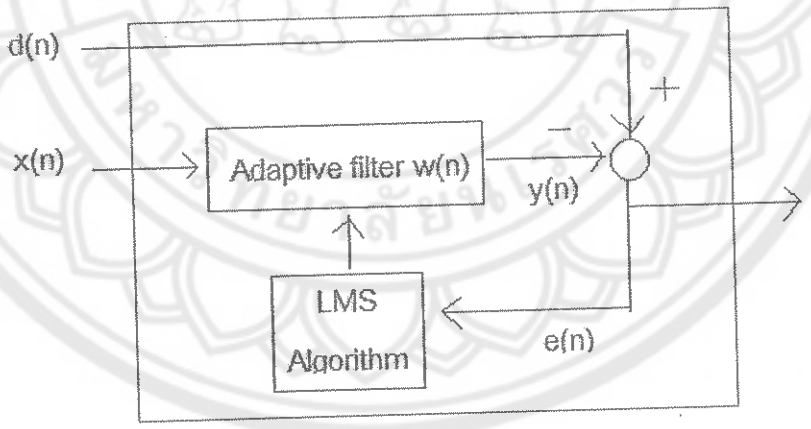
บทที่ 4

การคัดเลือกอัลกอริธึมที่จะใช้ในเครื่องช่วยฟัง

การทดสอบนี้เป็นการจำลองการทำงานของอัลกอริธึมที่จะนำไปใช้บรรจุในเครื่องช่วยฟัง เพื่อจำลองคุณสมบัติในการลดสัญญาณรบกวนของเครื่องช่วยฟังแทนการใช้เครื่องช่วยฟังจริง ซึ่งทำให้การทดสอบง่ายขึ้น

การทดลองเบื้องต้นของตัวกรองแบบปรับตัวเองโดยใช้อัลกอริธึมปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด

ในการออกแบบการทดลองนี้ผู้วิจัยได้พิจารณาเองตามความเหมาะสมที่ได้รับจากการทดสอบในเบื้องต้น เพื่อให้การพิจารณาผลการทดลองง่ายขึ้น ในการทดลองนี้จะใช้อัลกอริธึมปรับตัวสำหรับตัวกรองสัญญาณ โดยใช้การปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด ซึ่งมีโครงสร้างการทำงานของอัลกอริธึมดังภาพ 27



ภาพ 27 แสดงรูปแบบตัวกรองแบบปรับตัวเองโดยใช้อัลกอริธึมปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด

จากภาพ 27 มีสมการของระบบที่สำคัญดังนี้

$$y(n) = \sum_{l=0}^{L-1} w(n)x(n-l) \quad (22)$$

$$e(n) = d(n) - y(n) \quad (23)$$

$$w(n+1) = w(n) + \mu X(n)e(n) \quad (24)$$

โดยที่

$x(n)$ = สัญญาณอ้างอิงที่มีความถี่เดียวกับสัญญาณที่เข้ามารบกวน

$d(n)$ = สัญญาณข้อมูลที่ต้องการ $s(n)$ +สัญญาณที่เข้ามารบกวนที่ความถี่หนึ่ง $n(n)$

$y(n)$ = สัญญาณขาออกจากวงจรกรองที่ปรับตัวจนมีลักษณะคล้ายสัญญาณรบกวน

$e(n)$ = สัญญาณข้อมูลที่ต้องการขาออก

$_n$ คือค่าเวลา และ L คือค่าลำดับของสัมประสิทธิ์ตัวกรอง (Order)

μ = อัตราการปรับตัว

จากนั้นจะเริ่มทำการทดลองโดยกำหนดค่าตัวแปรที่ใช้ในการทดสอบเบื้องต้นก่อนที่จะทำการทดลองว่า อัลกอริธึมนี้สามารถใช้งานกำจัดสัญญาณรบกวนได้จริง ดังนี้

$L = 20$ ตัว

$s(n) = A_s \sin(2\pi f_s)$

$n(n) = A_n \sin(2\pi f_n)$ + สัญญาณรบกวนรูปแบบขาว

$x(n) = \sin(2\pi f_n)$ + สัญญาณรบกวนรูปแบบขาว

$d(n) = s(n) + n(n)$

เมื่อได้ทำการกำหนดค่าเริ่มต้นของตัวแปรต่างๆที่จะนำมาใช้กับสมการจนครบ จะเริ่มทำการสร้างอัลกอริธึม แล้วทดสอบโดยการปรับค่าของสัญญาณที่เข้ามารบกวน แล้วสังเกตผลการทดลองที่ได้ดังภาพ 28 ซึ่งจะมีการปรับค่าดังนี้

ขนาดของสัญญาณรบกวน (Amplitude noise) หรือ A_n ให้มีค่า 1 โวลท์

ความถี่ของสัญญาณรบกวน (Frequency noise) หรือ f_n ให้มีค่า 100 เฮิรตซ์

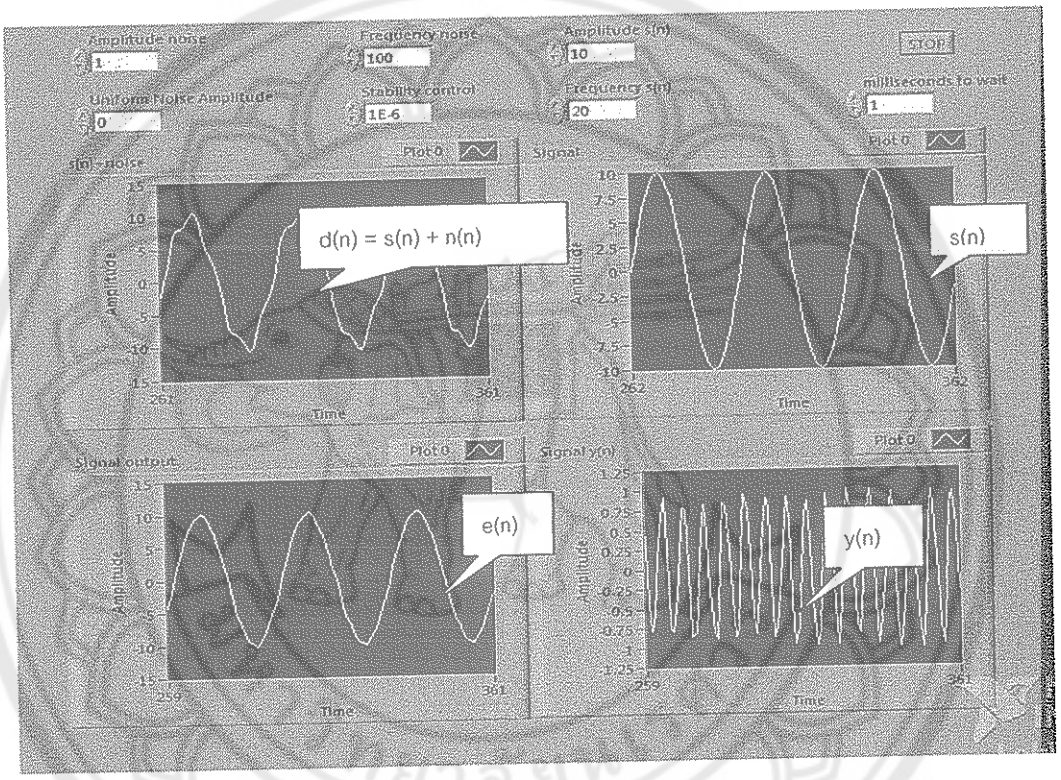
ขนาดของ (Amplitude) $s(n)$ หรือ A_s ให้มีค่า 10 โวลท์

ขนาดของสัญญาณรบกวนแบบขาว (Uniform Noise Amplitude) เป็น 0 โวลท์

ตัวควบคุมเสถียรภาพของระบบ (Stability control) หรือ μ ให้มีค่า 0.000001

ความถี่ (Frequency) ของ $s(n)$ หรือ f_s ให้มีค่า 20 เฮิรตซ์ (เหตุผลที่ต้องใช้ความถี่สัญญาณต่ำ ๆ ในการทดลองก็เพื่อให้ง่ายต่อการพิจารณา เพราะถ้าความถี่ใกล้สัญญาณรบกวนจะทำให้พิจารณายากมาก)

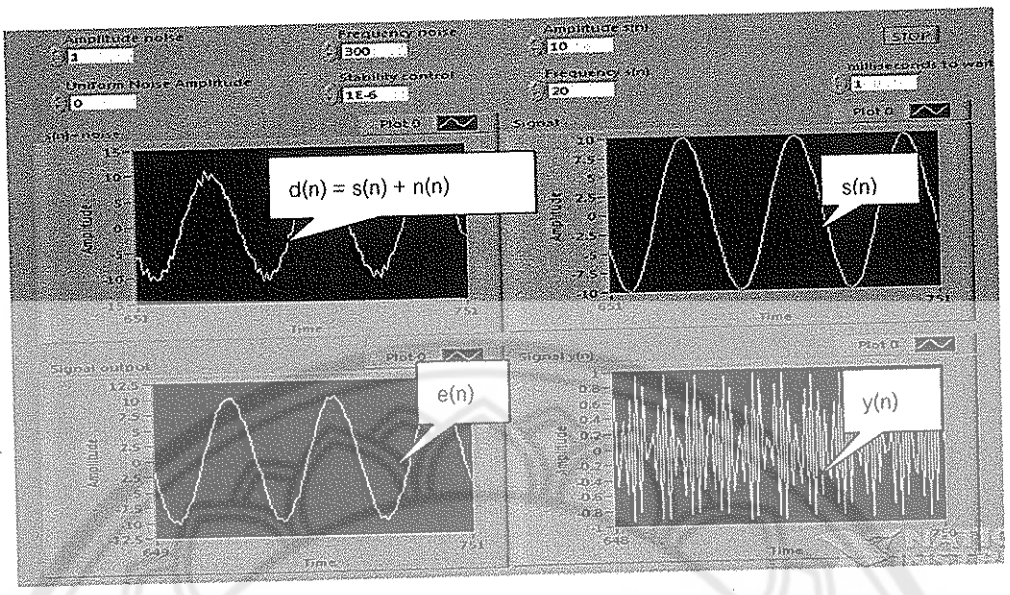
เวลาหน่วง (Milliseconds to wait) ให้มีค่า 1 มิลลิวินาที



ภาพ 28 แสดงกราฟเมื่อปรับค่า $f_n = 100Hz$

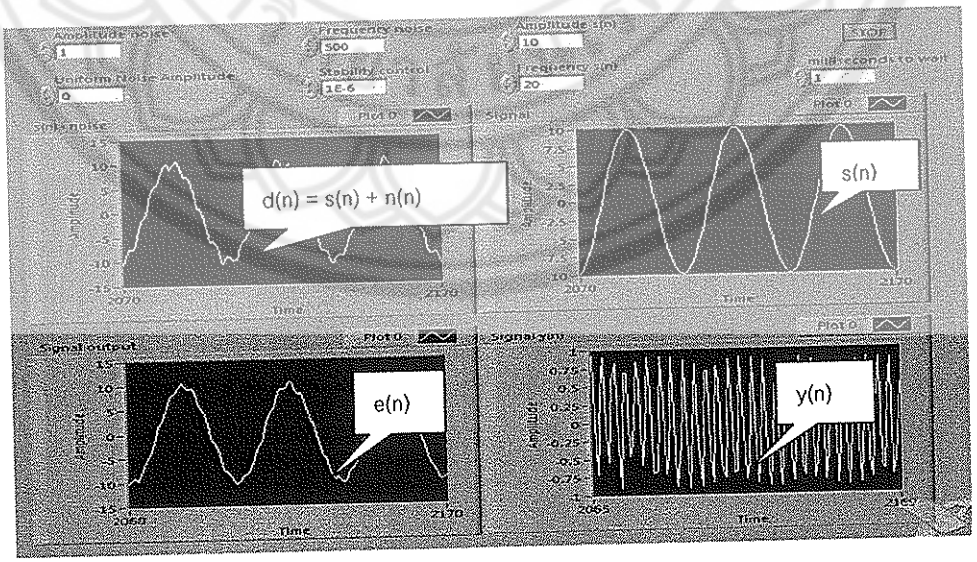
จากภาพ 28 จะเห็นได้ว่าอัลกอริธึมสามารถปรับตัวกำจัดสัญญาณรบกวนรูปคลื่นไซน์ออกไปได้จากกราฟสัญญาณขาออก $e(n)$ ดังในภาพที่ 28 ตอนแรกระบบอาจจะยังปรับตัวไม่ได้เลยทีเดียว แต่ในเวลาต่อมาระบบเริ่มปรับตัวได้ ก็จะสามารถกำจัดสัญญาณรบกวนออกไปได้ ทำให้สัญญาณขาออก $e(n)$ ที่ได้มีความคมชัดขึ้น คล้ายสัญญาณ $s(n)$

จากนั้นทดลองต่อโดยการปรับค่าความถี่ของสัญญาณที่เข้าไปรบกวน (Frequency noise หรือ f_n) ให้มีค่า 300 เฮิรตซ์ จะทำให้สัญญาณขาออก $e(n)$ จะมีสัญญาณรบกวนก่อนเล็กน้อยในตอนแรก แต่ในเวลาต่อมาระบบเริ่มปรับตัวได้ก็จะสามารถกำจัดสัญญาณรบกวนออกไปได้ ทำให้สัญญาณขาออก $e(n)$ ที่ได้มีความคมชัด เหมือนสัญญาณ $s(n)$ ดังภาพ 29



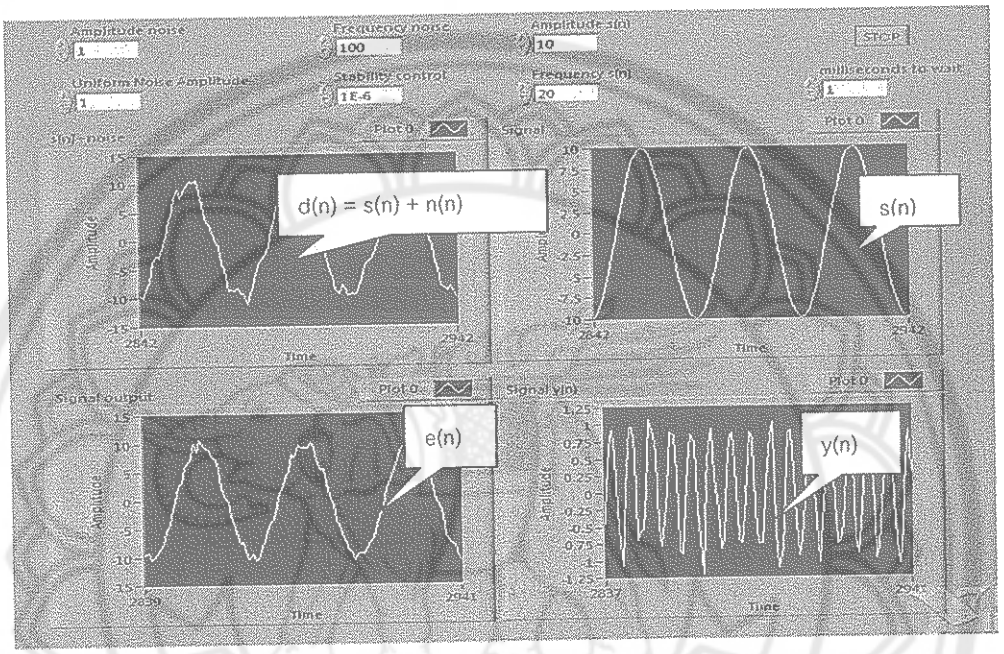
ภาพ 29 แสดงกราฟเมื่อปรับค่า $f_n = 300Hz$

จากนั้นทดลองต่อโดยการปรับค่าความถี่ของสัญญาณที่เข้าไปรบกวน (Frequency noise หรือ f_n) ให้มีค่า 500 เฮิรตซ์ ทำให้สัญญาณขาออก $e(n)$ จะมีสัญญาณรบกวนออกมาตลอดจะเห็นได้ว่าระบบไม่สามารถปรับตัวกำจัดสัญญาณรบกวนที่ 500 เฮิรตซ์ ออกไปได้ดังภาพ 30 และได้ทดลองโดยปรับความถี่สัญญาณที่เข้าไปรบกวนให้สูงขึ้นไปเรื่อย ๆ พบว่าระบบไม่สามารถปรับตัวกำจัดสัญญาณที่รบกวนได้เลย



ภาพ 30 แสดงกราฟเมื่อปรับค่า $f_n = 500Hz$

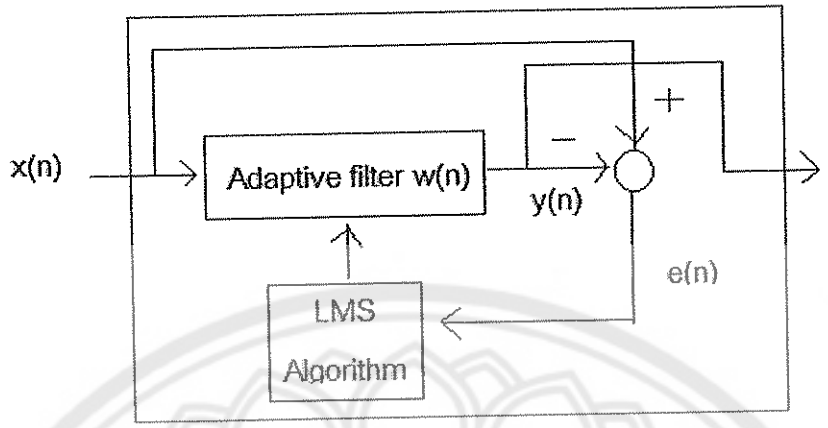
ทำการเพิ่มขนาดของสัญญาณรบกวนรูปแบบขาว (Uniform Noise Amplitude) ให้มีค่าเป็น 1 โวลท์ แล้วสังเกตผลการทดลองจากภาพ 31 จะเห็นได้ว่า สัญญาณขาออก $e(n)$ จะมีสัญญาณรบกวนรูปแบบขาวปนอยู่ ดังนั้นระบบไม่สามารถกำจัดสัญญาณรบกวนรูปแบบขาวออกไปได้



ภาพ 31 แสดงกราฟเมื่อปรับค่า $f_n = 100Hz$ และสัญญาณรบกวนรูปแบบขาว = 1

การทดลองเบื้องต้นของตัวปรับความคมชัดเส้นสัญญาณโดยใช้อัลกอริธึมปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด

ในการทดลองนี้จะใช้อัลกอริธึมที่เป็นตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง โดยใช้การปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด ซึ่งมีโครงสร้างการทำงานของอัลกอริธึมดังภาพ 32



ภาพ 32 โครงสร้างของตัวปรับความคมชัดเส้นสัญญาณโดยใช้อัลกอริธึมปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด

จากภาพ 32 มีสมการของระบบที่สำคัญดังนี้

$x(n)$ = สัญญาณข้อมูลที่ต้องการ $s(n)$ +สัญญาณที่เข้ามารบกวน $n(n)$

$y(n)$ = สัญญาณข้อมูลที่ต้องการขาออกหรือสัญญาณขาออกจากวงจรกรอง

$e(n)$ = ข้อผิดพลาดของระบบ

โดยที่

$$y(n) = \sum_{l=0}^{L-1} w(n)x(n-l-d) \quad (25)$$

$$e(n) = x(n) - y(n) \quad (26)$$

$$w(n+1) = w(n) + \mu X(n)e(n) \quad (27)$$

n คือค่าเวลา และ L คือค่าลำดับของสัมประสิทธ์ตัวกรอง

μ = อัตราการปรับตัว

จากนั้นจะเริ่มทำการทดลองโดยกำหนดค่าตัวแปรที่ใช้ในการทดสอบเบื้องต้นก่อนที่จะทดลองว่า อัลกอริธึมนี้สามารถใช้งานกำจัดสัญญาณรบกวนได้จริง ดังนี้

L = 20 ตัว

$s(n) = A_s \sin(2\pi f_s)$

$n(n) = A_n \sin(2\pi f_n)$ + สัญญาณรบกวนรูปแบบขาว

$x(n) = s(n) + n(n)$

เมื่อได้ทำการกำหนดค่าเริ่มต้นของตัวแปรต่าง ๆ ที่จะนำมาใช้กับสมการจนครบ จะเริ่มทำการสร้างอัลกอริธึม โดยทำการปรับค่าของสัญญาณที่เข้ามารบกวนดังภาพที่ 33 แล้วสังเกตผลการทดลองที่ได้ ซึ่งจะมีการปรับค่าดังนี้

ขนาดของสัญญาณรบกวน (Amplitude noise) หรือ A_n ให้มีค่า 0 โวลท์

ความถี่ของสัญญาณรบกวน (Frequency noise) หรือ f_n ให้มีค่า 100 เฮิรตซ์

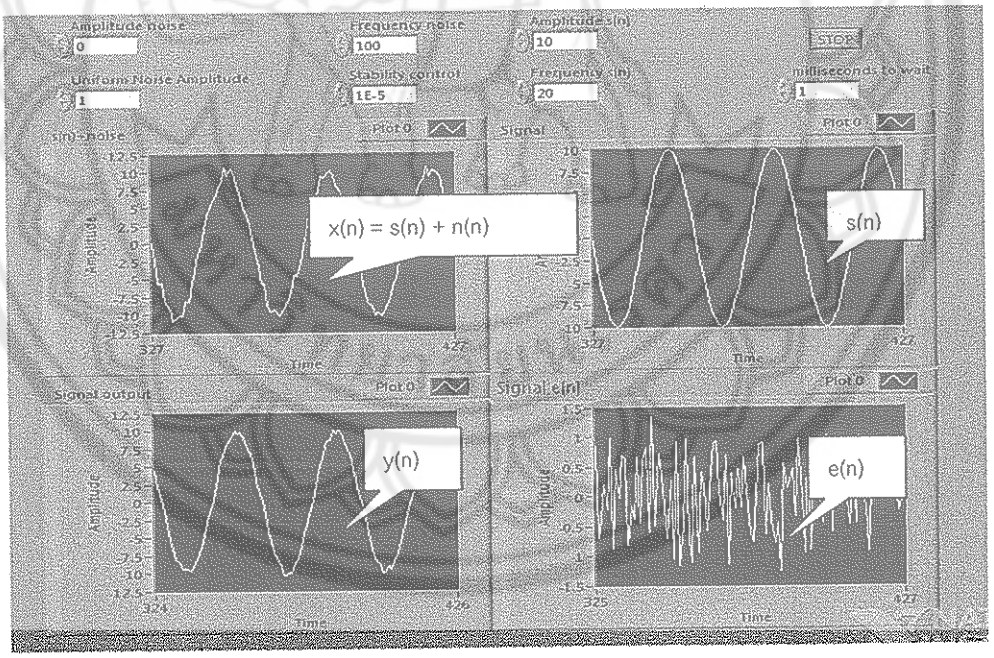
ขนาดของสัญญาณ (Amplitude) s(n) หรือ A_s ให้มีค่า 10 โวลท์

ขนาดของสัญญาณรบกวนรูปแบบขาว (Uniform Noise Amplitude) ให้มีค่า 1 โวลท์

ตัวควบคุมเสถียรภาพของระบบ (Stability control) หรือ μ ให้มีค่า 0.00001

ความถี่ของสัญญาณ (Frequency) s(n) หรือ f_s ให้มีค่า 20 เฮิรตซ์ (เหตุผลที่ต้องใช้ความถี่สัญญาณต่ำ ๆ ในการทดลองก็เพื่อให้ง่ายต่อการพิจารณา เพราะถ้าความถี่ใกล้สัญญาณรบกวนจะทำให้พิจารณายากมาก)

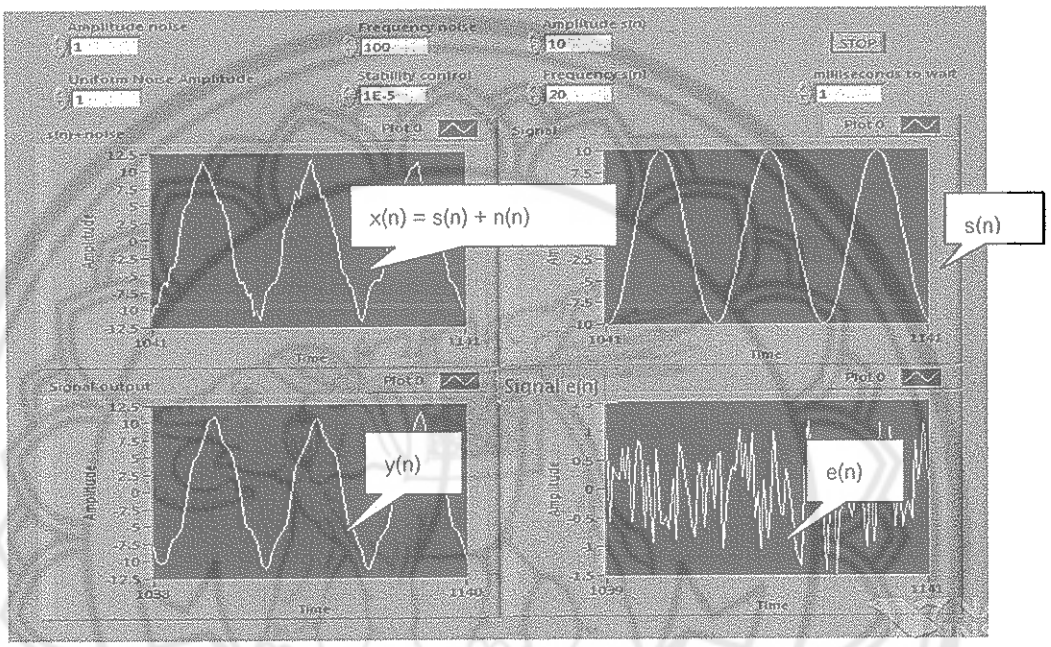
เวลาหน่วง (Milliseconds to wait) ให้มีค่า 1 มิลลิวินาที



ภาพ 33 แสดงกราฟเมื่อปรับค่าสัญญาณรบกวนรูปแบบขาว = 1 โวลท์

จากภาพ 33 จะเห็นได้ว่าอัลกอริธึมสามารถปรับตัวกำจัดสัญญาณรบกวนรูปแบบขาวออกไปได้จากกราฟสัญญาณขาออก $y(n)$ ในภาพ 33 แต่ก็พบว่ายังคงมีปัญหาตรงบริเวณยอดคลื่นยังมีความผิดเพี้ยนอยู่ ยิ่งความถี่ของ $s(n)$ สูง ๆ จะเจอปัญหานี้มากขึ้น

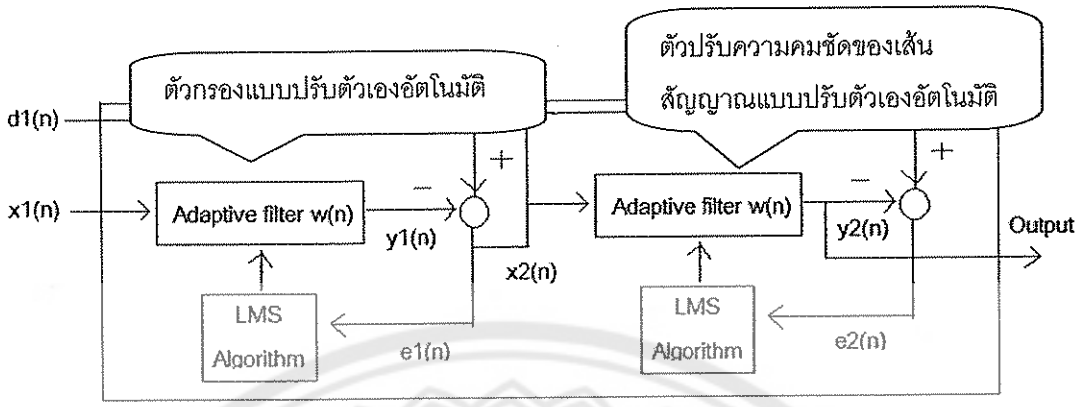
จากนั้นทำการทดลองต่อ โดยการเพิ่มสัญญาณที่เข้าไปรบกวนด้วยความถี่ 100 เฮิรตซ์ ทำได้โดยการปรับค่าความถี่ของสัญญาณที่เข้าไปรบกวน (Frequency noise หรือ f_n) ให้มีค่า 100 เฮิรตซ์ แล้วสังเกตผลการทดลองดังภาพ 34 จะเห็นได้ว่า กราฟของสัญญาณขาออก $y(n)$ ไม่สามารถกำจัดสัญญาณรบกวนรูปคลื่นไซน์ออกไปได้



ภาพ 34 แสดงกราฟเมื่อปรับค่า $f_n = 100Hz$ และ Uniform white noise = 1

การทดลองเบื้องต้นของตัวกรองแบบปรับตัวเองทำงานร่วมกับตัวปรับความคมชัดเส้นสัญญาณ โดยใช้อัลกอริธึมปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด

ในการทดลองนี้จะใช้อัลกอริธึมปรับตัวสำหรับตัวกรองสัญญาณ โดยใช้การปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุดทำงานร่วมกับตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง ซึ่งมีโครงสร้างการทำงานของอัลกอริธึมดังภาพ 35 จะเห็นได้ว่าระบบแรกจะเป็นการทำงานของอัลกอริธึมปรับตัวสำหรับตัวกรองสัญญาณ เพื่อทำการกำจัดสัญญาณรูปคลื่นไซน์ที่เข้าไปรบกวน ต่อจากนั้นจะนำสัญญาณขาออกของอัลกอริธึมปรับตัวสำหรับตัวกรองสัญญาณไปป้อนเข้าให้กับตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง เพื่อที่จะทำการกำจัดสัญญาณรบกวนรูปแบบขาว ออกจากสัญญาณข้อมูล



ภาพ 35 โครงสร้างของตัวกรองแบบปรับตัวเองทำงานร่วมกับตัวปรับความคมชัดเส้นสัญญาณ โดยใช้อัลกอริธึมปรับค่าน้ำหนักแบบกำลังสองเฉลี่ยน้อยสุด

จากภาพ 35 มีจะมีสมการของระบบที่สำคัญดังนี้

$x1(n)$ = สัญญาณอ้างอิงที่มีความถี่เดียวกับสัญญาณที่เข้ามารบกวน

$d1(n)$ = สัญญาณข้อมูลที่ต้องการ $s(n)$ +สัญญาณที่เข้ามารบกวน $n(n)$

$y1(n)$ = สัญญาณขาออกจากวงจรกรอง

$e1(n)$ = สัญญาณป้อนกลับ

$x2(n)$ = สัญญาณข้อมูลที่ต้องการ+สัญญาณที่เข้ามารบกวน

$y2(n)$ = สัญญาณข้อมูลที่ต้องการขาออก

$e2(n)$ = สัญญาณที่ป้อนกลับ

โดยที่

$$y1(n) = \sum_{l=0}^{L-1} w_1(n)x(n-l) \tag{28}$$

$$y2(n) = \sum_{l=0}^{L-1} w_1(n)x(n-l-d) \tag{29}$$

$$e1(n) = d1(n) - y1(n) \tag{30}$$

$$e2(n) = x2(n) - y1(n) \tag{31}$$

$$w(n+1) = w(n) + \mu X(n)e(n) \tag{32}$$

n คือค่าเวลา และ L คือค่าลำดับของสัมประสิทธ์ตัวกรอง

μ = อัตราการปรับตัว ในการทดลองนี้จะมี 2 ส่วน คือ ส่วนแรกใช้ปรับในอัลกอริธึมตัวกรองแบบปรับตัวเองจะเรียกว่า ตัวควบคุมเสถียรภาพของระบบ (Stability control) ตัวกรองแบบปรับตัวเอง ส่วนที่สองจะใช้ในตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง จะเรียกว่าตัวควบคุมเสถียรภาพของระบบตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง

จากนั้นจะเริ่มทำการทดลองโดยกำหนดค่าตัวแปรที่ใช้ในการทดสอบเบื้องต้นก่อนที่จะทดลองว่า อัลกอริธึมนี้สามารถใช้งานกำจัดสัญญาณรบกวนได้จริง ดังนี้

$L = 20$ ตัว

$s(n) = A_s \sin(2\pi f_s)$

$n(n) = A_n \sin(2\pi f_n)$ + สัญญาณรบกวนรูปแบบขาว

$x1(n) = \sin(2\pi f_n)$ + สัญญาณรบกวนรูปแบบขาว

$d1(n) = s(n) + n(n)$

เมื่อได้ทำการกำหนดค่าเริ่มต้นของตัวแปรต่าง ๆ ที่จะนำมาใช้กับสมการจนครบ จะทำการสร้างอัลกอริธึม ทดสอบโดยการปรับค่าของสัญญาณที่เข้ามารบกวนดังภาพ 36 ซึ่งจะมีการปรับค่าดังนี้

ขนาดของสัญญาณรบกวน (Amplitude noise) หรือ A_n ให้มีค่า 1 โวลท์

ความถี่ของสัญญาณรบกวน (Frequency noise) หรือ f_n ให้มีค่า 100 เฮิรตซ์

ขนาดของสัญญาณ (Amplitude) $s(n)$ หรือ A_s ให้มีค่า 10 โวลท์

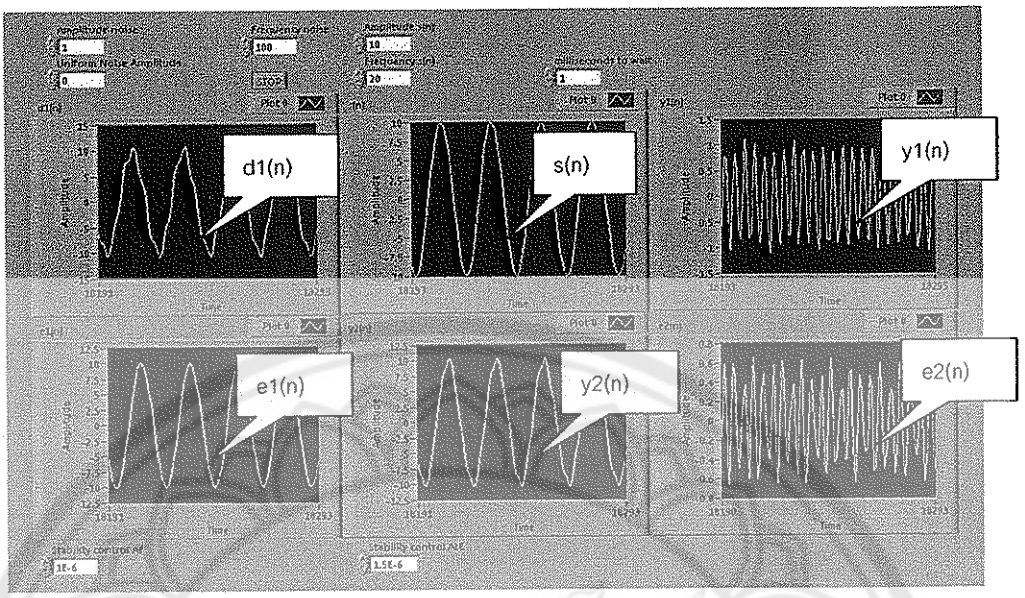
ขนาดของสัญญาณรบกวนรูปแบบขาว (Uniform Noise Amplitude) หรือ Uniform white noise ให้มีค่า 0 โวลท์

ความถี่ของสัญญาณ (Frequency) $s(n)$ หรือ f_s ให้มีค่า 20 เฮิรตซ์ (เหตุผลที่ต้องใช้ความถี่สัญญาณต่ำ ๆ ในการทดลองก็เพื่อให้ง่ายต่อการพิจารณา เพราะถ้าความถี่ใกล้สัญญาณรบกวนจะทำให้พิจารณายากมาก)

เวลาหน่วง (Milliseconds to wait) ให้มีค่า 1 มิลลิวินาที

ตัวควบคุมเสถียรภาพของระบบตัวกรองแบบปรับตัวเอง (Stability control AF) หรือ μ ให้มีค่า 0.000001

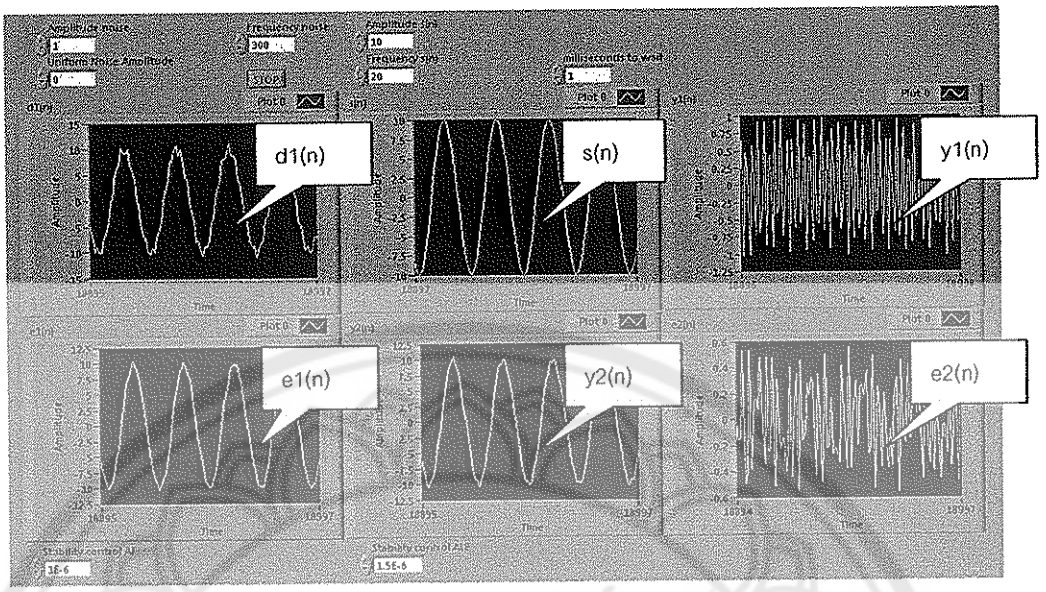
ตัวควบคุมเสถียรภาพของระบบตัวปรับความคมชัดของเส้นสัญญาณ (Stability control ALE) หรือ μ ให้มีค่า 0.000015



ภาพ 36 แสดงกราฟเมื่อปรับค่า $f_n = 100Hz$

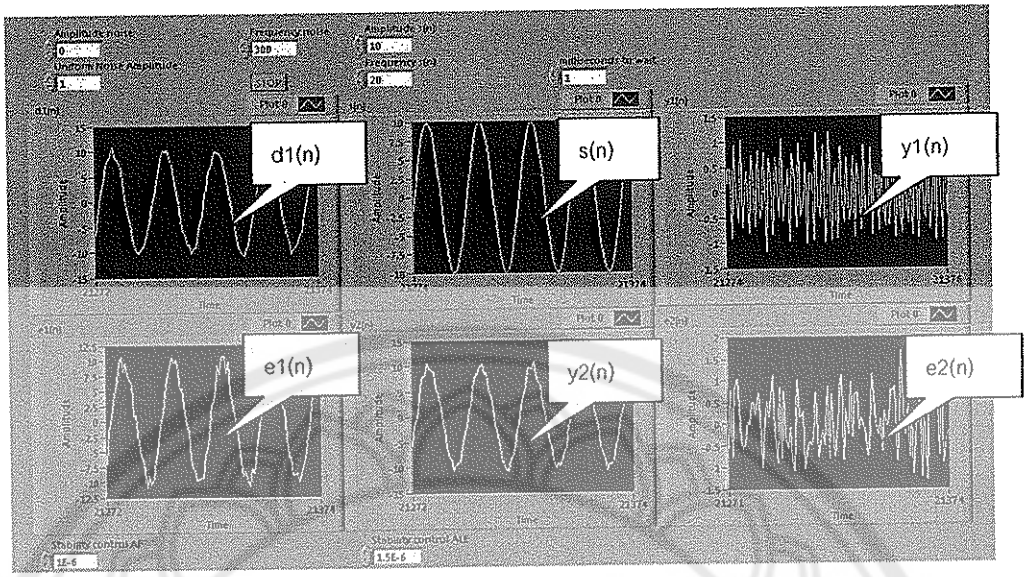
จากภาพ 36 จะเห็นได้ว่าอัลกอริธึมสามารถปรับตัวกำจัดสัญญาณรบกวนที่เป็นรูปคลื่นไซน์ที่ความถี่ 100 เฮิรตซ์ออกไปได้ จากกราฟสัญญาณขาออก $y2(n)$ ในภาพ 36 ปัญหาหนึ่งที่พบจากอัลกอริธึมนี้คือ การทำงานของมันช้าลงมีผลทำให้สัญญาณเกิดเวลาหน่วงมากขึ้น อาจทำให้อัตราการสุ่มเก็บสัญญาณของมันลดลงได้

จากนั้นทำการทดลองต่อ โดยการปรับความถี่สัญญาณที่เข้าไปรบกวน 300 เฮิรตซ์ ทำได้โดยการปรับค่าความถี่ของสัญญาณที่เข้าไปรบกวน (Frequency noise หรือ f_n) ให้มีค่า 300 เฮิรตซ์ แล้วสังเกตผลการทดลองดังภาพ 37 จะเห็นได้ว่า กราฟของสัญญาณขาออก $y2(n)$ เริ่มผิดเพี้ยนไปเล็กน้อย และเมื่อเพิ่มความถี่สัญญาณรบกวนสูงขึ้น พบว่ายังคงมีปัญหาบริเวณยอดคลื่นยังมีความผิดเพี้ยนอยู่ ยิ่งความถี่ของ $s(n)$ สูง ๆ จะเจอปัญหานี้มากขึ้น จนอัลกอริธึมไม่สามารถปรับตัวกำจัดออกได้



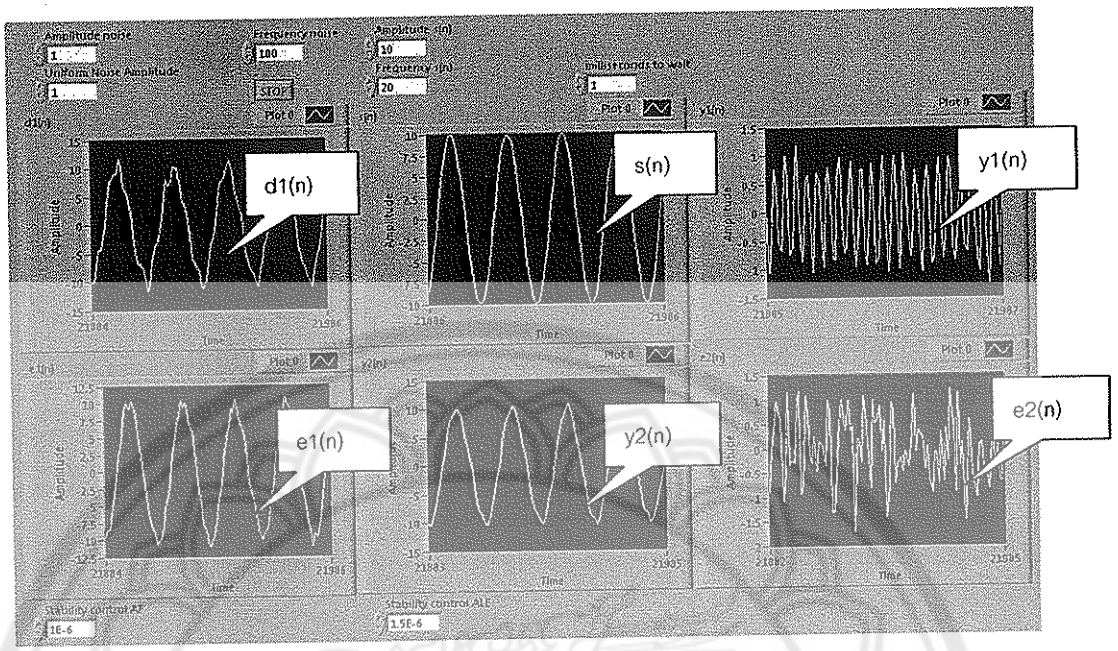
ภาพ 37 แสดงกราฟเมื่อปรับค่า $f_n = 300Hz$

จากนั้นทำการทดลองต่อ โดยทำการทดลองเพิ่มสัญญาณรบกวนรูปแบบขาว (Uniform noise amplitude) ให้มีค่าเป็น 1 โวลท์ ป้อนเข้าไปผสมกับสัญญาณข้อมูลของ $s(n)$ แต่ไม่มีการป้อนสัญญาณรบกวนรูปคลื่นไซน์ ดังภาพ 38 จะเห็นได้ว่า สัญญาณขาออก $y2(n)$ สามารถกำจัดได้บ้าง แต่ไม่ค่อยจะดีมากนักเมื่อเทียบกับการทดลองก่อนหน้านี้ ที่ใช้ตัวปรับความคมชัดของเส้นสัญญาณร่วมกับอัลกอริธึมกำลังสองเฉลี่ยน้อยสุด โดยเฉพาะยอดคลื่นมีการสูญเสียความคมชัดของสัญญาณไป



ภาพ 38 แสดงกราฟเมื่อปรับค่า $n(n)=0$ และสัญญาณรบกวนรูปแบบขาว = 1

จากนั้นทำการทดลองต่อ โดยทำการทดลองเพิ่มสัญญาณรบกวนรูปแบบขาว (Uniform noise amplitude) ให้มีค่าเป็น 1 โวลท์ และเพิ่มสัญญาณรบกวนรูปคลื่นไซน์ที่ความถี่ 100 เฮิรตซ์ ป้อนเข้าไปผสมกับสัญญาณข้อมูลของ $s(n)$ ดังภาพ 39 จะเห็นได้ว่า สัญญาณขาออก $y2(n)$ สามารถกำจัดสัญญาณรบกวนทั้งสองประเภทได้ แต่ยังพบว่าสัญญาณขาออก $y2(n)$ ยังคงผิดเพี้ยนจาก $s(n)$ ไปเล็กน้อย และที่ความถี่สูง ๆ พบว่ายังไม่สามารถทำงานกำจัดสัญญาณรบกวนออกได้หมด



ภาพ 39 แสดงกราฟเมื่อปรับค่า $f_n = 100Hz$ และสัญญาณรบกวนรูปแบบขาว = 1

สรุปผลการคัดเลือกอัลกอริธึมที่จะใช้ในเครื่องช่วยฟังแบบ 1 ไมโครโฟน

1. ตัวกรองแบบปรับตัวเอง : จากการทดลองที่ผ่านมาจะเห็นได้ว่าตัวกรองแบบปรับตัวเองสามารถปรับตัวกำจัดสัญญาณรบกวนที่มีคาบหรือความถี่ได้ แต่ยังพบปัญหาเรื่องการกำจัดสัญญาณรบกวนแบบไม่มีคาบหรือไม่มีความถี่ (Uniform white noise) ยังคงไม่สามารถปรับตัวกำจัดออกได้

ปัญหา : ยังคงไม่สามารถกำจัดสัญญาณรบกวนแบบไม่มีคาบ เช่น สัญญาณรบกวนรูปแบบขาวออกได้ และเมื่อความถี่ของสัญญาณรบกวนนั้นสูงขึ้น ตัวกรองแบบปรับตัวเองยังคงไม่สามารถปรับตัวและกำจัดสัญญาณรบกวนนั้นออกได้ และอาจจะมีความจำเป็นต้องใช้กับเครื่องช่วยฟังแบบ 2 ไมโครโฟนที่มีการแยกสัญญาณรบกวนออกจากเสียงพูดชัดเจน

2. ตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง : จากการทดลองที่ผ่านมาจะเห็นได้ว่า ตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเองสามารถปรับตัวกำจัดสัญญาณรบกวนที่ไม่มีคาบหรือไม่มีความถี่ได้ แต่ยังพบปัญหาเรื่องการกำจัดสัญญาณรบกวนแบบมีคาบหรือมีความถี่ยังคงไม่สามารถปรับตัวกำจัดออกได้ และสามารถใช้กับเครื่องช่วยฟังแบบ 1 ไมโครโฟน

ปัญหา : ยังคงไม่สามารถกำจัดสัญญาณรบกวนแบบมีคาบ เช่น สัญญาณรูปคลื่นไซน์ได้

3 ตัวกรองแบบปรับตัวเองทำงานร่วมกับตัวปรับความคมชัดของเส้นสัญญาณเสียงแบบปรับตัวเอง : จากการทดลองที่ผ่านมาจะเห็นได้ว่า ตัวกรองแบบปรับตัวเองทำงานร่วมกับตัวปรับความคมชัดของเส้นสัญญาณเสียงแบบปรับตัวเองสามารถปรับตัวกำจัดสัญญาณรบกวนที่มีคาบหรือมีความถี่ได้ และยังสามารถกำจัดสัญญาณที่ไม่มีคาบ เช่น สัญญาณรบกวนรูปแบบขาวได้

ปัญหา : ที่ความถี่สูงยังคงไม่สามารถปรับตัวกำจัดสัญญาณรูปคลื่นไซน์ออกได้ แต่พบอีกปัญหา ถ้าสัญญาณรบกวนมีเพียงอย่างใดอย่างหนึ่ง ระบบไม่สามารถกำจัดออกได้หมด และต้องใช้กับเครื่องช่วยฟังแบบ 2 ไมโครโฟนที่มีการแยกสัญญาณรบกวนออกจากเสียงพูดชัดเจน

ตารางราง 2 แสดงผลสรุปการกำจัดสัญญาณรบกวนเบื้องต้น

อัลกอริธึม	สัญญาณรบกวนรูปคลื่นไซน์	สัญญาณรบกวนรูปแบบขาว
ตัวกรองแบบปรับตัวเอง	ปรับตัวกำจัดได้	กำจัดไม่ได้
ตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง	กำจัดไม่ได้	ปรับตัวกำจัดได้
ตัวกรองแบบปรับตัวเองทำงานร่วมกับตัวปรับความคมชัดของเส้นสัญญาณเสียงแบบปรับตัวเอง	ปรับตัวกำจัดได้	ปรับตัวกำจัดได้

ดังนั้นจากตาราง 2 และผลการวิเคราะห์เบื้องต้น จะเห็นได้ว่า ตัวกรองแบบปรับตัวเองทำงานร่วมกับตัวปรับความคมชัดของเส้นสัญญาณเสียงแบบปรับตัวเองสามารถทำงานกำจัดสัญญาณรบกวนรูปคลื่นไซน์ได้ แต่สัญญาณรบกวนที่นำเข้าระบบจะต้องไม่มีสัญญาณข้อมูลปะปนมาด้วย หรือมีการแยกกันออกจากสัญญาณข้อมูลที่ชัดเจน และต้องใช้กับเครื่องช่วยฟังแบบ 2 ไมโครโฟน ซึ่งตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเองถึงแม้นว่าจะไม่สามารถกำจัดสัญญาณรบกวนรูปคลื่นไซน์ได้ แต่ก็สามารถกำจัดสัญญาณรบกวนแบบซ่า (Uniform white noise) ได้ และยังสามารถใช้กับเครื่องช่วยฟังแบบ 1 ไมโครโฟนซึ่งมีอยู่ทั่วไปได้อีกด้วย

ดังนั้นตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเองจะมีความเหมาะสมกับเครื่องช่วยฟังแบบ 1 ไมโครโฟน ซึ่งจะนำไปใช้จำลองฟังค์ชันการลดสัญญาณรบกวนในเครื่องช่วยฟังเพื่อจะทำการทดสอบระบบวัดประสิทธิภาพการลดสัญญาณรบกวนต่อไป

แบบจำลองการทำงานของเครื่องช่วยฟังแบบ 1 ไมโครโฟน

ในการทดสอบประสิทธิภาพของระบบวัดที่ออกแบบในการจำแนกคุณภาพการตัดเสียงรบกวนในเครื่องช่วยฟังนั้น แทนการใช้เครื่องช่วยฟังรุ่นต่าง ๆ มาทดสอบจริงซึ่งจะควบคุมตัวแปรได้ยาก การทดสอบในที่นี้ใช้การจำลองฟังก์ชั่นตัดเสียงรบกวนลงบนบอร์ดประมวลผลสัญญาณดิจิตอลแทน ซึ่งข้อดีของการดำเนินการดังกล่าวทำให้มีความยืดหยุ่นในการจำลองอัลกอริธึมการตัดเสียงรบกวนที่มีความซับซ้อนต่าง ๆ กัน ซึ่งในทางทฤษฎีทราบถึงความคาดหวังในเรื่องประสิทธิภาพได้ โดยนำมาทำการทดสอบในกระบวนการวัดจริงเพื่อยืนยันความแม่นยำของระบบ โดยในเบื้องต้นได้มีการจำลองลักษณะการตัดเสียงรบกวนของเครื่องช่วยฟังแบบ 1 ไมโครโฟน โดยใช้ตัวปรับความคมชัดของเส้นสัญญาณแบบปรับตัวเอง (Adaptive Line Enhancer) ร่วมกับอัลกอริธึมกำลังสองเฉลี่ยน้อยสุด (Least- Mean-Square Algorithm) ในการโปรแกรมลงบนบอร์ดประมวลผลสัญญาณดิจิตอล