

รายงานวิจัยฉบับสมบูรณ์ ประจำปีงบประมาณ 2546  
โครงการวิจัยย่อยลำดับที่ 8 เรื่อง  
การจัดสัญญาณสะท้อนทางเสียงเมื่อมีสถานการณ์ Double-Talk  
ในระบบการสัมมนาแบบสเตอริโอ  
(Stereophonic Acoustic Echo Cancellation in the Double-Talk Situation)

ผู้รับผิดชอบโครงการ      ดร.นิสาชล ตั้งเสงี่ยมวิสัย

วัตถุประสงค์ของโครงการ

- ศึกษา วิเคราะห์ และหาวิธีแก้ไขสถานการณ์ Double Talk สำหรับการกำจัดสัญญาณสะท้อนทางเสียงในระบบการสัมมนาแบบสเตอริโออย่างมีประสิทธิภาพ
- ศึกษาหาตัวตรวจวัด เมื่อเกิดสถานการณ์ Double Talk ที่มีประสิทธิภาพ
- นำไปใช้สร้างในอุปกรณ์แบบจำลอง

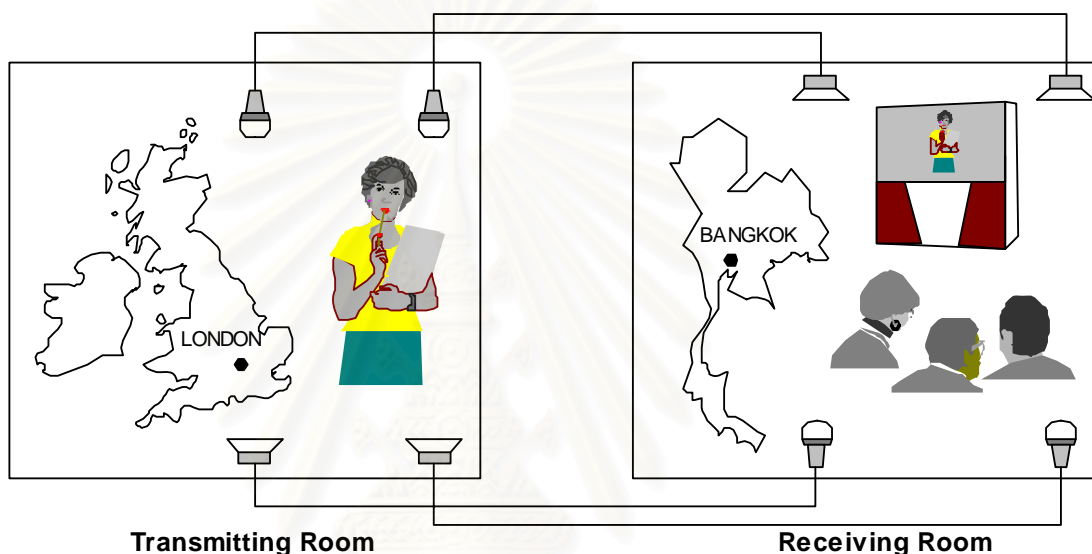
ขอบเขตของโครงการ

ในโครงการวิจัยนี้ จะเน้นไปที่การวิเคราะห์หากระบวนการวิธีที่ใช้ในการกำจัดสัญญาณสะท้อนทางเสียงของระบบการสัมมนาแบบสเตอริโอเมื่ออยู่ในสถานการณ์ Double Talk รวมถึงการหาตัวตรวจค่า (detector) เพื่อบ่งชี้ว่าเกิดสถานการณ์ Double Talk ขึ้น  
หลังจากนั้นจะเป็นการนำไปสร้างลงบนอุปกรณ์แบบจำลองเพื่อทดสอบกับสถานการณ์จริงของการสัมมนาแบบสเตอริโอ

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## 1. บทนำ

ในระบบการสัมมนาแบบสเตอริโอ (Stereo Conferencing System) ปัญหาการสะท้อนที่เกิดขึ้นจากการเชื่อมต่อทางเสียง (Acoustic Coupling) ระหว่างลำโพงกับไมโครโฟนภายในห้องรับ (Receiving Room) ดังรูปที่ 1 ทำให้การสนทนาที่มีประสิทธิภาพลดลง กล่าวคือ ผู้พูดเกิดการรำคาญเนื่องจากได้ยินเสียงตนเองวกกลับออกมาที่ลำโพงทางฝั่งห้องส่ง (Transmitting Room) ในทางตรงกันข้าม เมื่อคู่สนทนาทางฝั่งห้องรับเป็นฝ่ายพูด ปัญหาการสะท้อนก็เกิดขึ้นจากการเชื่อมต่อทางเสียงระหว่างลำโพงกับไมโครโฟนภายในห้องส่งในลักษณะเดียวกัน หรืออาจเรียกว่า วิธีสะท้อนทางเสียง (Acoustic Echo Path - AEP) ดังนั้นจึงมีความจำเป็นในการกำจัดสัญญาณสะท้อนทางเสียงในระบบการสัมมนาแบบสเตอริโอ (Stereophonic Acoustic Echo Cancellation - SAEC)



รูปที่ 1 แบบจำลองของระบบการสัมมนาแบบสเตอริโอ

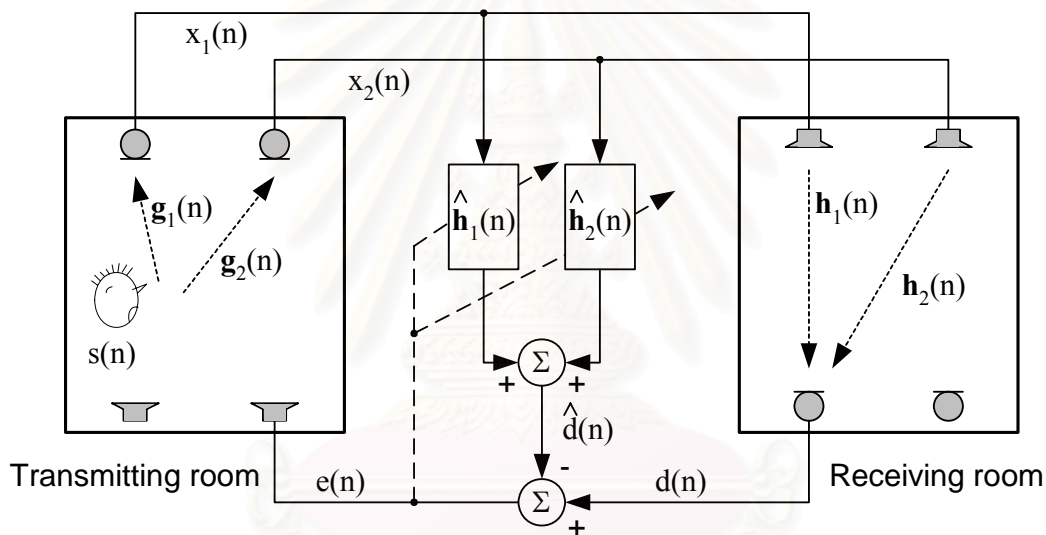
วิธีหนึ่งที่ใช้ในการกำจัดสัญญาณสะท้อนทางเสียงในระบบการสัมมนาแบบสเตอริโอ คือการใช้วงจรกรองแบบปรับตัว (Adaptive Filter) เพื่อหาแบบจำลองของระบบ (ซึ่งในที่นี้ หมายถึงแบบจำลองของการเชื่อมต่อทางเสียง) โดยมีระเบียบวิธีต่าง ๆ สำหรับใช้ในการควบคุมการทำงานของวงจรกรองแบบปรับตัว เช่น ระเบียบวิธี Normalised Least Mean Square (NLMS) ระเบียบวิธี Recursive Least Squares (RLS) เป็นต้น เมื่อวงจรกรองแบบปรับตัวสามารถจำลองการเชื่อมต่อทางเสียงได้อย่างถูกต้อง สัญญาณเสียงสะท้อนดังกล่าวก็จะสามารถถูกกำจัดได้ ไม่เกิดการรบกวนการสนทนาอีกต่อไป

อย่างไรก็ดี ถึงแม้ปัญหาสัญญาณเสียงสะท้อนสามารถถูกกำจัดไปได้โดยการนำวงจรกรองแบบปรับตัวมาใช้ร่วมในการส่งสัญญาณเสียงของระบบการสัมมนาแบบสเตอริโอก็ตาม อีกปัญหาหนึ่งก็ยังคงเป็นสิ่งที่ท้าทายการวิจัยในด้านนี้อยู่ เมื่อคู่สนทนาทั้งในห้องส่งและในห้องรับทำการพูดพร้อมกัน สถานการณ์ดังกล่าวเรียกว่า สถานการณ์ “Double Talk (DT)” ซึ่งจะรบกวนการปรับตัว ของวงจรกรองแบบปรับตัวไม่ว่าจะใช้ระเบียบวิธีใดๆ โดยมีผลทำให้เกิดการลู่ออกของระบบ SAEC จากสถานะอยู่ตัว (steady state) และนำไปสู่ความไม่มีเสถียรภาพของระบบได้ ถึงแม้ว่าจะมีเทคนิคต่าง ๆ ที่เสนอการจัดการกับสถานการณ์ DT นี้ [1-5] เพื่อป้องกันการลู่ออกของระบบ SAEC โดยที่เทคนิคส่วนมากเสนอให้ระบบ SAEC หยุดการทำงานของวงจรกรองแบบปรับตัวในช่วงเวลาที่มีสถานการณ์ DT เกิดขึ้น หากแต่ว่าการกระทำดังกล่าวมีข้อเสีย คือ ถ้าการเชื่อมต่อทางเสียงระหว่างไมโครโฟน

และถ้าโพงเกิดการเปลี่ยนแปลงในช่วงสถานการณ์ DT ถ้าระบบ SAEC ถูกหยุดการทำงานในช่วงดังกล่าว วงจรกรองแบบปรับตัวจะไม่สามารถปรับตัวเข้าหาคำตอบของการเชื่อมต่อทางเสียงได้ทันการเปลี่ยนแปลง ส่งผลให้ประสิทธิภาพในการกำจัดสัญญาณเสียงสะท้อนของระบบ SAEC ดังกล่าวไม่ดีนัก ดังนั้นตัวตรวจวัดสถานการณ์ DT (DT detector -- DTD) ควรได้รับการพัฒนาขึ้นให้มีประสิทธิภาพในการบ่งชี้เมื่อเกิดสถานการณ์ DT และระบบ SAEC ควรได้รับการปรับปรุงให้วงจรกรองแบบปรับตัวทำงานตลอดช่วงเวลาที่เกิดสถานการณ์ DT เพื่อให้มีประสิทธิภาพในการกำจัดสัญญาณเสียงสะท้อนได้มากกว่าการหยุดการทำงานของวงจรกรองแบบปรับตัว และเพื่อป้องกันความไม่มีเสถียรภาพของระบบ SAEC จากสถานการณ์ DT

## 2. ผลการดำเนินการ

จากเฟสที่หนึ่งของโครงการ ระบบ SAEC สำหรับการสัมมนาแบบสเตอริโอ ดังแสดงในรูปที่ 2 ได้ถูกทำการศึกษา วิเคราะห์ เพื่อหาวิธีการกำจัดสัญญาณเสียงสะท้อน โดยเฉพาะอย่างยิ่งในกรณีที่เกิดสถานการณ์ DT



รูปที่ 2 ระบบการกำจัดสัญญาณเสียงสะท้อนในระบบการสัมมนาแบบสเตอริโอ

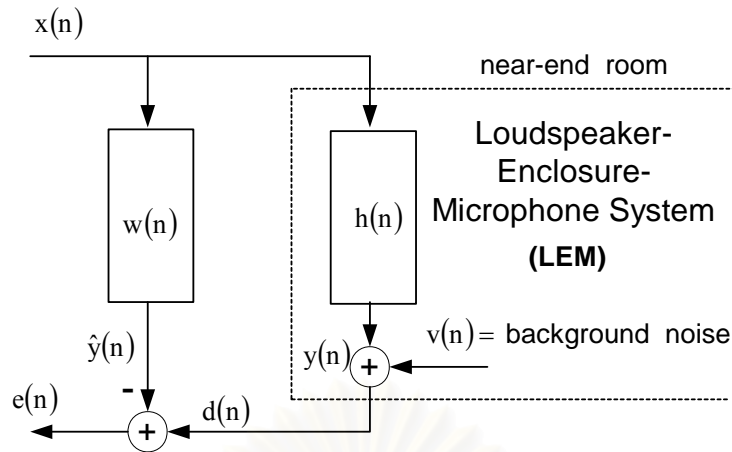
จากรูปที่ 2 จะเห็นว่าระบบการกำจัดสัญญาณเสียงสะท้อนแบบสองช่องสัญญาณสื่อสารนี้ถูกปรับปรุงมาจากระบบการกำจัดสัญญาณเสียงสะท้อนที่มีเพียงหนึ่งช่องสัญญาณสื่อสาร (Acoustic Echo Cancellation -- AEC) ได้โดยตรง ดังแสดงได้ในรูปที่ 3 สัญญาณความผิดพลาดของการประมาณ  $e(n)$  หาได้จากความแตกต่างระหว่างสัญญาณออกของวงจรกรองแบบปรับตัว  $\hat{y}(n)$  กับสัญญาณไมโครโฟน  $d(n)$

$$e(n) = d(n) - \hat{y}(n) \quad (1)$$

เมื่อค่าสัมประสิทธิ์ของวงจรกรองแบบปรับตัว จะถูกปรับปรุงทุกๆ เวลา  $n$  ดังต่อไปนี้

$$\underline{w}(n+1) = \underline{w}(n) + \frac{\tilde{\mu} \underline{x}(n) e(n)}{\mathcal{E} + \|\underline{x}(n)\|^2} \quad (2)$$

โดยที่  $\mathcal{E}$  คือ ค่าคงที่น้อยๆ เพื่อป้องกันการลู่ออกของวงจรกรองแบบปรับตัวในกรณีที่สัญญาณเข้ามีขนาดเล็กๆ และค่าช่วงก้าว (step-size)  $\tilde{\mu} \in (0,2)$  จะอยู่ภายใต้เงื่อนไขที่จำเป็นและพอเพียงสำหรับการลู่เข้าของระเบียบวิธี Normalised Mean Square (NLMS) [7]



รูปที่ 3 ระบบการกำจัดสัญญาณเสียงสะท้อนในระบบห้องสัญญาณเดียว

ดังนั้น ในเฟสที่สองของโครงการฯนี้จึงได้ทำการพัฒนาตัวตรวจวัด DTD เพื่อใช้ในระบบ SAEC และการจำลองระบบ SAEC เมื่อมีตัวตรวจวัดสถานการณ์ DT ลงบน DSK board เพื่อทดสอบกับสถานการณ์จริง (real-time) ของการสัมมนาแบบสตรีโอ โดยสังเกตผลของการจำลองการเชื่อมต่อทางเสียงทั้งในกรณีที่เกิดและไม่เกิดสถานการณ์ DT

## 2.1 ตัวตรวจวัด DTD

จากการศึกษาตัวตรวจวัด DTD ชนิดต่างๆ ที่ได้มีการนำเสนอใน [2,6] ในโครงการฯนี้จึงได้ปรับปรุงและนำเสนอตัวตรวจวัด DTD ขึ้นใหม่ 2 ตัว ซึ่งมีหลักการทำงานดังต่อไปนี้

ตัวตรวจวัด DTD ตัวที่หนึ่งจะใช้เพื่อแบ่งแยก (discriminate) สถานการณ์ DT ออกจากสถานการณ์ที่เกิดการเปลี่ยนแปลงของช่องเชื่อมต่อทางเสียงอย่างฉับพลัน (Abrupt Change in the Acoustic Echo Path -- ACEP) โดยอาศัยค่าสหสัมพันธ์ไขว้ระหว่างค่าเฉลี่ยของเวกเตอร์เกรเดียนท์ ของระเบียบวิธี NLMS ที่เวลา n และ n-1

$$g_1(n) = \bar{\nabla}(n) \cdot \bar{\nabla}(n-1) \quad (3)$$

สำหรับตัวตรวจวัด DTD ตัวที่สองจะใช้เพื่อแยกสถานการณ์ DT ออกจากสถานะอยู่ตัวของระบบหรือเรียกว่าช่วง Steady State (SS) โดยขึ้นอยู่กับ ค่ากำลังสองของค่าเฉลี่ยของเวกเตอร์เกรเดียนท์ที่เวลา n นั้น ๆ หรือเรียกว่าค่าอัตสหสัมพันธ์ (autocorrelation) ของค่าเฉลี่ยของเวกเตอร์เกรเดียนท์

$$g_2(n) = \bar{\nabla}^2(n) \quad (4)$$

การตัดสินใจของตัวตรวจวัด DTD ทั้งสองนี้ สามารถแสดงได้ดัง Flow Chart ในรูปที่ 4 เมื่อ  $\theta_1$  และ  $\theta_2$  เป็นค่าเริ่มเปลี่ยน (threshold) และ  $\mu_{max}$  เป็นค่าสูงสุดของค่าช่วงก้าวที่ใช้ในระเบียบวิธี NLMS โดยแบ่งการตัดสินใจออกเป็น 3 กรณีของสมมติฐานในรูปแบบความสัมพันธ์ระหว่าง  $g_1(n)$  และ  $g_2(n)$  ดังนี้

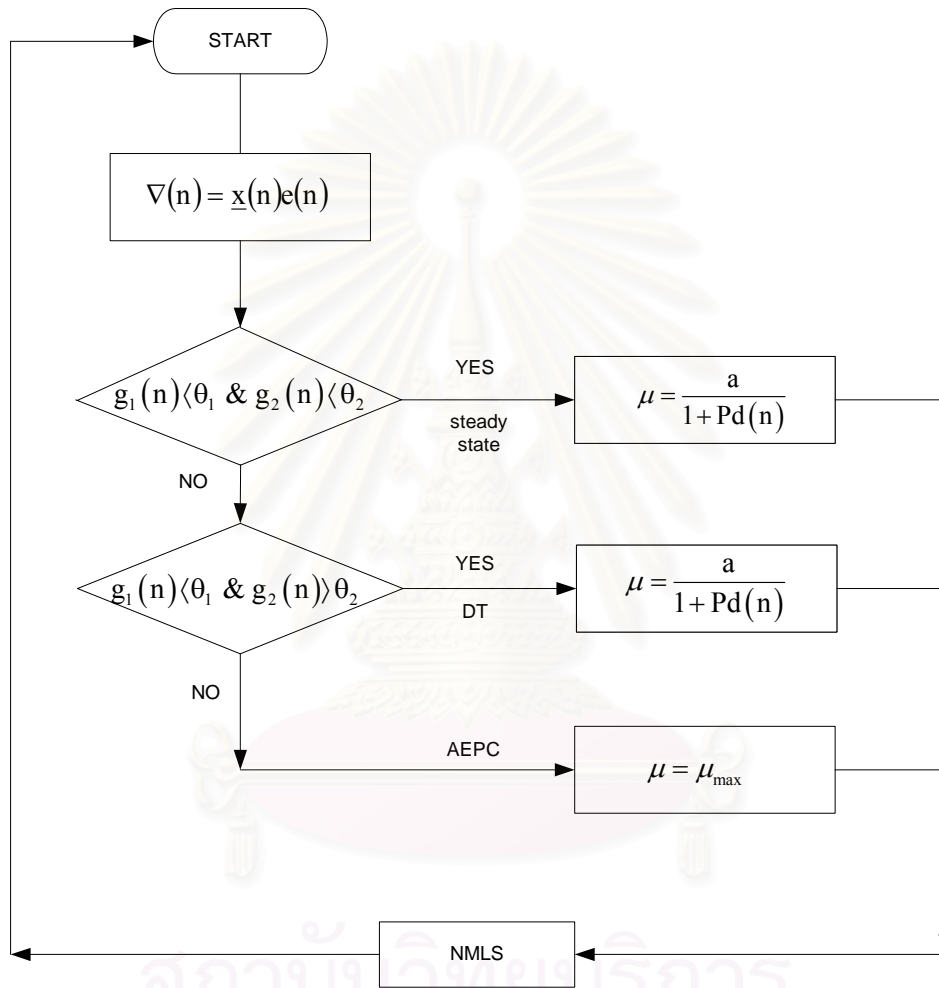
$$\begin{aligned} H_0 : \text{Steady State,} & \quad \text{if } g_1(n) < \theta_1 \text{ and } g_2(n) < \theta_2 \\ H_1 : \text{Double Talk,} & \quad \text{if } g_1(n) < \theta_1 \text{ and } g_2(n) > \theta_2 \\ H_2 : \text{AEPC,} & \quad \text{if } g_1(n) > \theta_1 \text{ and } g_2(n) > \theta_2 \end{aligned}$$

- เมื่อเกิดสถานการณ์ AEPC การปรับตัวของวงจรกรองแบบปรับตัวควรมีค่ามาก เพื่อลู่เข้าสู่ค่าตอบโดยเร็วที่สุด ดังนั้นค่าช่วงก้าวจึงถูกเลือกเป็น  $\mu = \mu_{max}$

- เมื่อเกิดสถานการณ์ DT วงจรกรองแบบปรับตัวควรทำการปรับตัวอย่างต่อเนื่อง แต่ด้วยอัตราการปรับตัวที่ช้า เพื่อป้องกันการลู่ออกของระบบ โดยมีค่าช่วงก้าว  $\mu = \frac{a}{1 + P_d(n)}$  ที่ a เป็นค่าคงที่ใดๆ ค่าช่วงก้าวถูกปรับเป็นฟังก์ชันของค่าพลังงานของสัญญาณไมโครโฟน d(n) เมื่อ

$$P_d(n) = \lambda_d P_d(n-1) + (1 - \lambda_d) d^2(n) \quad (5)$$

และ  $\lambda_d$  เป็นค่า forgetting factor



รูปที่ 4 การตัดสินใจของ DTD ที่นำเสนอ

จากนั้นจึงทำการทดลองศึกษาตัวตรวจวัด DTD ที่นำเสนอในโครงการวิจัยนี้เมื่อใช้ในระบบ SAEC โดยเฉพาะเมื่อเกิดสถานการณ์ DT โดยจำลองและทดสอบระบบบนคอมพิวเตอร์ด้วยโปรแกรม MATLAB สำหรับสมรรถนะของวงจรกรองแบบปรับตัวเมื่อใช้ระเบียบวิธี NLMS จะถูกพิจารณาจากค่า Weight Error Vector Norm (WEVN) หาได้จาก

$$WEVN(n) = 10 * \log_{10} \frac{\|\underline{h} - \underline{w}(n)\|^2}{\|\underline{h}\|^2} \quad (6)$$

ในการจำลองแบบการทำงานของวงจรกรองปรับตัวในระบบ SAEC นั้น กำหนดให้

- สัญญาณเข้ามีทั้งกรณีที่เป็นที่เป็นสัญญาณเสียงพูด 2 สัญญาณ (สเตริโอ) และสัญญาณรบกวนขาว (White Gaussian Noise -- WGN)

- สัญญาณเสียงพูดอีกชุดหนึ่งเป็นสัญญาณ DT

ทำการทดลองโดยใช้ระเบียบวิธี NLMS ที่มีความยาวของวงจรรองเป็น  $L = 256$  และมีค่าช่วงก้าวเป็น  $\mu = 0.3$  ผลการทดลองดังต่อไปนี้

รูปที่ 5 แสดงสัญญาณเข้าที่เป็นเสียงพูด  $x(n)$  เมื่อไม่มีสัญญาณรบกวนเบื้องหลัง

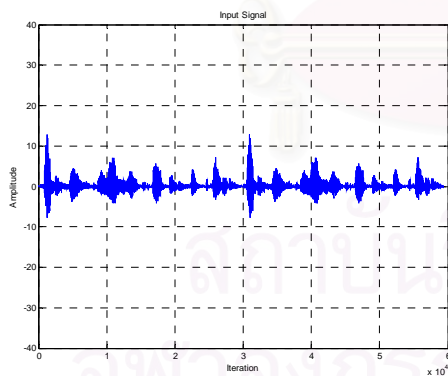
รูปที่ 6 แสดงสัญญาณเสียงสะท้อนหรือสัญญาณไมโครโฟน  $d(n)$  เมื่อระบบมีสัญญาณรบกวนเบื้องหลัง (background noise) ที่มีค่า  $SNR = 30$  dB

รูปที่ 7 แสดงสัญญาณ DT ที่ถูกสร้างให้เกิดขึ้นในช่วง 30,000-35,000 รอบการจำลองแบบ ซึ่งกำหนดให้ค่าความแปรปรวน (Variance) ของสัญญาณ DT มีค่าเท่ากับสัญญาณเข้า

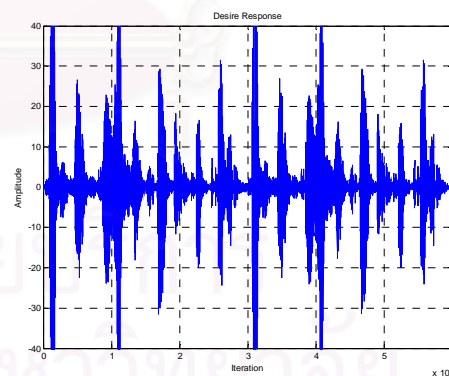
รูปที่ 8 แสดงค่าสัญญาณความผิดพลาด (error signal) เมื่อเกิดสถานการณ์ DT เปรียบเทียบกับรอบการจำลอง สัญญาณความผิดพลาดนี้เกิดจากการจำลองของวงจรรองแบบปรับตัวในระบบ SAEC โดยที่ในช่วงแรกวงจรรองจะปรับตัวให้มีความใกล้เคียงกับการเชื่อมต่อทางเสียงหรือ AEP จึงทำให้ค่าสัญญาณความผิดพลาดมีขนาดลดน้อยลงเรื่อยๆ แต่เมื่อเกิดสถานการณ์ DT ในช่วง 30,000-35,000 รอบการจำลองแบบ วงจรรองจะไม่สามารถปรับค่าสัมประสิทธิ์ให้ใกล้เคียงกับ AEP ได้ จึงทำให้ค่าสัญญาณความผิดพลาดมีขนาดใหญ่ขึ้น

รูปที่ 9 แสดงค่า Weight Error Vector Norm (WEVN) เมื่อไม่เกิดสถานการณ์ DT ในระบบ เปรียบเทียบกับรอบการจำลอง จะสังเกตได้ว่าเมื่อวงจรรองปรับตัวมีความใกล้เคียงกับ AEP มากขึ้น ค่า WEVN จะค่อยๆ เข้าสู่ภาวะ SS

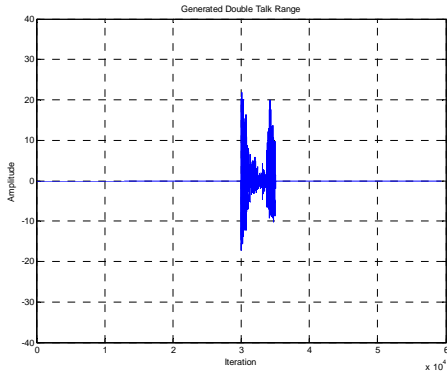
รูปที่ 10 แสดงค่า WEVN เมื่อเกิด Double Talk เปรียบเทียบกับรอบการจำลอง จะเห็นได้ว่าเมื่อระบบถูกรบกวนด้วยสัญญาณ DT จะทำให้ค่า WEVN เพิ่มมากขึ้นจากสถานะอยู่ตัว แล้วจะลู่ลงสู่สถานะอยู่ตัวตามเดิม



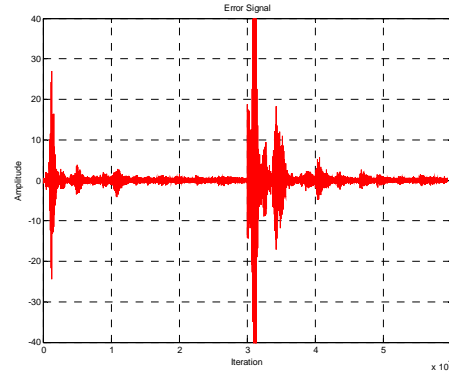
รูปที่ 5 แสดงสัญญาณเข้าเมื่อไม่มีสัญญาณรบกวนภายในระบบ



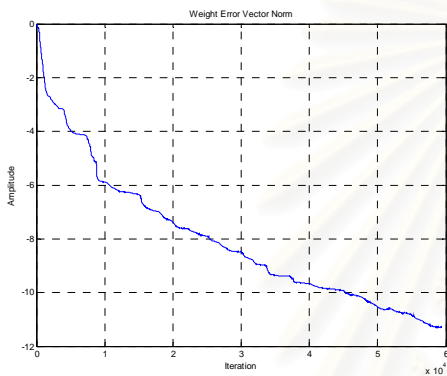
รูปที่ 6 แสดงสัญญาณเสียงสะท้อนเมื่อมีสัญญาณรบกวนภายในระบบ



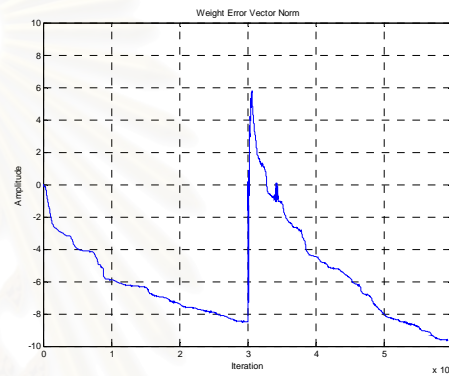
รูปที่ 7 แสดงสัญญาณ DT



รูปที่ 8 แสดงค่า error signal เมื่อเกิดสถานการณ์ DT

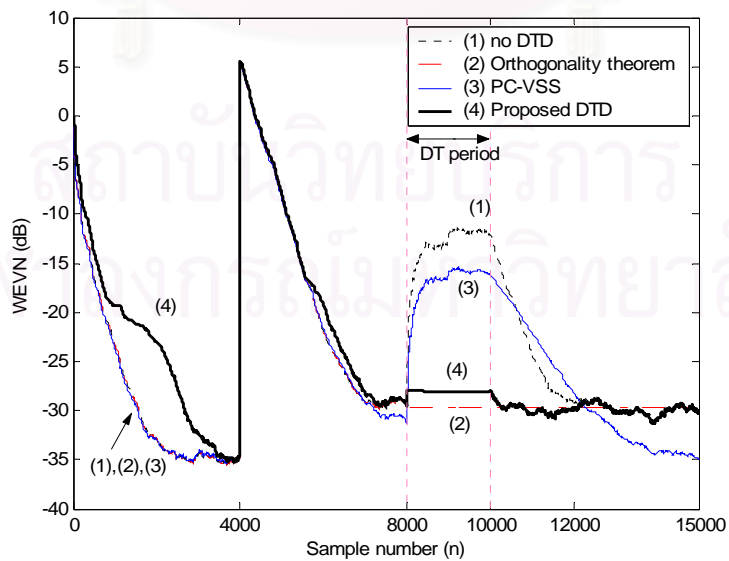


รูปที่ 9 แสดงค่า WEVN เมื่อยังไม่เกิดสถานการณ์ DT



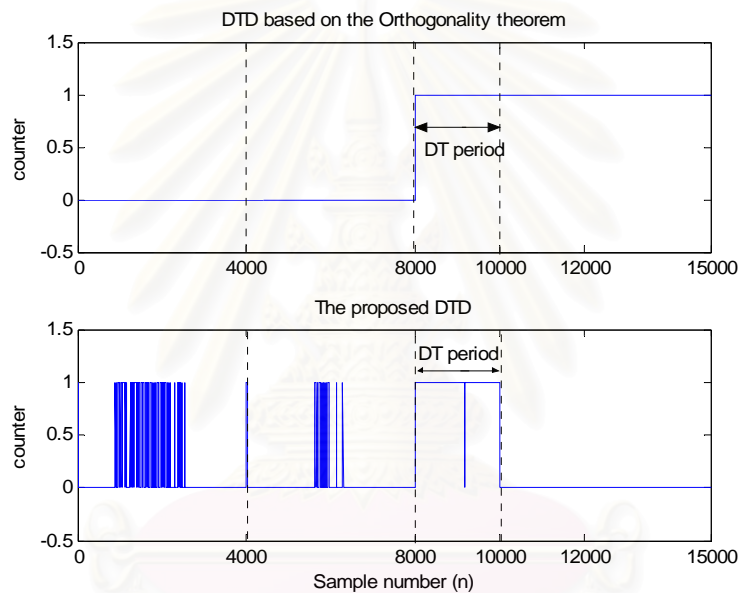
รูปที่ 10 แสดงค่า WEVN เมื่อเกิดสถานการณ์ DT

นอกจากนี้ ยังได้ทำการทดลองศึกษาเปรียบเทียบประสิทธิภาพตัวตรวจวัด DTD ที่นำเสนอในโครงการวิจัยนี้กับตัวตรวจวัดอื่นๆ ที่มีการนำเสนอใช้งานแล้ว คือ DTD ที่อาศัย Orthogonality Theorem ใน [2] และ DTD ที่ใช้ Projection Correlation และ Variable Step-Size ใน [7] หรือเรียกว่าเทคนิค PC-VSS



รูปที่ 11 การเปรียบเทียบประสิทธิภาพของระบบ SAEC ที่ใช้ระเบียบวิธี NLMS เมื่อไม่มีและมี DTD ชนิดต่างๆ ผ่านทางค่า WEVN

จากรูปที่ 11 จะเห็นว่าระบบ SAEC ลู่ออกจากสภาวะอยู่ตัวเมื่อไม่มี DTD ดังกราฟเส้นที่ 1 แต่จะถูกรบกวนจากสภาวะอยู่ตัวน้อยลงเมื่อมีการใช้ DTD ชนิดต่างๆ โดยเทคนิคที่นำเสนอในโครงการฯ นี้เป็นดังกราฟเส้นที่ 4 มีประสิทธิภาพใกล้เคียงกับ DTD ที่อาศัย Orthogonality Theorem ใน [2] และดีกว่าเทคนิค PC-VSS ใน [7] นอกจากนี้ การตัดสินใจของ DTD ระหว่างเทคนิคที่นำเสนอและเทคนิคใน [2] สามารถแสดงได้ดังรูปที่ 12 เมื่อให้ตัวนับ (counter) เป็น '1' แสดงถึงการเกิดสถานการณ์ DT และตัวนับ (counter) เป็น '0' แสดงถึงการไม่เกิดสถานการณ์ DT จากรูปที่ 12 จะเห็นได้ว่าความผิดพลาดในการตัดสินใจสถานการณ์ DT ของวิธีที่นำเสนอในช่วง Transient มีอยู่มากกว่าเทคนิคใน [2] หากแต่เมื่อพิจารณาการตัดสินใจภายหลังช่วง DT ประสิทธิภาพของเทคนิคที่นำเสนอจะดีกว่าเทคนิคใน [2] อยู่มาก ซึ่งหาก DTD ใน [2] ตัดสินว่ายังอยู่ในสถานการณ์ DT และทำการยับยั้งการปรับปรุงสัมประสิทธิ์ของวงจรกรองแบบปรับตัวในช่วงดังกล่าว ระบบ SAEC จะไม่สามารถจำลอง AEP ที่เปลี่ยนแปลงไปจากก่อนเกิดสถานการณ์ DT ได้ (หมายเหตุ DTD ที่ใช้เทคนิค PC-VSS ใน [7] ไม่มีการตัดสินใจสถานการณ์ DT ดังกล่าว)



รูปที่ 12 การตัดสินใจสถานการณ์ DT ของ DTD

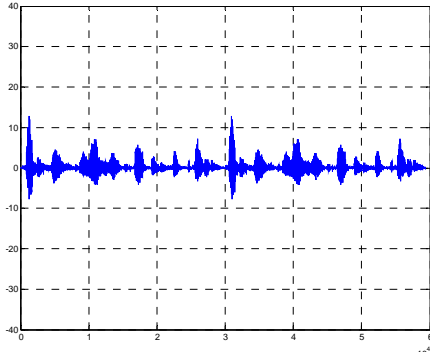
## 2.2 การจำลองระบบ SAEC ลงบน DSK board

สำหรับรายละเอียดต่างๆ ในส่วนนี้ มีดังต่อไปนี้

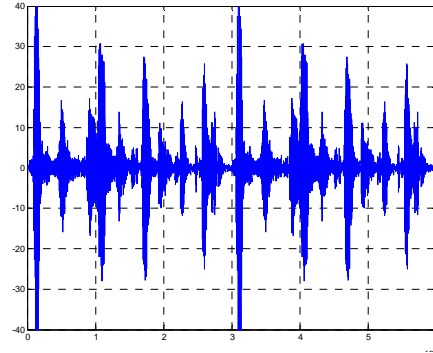
- ศึกษารายละเอียดการใช้งานบอร์ด DSK C6711 จากคู่มือและ Tutorial ของบริษัท TI ที่มีให้
- จำลองแบบการทำงานของวงจรกรองปรับตัวในระบบ SAEC โดยเฉพาะเมื่อเกิดสถานการณ์ DT โดยจำลองแบบด้วยคอมพิวเตอร์ด้วยโปรแกรมภาษา C
- ประยุกต์โปรแกรมภาษา C เพื่อนำไปใช้งานกับบอร์ด DSK C6711

ผลการทดลองการจำลองระบบ SAEC ในสถานการณ์ Double Talk ด้วยโปรแกรมภาษา C ที่พัฒนาขึ้น เมื่อสัญญาณต่างๆ ที่เกี่ยวข้องในระบบเป็นดังเช่นในการจำลองแบบด้วยโปรแกรม MATLAB เป็นดังนี้

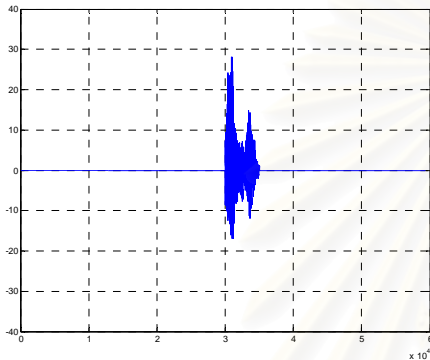




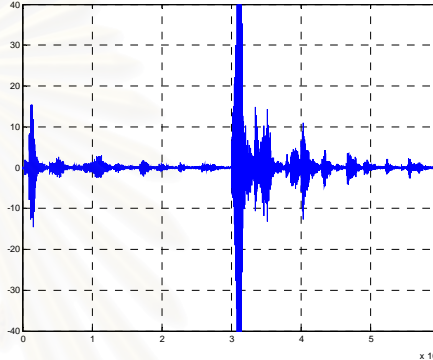
รูปที่ 13 แสดงสัญญาณขาเข้า



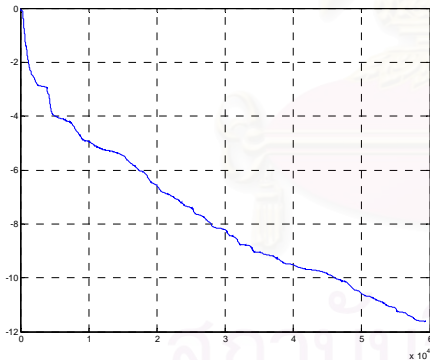
รูปที่ 14 แสดงสัญญาณเสียงสะท้อน



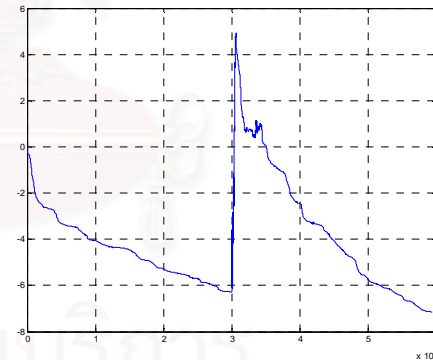
รูปที่ 15 แสดงสัญญาณ Double Talk



รูปที่ 16 แสดงสัญญาณ Error Signal เมื่อมี  
สถานการณ์ DT



รูปที่ 17 แสดงค่า WEVN เมื่อยังไม่เกิดสถานการณ์ DT



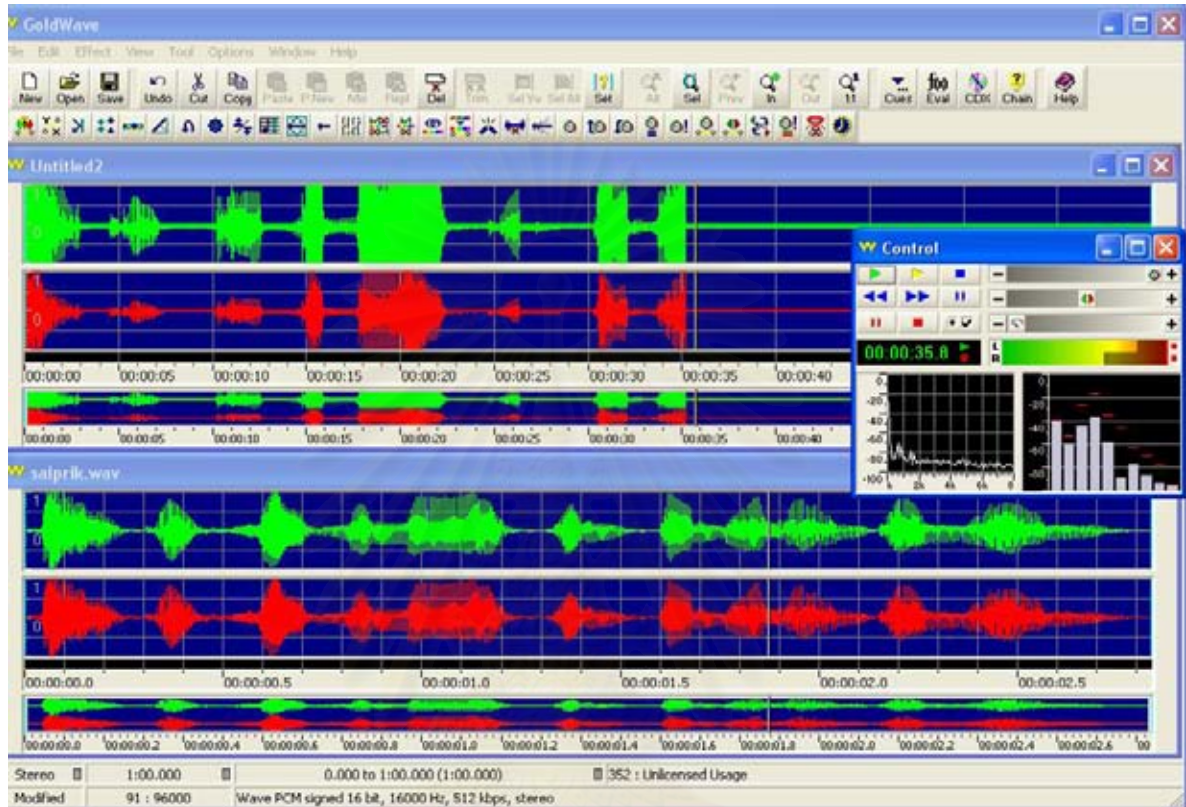
รูปที่ 18 แสดงค่า WEVN เมื่อเกิดสถานการณ์ DT

### 2.3 ผลการทดสอบระบบ SAEC บน DSK board

ในส่วนนี้ จะนำโปรแกรมภาษาซีสำหรับระบบ SAEC ที่ได้ไปประยุกต์ใช้กับบอร์ด DSK C6711 โดยได้ทำการศึกษาการใช้งานบอร์ด DSK C6711 จากหนังสือ “การประมวลผลสัญญาณดิจิทัล โดย TMS320 C6000” โดย รองศาสตราจารย์ กฤษดา วิสาริรานนท์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย พฤษภาคม 2545 และได้ทำการสรุปเรียบเรียงใหม่ดังแสดงในภาคผนวก ก นอกจากนี้ ก็มีแบบฝึกหัดให้ทำการทดลองด้วยโปรแกรม Code Composer Studio ในตอนท้ายของหนังสือเล่มนี้ด้วย ทำให้สามารถใช้งานบอร์ด DSK C6711 อย่างเข้าใจมากขึ้น

การส่งสัญญาณผ่านบอร์ด DSK C6711 โดยใช้โปรแกรม Goldwave ทำการพลอตกราฟดังแสดงในรูปที่ 19 โดยสัญญาณสเตอริโอขาเข้า (รูปล่าง) เป็นสัญญาณเสียงพูด จะถูกประมวลผลผ่านบอร์ด DSK c6711 ได้

สัญญาณขาออก แล้วนำสัญญาณขาออกส่งไปยัง sound card ในคอมพิวเตอร์เพื่อนำไปพลอตกราฟโดยใช้โปรแกรม Goldwave จะได้สัญญาณสเตริโอขาออก (รูปบน) จะเห็นว่าสัญญาณขาออกที่มีมีขนาดใหญ่มากกว่าสัญญาณขาเข้า เนื่องจาก sound card ของคอมพิวเตอร์มีวงจรขยายสัญญาณ (amplifying circuit) และเกิดการหน่วงเวลา (delay time) กับสัญญาณรบกวนเบื้องหลัง (background noise) ด้วย



รูปที่ 19 สัญญาณเข้าบอร์ดแบบสเตริโอ(รูปล่าง)และสัญญาณออกจากบอร์ด(รูปบน)

ขณะนี้สามารถเขียนโปรแกรมลงบอร์ด DSK C6711 ให้รับสัญญาณเข้าจาก port input แล้วนำไปประมวลผลด้วยระเบียบวิธี LMS จากนั้นนำข้อมูลที่ต้องการออกทาง port output ได้แล้ว (รายละเอียดดังแสดงในภาคผนวก ข) แต่โปรแกรมยังไม่สมบูรณ์ จะต้องทำการพัฒนาต่อไปให้เป็นระเบียบวิธี NLMS ที่สามารถตรวจจับและแก้ไขความผิดพลาดได้ในสถานการณ์ DT

ปัญหาที่เกิดขึ้นเนื่องจากการไม่สามารถประยุกต์โปรแกรมภาษาซีที่เขียนขึ้นบนบอร์ด DSK C6711 เนื่องจาก

- หน่วยความจำของบอร์ด DSK C6711 มีจำนวนน้อย ซึ่งอัลกอริทึมที่เขียนขึ้นเรียกใช้หน่วยความจำมากไป จะต้องทำการปรับปรุงโปรแกรมภาษาซีที่เขียนไว้
- ไม่สามารถเก็บค่าผลการทดลองที่ต้องการได้ เนื่องจากต้องทำการเก็บลงขา register ของบอร์ด DSK C6711 ดังนั้น ต้องศึกษาเพิ่มเติมเกี่ยวกับการเก็บข้อมูลบนบอร์ด
- ยังไม่สามารถทำการทดลองได้อย่างมีประสิทธิภาพ เนื่องจากยังขาดความรู้หลายอย่างเช่น การ plot graph, การ include file, คำสั่งต่างๆ ในการ interface กับบอร์ด โดยต้องศึกษาการใช้ฟังก์ชันต่างๆ ของบอร์ด DSK C6711 ต่อไป

### 3. สรุปผลการทดลองและข้อเสนอแนะ

ถึงแม้ว่าจะสามารถทำการกำจัดสัญญาณเสียงสะท้อนในระบบ SAEC เมื่อมีสถานการณ์ DT ได้แล้วโดยการจำลองระบบบนคอมพิวเตอร์ผ่านโปรแกรม MATLAB และมีการนำเสนอ DTD ที่มีประสิทธิภาพในการตรวจวัดและแยกแยะสถานการณ์ DT ได้ หากแต่ว่าการจำลองระบบดังกล่าวลงบนบอร์ด DSK 6711 นั้นยังไม่มี ความคืบหน้าเท่าที่ควร เนื่องจากผู้วิจัยและผู้ช่วยวิจัยขาดประสบการณ์การใช้งานบอร์ด DSK 6711 จึงต้องศึกษาหาความรู้เพิ่มเติมให้มากขึ้นอีก

#### ส่วนที่จะดำเนินการต่อไป

- ศึกษาการใช้งานบอร์ด DSK C6711 ให้ละเอียดมากขึ้น
- พัฒนาโปรแกรมสำหรับระเบียบวิธี LMS ที่สามารถทำงานบนบอร์ดได้แล้วให้เป็นระเบียบวิธี NLMS ที่สามารถตรวจจับและแก้ไขความผิดพลาดได้ในสถานการณ์ DT
- ทำการปรับปรุงโปรแกรมให้ทดสอบกับสถานการณ์จริง (real-time) ได้

### 4. ผลิตผลและหรือความสัมฤทธิ์ผลของงานที่ได้ดำเนินการไปแล้ว

- 4.1 บทความทางวิชาการเรื่อง “Double-Talk Detection for Stereophonic Acoustic Echo Cancellation in a Time-Varying Environment” ในการประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 25 (EECON-25) วันที่ 21-22 พฤศจิกายน 2545, หน้า 5 – 9. (โปรดดูภาคผนวก ก)
- 4.2 นิติตยสารศึกษาระดับมหัศจรรย์ 1 คน (นาย ฌตพร อธิธิโสภณกุล) เมื่อมีนาคม พ.ศ. 2547

### 5. เอกสารอ้างอิง

1. S. Minami and T. Kawasaki, “A Double Talk Detection Method for an Echo Canceller”, IEEE International Conference on Communications (ICC), pp. 1492-1497, 1985.
2. H. Ye and B. Wu, “A New Double-Talk Detection Algorithm Based on the Orthogonality Theorem”, IEEE Transactions on Communications, vol. 39, no. 11, pp. 1542-1545, 1991.
3. C. Carlemalm, F. Gustaffson and B. Wahlberg, “On the problem of detection and discrimination of double talk and change in the echo path”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’96, Atlanta, USA.), vol. 5, pp. 2742-2745, 1996.
4. C. Carlemalm and A. Logothetis, “On detection of double talk and changes in the echo path using a markov modelulated channel model”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’97, Munich, Germany), vol. 5, pp. 3869-3872, 1997.
5. S. J. Pushparajah and J. A. Chambers, “A smarter method for Acoustic Echo Cancellation in the presence of Double Talk”, International Workshop on Acoustic Echo and Noise Control (IWAENC’99, Pennsylvania, USA.), pp. 184-186, 1999.
6. T. Creasy and T. Aboulnasr, “A Projection-Correlation Algorithm for Acoustic Echo Cancellation in the presence of Double Talk”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’2000, Istanbul, Turkey), vol. 1, pp. 436 – 439, 2000.
7. S. Haykin, “Adaptive Filter Theory”, Prentice Hall, 1996.

## 6. ภาคผนวก

### 6.1 ภาคผนวก ก

#### การเริ่มใช้งาน CCS

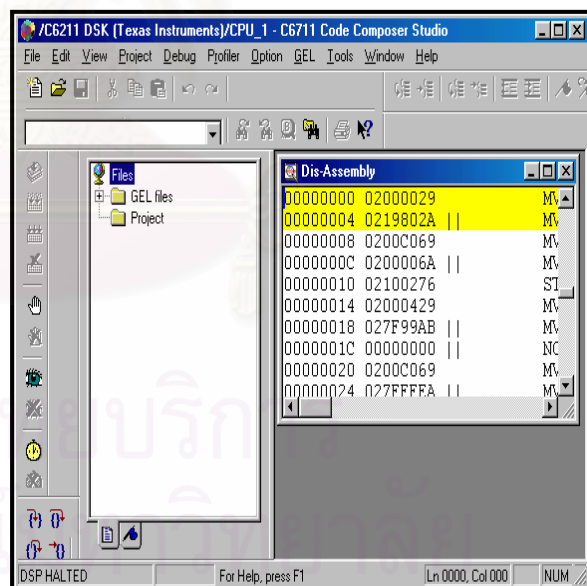
ต่อบอร์ด C6711 DSK Board เข้ากับ Printer Port (LPT1) ของคอมพิวเตอร์ และจ่ายไฟเลี้ยง(power supply) ให้กับ C6711 DSK Board

- เลือก Start → Programs → C6211-C6711 DSK Development tools  
→ CCS Studio DSK ' C6000 1.23



กรณีที่โปรแกรมไม่สามารถติดต่อกับบอร์ดได้จะมีบลิ๊คแสดงบนหน้าจอเพื่อแจ้งให้ทราบ ให้ทำการคลิกที่ Abort หลังจากนั้นให้ทำการตรวจสอบการต่อสายที่ Port Printer กับบอร์ด DSK หลังจากนั้นให้ทำการปิดแหล่งจ่ายไฟสักรุ่นแล้วเริ่มจ่ายไฟให้กับบอร์ด DSK อีกครั้ง แล้วทำการเรียกโปรแกรมเหมือนเดิมอีกครั้ง

กรณีที่โปรแกรมสามารถติดต่อกับบอร์ดได้ ก็จะมีรันโปรแกรม CCS โดยมีรูปแบบ Windows แสดงดังรูป

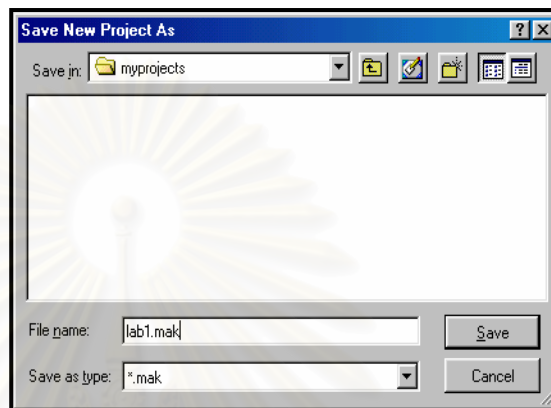


และในขณะที่เดียวกันให้สังเกตที่ LED 3 ดวงบนบอร์ด DSK จะกระพริบเป็นลักษณะของไฟวิ่งอยู่สักรุ่นหนึ่ง แล้วจึงจะดับ ซึ่งแสดงว่าบอร์ดพร้อมสำหรับใช้งานแล้ว และขณะเดียวกันโปรแกรมก็พร้อมที่จะใช้งานในขั้นต่อไป แต่ถ้กรณีที่ LED อยู่ในสภาวะค้างแสดงว่าโปรแกรมเกิดการผิดพลาดให้ทำการปิดแหล่งจ่ายไฟสักรุ่นจึงเริ่มจ่ายไฟให้กับบอร์ดอีกครั้งหนึ่ง จากนั้นจึงเริ่ม Start Program ใหม่ ถ้าไม่มีข้อผิดพลาดก็จะสามารถใช้งานในขั้นต่อไป

## การสร้างโปรเจกงาน (Creating a New Project)

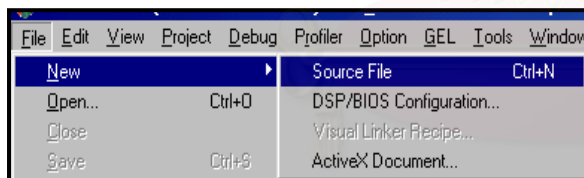
ในการใช้งาน CCS ขั้นแรกต้องสร้าง Project (\*.mak)

1. เลือกเมนู **Project** → **New** : จะปรากฏ window “**Save New Project As**” ให้เลือก directory (C:\ti\myproject\..) ที่ต้องการเก็บไฟล์ จากนั้นให้พิมพ์ชื่อ Lab1.mak (\*.mak) ของ Project ในช่อง “**File name**” และ click “**Save**” ก็จะได้ Project ชื่อ Lab1.mak

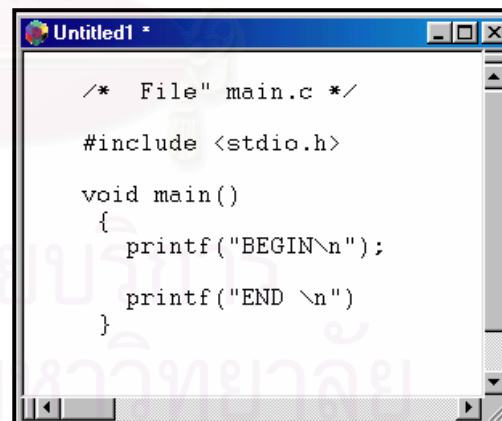


2. สร้างไฟล์ “main.c” (\*.c) ด้วยภาษา C ตัวอย่างง่ายๆ โปรแกรมนี้จะประกอบไปด้วย Include Header files , Function main ซึ่งเป็นฟังก์ชันหลักในการทำงาน ขั้นตอนการสร้างมีดังนี้

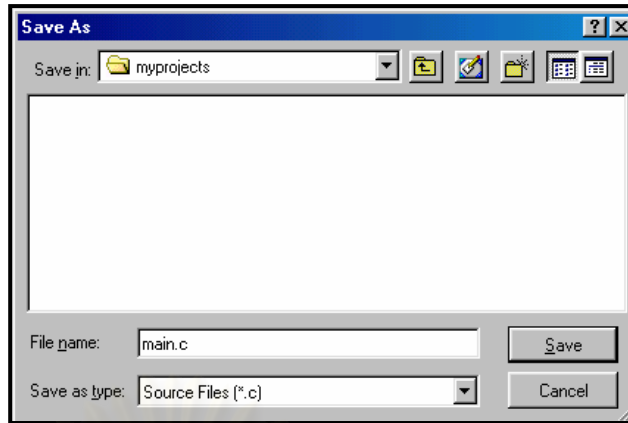
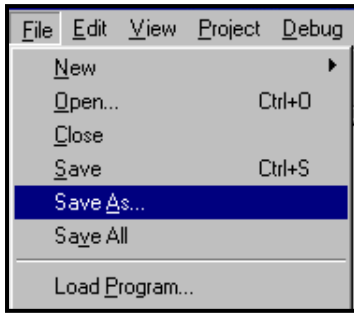
2.1 Menu “File” → ”New” → ”Source file” : หรือกด Ctrl+N จะปรากฏเป็น Text editor ว่าง สำหรับให้พิมพ์ซอร์สโค้ด จากนั้นให้เขียนพิมพ์โค้ดภาษาซีตามโปรแกรม “main.c”



การทำงานของโปรแกรม “main.c” จะพิมพ์คำว่า “BEGIN” และ “END” ที่หน้าจอคอมพิวเตอร์ เมื่อเรารันโปรแกรมนี



2.2 Menu “File” → ”Save as” จะปรากฏเป็นไดอะล็อกบ็อก Save as จากนั้นพิมพ์ชื่อ “main.c” (\*.c) เลือกไดเรกทอรีที่ต้องการเก็บไฟล์ จากนั้นคลิกที่ “Save” จะได้ไฟล์ main.c



3. ส

สร้างไฟล์ “vectors.asm”(\*.asm) เป็นไฟล์ภาษาแอสเซมบลีทำหน้าที่เป็นรีเซตเวกเตอร์ในกรณีที่เราทำการ Restart Program หรือรีเซตโปรแกรม TMS320C6711 จะเริ่มทำงานที่ตำแหน่งรีเซตตามเวกเตอร์นี้ และโปรแกรมนี้อาจจะสั่งให้กระโดดไปเริ่มทำงานที่ Address “\_c\_int00” ซึ่งเป็นจุดเริ่มต้นของโปรแกรม main.c นั่นเอง การสร้างโปรแกรมนี้อาจมีขั้นตอนเหมือนการสร้างไฟล์“main.c”แต่เปลี่ยน เป็นไฟล์ “vectors.asm” แทน

```

;File "vector.asm"
.title "vectors.asm"
.ref _c_int00

.sect "vectors"
rst: mvgl .s2 _c_int00,b0
      mvkh .s2 _c_int00,b0
      b .s2 b0
      nop
      nop
      nop
      nop
      nop

```

```

/* C6711dsk.cmd */
MEMORY
{
  VECS:      org =      0h,   len =    0x220
  I_HS_MEM:  org = 0x00000220, len = 0x00000020
  IRAM:      org = 0x00000240, len = 0x0000FDC0
  SDRAM:     org = 0x80000000, len = 0x01000000
  FLASH:    org = 0x90000000, len = 0x00020000
}
SECTIONS
{
  /* Created in vectors.asm */
  vectors  >> VECS
  /* Created by C Compiler */
  .text    >> IRAM
  .bss     >> IRAM
  .cinit   >> IRAM
  .stack   >> IRAM
  .system  >> IRAM
  .const   >> IRAM
  .switch  >> IRAM
  .cio     >> IRAM
  .mydata  >> SDRAM
  .far     >> SDRAM
}

```

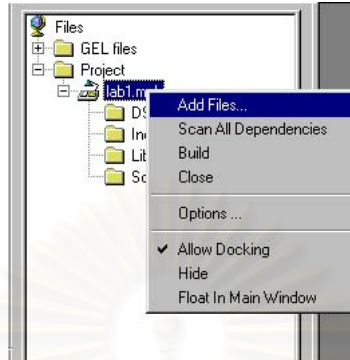
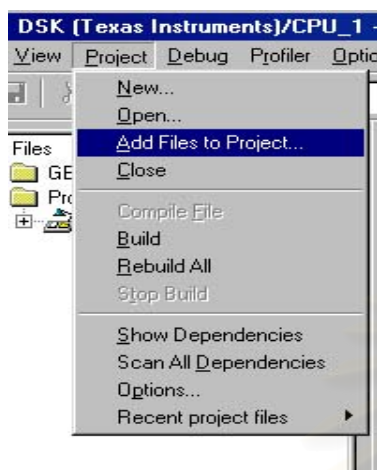
สร้างไฟล์ “C6711dsk.cmd” (\*.cmd) เป็นไฟล์ที่ใช้ในการกำหนดผังหน่วยความจำของ บอร์ด C6711 DSK วิธีการสร้างใช้ text editor และทำเหมือนขั้นตอนการสร้างไฟล์ “main.c” แต่เปลี่ยนไฟล์เป็น “C6711dsk.cmd”

- \*.h เป็น Include Header file ต่างๆ เช่น stdio.h จะมีมาให้พร้อมในโปรแกรม CCS (ไม่ต้องสร้างเอง)
- rts6701.lib เป็น Library File สำหรับ TMS320C6711 ไฟล์นี้จะให้มาพร้อมกับโปรแกรม CCS (ไม่ต้องสร้างเอง)

## การเพิ่มไฟล์ในโปรเจก (Adding Files to a Project)

หลังจากที่เราสร้างไฟล์ต่างๆ เสร็จแล้วขั้นตอนต่อไปก็คือการเพิ่มไฟล์เข้าไปในโปรเจกเพื่อจะได้ทำการคอมไพล์ในขั้นตอนต่อไป

### 1. การเพิ่มไฟล์ในโปรเจกสามารถทำได้ 2 วิธี



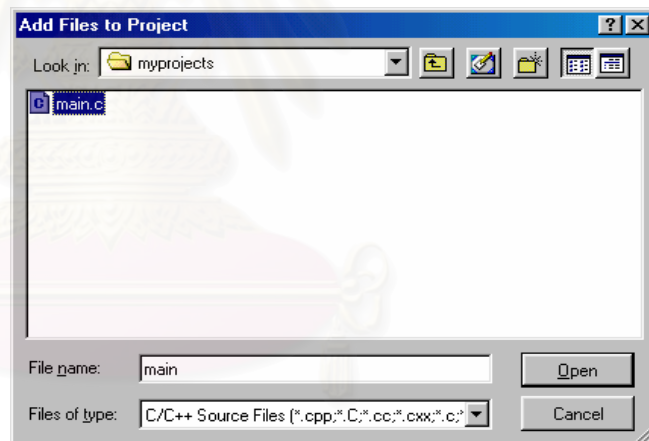
1.1 เลือกเมนู “Project” → “Add Files to Project” หรือ

1.2 คลิกขวาที่ windows Project จากนั้นเลือก “Add File”

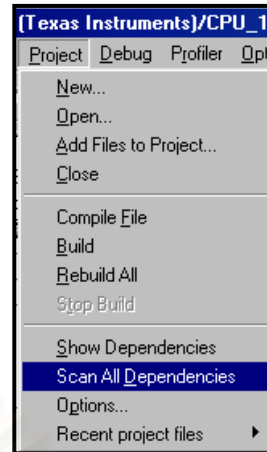
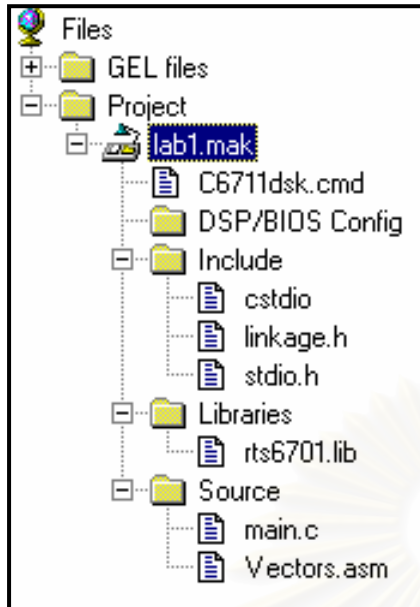
2. จากนั้นให้เลือกไฟล์ที่ได้สร้างไว้คือ “main.c” แล้วคลิกที่ Open จากนั้น Add fileอื่นๆ ตามขั้นตอนที่ผ่านมาให้ครบทุก

ไฟล์ “vector.asm” “c6711dsk.cmd” “rts6701.lib”

( สำหรับไฟล์ rts6701.lib จะอยู่ใน C:\ti\c6000\cgtools\lib) ตามลำดับ

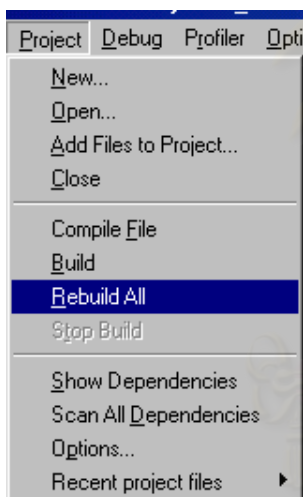


3. ให้คลิกที่เครื่องหมาย “+ Project” เพื่อดูไฟล์โปรแกรมทั้งหมดที่เรา Add เข้าไปว่าครบหรือยัง ในส่วนของ Include Header File โปรแกรม CCS จะเพิ่มให้อัตโนมัติหลังจากที่เรา Build Program หรือเราสามารถตรวจสอบได้โดยเลือกเมนู “Project” → “Scan All Dependencies” ก็จะเห็น Include header file และสามารถดูซอร์สโค้ดภายในของไฟล์แต่ละไฟล์ได้โดยการดับเบิ้ลคลิกที่ชื่อไฟล์นั้นๆ



การ Build และการรันโปรแกรม

ในขั้นตอนนี้เราจะทำการ Build โปรแกรม

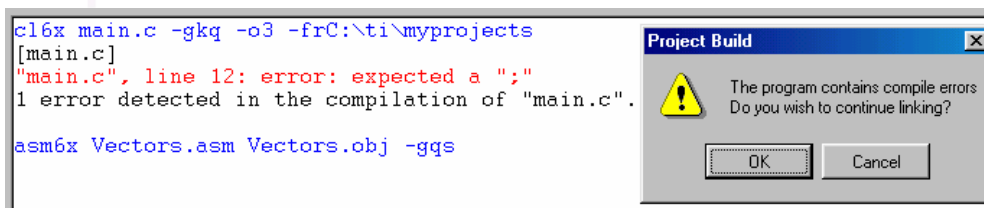


1. วิธีการ Build โปรแกรมทำได้ 2 วิธี คือ
  - 1.1 เลือกเมนู Project → Rebuild All
  - 1.2 คลิกที่ toolbar button (Rebuild All)



2. หลังจากการที่เราทำการ Build All โปรแกรมแล้วสังเกตที่บริเวณ

Build Area จะแสดงผลการ Build โปรแกรม ถ้ากรณีที่มีการเขียนโปรแกรมผิดพลาด ก็จะแสดงข้อความ Error บอกว่ามีการผิดพลาดที่ไฟล์ใดและบรรทัดที่เท่าไร (สมมุติว่าเราลืมเขียน เครื่องหมายเซมิคอลตอน ";" ทำขบรทัดใดสักบรรทัด โปรแกรมก็จะแจ้งให้เราทราบว่ามีการผิดพลาดที่ใด จากนั้นก็ทำการแก้ไขและทำการ Rebuild All โปรแกรมจนกว่าจะผ่าน



ถ้าเราเขียนโปรแกรมถูกต้องก็จะไม่เกิดข้อผิดพลาดใดๆ โปรแกรมก็จะแจ้ง



0 Errors 0 Warnings. ก็แสดงว่าเราสามารถสร้างไฟล์ได้สำเร็จ จะได้เอาต์พุตไฟล์เป็น Lab1.Out (\*.out) ซึ่งจะใช้ในการดาวน์โหลดไปยังบอร์ด C6711 DSK ต่อไป

```
[main.c]
asm6x Vectors.asm Vectors.obj -gqs

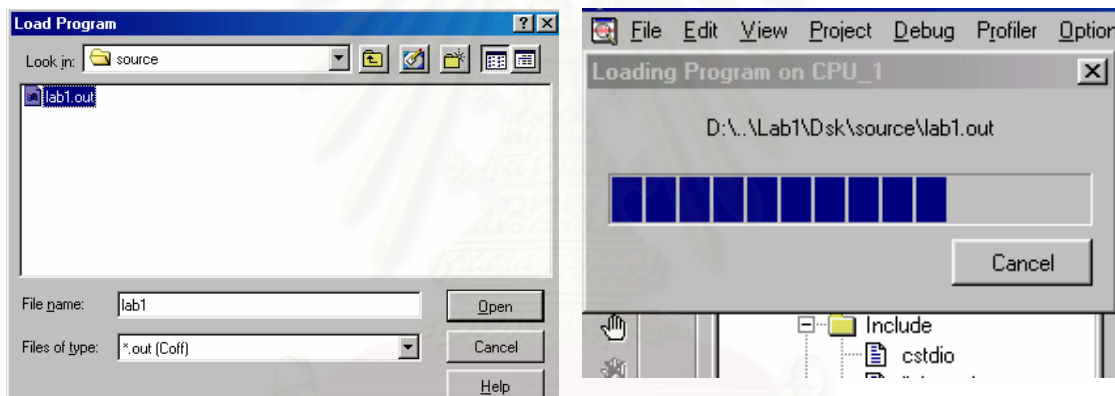
lnk6x "lab1.mak"
TMS320C6x COFF Linker          Version 4.00
Copyright (c) 1996-2000 Texas Instruments Incorporated
Build Complete,
  0 Errors, 0 Warnings.
```

Build / Stdout /

DSP HALTED For Help, press F1

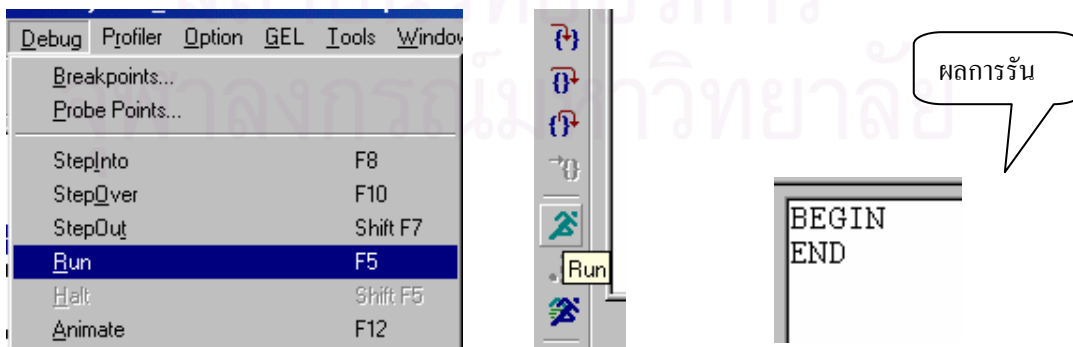
3. หลังจากที่เรา build All โปรแกรมผ่านแล้วก็จะได้ไฟล์ Lab1.out ซึ่งเป็นไฟล์ที่ Load ไปยังบอร์ด DSK วิธีการ Download ไฟล์ Lab1.out (\*.out) นั้นมีขั้นตอนดังนี้

3.1 เลือก “File” → “Load Program” ; ให้เลือก File “Lab1.out” คลิก “Open” โปรแกรมก็จะ Load File Lab1.out ไปยัง DSK Board ทันที



4. หลังจากที่เรา Load files เสร็จก็พร้อมที่จะรันโปรแกรม การรันโปรแกรมทำได้ดังนี้คือ

4.1 เลือกเมนู “Debug” → “Run” หรือ กด “F5” หรือคลิกที่ toolbar button “Run” จากนั้นผลการรันโปรแกรมก็จะปรากฏตามที่เราได้เขียนโปรแกรม



## การ Debug โปรแกรม (Debugging Tools)

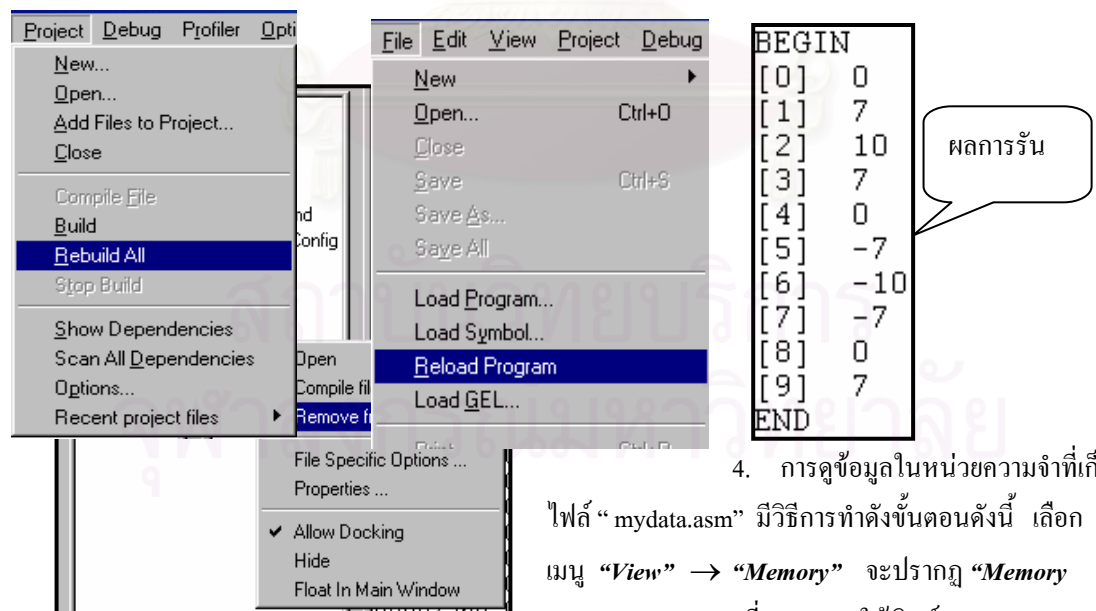
CCS มีเครื่องมือที่ใช้ในการ Debug Program หลายอย่าง เช่น Breakpoints, Watch Windows, Profiling, Graph display ฯลฯ ซึ่งจะได้แนะนำการใช้ในขั้นตอนถัดไป

1. ทำการพิมพ์ไฟล์ “mydata.asm” และ Add file to project และสร้างไฟล์ “main\_point.c” ตามตัวอย่างการทำงานของโปรแกรม “main\_point.c” คือจะใช้ตัวแปรแบบพอยน์เตอร์ไปชี้ที่ไฟล์ของ “mydata.asm” ที่ตำแหน่งแอดเดรส 0x80000000 จากนั้นก็จะพิมพ์ข้อมูลในไฟล์ “mydata.asm” จำนวน 10 ตัวเลขที่หน้าจอ
2. Remove File “main.c” ที่มีอยู่ก่อนแล้วออกจากโปรเจกงาน จากนั้นทำการ Add files “main\_point.c”

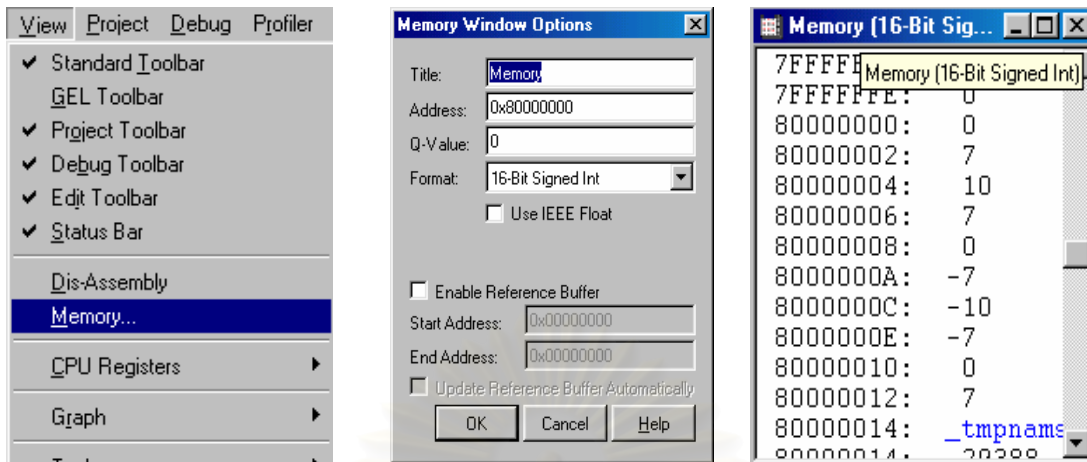
|  |  |
|--|--|
| <pre> ; mydata.asm  .sect ".mydata" .short 0 .short 7 .short 10 .short -7 .short 0 .short -7 .short -10 .short -7 .short 0 .short 7         </pre> | <pre> /* main_point.c */ #include &lt;stdio.h&gt;  void main() {     int i;     short *point;      point=(short*)0x80000000; /*address file mydata*/     printf("BEGIN\n");     for(i=0;i&lt;10;i++)     {         printf("[%d] %d\n",i,point[i]);     }     printf("END\n"); }         </pre> |
|--|--|

ไปยัง Projects

3. จากนั้นทำการ “Rebuild All” และ “Reload Program” เครื่องจะทำการ Load file “lab1.out” ให้อัตโนมัติ จากนั้นทำการรันโปรแกรม (กด F5) เพื่อดูผลการทำงานของโปรแกรม



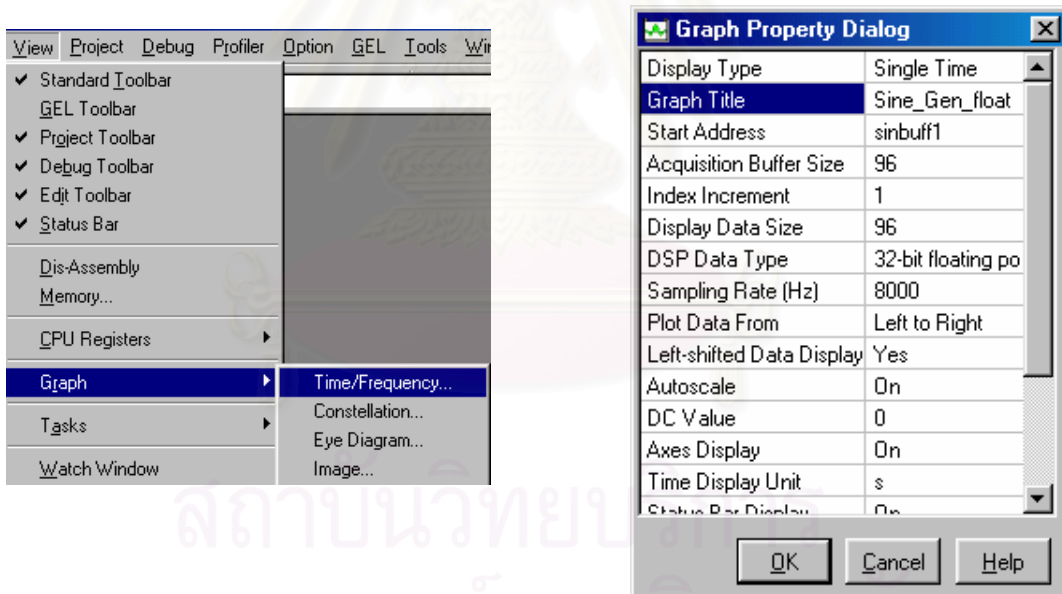
4. การดูข้อมูลในหน่วยความจำที่เก็บไฟล์ “mydata.asm” มีวิธีการทำดังขั้นตอนดังนี้ เลือกเมนู “View” → “Memory” จะปรากฏ “Memory Windows Options” ที่ Address ให้พิมพ์ 0x80000000 เพราะเป็น Address ที่เราเก็บข้อมูลไฟล์ “mydata” ที่ Format ให้เลือกเป็นแบบ “16-Bit Signed Int” จากนั้นคลิก OK. จะปรากฏ Windows Memory แสดงข้อมูลของไฟล์ “mydata” เริ่มที่ Address 0x80000000



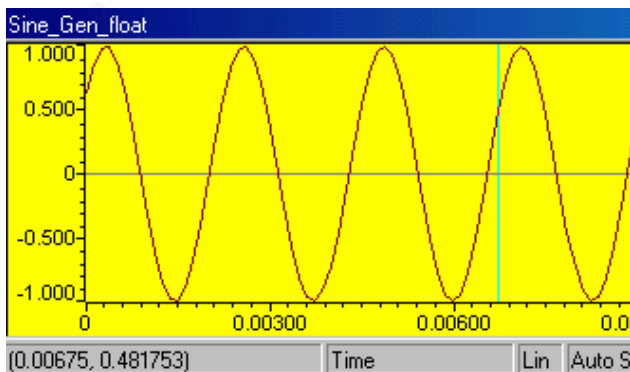
**การแสดงผลด้วยกราฟ (Displaying Graphs)**

เราสามารถดูรูปสัญญาณหรือข้อมูลที่แสดงในลักษณะของกราฟตามขั้นตอนดังนี้

1. เลือกเมนู View → Graph → Time/Frequency จะปรากฏ “Graph Property dialog” Set ค่า



ต่างๆ ตามตัวอย่างจากนั้นคลิก OK.

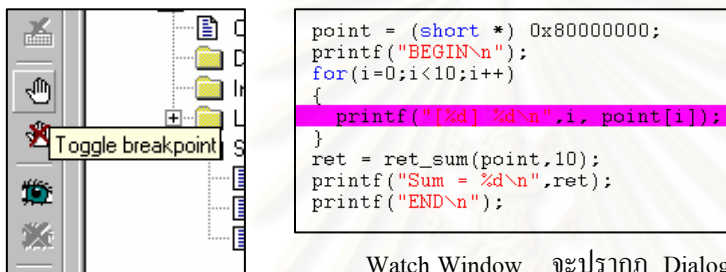


ข้อมูลที่แอดเดรส sinbuffl จะถูกพล็อตเป็นรูปภาพแสดงเป็นรูปสัญญาณ Sine wave ตามค่าของข้อมูลที่อยู่ในไฟล์ “mydata.asm”

### การใช้ Breakpoints และ Watch Window

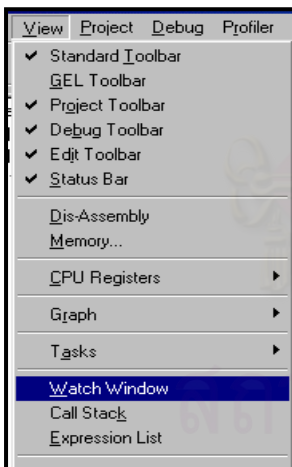
การใช้ Breakpoints เพื่อหยุดโปรแกรมและ Watch Window เพื่อดูค่าตัวแปรต่างๆ ในขณะที่รันโปรแกรมเพื่อจะได้ตรวจสอบว่าโปรแกรมที่เราเขียนมีการรันในแต่ละรอบมีค่าที่ถูกต้องหรือไม่ วิธีการใช้ Breakpoints และ Watch Windows มีขั้นตอนดังนี้

1. เลือก File → Reload Program
2. ดับเบิ้ลคลิกที่ไฟล์ “main\_ret.c.c”
3. คลิกที่บรรทัด printf(“[%d] %d\n”,i, point[i]);
4. คลิกที่ “ToggleBreakpoint” toolbar button หรือ กด F9 จะปรากฏแถบสีม่วงแดงเข้ม (magenta)

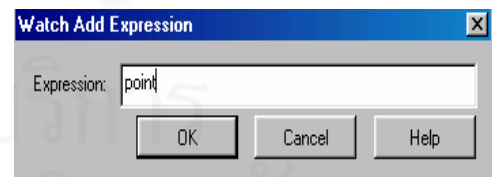
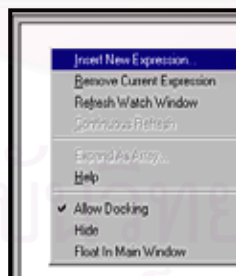


5. เลือก “View” →

Watch Window จะปรากฏ Dialog ตรงมุมขวาด้านล่างของ Window จากนั้นให้คลิกขวาตรงบริเวณพื้นที่ของ Dialog แล้วเลือก “Insert New Expression” จากนั้นจะปรากฏ

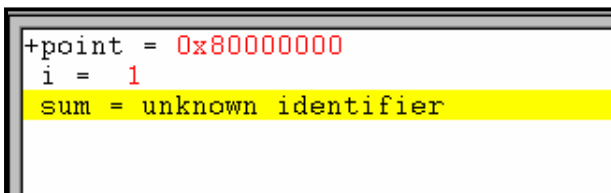


Dialog “Watch Add Expression” พิมพ์ตัวแปรที่ต้องการดูค่า เช่น point , i , sum แล้วคลิก OK จากนั้นทำการ “Insert New Expression” ให้ครบทุกตัวแปรตามที่เรากำลังจะดูค่า เช่น ค่าของตัวแปร “i” , “sum”



6. เลือก menu Debug → Run (F5) เมื่อโปรแกรมรันมาถึงตรงจุด Breakpoint โปรแกรมก็จะหยุดตรง

บรรทัดที่เรา Break point ไว้และแสดงค่าของตัวแปรต่างๆ ที่เลือกไว้ ดังตัวอย่าง



## 6.2 ภาคผนวก ข

### ทดสอบการใช้งานบอร์ด TMS 320 C6000

#### สร้างสัญญาณไซน์โดยใช้หลักการของฟิลเตอร์แบบ IIR (Sine wave generator using IIR filter)

การสร้างสัญญาณไซน์ความถี่ 440 Hz โดยใช้หลักการของฟิลเตอร์แบบ IIR

จากสมการพื้นฐานจะได้

$$y(n) = a_1 y(n-1) + a_2 y(n-2)$$

จากสมการกำหนดให้

$$\text{ค่าคงที่(Constant)} \quad a_1 = 2\cos(\omega T_s) \quad a_2 = -1$$

$$\text{ค่าเริ่มต้น(Initial)} \quad y(0) = 0 \quad y(1) = \sin(\omega T_s)$$

คำนวณหาค่า  $\omega$  จาก

$$\omega = 2\pi f = 2\pi \times 440 = 880\pi$$

คำนวณหาค่า  $T_s$  จาก  $f_s = 8000$  Hz จะได้  $T_s = 1/8000$

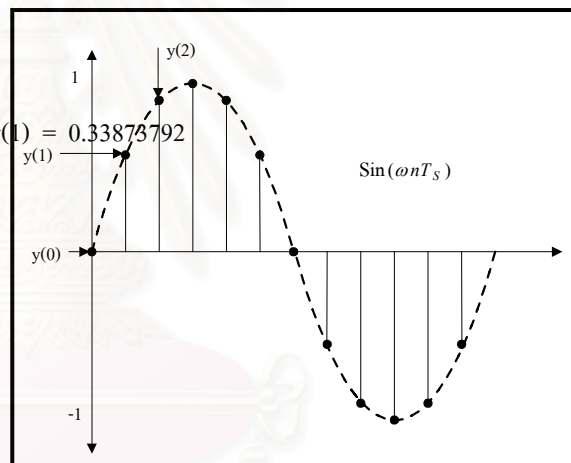
แทนค่าเพื่อหาค่า  $a_1$  จะได้  $a_1 = 2\cos(\omega T_s) = 2\cos(880\pi/8000) = 1.8817615$

คำนวณหาค่า  $y(1)$  จากสมการ  $y(1) = \sin(\omega T_s) = \sin(880\pi/8000) = 0.33873792$

ดังนั้นค่าเริ่มต้นที่ใช้ในการคำนวณ

$$a_1 = 1.8817615 \quad : \quad a_2 = -1$$

$$y(0) = 0$$



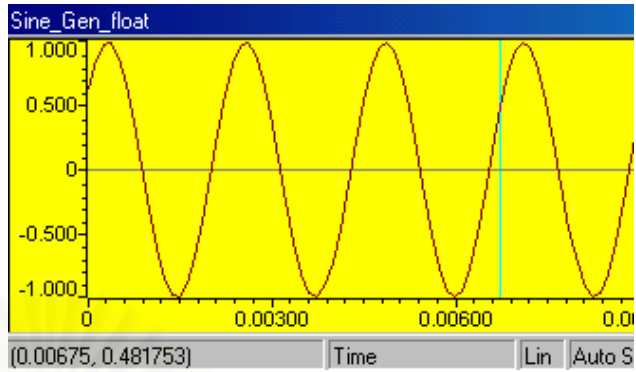
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ผลการทดลอง

```

sine_float 0 = 0.637424
sine_float 1 = 0.860742
sine_float 2 = 0.982287
sine_float 3 = 0.987688
sine_float 4 = 0.876307
sine_float 5 = 0.661312
sine_float 6 = 0.368125
sine_float 7 = 0.031411
sine_float 8 = -0.309017
sine_float 9 = -0.612907
sine_float 10 = -0.844328
sine_float 11 = -0.975917
sine_float 12 = -0.992115
sine_float 13 = -0.891007
sine_float 14 = -0.684547
sine_float 15 = -0.397148
sine_float 16 = -0.062791
sine_float 17 = 0.278991
sine_float 18 = 0.587785
sine_float 19 = 0.827080
sine_float 20 = 0.968583
sine_float 21 = 0.995562
sine_float 22 = 0.904827
sine_float 23 = 0.707107
sine_float 24 = 0.425779
sine_float 25 = 0.094108
sine_float 26 = -0.248690
    
```

ผลการรัน



กราฟรูปสัญญาณไซน์ที่ได้จากการคำนวณ

| Memory (32-Bit Floating Point) |              |             |
|--------------------------------|--------------|-------------|
| 00007720:                      | sinbuff1     |             |
| 00007720:                      | 0.63742399   | 0.86074197  |
| 00007728:                      | 0.98228717   | 0.9876883   |
| 00007730:                      | 0.87630665   | 0.66131186  |
| 00007738:                      | 0.3681246    | 0.031410873 |
| 00007740:                      | -0.30901682  | -0.61290687 |
| 00007748:                      | -0.84432781  | -0.97591668 |
| 00007750:                      | -0.99211466  | -0.89100653 |
| 00007758:                      | -0.68454719  | -0.39714807 |
| 00007760:                      | -0.062790811 | 0.27899075  |
| 00007768:                      | 0.58778489   | 0.82708025  |

| Memory (32-Bit Binary) |                                  |
|------------------------|----------------------------------|
| 00007720:              | sinbuff1                         |
| 00007720:              | 00111111001000110010111000111000 |
| 00007724:              | 00111111010111000101100110010110 |
| 00007728:              | 00111111011110110110111001011100 |
| 0000772C:              | 0011111101111001101100100100100  |
| 00007730:              | 00111111011000000101010110100010 |
| 00007734:              | 00111111001010010100101110111100 |
| 00007738:              | 00111110101111000111101011010100 |
| 0000773C:              | 00111101000000001010100010110000 |
| 00007740:              | 10111110100111100011011101110100 |
| 00007744:              | 10111111000111001110011101110111 |
| 00007748:              | 10111111010110000010010111011110 |
| 0000774C:              | 10111111011110011101010110101101 |

แสดงข้อมูลในหน่วยความจำ แสดงในรูปแบบของเลขฐานสอง

### 9.3 ภาคผนวก ค

#### บทความทางวิชาการ

N. Tangsangiumvisai and N. Ithisoponkul, “Double-Talk Detection for Stereophonic Acoustic Echo Cancellation in a Time-Varying Environment” ในการประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 25 (EECON-25) วันที่ 21-22 พฤศจิกายน 2545, หน้า 5 – 9.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย