

บทที่ 4

ทฤษฎีการประมวลผลสัญญาณ

ในบทนี้จะกล่าวถึงทฤษฎีการประมวลผลของสัญญาณกล้ำมเนื้อเพื่อที่จะนำไปใช้ ซึ่งในการประมวลผลจะเริ่มจากการหาตัวแทนสัญญาณเกี่ยวข้องกับงานวิจัย ซึ่งนำไปใช้ในส่วนของ การหาตัวแทนสัญญาณ (Feature Extraction) และการแยกแยะสัญญาณกล้ำมเนื้อ

4.1 การออกแบบตัวกรองแบบดิจิตอล (Digital Filter Design)

ตัวกรองแบบดิจิตอล (Digital Filter Design) สามารถแบ่งออกเป็น 2 ประเภทด้วยกันคือ ตัวกรองแบบ IIR (Infinite impulse response Filter) และ ตัวกรองแบบ FIR (Finite impulse response Filter) ตัวกรองแบบ FIR มีข้อดีคือมีความเป็นเชิงเส้นของเฟส และมีเสถียรภาพ แต่มีข้อเสียคือมีอันดับของตัวกรอง FIR มากกว่าของตัวกรอง IIR เมื่อต้องการผลตอบของขนาด (Magnitude Response) ของตัวกรองเหมือนกัน

4.1.1 ตัวกรองแบบ IIR

ในกรณีการออกแบบของตัวกรองแบบ IIR ที่นิยมในทางปฏิบัติ คือ การออกแบบตัวกรองแอนะล็อกก่อนซึ่งจะได้ฟังก์ชันถ่ายโอน เป็น $H_a(s)$ หลังจากนั้นแปลงไปเป็นฟังก์ชันถ่ายโอนของตัวกรองดิจิตอลเป็น $G(z)$

การแปลงดังกล่าวก็คือการแปลงจากระนาบ S (S -plane) ไปเป็นระนาบ z (z -plane) คุณสมบัติที่สำคัญของผลตอบเชิงความถี่ของแอนะล็อกที่ได้ออกแบบไว้ยังคงอยู่ ดังนั้นในการแปลงดังกล่าวจึงควรมีคุณสมบัติดังนี้

- (1) จุดบนแกนจินตภาพของระนาบ S จะย้าย (map) ไปบนวงกลมหนึ่งหน่วยของระนาบ z
- (2) ฟังก์ชันถ่ายโอนที่เสถียรของแอนะล็อกจะย้ายไปเป็นฟังก์ชันถ่ายโอนที่เสถียรของดิจิตอล

วิธีการแปลงดังกล่าวมีหลายวิธี แต่ที่นิยมใช้มี 2 วิธีคือ

- 1) Impulse Invariance method หลักการพื้นฐานของวิธีนี้ก็คือการหาฟังก์ชันถ่ายโอนของ IIR ที่มีผลตอบสนองต่ออิมพัลส์เหมือนกับผลตอบสนองต่ออิมพัลส์ของฟังก์ชันถ่ายโอนแบบแอนะล็อกที่ถูกสุ่มอย่างคงที่ เริ่มจากการหาฟังก์ชันถ่ายโอนแบบแอนะล็อกที่มีเสถียรภาพและ causal $H_a(s)$ ซึ่งจะได้ผลตอบสนองต่ออิมพัลส์ $h_a(t)$ โดยการทำการแปลงกลับลาปลาซ (inverse laplace transform) $H_a(s)$

$$h_a(t) = L^{-1}\{H_a(s)\} \quad (4.1)$$

เราต้องการหา unit sample response $g[n]$ ของฟังก์ชันถ่ายโอนแบบดิจิทัล $G(z)$ โดยการสุ่มค่า $h_a(t)$ ด้วยคาบเวลาในการสุ่ม T วินาที

$$g[n] = h_a(nT), \quad n = 0, 1, 2, \dots \quad (4.2)$$

เราสามารถแสดงความสัมพันธ์ระหว่างฟังก์ชันถ่ายโอนแบบดิจิทัล $G(z)$ กับฟังก์ชันถ่ายโอนแบบแอนะล็อก $H_a(s)$ ได้ดังนี้

$$\begin{aligned} G(z) &= Z\{g[n]\} = Z\{h_a(nT)\} \\ &= \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(s + j\frac{2\pi k}{T}\right) \Big|_{s=(1/T)\ln z} \end{aligned} \quad (4.3)$$

ซึ่งจะได้เป็นผลตอบเชิงความถี่ โดยการกำหนดให้ $z = e^{j\omega}$ และ $s = j\Omega$

$$G(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(j\frac{\omega}{T} + j\frac{2\pi k}{T}\right) \quad (4.4)$$

จากการทำการแปลงสมการที่(4.3) โดยใช้ $s=(1/T)\ln z$ ซึ่งสามารถเขียนได้ใหม่เป็น

$$z = e^{sT} \quad (4.5)$$

จาก $s = \sigma_0 + j\Omega_0$

$$z = re^{j\omega} = e^{\sigma_0 T} e^{j\Omega_0 T} \quad (4.6)$$

จะได้ว่า

$$|z| = r = e^{\sigma_0 T} \quad (4.7)$$

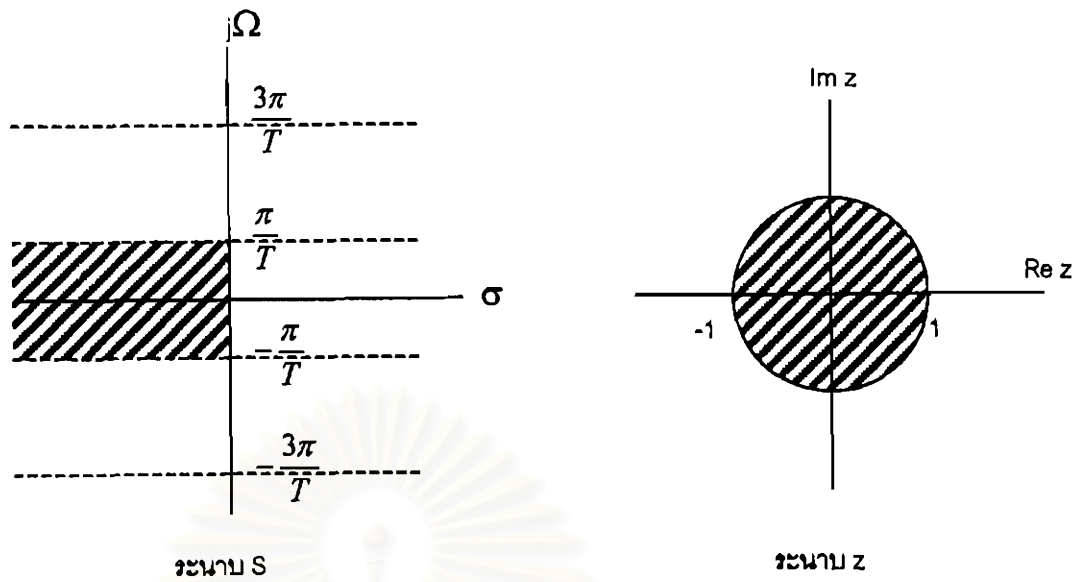
ดังนั้นจุดบนแกน $j\Omega$ ในระนาบ s สามารถหาได้โดยให้ $\sigma_0 = 0$ เมื่อแทนค่าลงในสมการที่ (4.7) จะได้ว่าจุดเหล่านี้จะถูกย้ายไปเป็นจุดบนวงกลมหนึ่งหน่วยในระนาบ z ($|z|=1$) เช่นเดียวกับจุดที่อยู่ฝั่งซ้ายของระนาบ s ซึ่งกำหนดโดย $\sigma_0 < 0$ โดยใช้สมการที่(4.7) จุดเหล่านี้ก็จะถูกย้ายไปเป็นจุดในวงกลมหนึ่งหน่วยในระนาบ z และโดยใช้สมการเดียวกันนี้ก็จะได้ว่าจุดบนฝั่งขวาของระนาบ s ก็จะถูกย้ายไปเป็นจุดนอกวงกลมหนึ่งหน่วยของ ระนาบ z ซึ่งเป็นไปตามคุณสมบัติ 2 ข้อที่ได้กล่าวไว้แล้ว อย่างไรก็ตามการย้ายจุดดังกล่าวเป็นการย้ายจุดบนระนาบ s

$$s = \sigma_0 + j\Omega_0 + j\frac{2\pi k}{T}, \quad k=0, \pm 1, \pm 2, \dots \quad (4.8)$$

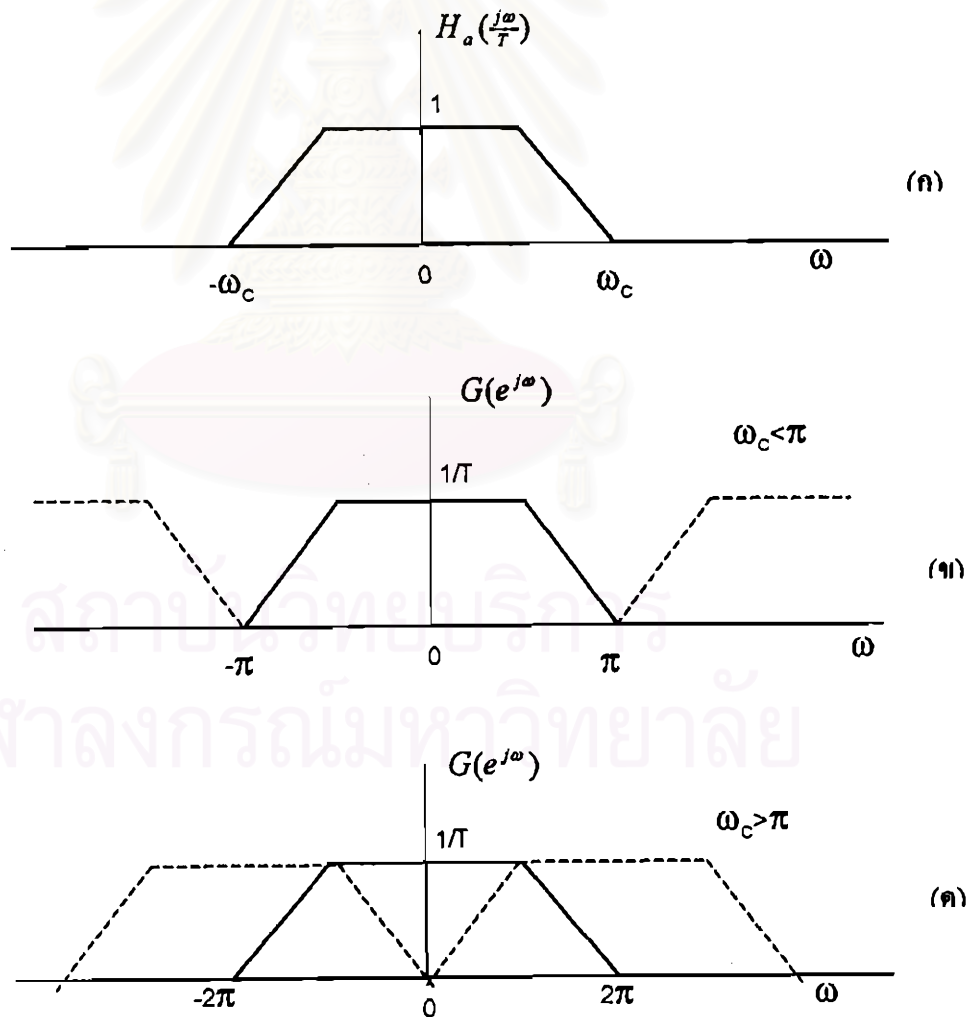
ไปเป็นจุดบน ระนาบ z

$$e^{\sigma_0 T} e^{j(\Omega_0 + \frac{2\pi k}{T})T} = e^{\sigma_0 T} e^{j\Omega_0 T} \quad (4.9)$$

รูปที่ 4.1 แสดงการแปลงโดยใช้วิธี impulse invariance method จากรูป แถบความกว้าง $2\pi/T$ ในระนาบ s ในช่วง $-\pi/T \leq \Omega \leq \pi/T$ จะถูกย้ายไปเป็นบริเวณ ระนาบ z ทั้งหมด เช่นเดียวกับ $2\pi/T$ ในช่วง $\pi/T \leq \Omega \leq 3\pi/T$ ก็ ย้าย ไปใน ระนาบ z ทั้งหมด และแถบ $2\pi/T$ ในช่วง $-\pi/T \leq \Omega \leq -3\pi/T$ ดังนั้นการ ย้าย แบบนี้จึงเป็นแบบ many to one



รูปที่ 4.1 Impulse invariance mapping



รูปที่ 4.2 ความสัมพันธ์ระหว่างผลตอบสนองเชิงความถี่ของตัวกรองแอนะล็อกกับตัวกรองดิจิทัล

จากการตรวจสอบผลของการ ชำ ในแง่ของ ผลตอบเชิงความถี่ จากสมการ(4.4) เราสังเกต ได้ว่าผลตอบเชิงความถี่ของฟังก์ชันโอนย้ายแบบดิซครีตก็คือผลรวมของผลตอบเชิงความถี่ของ ฟังก์ชันถ่ายโอนแบบแอนะล็อกตัวเดิมกับตัวมันเองที่ถูกเลื่อนออกไป $\pm 2\pi k/T$ แล้วทำการสเกล ด้วย $1/T$ จากสมการ(4.4) และจากทฤษฎีสุ่ม หากกำหนดให้ ผลตอบเชิงความถี่แบบแอนะล็อก $H_a(j\Omega)$ เป็นแถบความถี่จำกัด (bandlimited) ดังนี้คือ

$$H_a(j\Omega) = 0 \quad \text{เมื่อ } |\Omega| \geq \frac{\pi}{T}, \quad (4.10)$$

ดังนั้นจะได้

$$G(e^{j\omega}) = \frac{1}{T} H_a(j\frac{\omega}{T}), \quad \text{เมื่อ } |\omega| \leq \pi \quad (4.11)$$

และจะไม่เกิดการเคลือบแฝง แต่ถ้าสมการ(4.10) ไม่เป็นจริงจะเกิดการเคลือบแฝง รูปที่ 4.2(ข) แสดงการไม่เกิดการเคลือบแฝง และรูปที่ 4.2 (ค) การเกิดการเคลือบแฝง

2) Bilinear transformation

ในวิธีของ impulse invariance mapping แถบในระนาบ S ระหว่างค่า $\Omega = k\pi/T$ กับ $\Omega = (k+2)\pi/T$ จะถูกย้ายไปเป็นระนาบ z ซึ่งเป็นผลให้เกิดการเคลือบแฝง (aliasing) เพื่อที่จะหลีกเลี่ยง จึงต้องการแปลงด้วยการย้ายแบบหนึ่งต่อหนึ่ง (one to one mapping) ซึ่งได้แก่ bilinear transform การแปลงเป็นไปตามสมการดังนี้

$$s = \frac{2}{T} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \quad (4.12)$$

โดยใช้ความสัมพันธ์ตามสมการ(4.12) แปลงฟังก์ชันถ่ายโอนแบบแอนะล็อก $H_a(s)$ ไปเป็นฟังก์ชันถ่ายโอนแบบดิซครีต $G(z)$ ดังในสมการที่(4.13)

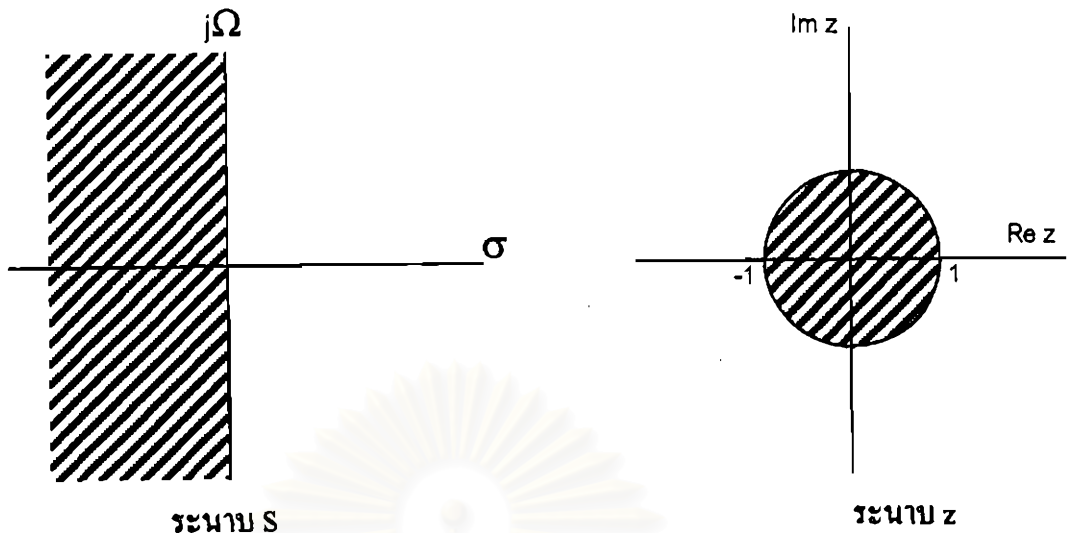
$$G(z) = H_a(s) \Big|_{s=\frac{2}{T} \left(\frac{1-z^{-1}}{1+z^{-1}} \right)} \quad (4.13)$$

bilinear transformation ได้มาจากการทำการอินทิเกรตเชิงตัวเลข แบบ trapezoidal (trapezoidal numerical integration) กับสมการเชิงอนุพันธ์ ที่ได้มาจาก $H_a(s)$ จะได้สมการเชิงอนุพันธ์ที่ตรงกับ $G(z)$ พารามิเตอร์ T แสดง step size ของการอินทิเกรตเชิงตัวเลข (numerical integration) ดังนั้น พารามิเตอร์ T ไม่มีผลต่อ $G(z)$ และเพื่อความสะดวกจะเลือกให้ $T=2$ ดังนั้นจะได้ว่า

$$z = \frac{1+s}{1-s} \quad (4.14)$$

ลองตรวจสอบการ ชำ ของ แกนจินตภาพของ ระนาบ S ซึ่งก็คือ $s=j\Omega_0$,

$$z = \frac{1+j\Omega_0}{1-j\Omega_0} \quad (4.15)$$



รูปที่ 4.3 bilinear transformation mapping

ซึ่งมีขนาดเท่ากับหนึ่ง สรุปได้ว่า จุดบนแกนจินตภาพในระนาบ s จะย้ายไปเป็นจุดบนวงกลมหนึ่งหน่วยในระนาบ z ในกรณีทั่วไปจะได้ว่า $s = \sigma_0 + j\Omega_0$

$$z = \frac{1 + (\sigma_0 + j\Omega_0)}{1 - (\sigma_0 + j\Omega_0)} = \frac{(1 + \sigma_0) + j\Omega_0}{(1 - \sigma_0) - j\Omega_0} \quad (4.16)$$

ดังนั้น

$$|z|^2 = \frac{(1 + \sigma_0)^2 + (\Omega_0)^2}{(1 - \sigma_0)^2 + (\Omega_0)^2} \quad (4.17)$$

จากสมการสรุปได้ว่า ครึ่งซ้ายของระนาบ s จะย้ายไปเป็นจุดในวงกลมหนึ่งหน่วย และครึ่งขวาของระนาบ s ก็จะย้ายไปเป็นจุดนอกวงกลมหนึ่งหน่วยซึ่งการย้ายนี้เป็นการย้ายแบบหนึ่งต่อหนึ่ง ดังนั้นจึงไม่เกิดการเคลือบแฝงขึ้น

4.1.2 ตัวกรอง FIR

กรณีการออกแบบตัวกรอง FIR มี 2 แบบด้วยกันคือ

- 1) การตัดสัมประสิทธิ์อนุกรมฟูรีเยร์ (truncated fourier series)
- 2) Frequency sampling approach

ในที่นี้จะกล่าวแต่เฉพาะ truncated fourier series ซึ่งใช้หลักการออกแบบ Least Integral-Squared Error design โดยกำหนดให้ $H_d(e^{j\omega})$ เป็นผลตอบของตัวกรองที่ต้องการ ซึ่งสามารถขยายออกเป็นอนุกรมฟูรีเยร์ได้ดังนี้

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n} \quad (4.18)$$

โดยที่สัมประสิทธิ์ฟูรีเยร์ $h_d[n]$ ก็คือ ผลตอบสนองต่ออิมพัลส์

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega, \quad -\infty \leq n \leq \infty \quad (4.19)$$

กำหนดให้ $H_i(e^{j\omega})$ เป็นผลตอบของ FIR ที่เราต้องออกแบบ เพราะฉะนั้นเพื่อให้ผลตอบที่ใกล้เคียงกับกรณีอุดมคติมากที่สุด ซึ่งเราจะใช้ integral square error เป็นตัววัด

$$\phi = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H_i(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega \quad (4.20)$$

โดยที่

$$H_i(e^{j\omega}) = \sum_{n=-M}^M h_i[n] e^{-j\omega n} \quad (4.21)$$

$$\begin{aligned} \phi &= \sum_{n=-\infty}^{\infty} |h_i[n] - h_d[n]|^2 \\ &= \sum_{n=-M}^M |h_i[n] - h_d[n]|^2 + \sum_{n=-\infty}^{-M-1} h_d^2[n] + \sum_{n=M+1}^{\infty} h_d^2[n] \end{aligned} \quad (4.22)$$

จากสมการพบว่าค่าผิดพลาดจะน้อยที่สุดเมื่อ $h_i[n] = h_d[n]$ สำหรับ $-M \leq n \leq M$ และสามารถทำให้ตัวกรองเป็น causal โดยการเลื่อนไปทางขวาด้วยค่า M ที่พอเหมาะ สรุปลวิธีนี้ก็คือ กรณีอุดมคติที่ถูกตัดสัมพันธ์ผลตอบสนองต่ออิมพัลส์ออกไปนั่นเอง

ผลของการตัดสัมพันธ์ผลตอบสนองต่ออิมพัลส์ ทำให้เกิดการสั่น หรือแกว่งของขนาดการตอบสนอง (Magnitude response) เรียกผลนี้ว่าเกิดปรากฏการณ์ Gibbs Phenomenon ซึ่งขึ้นอยู่กับค่า M ที่เลือกในสมการ (4.22) ยิ่งเลือก M มากๆ จะทำให้เกิด ripple ที่ขึ้นทั้งในแถบผ่าน (pass band) และแถบหยุด (stop band) แต่ขนาดของความสูงของ ripple ที่ใหญ่ที่สุดจะไม่เปลี่ยนไปตามความยาวของตัวกรอง

เหตุผลของการเกิด Gibbs คือ window $w[n]$ ที่ใช้ตัดสัมพันธ์

$$h_i[n] = h_d[n] \cdot w[n] \quad (4.23)$$

จาก modulation theorem ได้ใช้ผลการแปลงฟูริเยร์ กับสมการที่(4.23) จะได้

$$H_i(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\phi}) \Psi(e^{j(\omega-\phi)}) d\phi \quad (4.24)$$

โดยที่ $H_i(e^{j\omega})$ และ $\Psi(e^{j\omega})$ เป็นผลการแปลงฟูริเยร์ของ $h_i[n]$ และ $w[n]$ ตามลำดับ ซึ่งก็คือการประสาน (convolution) กันทางความถี่นั่นเอง $H_i(e^{j\omega})$ คือผลตอบเชิงความถี่ของตัวกรองซึ่งเป็นแบบอุดมคติมีลักษณะเป็นสี่เหลี่ยม ดังนั้นคุณภาพของตัวกรองหรือลักษณะสมบัติของตัวกรองขึ้นอยู่กับ windows ที่เลือกใช้ เช่น ถ้าเลือก windows เป็นสี่เหลี่ยม ก็จะเกิด ripple ขึ้นทั้งในช่วง pass band และ stop band เนื่องจากผลตอบเชิงความถี่ของสี่เหลี่ยมเป็น Sinc function ซึ่งมีการแกว่งสูงมาก

window มี 2 ประเภทคือ

1) Fixed Window Function

ได้แก่ windows ที่มีความยาว $2M+1$ ดังนี้

$$\text{Hann: } w[n] = \frac{1}{2} \left[1 + \cos\left(\frac{2\pi n}{2M+1}\right) \right], \quad -M \leq n \leq M \quad (4.25)$$

$$\text{Hamming: } w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{2M+1}\right) \quad -M \leq n \leq M \quad (4.26)$$

$$\text{Black man: } w[n] = 0.42 + 0.5 \cos\left(\frac{2\pi n}{2M+1}\right) + 0.08 \cos\left(\frac{4\pi n}{2M+1}\right), \quad -M \leq n \leq M \quad (4.27)$$

2) Adjustable window Functions คือ window ที่สามารถปรับเปลี่ยนพารามิเตอร์ได้

2.1 Dolph-Chebyshev window ความยาว $2M+1$ กำหนดโดย

$$w[n] = \frac{1}{2M+1} \left[\frac{1}{\gamma} + 2 \sum_{k=1}^M T_k \left(\beta \cos \frac{k\pi}{2M+1} \right) \cos \frac{2nk\pi}{2M+1} \right], \quad -M \leq n \leq M \quad (4.28)$$

โดยที่ γ คือขนาดพูข้างสัมพัทธ์ (relative side lobe amplitude) มีค่าดังต่อไปนี้

$$\gamma = \frac{\text{amplitude of sidelobe}}{\text{main lobe amplitude}} \quad (4.29)$$

$$\beta = \cosh \left(\frac{1}{2M} \cosh^{-1} \frac{1}{\gamma} \right) \quad (4.30)$$

และ T_l เป็นโพลิโนเมียลของ chebyshev อันดับที่ l กำหนดโดย

$$T_l(x) = \begin{cases} \cos(l \cos^{-1} x), & \text{for } |x| \leq 1 \\ \cosh(l \cosh^{-1} x), & \text{for } |x| \geq 1 \end{cases} \quad (4.31)$$

2.2 kaiser window

$$w[n] = \frac{I_0 \left\{ \beta \sqrt{1 - (n/M)^2} \right\}}{I_0(\beta)}, \quad -M \leq n \leq M, \quad (4.32)$$

โดยที่ β เป็นพารามิเตอร์ที่สามารถปรับเปลี่ยนได้ และ $I_0(x)$ ดัดแปลงมาจากเบสเซลฟังก์ชันอันดับ 0 เป็นสมการดังนี้

$$I_0(u) = 1 + \sum_{r=1}^{\infty} \left[\frac{(u/2)^r}{r!} \right]^2 \quad (4.33)$$

จะเห็นได้ว่ามีค่าเป็นบวกเสมอ ในทางปฏิบัติการหาผลบวกแค่ 20 พจน์แรกก็เพียงพอจะแทนค่าดังกล่าวได้

พารามิเตอร์ β เป็นตัวควบคุมค่าคutoffที่ต่ำที่สุด ในช่วงแถบหยุดซึ่งคำนวณได้จาก

$$\beta = \begin{cases} 0.1102(\alpha_s - 8.7), & \text{for } \alpha_s > 50 \\ 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21), & \text{for } 21 \leq \alpha_s \leq 50 \\ 0 & \text{for } \alpha_s < 21 \end{cases} \quad (4.34)$$

ขนาดของ ความยาว N คำนวณได้จาก

$$N \approx \begin{cases} \frac{\alpha_s - 7.95}{14.36\Delta f} + 1, & \text{for } \alpha_s > 21 \\ \frac{0.9222}{\Delta f} + 1, & \text{for } \alpha_s \leq 21 \end{cases} \quad (4.35)$$

4.1.3 การออกแบบตัวกรองดิจิทัล IIR โดยใช้ MATLAB

ในการออกแบบตัวกรองดิจิทัลโดยใช้ MATLAB จะใช้ฟังก์ชันที่มีอยู่ใน Signal Processing Toolbox ซึ่งฟังก์ชันใน MATLAB จะใช้วิธีการออกแบบจากตัวกรอง แอนะล็อก ไปเป็นตัวกรองดิจิทัล โดยใช้ bilinear transformation ซึ่งได้แก่

$$[b,a] = \text{butter}(N,Wn)$$

$$[b,a] = \text{cheby1}(N,Rp,Wn)$$

$$[b,a] = \text{cheby2}(N,Rs,Wn)$$

$$[b,a] = \text{ellip}(N,Rp,Rs,Wn)$$

N คืออันดับของตัวกรอง Wn คือ normalized passband edge frequency. Wn มีค่า 0-1 โดยคิดว่ามี ความถี่ศูนย์กลางเท่ากับ 2 Hz Rp คือ Ripple Rs คือ minimum stopband attenuation ซึ่งฟังก์ชันนี้จะให้ค่าเป็นสัมประสิทธิ์ของเวกเตอร์ b และเวกเตอร์ a ซึ่งเป็นพารามิเตอร์ของ ฟังก์ชันถ่ายโอน

$$G(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(N+1)z^{-N}}{1 + a(2)z^{-1} + \dots + a(N+1)z^{-N}} \quad (4.36)$$

4.1.4 FIR Digital Filter Design using Matlab

ใน MATLAB มีฟังก์ชันให้เลือกในการออกแบบ 2 วิธีคือ แบบ Parks-McClellan algorithm และวิธี windows Fourier series method ซึ่งในงานวิจัยนี้ใช้วิธี windowed Fourier series method ในการและการออกแบบ window ใน MATLAB มีฟังก์ชันให้ดังนี้

w=blackman(N) w=hamming(N) w=hanning(N)

w=chebwin(N,Rs) w=kaiser(N,beta)

N คือ ความยาวของตัวกรองเป็นเลขที่ beta ก็คือค่า β ในสมการ(4.32)

windows ที่ได้จะนำไปใช้คำนวณตัวกรอง FIR ต่อไปซึ่งมีฟังก์ชันดังนี้

b=fir1(L,Wn>window)

L คืออันดับของตัวกรองเป็นเลขคู่ มีค่าน้อยกว่า N อยู่ 1, Wn คือ normalized passband edge frequency. Wn มีค่า 0-1

4.2 Discrete-Time Fourier Transform

Discrete-Time Fourier Transform ก็คือผลการแปลงฟูริเยร์ (Fourier Transform) ของลำดับดิสครีตทางเวลา (discrete time sequence) $x[n]$ เป็นไปตามสมการดังต่อไปนี้

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}, \quad (4.37)$$

โดยที่ $x(n)$ คือ สัญญาณดิสครีตที่ถูกสุ่มเข้ามา

$X(e^{j\omega})$ คือ ผลการแปลงฟูริเยร์ของสัญญาณดิสครีต $x(n)$

ω คือ ตัวแปรทางความถี่

จะสังเกตได้ว่าสัญญาณในสมการ (4.35) มีความยาวของสัญญาณเป็นอนันต์ แต่ในทางปฏิบัติเป็นไปไม่ได้ ความยาวของสัญญาณต้องมีจำกัดถึงพจน์ที่ $x[n]$ โดยกำหนดให้ $0 \leq n \leq N-1$ และ $0 \leq \omega \leq 2\pi$ โดยที่ $\omega_k = 2\pi k/N$, $k = 0, 1, \dots, N-1$ จากสมการที่(4.35) จะได้ว่า

$$X[k] = X(e^{j\omega}) \Big|_{\omega=2\pi k/N} = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N} \quad k = 0, 1, \dots, N-1 \quad (4.38)$$

ซึ่งปกติจะกำหนดให้

$$W_N = e^{-j2\pi/N} \quad (4.39)$$

เขียนเป็นสมการได้ใหม่เป็น

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad k = 0, 1, \dots, N-1 \quad (4.40)$$

ใน MATLAB มีฟังก์ชันสำเร็จรูปในการคำนวณ DFT คือ

$$\text{fft}(x), \quad \text{fft}(x, N)$$

ฟังก์ชัน $\text{fft}(x)$ จะทำการคำนวณค่าผลการแปลงดิคคริตฟูรีเยร์อย่างรวดเร็ว (Fast Fourier Transform :FFT เป็นอัลกอริธึมอันหนึ่งในการหาค่าดิคคริตฟูรีเยร์ซึ่งใช้จำนวนครั้งในการคำนวณน้อย) ของเวกเตอร์ x ได้เวกเตอร์ที่มีความยาวเท่ากับความยาวของเวกเตอร์ x แต่ถ้าต้องการจะกำหนดความยาวของ DFT ของเวกเตอร์ x ให้ยาวเท่ากับ N ก็ให้เลือกใช้ฟังก์ชัน $\text{fft}(x, N)$ โดยที่ถ้าความยาวของเวกเตอร์ x มากกว่า N เวกเตอร์ x ก็จะถูกตัดให้เหลือ N แต่ถ้าน้อยกว่าก็จะเติมเลขศูนย์เข้าไป

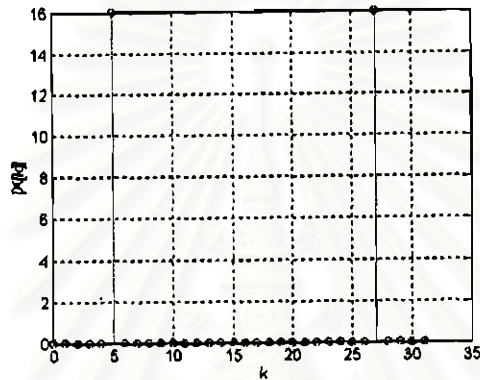
ใน MATLAB ใช้อัลกอริธึม radix-2 เมื่อความยาวของเวกเตอร์ x เป็นเลขยกกำลังสอง ซึ่งจะใช้เวลาคำนวณเร็วกว่ากรณีที่เวกเตอร์ x ไม่เป็นเลขยกกำลังสอง ซึ่งจะใช้วิธีคำนวณแบบ mixed-radix FFT

เนื่องจากเวกเตอร์ใน MATLAB มีดัชนีจาก 1 ถึง N แทนที่จะเป็น 0 ถึง $N-1$ ดังสมการที่ (4.41) ดังนี้

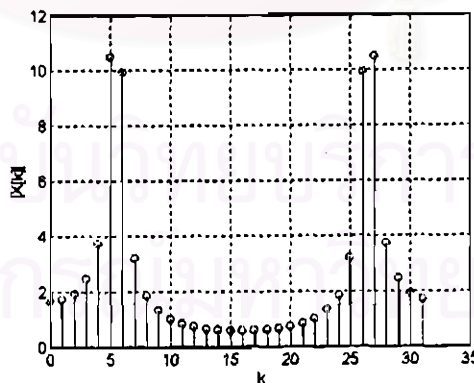
$$X[k] = \sum_{n=1}^N x[n]W_N^{(n-1)(k-1)}, \quad k = 1, 2, \dots, N \quad (4.41)$$

ในการคำนวณผลแปลงฟูรีเยร์อย่างรวดเร็วเป็นการแปลงข้อมูลทางเวลาไปเป็นข้อมูลทางความถี่นั้น ความละเอียดในการแสดงผลในเชิงความถี่จะขึ้นอยู่กับค่า N หรือจำนวนข้อมูล และความถี่สุ่ม โดยความละเอียด (resolution) คือความถี่สุ่มหารด้วยความยาวของข้อมูล ถ้าสัญญาณมีองค์ประกอบของความถี่ที่เป็นจำนวนเท่าของความละเอียด ก็จะได้สเปกตรัมของสัญญาณที่ถูกต้องคือการทำ FFT จะได้เวกเตอร์มีค่าแต่เฉพาะความถี่ที่เป็นความถี่ของสัญญาณเท่านั้นส่วนที่เหลือจะเป็น 0 หหมด ยกตัวอย่างเช่น สัญญาณไซน์ความถี่ 10 Hz ขนาดความยาวสัญญาณ 32 สุ่มที่ความถี่

64 Hz ทำการแปลงฟูรีเยอร์อย่างรวดเร็วจะได้เป็นเวกเตอร์ x ความยาว 32 ที่มีค่าเฉพาะที่ $x[5]$ และที่ $x[32-5=27]$ ที่เหลือจะมีค่าเป็นศูนย์หมดดังแสดงในรูปที่ 4.4 ที่ตำแหน่งที่ 5 จะแทนความถี่บวก $(64/32)*10 = 5$ ส่วนอีกตำแหน่งเป็นส่วนของความถี่ลบนั่นเอง ซึ่งโดยปกติในการนำไปวิเคราะห์ก็จะคิดแค่ครึ่งบวกครึ่งเดียว กรณีแรกนี้เป็นกรณีที่ความถี่ของสัญญาณเป็นจำนวนที่ลงตัวกับความถี่เซด แต่ถ้าเปลี่ยนเป็นสัญญาณชายน์ความถี่ 11 Hz เมื่อทำการแปลงฟูรีเยอร์อย่างรวดเร็วจะได้เวกเตอร์ที่ไม่มีค่าใดที่เป็นศูนย์ และจะมีมากที่ตำแหน่ง $x[5], x[6], x[26], x[27]$ เรียกว่าเกิดการ leakage ดังแสดงในรูปที่ 4.5



รูปที่ 4.4 ขนาด (magnitude) ของการแปลงฟูรีเยอร์ 32 จุด ของสัญญาณชายน์ความถี่ 10 Hz ด้วยอัตราสุ่ม 64 Hz



รูปที่ 4.5 ขนาด (magnitude) ของการแปลงฟูรีเยอร์ 32 จุด ของสัญญาณชายน์ความถี่ 11 Hz ด้วยอัตราสุ่ม 64 Hz

4.3 การแสดงตัวเลขในระบบไบนารี (Number Representation)

ในการเขียนโปรแกรมภาษาแอสเซมบลีเพื่อที่จะนำไปใช้งานนั้น จะต้องแสดงตัวเลขเป็นรูปแบบของเลขไบนารี ซึ่งจะแบ่งเป็น 2 ส่วนคือส่วนที่เป็นจำนวนเต็ม กับส่วนที่เป็นเศษส่วน ยกตัวอย่างเช่น เลข 11.625 สามารถแสดงเป็นเลขไบนารีได้ 1011.101 ซึ่งสามารถคำนวณกลับได้ดังนี้

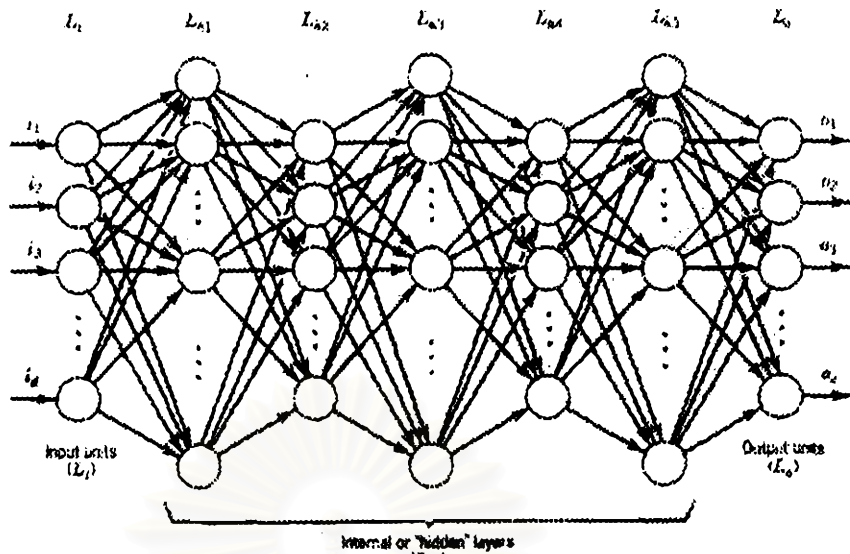
$$1011.101 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 + 1x2^{-1} + 0x2^{-2} + 1x2^{-3}$$

เราเรียกรูปแบบในตัวอย่างนี้ว่าเป็นรูปแบบ Q3 format (มีทศนิยม 3 บิต) ซึ่งในตัวอย่างข้างต้นนี้เป็นตัวเลขไบนารีที่แสดงตัวเลขที่เป็นบวกเท่านั้น ถ้าต้องการจะแสดงเลขที่เป็นลบ จะใช้บิตแรก(ซ้าย) เป็นบิตที่ใช้บอกเครื่องหมาย ถ้าเป็น 0 แสดงว่าเป็นเลขบวกบิตที่ตามมาก็สามารถนำมาคำนวณได้ตามตัวอย่างข้างต้น แต่ถ้าเป็น 1 แสดงว่าเป็นเลขลบ ในการคำนวณหาค่าให้ทำการหา 2's complement ก่อนแล้วค่อยใช้วิธีตามตัวอย่างข้างต้น ยกตัวอย่างเช่น 1111110011000000 เก็บในรูปแบบ Q6 หาค่าโดยอันดับแรกหา 2's complement จะได้ 0000001100100000 คิดเป็นเลขฐานสิบได้ 13.5

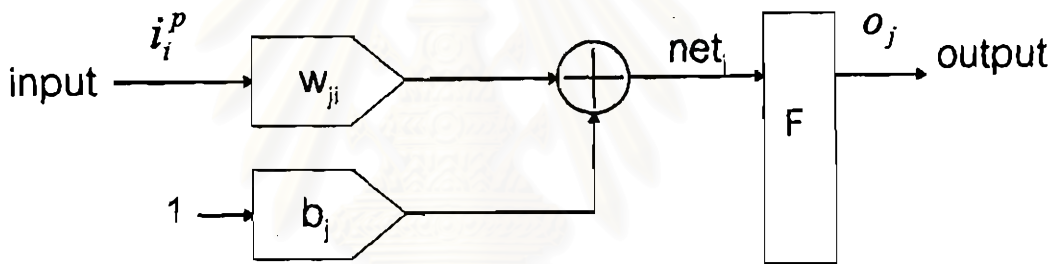
ในการแสดงตัวเลขโดยปกติตำแหน่งของจุดทศนิยมจะอยู่ตำแหน่งเดิม ดังนั้นในการบวกหรือการคูณเมื่อทำการคำนวณเสร็จแล้วจะต้องมีการเปลี่ยนตำแหน่งจุดทศนิยมใหม่

4.4 ระบบเครือข่ายประสาท (Neural Network) [25]

สถาปัตยกรรมของระบบเครือข่ายประสาทแสดงได้ดังรูปที่ 4.6. ซึ่งเป็นเครือข่ายแบบป้อนไปข้างหน้า (feedforward) แต่ละเส้นประสาทประกอบไปด้วยพารามิเตอร์ประจำโหนดคือ น้ำหนัก (weight: w), ไบแอส (bias: b) และฟังก์ชันถ่ายโอน (f) รูปที่ 4.7 แสดงแบบจำลองของระบบเครือข่ายประสาท 1 เลเยอร์ สมการคำนวณค่า output ของเส้นประสาทแสดงได้ดังสมการที่ (4.42) , (4.43) หลักการของระบบเครือข่ายประสาทก็คือพยายามหาเมตริกน้ำหนัก และ เมตริกไบแอสที่เหมาะสมเพื่อนำมาแทนระบบได้อย่างถูกต้อง การจะหาเมตริกดังกล่าวได้จะต้องมีชุดข้อมูลมาสอนระบบ เรียกว่าชุดฝึก (training set) ซึ่งข้อมูลเหล่านี้ประกอบไปด้วยค่าอินพุต และค่าเอาต์พุตของระบบ เพื่อที่ระบบจะได้ปรับเปลี่ยนพารามิเตอร์ให้ได้ค่าเอาต์พุต ที่ถูกต้อง ลักษณะในการเรียนรู้ข้อมูลนี้ได้หลายวิธี แต่วิธีที่ใช้ในงานวิจัยนี้เป็นแบบเคลื่อนกลับ (Backpropagation)



รูปที่ 4.6 โครงข่ายระบบประสาท แบบป้อนไปข้างหน้า (feed forward)



รูปที่ 4.7 แบบจำลองของโครงข่ายระบบประสาทจำนวน 1 ชั้น

$$net_j = \sum_i w_{ji} i_i + b_j \tag{4.42}$$

$$o_j^p = f_j(net_j^p) \tag{4.43}$$

การกำหนดค่าเริ่มต้น

ค่าน้ำหนัก และ ไบแอส เริ่มต้นก่อนที่จะทำการสอนระบบ กำหนดโดยการสุ่มค่าตัวเลขโดยมีค่าตั้งแต่ -1 ถึง 1

การเรียนรู้แบบเคลื่อนกลับ (backpropagation) ที่ใช้ในการปรับค่าน้ำหนัก และค่าไบแอสของโครงข่ายนั้นจะพยายามปรับระบบเพื่อให้ได้ค่าผลบวกกำลังสองของค่าผิดพลาดต่ำสุด โดยทำการเปลี่ยนค่าพารามิเตอร์ของโครงข่ายด้วยวิธีการลดระดับเกรเดียนต์ (gradient descent procedure) เพื่อให้ค่าผิดพลาดลดลงเร็วที่สุด ซึ่งมีสมการการปรับเปลี่ยนค่าเป็นดังต่อไปนี้

$$\Delta^p w_{ji} = \varepsilon \delta_j^p i_i^p \quad (4.44)$$

$$\delta_j^p = e f_j'(net_j^p) \quad (4.45)$$

โดยที่ ε คือ อัตราการเรียนรู้ (learning rate)

δ_j^p คือ เดลตาเวกเตอร์

e คือ ค่าผิดพลาดระหว่างเอาต์พุต ที่ได้จากเครือข่ายกับค่าที่ต้องการ

i_i^p คือ อินพุต

f_j' คือ อนุพันธ์ของฟังก์ชันถ่ายโอนเทียบกับ net_j^p

ในการสอนระบบเราจะคำนวณหาผลบวกกำลังสองของค่าผิดพลาดในแต่ละโหนด ถ้าได้น้อยกว่าค่าที่ตั้งเอาไว้ (error goal) การสอนก็จะสิ้นสุด แต่ถ้าได้ค่ามากกว่าที่ตั้งเอาไว้ ก็จะมีการปรับพารามิเตอร์ ซึ่งเป็นเสมือนกับการเรียนรู้ของระบบเครือข่ายประสาท โดยจะปรับจากเลเยอร์หลังสุดแล้วไล่ไปถึงเลเยอร์แรก ซึ่งเป็นที่มาของคำว่าเคลื่อนกลับ (backpropagation) ซึ่งขั้นตอนการปรับเป็นดังนี้

1. Presentation phase - ให้อินพุตที่อินพุตเลเยอร์ ของระบบเครือข่ายประสาท แล้วให้เครือข่ายคำนวณค่าเอาต์พุต ของเอาต์พุตเลเยอร์ออกมา
2. Check phase - คำนวณหาผลบวกกำลังสองของค่าผิดพลาดของค่าเอาต์พุต ที่คำนวณได้ในข้อ 1
3. Backpropagation phase - คำนวณหาเดลตาเวกเตอร์ เพื่อใช้ในการปรับเปลี่ยนน้ำหนัก และไบแอสของเลเยอร์ข้างหน้า ทำย้อนไปจนถึงอินพุตเลเยอร์
4. Learning phase - ปรับเปลี่ยนค่าน้ำหนัก และไบแอส แล้วกลับไปทำข้อ 1 ใหม่

ความหมายของตัวแปรต่างๆเป็นดังต่อไปนี้

อัตราการเรียนรู้ (learning rate) เป็นตัวบอกขนาดที่จะปรับเปลี่ยนน้ำหนัก, ไบแอส ในแต่ละครั้ง การกำหนดให้ค่าอัตราการเรียนรู้ ต่ำๆ จะทำให้ต้องใช้เวลาในการสอนระบบนาน แต่รับประกันว่าจะไม่กระโดดข้ามหลุมของพื้นผิวความผิดพลาด (error surface) ไป

โมเมนตัม (momentum) เป็นตัวช่วยลดความไวของระบบประสาทแบบเคลื่อนกลับ ในกรณีที่ไปติดอยู่ในหลุมของพื้นผิวความผิดพลาด ที่เป็นหลุมตื้น โดยปกติในโปรแกรม Matlab ตั้งค่าไว้ที่ 0.95 มีสมการดังต่อไปนี้

$$\Delta^p w_{ji}(n+1) = \varepsilon \delta_j^p i_i^p + \alpha \Delta^p w_{ji}(n) \quad (4.46)$$

พจน์ที่ 2 ในสมการ(4.46) คือส่วนที่เพิ่มเข้าซึ่งเป็นค่าเท่ากับ α ซึ่งเป็นตัวบอกความมากน้อยของโมเมนต์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย