd, 1975.
3. D. J. Todd, "Walking Machines : An introduction to legged robot." New York, NY; Chapman and Hall, 1985
4. M. H. Raibert, "Legged Robots That Balance," The MIT Press, Cambridge, Massachusetts, 1986
5. S. Kajita and K. Tani. "Experimental Study of Biped Dynamic Walking," In *Proc. IEEE International Conference on Robotics and Automation*, May, pp. 13-19, 1995.
6. A. Sano and J. Furusho, "Realization of Natural Dynamic Walking Using The Angular Momentum Information." In *Proc. IEEE International Conference on Robotics and Automation*, May, pp. 1476-1481, 1990.
7. H. Benbrahim and J. Franklin, "Biped Dynamic Walking Using Reinforcement Learning." *Robotics and Autonomous System*. 22, pp. 283-302, 1997.
8. W. T. Miller., P. W. Latham. and S. M. Scalera, "Bipedal gait adaptation for walking with dynamic balance." In *Proc. 1991 Amer. Control Conf.*, Boston, MA, June 26-28, pp. 1603-1608, 1991.
9. J. Hu., J. Pratt and G. Pratt., "Adaptive Dynamic Control of a Bipedal Walking Robot with Radial Basis Function Neural Networks," *IEEE International Conference on intelligent Robots and Systems*, Victoria, BC, Canada, October, 1998.
10. J. Hu., J. Pratt and G. Pratt., "Adaptive CMAC Neural Network Control of Bipedal Locomotion." *IEEE International Conference on Robotics and Automation*, May, 1999.
11. E. Tunstel, M.R. Akbarzadeh-T, K. Kumbla and M. Jamshidi., "Soft Computing Paradigm for Learning Fuzzy Controller with applications to robotics," In *Proc. IEEE 1996 International Conference on Robotics & Automation*, May, pp. 355-359, 1996.
12. P. Arena and G. Muscato. "New-trads in the control of walking robots," In *Proc. IEEE 1998 International Conference on Control Applications*, September, 1998.
13. H. Kazuo, H. Masato, H. Yuji and T. Toru, " The Development of Honda Humanoid Robot," In *Proc. IEEE 1998 International Conference on Robotics & Automation*, May, pp. 1321-1326, 1998.
- 14 F. Yamazaki, T. Matsui, T. Miyashita and H. Kitano, PINO The Humanoid That Walk, *Proc. IEEE-RAS Int. Conf. on Humanoid Robot*, 2000.

15. Jang R, Neuro-Fuzzy Modeling: Architectures, Analyses and Applications, *PhD Thesis*, University of California, Berkeley, July 1992.
16. Tano S, Oyama T, Arnould T, Deep combination of Fuzzy Inference and Neural Network in Fuzzy Inference, *Fuzzy Sets and Systems*, 82(2) pp. 151-160, 1996.
17. Lin C T and Lee C S G, Neural Network based Fuzzy Logic Control and Decision System, *IEEE Transactions on Computation*. (40(12): pp. 1320-1336, 1991.
18. Nauck D, Kruse R, Neuro-Fuzzy Systems for Function Approximation, *4th International Workshop Fuzzy-Neuro Systems*, 1997.
19. Sulzberger SM, Tschicholg-Gurman NN, Vestli SJ, FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks, *In Proceedings of IEEE Conference on Neural Networks*, San Francisco, pp 312-316, March 1993.
20. Juang Chia Feng, Lin Chin Teng, An Online Self Constructing Neural Fuzzy Inference Network and its Applications, *IEEE Transactions on Fuzzy Systems*, Vol 6, No.1, pp. 12-32, 1998.
21. Abraham A & Nath B, Evolutionary Design of Neuro-Fuzzy Systems – A Generic Framework, *In Proceedings of the 4th Japan – Australia joint Workshop on Intelligent and Evolutionary Systems*, Japan, November 2000.
22. Abraham A & Nath B, Designing Optimal Neuro-Fuzzy Systems for Intelligent Control, *In proceedings of the Sixth International Conference on Control Automation Robotics Computer Vision*, (ICARCV 2000), Singapore, December 2000.
23. Kasabov N and Qun Song, Dynamic Evolving Fuzzy Neural Networks with 'm-out-of-n' Activation Nodes for On-line Adaptive Systems, Technical Report TR99/04, Department of information science, University of Otago, 1999.
24. Mackey MC, Glass L, Oscillation and Chaos in Physiological Control Systems, *Science* Vol 197, pp.287-289, 1977.
25. Bherenji H R and Khedkar P, Learning and Tuning Fuzzy Logic Controllers through Reinforcements, *IEEE Transactions on Neural Networks*, Vol (3), pp. 724-740, 1992.
26. S. Kitamura, Y. Kurematsu, Y. Nakai, Application of the Neural Network for the Trajectory Planing of a Biped Locomotion Robot, *Neural Networks*, vol. 1, pp. 344-356, 1988.
27. W. Salatan, K. Y. Yi, Y. F. Zheng, Reinforcement Learning for a Biped Robot to Climb Sloping Surfaces, *Journal of Robotic Systems*, Vol. 14, No. 4, April 1997, pp. 283-296.
28. C. L. Shih, Y.Z. Li, S. Churng, T. T. Lee, and W. A. Gruver, "Trajectory Synthesis and Physical Admissibility for a Biped Robot During the Single-Support Phase," *Proc. IEEE Int. Conference on Robotics and Automation*, Vol.3, pp. 1646 - 1652, 1990.

29. C. L. Shih, Yun Zhu, and W. A. Gruver, "Optimization of the Biped Robot Trajectory," *Proc. IEEE International Conference on Robotics and Automation*, pp. 899- 903, 1991.
30. C. L. Shih, Yun Zhu, and W. A. Gruver, "Fuzzy Logic Force Control for a Biped Robot," *Proc. IEEE International Conference on Robotics and Automation*, Vol.1, pp. 269 - 274, 1991.
31. T. Arakawa, T. Fukuda, Natural Motion Generation of Biped Locomotion Robot Using Hierarchical Trajectory Generation Method Consisting of GA, EP Layers, *Proc. IEEE Conf. on Robotics and Automation*, Albuquerque, pp 211-216, April 1997.
32. M-Y Cheng, C-S. Lin, Genetic Algorithm for control Design of Biped Locomotion, *Journal of Robotic Systems*, Vol. 14, No. 5, pp. 365-373, May 1997.
33. J-G Juang, Fuzzy Neural Network Approaches for Robotics Gait Synthesis, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, Vol. 30, No.4, pp. 594-601, 2000.
34. Zhou. C., Neuro-fuzzy gait synthesis with reinforcement learning for a biped walking robot, *Soft Computing* 4, pp. 238-250, 2000.
35. Li, Q., Takanishi, A. and Kato, I., Learning Control of Compensative Trunk Motion for biped Walking Robot based on ZMP Stability Criterion, *In Proc. IEEE/RSJ Int. Conf. on intelligent Robots and System*, pp.597-603, 1992.
36. Yamaguchi, J., Takanishi, A. and Kato, I., Development of a Biped Walking Robot Compensating for Three-Axis Moment by Trunk Motion, *In Proc. IEEE/RSJ Proc. Int. Conf. on intelligent Robots and System*, pp.561-566, 1993.
37. Park, J. H. and Rhee, Y. K., ZMP trajectory generation for reduced trunk motions of biped robots, *In IEEE Int. Conf. on Robotics and Automation*, pp. 2014-2021, 1998.
38. Park, Fuzzy-logic ZMP trajectory generation for reduced trunk motions of biped robots, *Fuzzy Sets and Systems*, Vol. 134, Issue 1, pp. 189-203, 2003.
39. Takagi, T, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, *In Proc. IEEE Transaction on SMC*, pp. 116-132, 1985.
40. Babuska, R., Neuro-fuzzy methods for modeling and identification, *In Recent Advances in intelligent Paradigms and Application*, pp. 161-186, Springer-Verlag, 2002.
41. Featherstone, R. and Orin, D., Robot Dynamics: Equations and Algorithms, *In Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 826-834, 2000.

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

1. การเสนอผลงานในที่ประชุมวิชาการนานาชาติ

Hataitep Wongsuwarn and Djitt Laowattana, ***“Bipedal Gait Synthesizer Using Adaptive Neuro-fuzzy Network,”*** in Proceedings of the First Asia International Symposium on Mechatronics (AISM 2004), September 27~30, 2004, Xi'an, China

Bipedal Gait Synthesizer Using Adaptive Neuro-fuzzy Network

Hataitep Wongsuwarn¹ and Djitt Laowattana²

Institute of Field roBOTics Development (FIBO)
King Mongkut's University of Technology Thonburi, Bangkok, Thailand 10140
e-mail : {hataitep¹, djitt²}@fibo.kmutt.ac.th

Abstract:

This paper presents a bipedal gait synthesis technique based on an adaptive neuro-fuzzy controller. At first, we developed an architecture which generates trunk trajectory in sagittal plane and lateral plane. Our proposed gait synthesizer was trained through sets of a Zero Moment Point (ZMP). In this paper, we experimented on an 8-link biped robot walking in sagittal plane and lateral plane. The robot is both headless and armless. The hierarchical control software for the biped robot was developed using MATLAB, SIMULINK, Fuzzy Logic Toolbox and Neural Network Toolbox. The main contribution of this paper is a general neuro-fuzzy synthesis of using gait generating algorithm. This algorithm implementation leads to a dynamic bipedal walking without requiring for dynamic equations and nominal data.

Keywords: Neuro-fuzzy System, Biped robot, Gait synthesis, Dynamic walking

Selected References

1. Vokobratovic M, Borovac B, Surla D, Stokic D (1990) Biped Locomotion: Dynamics, Stability, Control and Application, Springer-Verlag
2. Li Q, Takanishi A, Kato I (1992) Learning control compensative trunk motion for a biped walking robot based on ZMP stability criterion, In: Proc IEEE/RST Int Conf on Intelligent Robots and Systems, pp. 597-603.
3. R. Babuska & H. B. Verbruggen, An overview of fuzzy modeling for control, Control Eng. Practice, Vol. 4, No. 11, pp. 1593-1606, 1996, Elsevier Science Ltd.
4. Park JH, Rhee YK (1998) ZMP trajectory generation for reduced trunk motions of biped robots, In: Proc IEEE Conf Robotics and Automation, pp. 2014-2021

Bipedal Gait Synthesizer Using Adaptive Neuro-fuzzy Network

Hataitep Wongsuwarn¹ and Djitt Laowattana²

Institute of Field roBOtics Development (FIBO)
King Mongkut's University of Technology Thonburi, Bangkok, Thailand 10140
e-mail : {hataitep¹, djitt²}@fibo.kmutt.ac.th

Abstract: This paper presents a bipedal gait synthesis technique based on an adaptive neuro-fuzzy controller. At first, we developed an architecture which generates trunk trajectory in sagittal plane and lateral plane. Our proposed gait synthesizer was trained through sets of a Zero Moment Point (ZMP). In this paper, we experimented on an 8-link biped robot walking in sagittal plane and lateral plane. The robot is both headless and armless. The hierarchical control software for the biped robot was developed using MATLAB, SIMULINK, Fuzzy Logic Toolbox and Neural Network Toolbox. The main contribution of this paper is a general neuro-fuzzy synthesis of using gait generating algorithm. This algorithm implementation leads to a dynamic bipedal walking without requiring for dynamic equations and nominal data.

Keywords: Neuro-fuzzy System, Biped robot, Gait synthesis, Dynamic walking

1 Introduction

Bipedal walking robots have been studied by researchers for about three decades. Summarized discussions of the early history of bipedal walking machines have been reported in Todd[1] and Raibert[2]. Control approaches developed for bipeds are generally based on simplified robot dynamics [3,4]. Other investigators have benefited neural network as an on-line control method. Benbrahim and Franklin [5] have used a reinforcement learning for biped dynamic walking. Miller et al. [6] combined standard supervised and temporal difference learning schemes in order to achieve an on-line gait adaptation for a simulated two-dimensional biped with massless legs. Adaptive learning control approaches with neural networks i.e. Radial Basis Function Neural Networks and Cerebellar Model Articulation Controller (CMAC) Neural Networks have been developed in order to accommodate nonlinear dynamics and disturbances from environment [7,8]. However, it is difficult to maintain stable and smooth locomotion while the biped is in a dynamic walking mode.

In this paper, we propose an supervised learning based on neuro-fuzzy gait synthesis strategy for a biped walking robot. The architecture is applied to our eight Axis experimental bipedal robots, walking in the sagittal plane and lateral plane. we will illustrate and discuss several ways of implementing the architecture for steady walking. Then, we will extend its to a 14 axis bipedal robot.

In the following section, we introduce the model of the biped robot, its gait and a relevant control system structure. In the third section, Our Neuro-Fuzzy Controller based on Adaptive Neuro-Fuzzy Inference System is described. Moreover, in this section, a learning-based neuro-fuzzy gait synthesis scheme is also proposed for a biped walking robot. We then demonstrate how this gait synthesis method could be used for dynamic balance of the trunk sagittal plane and lateral plane. In the fourth section, experimental results of the fuzzy gait synthesis and neuro-fuzzy gait synthesis by the optimize learning are presented and compared. Finally, we discuss our some concluding remarks.

2 Robot Design and Dynamics of the biped locomotion

M. Vukobratovic [8] is credited for formulating a theoretical foundation of the biped robot locomotion. He numerically calculated the general trajectories for a stable biped locomotion, dividing the general stability into three groups: body stability, body path stability, and gait stability. He also introduced the concept of a Zero Moment Point (ZMP) and the method of compensative inverted pendulum stable walking. The model that he used had six degrees of freedom (DOFs). Kato [9, 10], using Vukobratovic's ideas, built a robot called WL-12R. This biped robot has eight DOFs and has trunk, acting like an inverted pendulum. The motion of the inverted pendulum helps matching the actual and the desired ZMP. This upper body movement is very important because the

position of the ZMP is used to achieve the stability of walking. Although the robot's walking is stable on a smooth ground profile, it is unable to walk on an uneven ground since the calculation of their control scheme assumes a simple model and is very time consuming. To solve this problem, Kato added a learning control algorithm of the compensatory trunk motion [11].

In this paper, we simplify an experimental biped robot to an eight link model on a study robot walking in sagittal plane and lateral plane (see Fig 1). The robot is also headless and armless. Each leg has three active DOFs of which one DOF is available at the knee (x_1, x_4), ankle (x_2, x_5), hip (x_3, x_6) and two DOFs at trunk (x_7, x_8). It has a total weight of about 1.5 kg (see Fig 2).

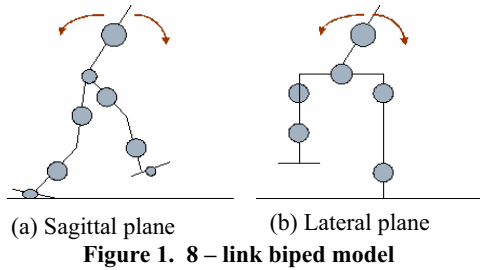


Figure 1. 8 – link biped model

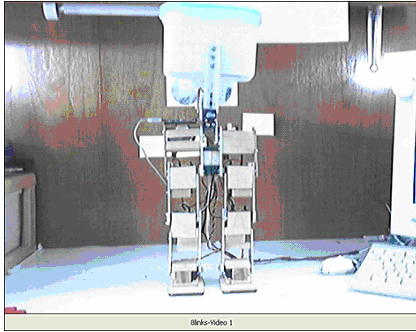


Figure 2. Experimental biped robot

Various approaches exist for modeling the multibody system (MBS) dynamics of a tree-structured legged robot, subject to unilateral contact constraints. They deal with quite different characteristics regarding efficiency and accuracy in simulation and optimization. Our selected MBS modeling and computational approach relies upon the Articulated Body Algorithm (ABA) due to its superior modularity and computational efficiency for high dimensional systems [12].

The basic equations of motion are general for a rigid, multibody system experiencing contact forces $\ddot{q} = M(q)^{-1}(Bu - C(q, \dot{q}) - g(q) + J_c(q)^T f_c)$, and $0 = g_c(q)$. (1)

Where N equals the number of links in the system, m equals the number of actively controlled joints, $M \in \mathbb{R}^{N \times N}$ is the square, positive-definite

mass-inertia matrix, $C \in \mathbb{R}^N$ contains the Coriolis and centrifugal forces, $g \in \mathbb{R}^N$ the gravitational forces, and $u(t) \in \mathbb{R}^m$ are the control input functions which are mapped with the constant matrix $B \in \mathbb{R}^{N \times m}$ to the actively controlled joints. The ground contact constraints $g_c \in \mathbb{R}^{n_c}$ represent holonomic constraints on the system from which the constraint Jacobian may be obtained $J_c = \frac{\partial g_c}{\partial q} \in \mathbb{R}^{n_c}$, while $f_c \in \mathbb{R}^{n_c}$ is the

ground constraint force. q , $\dot{q}$, and $\ddot{q}$ are the generalized position, velocity and acceleration vectors respectively.

Depending upon whether a statically stable or dynamically stable biped gait is desired, the optimization problem formulation will have different periodicity, symmetry, and kinematic phase boundary constraints depending on the foot contact positions (Fig. 3). The number of phases may also differ. We model the static and dynamically stable gaits as follows.

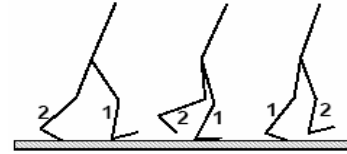


Figure 3. Three Phases of Dynamic Gait with Different Foot Contact Positions for Leg 1: (1) Heel Roll, (2) Flat Contact, (3) Toe Roll

Statically Stable Gait:

- 1: Foot 1 has a flat contact, Foot 2 is swinging freely.
- 2: Foot 1 has a flat contact, Foot 2 has a flat contact.

Dynamically Stable Gait:

- 1: Foot 1 heel has a roll contact, Foot 2 has a toe roll contact.
- 2: Foot 1 has a flat contact, Foot 2 is swinging freely.
- 3: Foot 1 has a toe roll contact, Foot 2 is swinging freely.

To synthesize the biped walking motion and to achieve the dynamic walking, the zero moment point (ZMP) is usually used as a criterion [8-11, 13, 14]. Therefore, we can determine the biped motion to minimize the following performance index [15].

$$\text{Minimize } \int_{t_i}^{t_f} \|P_{zmp}(t) - P_{zmp}^d(t)\|^2 dt, \quad (2)$$

subject to the boundary conditions of both trunk motion $x_7(t)$ and $x_8(t)$ at time t and t_f . The terms $P_{ZMP}(t)$ and $P_{ZMP}^d(t)$ are the actual and desired ZMP positions, respectively

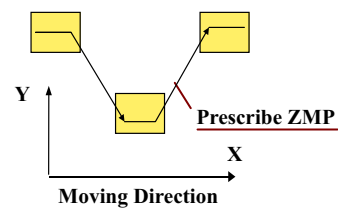


Figure 4. The prescribed ZMP of the biped robot

With respect to a fixed Cartesian coordinate system, 0-XYZ (see Fig. 4), we have the following ZMP equations [16] namely

$$x_{zmp} = \frac{\sum_{i=0}^{n-1} m_i (z_i + g) x_i - \sum_{i=0}^{n-1} m_i \ddot{x}_i z_i}{\sum_{i=0}^{n-1} (\ddot{z}_i + g)}, \text{ and } (3)$$

$$y_{zmp} = \frac{\sum_{i=0}^{n-1} m_i (z_i + g) y_i - \sum_{i=0}^{n-1} m_i \ddot{y}_i z_i}{\sum_{i=0}^{n-1} (\ddot{z}_i + g)}. \quad (4)$$

The term $m_i (i = 0, 1, \dots, n - 1)$ is the mass of the link i . The term (x_i, y_i, z_i) is the coordinate of the center of mass of the link i , $(x_{ZMP}, y_{ZMP}, 0)$ is the coordinate of the ZMP with respect to the 0-XYZ and g is the acceleration of gravity.

For practical measurement of reaction forces from the ground, Force Sensing Resistors (FSRs) are used. The FSR device is a polymer thick film which exhibits a decrease in resistance with an increase in the force. The FSR devices measure reaction forces from the ground as shown on Figure 5. The relation $p_i = [x_i, y_i]^T$ denotes the places where FSR devices are attached, while the location of the ZMP is at $p_{ZMP} = [x_{ZMP}, y_{ZMP}]^T$. The ZMP is equivalent to the center of the reaction force from the ground. Therefore, the ZMP is computed as follows:

$$x_{ZMP} = \frac{\sum_{i=1}^n f_i x_i}{\sum_{i=1}^n f_i}, \quad y_{ZMP} = \frac{\sum_{i=1}^n f_i y_i}{\sum_{i=1}^n f_i} \quad (5)$$

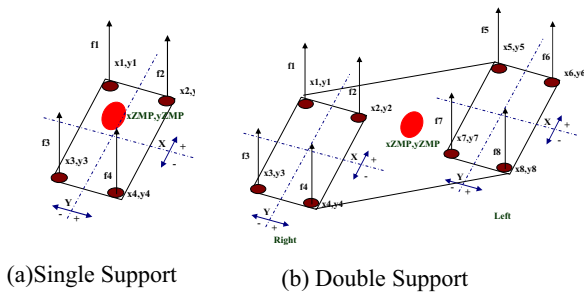


Figure 5. Practical measurement of reaction forces from the ground

Motion of the biped robot is considered into two parts: leg and trunk (or upper-body) motion. Leg motion is related to terrain and environment. For example, if there is an obstacle in the path. Its foot must be lifted up higher than normal amplitude. Under this circumstance, the trajectories of all legs are determined by terrain constraints. The trunk height is constant because of the stationary posture,

i.e. $\ddot{z}_7 = \dot{z}_7 = 0$. All masses are modeled as particles, the equation 3 and 4 become

$$m_7 \ddot{x}_7 z_7 + m_7 g_z x_7 = \sum_{i=1}^6 m_i (\ddot{z}_i - g_z) x_i - \sum_{i=1}^6 m_i (\ddot{x}_i - g_x) z_i + m_7 g_x z_7 + x_{zmp} \left\{ m_7 g_z - \sum_{i=1}^6 m_i (\ddot{z}_i - g_z) + \sum_k F_{z,k} \right\}, \text{ and } (6)$$

$$m_8 \ddot{x}_8 z_8 + m_8 g_z x_8 = \sum_{i=1}^6 m_i (\ddot{z}_i - g_z) x_i - \sum_{i=1}^6 m_i (\ddot{x}_i - g_x) z_i + m_8 g_x z_8 + y_{zmp} \left\{ m_8 g_z - \sum_{i=1}^6 m_i (\ddot{z}_i - g_z) + \sum_k F_{z,k} \right\}. \quad (7)$$

If the ZMP is determined and no external force exists, the terms on the right-hand side of the equation are known except for external forces and thus differential equation of the equation (6) and (7) can be solved numerically.

3 Neuro-Fuzzy Gait Synthesis Approach

3.1 Adaptive Neuro-Fuzzy Inference System

Both neural networks and fuzzy system are motivated by imitating human reasoning process. In fuzzy systems, relationships are represented explicitly in forms of if-then rules. In neural networks, the relations are not explicitly given, but are coded in designed networks and parameters. Neurofuzzy systems combine semantic transparency of rule-based fuzzy systems with a learning capability of neural networks. Depending on the structure of if-then rules, two main types of fuzzy models are distinguished as mamdani (or linguistic) and takagi-sugeno models. The mamdani model is typically used in knowledge-based (expert) systems, but the takagi-sugeno model used in data-driven systems.

In this paper, we consider only the Takagi - Sugeno-Kang (TSK) model. The TSK was proposed by Takagi, Sugeno, and Kang [16]. They formalized a systematic approach for generating fuzzy rules from an input-output data pairs. The fuzzy IF-THEN rules, for pure fuzzy inference system, are of the following form:

$$\text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and } x_n \text{ is } A_n \text{ then } y = f(x). \quad (8)$$

Where $x = [x_1, x_2, \dots, x_N]^T$, $A_1, A_2, \dots, A_N$ are fuzzy sets in the antecedent, while y is a crisp function in the consequent part. The function $y = f(x)$ is a polynomial in the input variables $x_1, x_2, x_3, \dots, x_N$.

They applied a linear form of this function. The aggregated values of the membership function for the vector x assumed either in a form of MIN operator or in the product form. For M fuzzy rules of the form Eq. (8), they have M membership functions $\mu_1, \mu_2, \mu_3, \dots, \mu_M$. They assume that each antecedent is followed by the consequent:

$$y_i = p_{i0} + \sum_{j=1}^N p_{ij} x_j. \quad (9)$$

Where p_{ij} are the adjustable coefficients, for $i = 1, 2, \dots, M$ and $j = 1, 2, 3, \dots, N$.

Applying fuzzy singleton, a generalized bell function as membership function and algebraic product aggregation of input variables, at the existence of M rules the neuro-fuzzy TSK system output signal $y(x)$ upon excitation by the vector x , are described by equation:

$$y(x) = \frac{1}{\sum_{r=1}^M \left[\prod_{j=1}^N \mu_r(x_j) \right]} \sum_{k=1}^M \left(\left[\prod_{j=1}^N \mu_r(x_j) \right] \left[p_{k0} + \sum_{j=1}^N p_{kj} x_j \right] \right). \quad (10)$$

The adjusted parameters of the system are the nonlinear parameters $(c_j^{(k)}, \sigma_j^{(k)}, b_j^{(k)})$ for $j = 1, 2, \dots, N$ and $k = 1, 2, \dots, M$ of the fuzzifier functions and the linear parameters (weight p_{kj}) of TSK function. In contrary to the mamdani fuzzy inference system, the TSK model generates a crisp output values instead of fuzzy one. This network is simplified, since the defuzzifier is not necessary.

The first-order TSK fuzzy model could be represented in a similar fashion. Consider again the example with two rules:

If x_1 is A_{11} and x_2 is A_{21} then $y_1 = a_{11}x_1 + a_{12}x_2 + b_1$, and

If x_1 is A_{12} and x_2 is A_{22} then $y_2 = a_{21}x_1 + a_{22}x_2 + b_2$.

For which the corresponding network is given in Fig 7. [17].

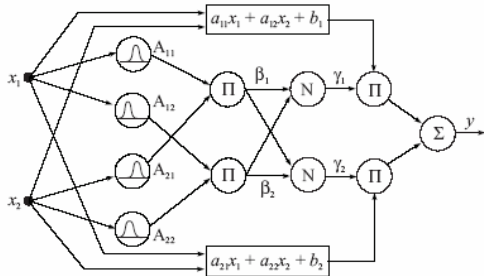


Figure 6. An example of a first-order TSK fuzzy model with two rules systems[17]

The learning of neuro-fuzzy network, which adapts parameters of the bell shape membership function $(c_j^{(k)}, \sigma_j^{(k)}, b_j^{(k)})$ and consequent coefficient, (p_{ij}) can be done either in supervised or self-organizing mode. In this study, we apply hybrid method which is one-shot least-squares estimation of consequent parameters with iterative gradient-based optimization of membership function.

The important problem in the TSK network is to determine of the number of rules that should be used in modeling of data. More rules mean better representation of data processing, but increase of complexity of network and high cost of data processing. Therefore, the procedure for

automatically determining number of rules is required. In our solution, each rule should be associated with one cluster of data. We apply fuzzy-subtractive clustering. The degree of similarity can be calculated by using suitable distance measure. Data vectors are clustered such that data within cluster are as similar, and data from different clusters are as dissimilar as possible.

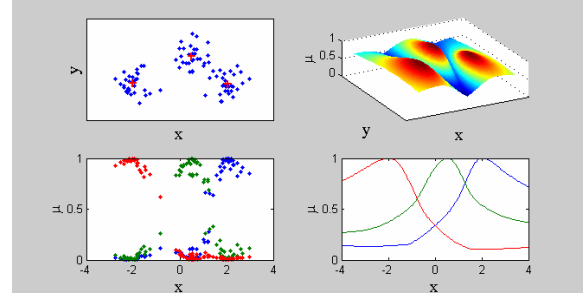


Figure 7. Identification of membership functions through fuzzy subtractive clustering.

Fig. 7 shows an example of three clusters in $\mathbb{R}^2$ with prototype v_1, v_2, v_3 . The data partitioning is expressed in the fuzzy partition matrix $U = [u_{ij}]$ whose elements are membership degrees of the data vector x_k , in the fuzzy clusters with prototypes v_j . The antecedent membership functions are then extracted by projecting the clusters onto the individual variables.

The advantage of the TSK fuzzy system is to provide a compact system equation (9). Therefore, some classical system identification methods, such as parameter estimation and order determination algorithms, can be developed to get the fuzzy inference rules by using input/output data. Similar to neural networks, fuzzy systems are universal approximators. Thus, the TSK fuzzy inference systems are generally for many complex nonlinear practical problems, such as multilink biped robots, and etc.

3.2 Neuro-Fuzzy Gait Synthesis with optimization

Our proposed fuzzy gait synthesis method can be summarized as follow:

Step 1. Prescribing all other joint's trajectories and ZMP trajectories are bounded, except the biped trunk.

There are several methods to prescribe reference trajectories for the biped robot according to the terrain and intuitive knowledge for human's walking. The human's walking pattern and minimizing input energies are most often used. To generate smooth biped locomotion, a curve transition from a line segment into another is required. The polynomial method determines joint, satisfying boundary conditions through optimization algorithms.

Step 2. Constructing gait synthesis rule base.

Human is capable of performing stable and adaptive locomotion by changing his or her ZMP appropriately. In typical human locomotion, the ZMP never stays in a fixed position and rather moves towards in the direction of locomotion [18]. The ZMP trajectory is somehow determined to maximize the stability of locomotion. Based on our observations on human locomotion, we propose that the ZMP should be changed as the body moves during a stride. The ZMP trajectory should be dependent on the leg trajectory. The desired ZMP trajectory is generated depending on the posture of biped robot, especially the positions of the hip and the swing leg. Initially, the fuzzy control rules are mainly derived from human intuitive knowledge on balancing. As an example, seven and five fuzzy membership functions are used for position of hip and any joints trajectory of swing leg, respectively. As show in Fig. 8

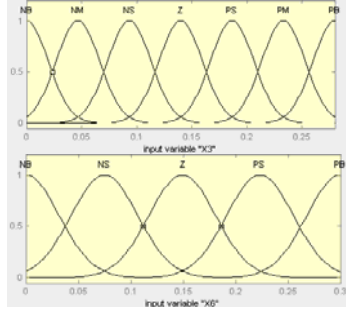


Figure 8. Fuzzy membership function for and (joint angle) systems

After, the desired ZMP are obtained from TSK Neuro-fuzzy inference system as shown in equation (10), the trunk trajectory is generated as shown by equation (6) and (7).

Step 3. Tuning consequent parameters of the fuzzy gait synthesizer

Consequent parameters of the neuro-fuzzy system are updated by using gradient descent algorithm. An error between actual ZMP and desired ZMP is used to adjust the consequent parameters for the biped balance. There is no unique control solution for the biped dynamic walking. Each joint trajectory has different error of ZMP. Thus, the gait synthesizer must be tuned accordingly.

4 Experimental Setup and Results

The biped robot used in the experiment is 50 cm tall. The detailed dimensions and masses of its links are listed in Table 1. Hierarchical control software for the biped robot was developed in visual basic, integrating MATLAB, SIMULINK, Fuzzy Logic Toolbox and Neural Network Toolbox Module. The software package consists of different component, such as biped joint controller, gait editor and gait synthesizer, as shown in Fig 9. Fig 9 shows information about each joint position, body attitude

and trunk motion. We use the ZMP as stabilizing index. For real-time control, the position of an actual ZMP can be calculated by using measured values from force sensors at each foot, as shown in Fig 5.

Link	Length (m)	Mass (kg)	Moment of inertia ($\text{kg} \cdot \text{m}^2$)
Trunk	0.25	0.50	0.15
Left leg	0.25	1.00	0.03
Right leg	0.25	1.00	0.03

Table 1. Parameters of the biped robot

In our experiment, the biped robot is controlled to walk at a constant speed. The step length is 30 mm. and the step height 10 mm. The neuro-fuzzy scheme to change the desired ZMP, according to the biped robot configuration, was compared with the conventional fuzzy scheme. Initial constructions of both schemes are same as equations (8) which are 6 inputs with $(x_1, x_2, x_3, x_4, x_5, x_6)$ and 2 outputs.

The inputs are assigned bell membership function as indicated in Fig 11. Our experimental goal is to control the biped robot with a stable gait on sagittal plane and lateral.

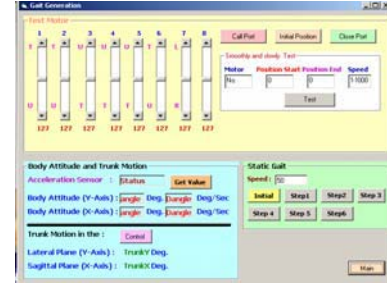


Figure 9. The Control Panel for the neuro-fuzzy gait synthesizer

Fig 10(a) shows the results of fuzzy gait synthesizing, using trunk balancing rules. The control objective is to make the ZMP moving towards the center of supporting foot as much as possible. Approximately 300 data points have been collected and illustrated in Fig. 10. Similarly, we applied our neuro-fuzzy based gait synthesizer. The results are in Fig10 (b). We discovered that the ZMP is much closer to center of supporting foot, under our neuro-fuzzy- based gait synthesizer. Therefore, the biped gait is improved.

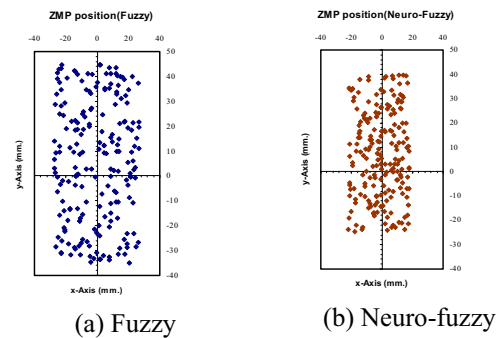


Figure 11 Comparison of ZMP position

Fig 10 shows an example of the learned membership functions for the proposed neuro-fuzzy system on trunk trajectory output. At first, we start initial membership function as show in Fig 11(a). After training, a learned strategy time successful if the ZMP is always within the supporting area is within a certain time. Notice that, in Fig 11, the membership functions were adjusted by the gradient-descent optimization method.

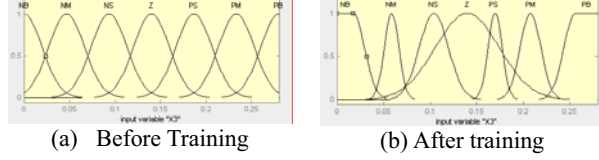


Figure 11 Membership function of X3 (position of trunk on x axis)

After training, the following rules has been obtained, for example

if $x_3(k)$ is NB and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.37x_3(k) + 0.05x_6(k) + 2.14$,
if $x_3(k)$ is NS and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.91x_3(k) + 0.06x_6(k) + 0.23$,
if $x_3(k)$ is Z and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.40x_3(k) + 0.03x_6(k) + 5.63$,
if $x_3(k)$ is PS and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.83x_3(k) + 0.09x_6(k) + 0.03$,
if $x_3(k)$ is PB and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.46x_3(k) + 0.02x_6(k) + 1.11$,
if $x_3(k)$ is NB and $x_6(k)$ is NB then $x_{zmp}(k+1) = 0.37x_3(k) + 0.05x_6(k) + 2.14$, and etc.

f $x_3(k)$ is NB and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.24x_3(k) + 0.05x_6(k) + 1.29$,
if $x_3(k)$ is NS and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.31x_3(k) + 0.11x_6(k) + 0.45$,
if $x_3(k)$ is Z and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.32x_3(k) + 0.07x_6(k) + 4.80$,
if $x_3(k)$ is PS and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.11x_3(k) + 0.12x_6(k) + 5.01$,
if $x_3(k)$ is PB and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.67x_3(k) + 0.09x_6(k) + 2.3$,
if $x_3(k)$ is NB and $x_6(k)$ is NB then $y_{zmp}(k+1) = 0.45x_3(k) + 0.23x_6(k) + 6.67$, and etc.

5 Concluding Remarks

We have studied and proposed a biped gait synthesizer based on neuro-fuzzy scheme. In this paper, it synthesizes trunk trajectories and varies ZMP rather than a fixed ZMP, while all other joint's trajectories are prescribed. The proposed gait synthesizer formulates initial dynamic balancing gaits from fuzzy rules and its membership function. The experimental results show that the neuro-fuzzy system significantly improves the stability of biped robot locomotion better than conventional fuzzy systems. Thus, the main contribution of this paper is a neuro-fuzzy synthesis of gait generated algorithm with optimal learning algorithms to provide a simple but general and systematic way of implementing dynamic bipedal walking. There is no need for dynamics formulation. The proposed control architecture provides another framework to analyze as attempt to understand how bipedal robots learn to walks. In the future, we will extend our scheme on a further complicated 14 DOFs biped robot.

Acknowledgements

The research described in this paper was made possible by the support of the Thailand Research

Fund (TRF). The first author is a recipient of the TRF Golden Jubilee PhD fellowship, completing his PhD dissertation at the Institute of Field Robotics (FIBO), King Mongkut's University of Technology Thonburi (KMUTT).

References

1. Todd, D. J., Walking Machines : An introduction to legged robot, New York, NY; Chapman and Hall, 1985
2. Raibert, M. H., Legged Robots That Balance, The MIT Press, Cambridge, Massachusetts, 1986
3. Kajita, S. and Tani K., Experimental Study of Biped Dynamic Walking, *In Proc. IEEE Int. Conf. on Robotics and Automation*, May, pp. 13-19, 1995.
4. Sano, A. and Furusho, J., Realization of Natural Dynamic Walking Using The Angular Momentum Information, *In Proc. IEEE Int. Conf. on Robotics and Automation*, May, pp. 1476-1481, 1990.
5. Benbrahim, H. and Franklin, J., Biped Dynamic Walking Using Reinforcement Learning, *Robotics and Autonomous System*. 22, pp. 283-302, 1997.
6. Miller, W. T., Latham, P. W. and Scalera, S. M, "Bipedal gait adaptation for walking with dynamic balance, *In Proc. 1991 Amer. Control Conf.*, Boston, MA, June 26-28, pp. 1603-1608, 1991.
7. Hu, J., Pratt, J. and Pratt, G., Adaptive Dynamic Control of a Bipedal Walking Robot with Radial Basis Function Neural Networks, *In Proc. IEEE Int. Conf. on intelligent Robots and Systems*, Victoria, BC, Canada, October, 1998.
8. Vokobratovic M, Borovac B, Surla D, Stokic D, Biped Locomotion: Dynamics, Stability, Control and Application, Springer-Verlag, 1990.
9. Li, Q., Takanishi, A. and Kato, I., Learning Control of Compensative Trunk Motion for biped Walking Robot based on ZMP Stability Criterion, *In Proc. IEEE/RSJ Int. Conf. on intelligent Robots and System*, pp.597-603, 1992.
10. Yamaguchi, J., Takanishi, A. and Kato, I., Development of a Biped Walking Robot Compensating for Three-Axis Moment by Trunk Motion, *In Proc. IEEE/RSJ Proc. Int. Conf. on intelligent Robots and System*, pp.561-566, 1993.
11. Li, Q., Takanishi, A. and Kato, I., Learning Control for biped Walking Robot with Trunk, *IEEE/RSJ Int. Conf. on intelligent Robots and System*, pp.1771-1777, 1992.
12. Featherstone, R. and Orin, D., Robot Dynamics: Equations and Algorithms, *In Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 826-834, 2000.
13. Park, J. H. and Rhee, Y. K., ZMP trajectory generation for reduced trunk motions of biped robots, *In IEEE Int. Conf. on Robotics and Automation*, pp. 2014-2021, 1998.
14. Shih, C.L., Gait synthesis for a robot, *Robotics* 15, pp. 599-607, 1997.
15. Zhou. C., Neuro-fuzzy gait synthesis with reinforcement learning for a biped walking robot, *Soft Computing* 4, pp. 238-250, 2000.
16. Takagi, T, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, *In Proc. IEEE Transaction on SMC*, pp. 116-132, 1985.
17. Babuska, R., Neuro-fuzzy methods for modeling and identification, *In Recent Advances in intelligent Paradigms and Application*, pp. 161-186, Springer-Verlag, 2002.
18. Park, Fuzzy-logic ZMP trajectory generation for reduced trunk motions of biped robots, *Fuzzy Sets and Systems*, Vol. 134, Issue 1, pp. 189-203, 2003.