สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง (Discrete-time Signal & System)

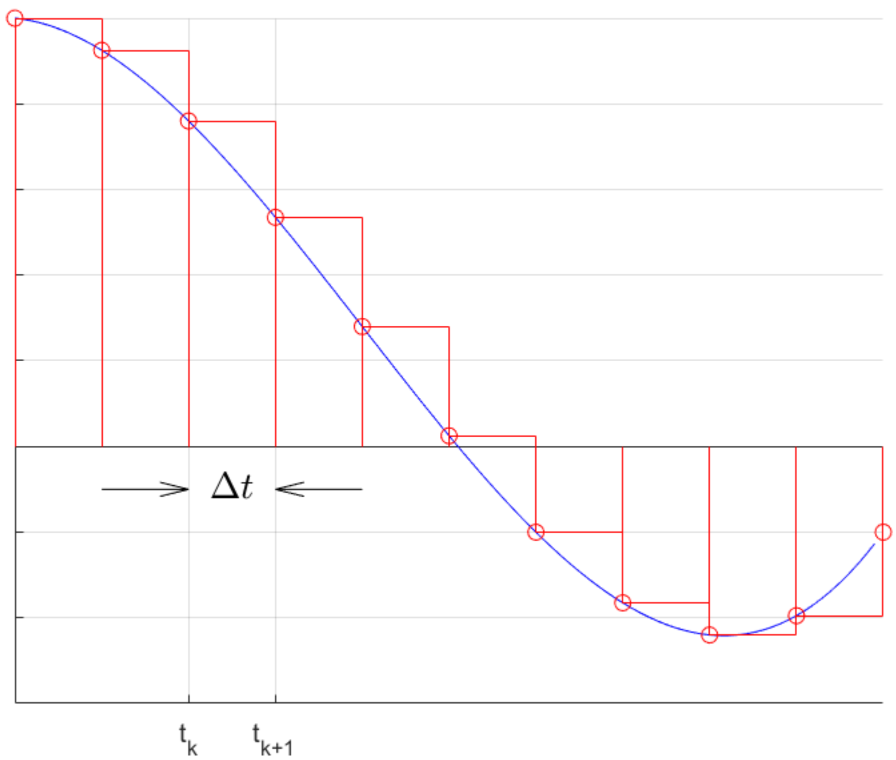
ตอนที่ 1 : การสุ่มตัวอย่าง (Signal Sampling)

คำว่า สุ่ม ในกรณีนี้ไม่ได้มีความหมายในเชิงสุ่มมั่ว การสุ่มตัวอย่างสัญญาณคือการอ่านสัญญาณแบบเวลาต่อเนื่อง ณ เวลาที่เป็นช่วงๆ เพื่อที่สังเคราะห์สัญญาณแบบเวลาไม่ต่อเนื่อง เราจะกำหนดให้จำนวนครั้งของการสุ่มสัญญาณเท่ากับ N (number of samples)

เพื่อความสม่ำเสมอในการวิเคราะห์และการสังเคราะห์ระบบต่างๆ **การสุ่มตัวอย่างจำเป็นต้องทำเป็นช่วงเวลาที่เท่าๆกัน (fixed time interval)** เราจะเรียกว่าช่วงระยะเวลาระหว่างการสุ่มว่า คาบการสุ่ม Δt (sampling period, sample time) หากเราทำการสุ่ม ณ เวลาที่ t_k และ เวลาถัดไปที่สุ่มคือ t_{k+1} ผลต่างของเวลาทั้งสองคือคาบการสุ่ม หรือเขียนเป็นสมการได้ดังต่อไปนี้

$$t_{k+1} - t_k = \Delta t \, \forall \, k \in \{1, \cdots, N\}$$

```
clf;
ax = axes;
hold(ax,'on')
grid(ax,'on')
t_max = 10;
N = 100;
t = linspace(0,t_max-t_max/N,N)';
f = @(t)(0.25*t.^3-3*t.^2-t+50)/10;
plot(ax,t,f(t),'b');
f_s = 1;
dt = 1/f_s;
n = (0:dt:t_max)';
stairs(ax,n,f(n),'r')
stem(ax,n,f(n),'r')
xticks(ax,[2 3]);
xticklabels(ax,{'t_k','t_{k+1}'});
yticklabels(ax,'')
text(ax,2.25,-0.5,'$\Delta t$','FontSize',14,'Interpreter',"latex")
quiver(ax,1,-0.5,1,0,1,'Color','k','MaxHeadSize',2)
quiver(ax,4,-0.5,-1,0,1,'Color','k','MaxHeadSize',2)
```



หลายครั้ง ความถี่การสุ่ม f_s (sampling frequency) มักถูกใช้ในการอธิบายอัตราการสุ่มตัวอย่างของสัญญาณซึ่งมีค่าเป็นส่วนกลับของคาบการสุ่ม Δt หรือ

$$f_s = \frac{1}{\Delta t}$$

ตัวอย่างที่ 1 การสุ่มตัวอย่าง

กำหนดให้สัญญาณแบบเวลาต่อเนื่องเป็นดังต่อไปนี้

$$y(t) = \cos(2\pi t)$$

สัญญาณนี้มีความถี่ที่ $1\,[\mathrm{Hz}]$

เนื่องจากไมโครคอนโทรลเลอร์ไม่สามารถประมวลผลสัญญาณแบบเวลาต่อเนื่องได้ เราจำเป็นต้องสุ่มตัวอย่าง ซึ่งเราจะกำหนดให้ความถี่การสุ่มเป็น $1.2\,[\mathrm{Hz}]$ หรือมีคาบการสุ่มเป็น $\frac{5}{6}\,[\mathrm{s}]$

กำหนดให้ $y_s[n]$ เป็นสัญญาณแบบเวลาไม่ต่อเนื่อง และ n เป็นตัวแปรอิสระที่อธิบายรอบการคำนวณ เราสามารถเขียนสัญญาณ y_s ณ รอบการคำนวณที่ nให้เท่ากับสัญญาณ y ณ เวลาที่ $n \cdot \Delta t$ ซึ่งเขียนปืนสมการได้ดังนี้

$$y_s[n] = y(n \cdot \Delta t)$$
$$y_s[n] = \cos(2\pi n\Delta t) = \cos\left(\frac{5}{3}\pi n\right) = \cos\left(2\pi n - \frac{1}{3}\pi n\right)$$

เนื่องจาก n เป็นจำนวนเต็ม เราจึงสามารถใช้คุณสมบัติดังต่อไปนี้ได้

$$\cos(2\pi n + \theta) = \cos(\theta)$$

ซึ่งทำให้เราได้สัญญาณที่อยู่ในรูปของฟังก์ชั่นดังต่อไปนี้

$$y_s[n] = \cos\left(-\frac{1}{3}\pi n\right) = \cos\left(\frac{1}{3}\pi n\right)$$

ตัวอย่างที่ 2 : การสังเคราะห์สัญญาณจากค่าที่สุ่มตัวอย่าง

กำหนดให้สัญญาณแบบเวลาไม่ต่อเนื่องเป็นดังต่อไปนี้

$$y_s[n] = \cos\left(\frac{1}{3}\pi n\right)$$

โดยที่มีความถี่การสุ่มเป็น $1.2\ [\mathrm{Hz}]$ หรือคาบการสุ่มเป็น $\frac{5}{6}\ [\mathrm{s}]$

เราทำการจัดรูปเพื่อให้ฟังก์ชันเขียนอยู่ในรูปของ Δt

$$y_s[n] = \cos\left(\frac{2}{2}\cdot\frac{1}{3}\pi n\cdot\frac{5}{6}\cdot\frac{6}{5}\right) = \cos\left(2\pi\cdot\frac{1}{5}\cdot n\cdot\frac{5}{6}\right) = \cos\left(2\pi\cdot\frac{1}{5}\cdot n\cdot\Delta t\right)$$

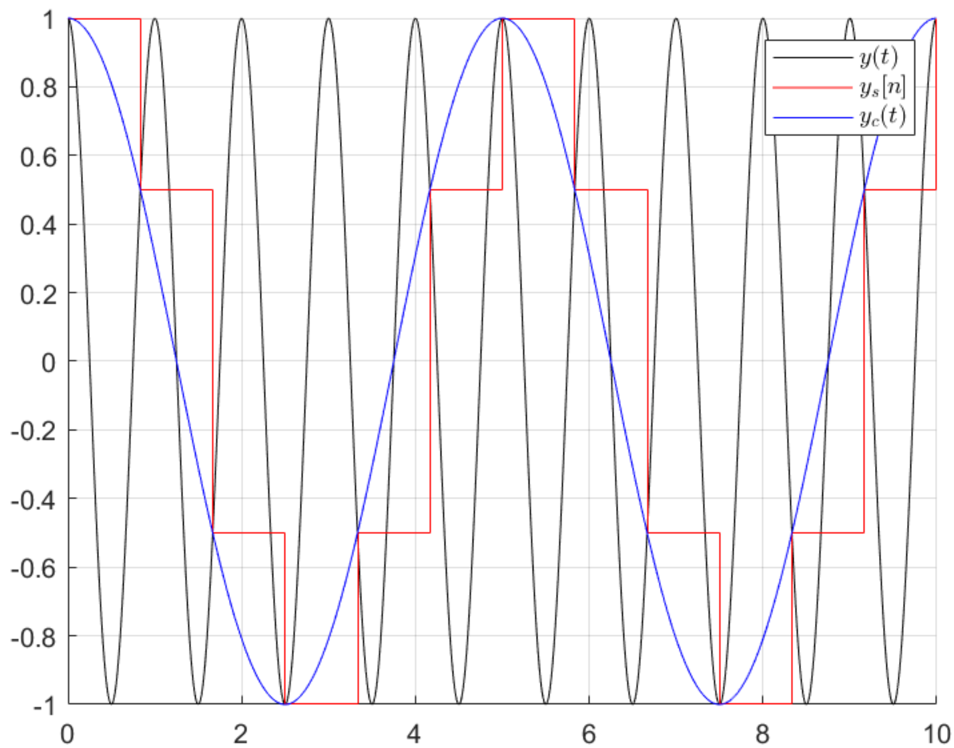
ในการสุ่มตัวอย่าง เราแทนค่า t ด้วย $n\Delta t$ ดังนั้นเมื่อเราแทนค่ากลับ เราจะได้สัญญาณแบบเวลาต่อเนื่อง y_c ดังนี้

$$y_c(t) = \cos\left(2\pi\cdot\frac{1}{5}\cdot t\right)$$

สัญญาณที่ได้สร้างขึ้นมาจากค่าที่สุ่มตัวอย่างมีความถี่เท่ากับ $\frac{1}{5}\ [\mathrm{Hz}]$

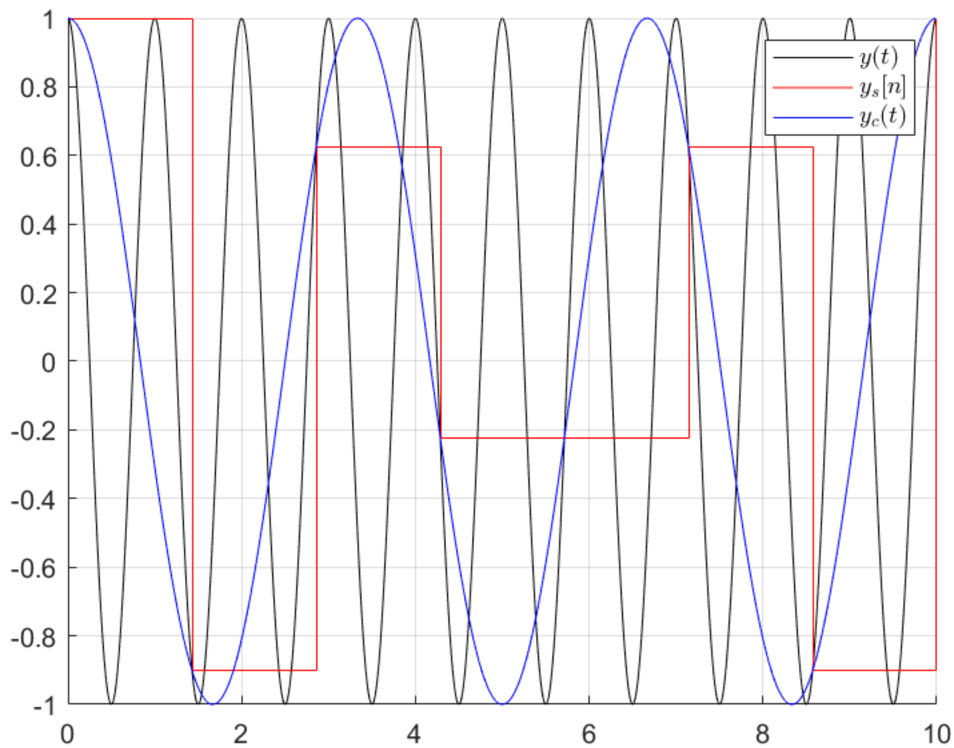
เราจะสังเกตได้ว่าสัญญาณเดิมจากตัวอย่างที่ 1 นั้นมีความถี่ที่ $1\ [\mathrm{Hz}]$ แต่การที่เราสุ่มตัวอย่างด้วยด้วยความถี่ไม่เหมาะสมทำให้สัญญาณที่ถูกสร้างใหม่มีความถี่เป็น $\frac{1}{5}[\mathrm{Hz}]$

```
clf;
ax = axes;
hold(ax,'on')
grid(ax,'on')
t_max = 10;
N = 1000;
t = linspace(0,t_max-t_max/N,N)';
y = cos(2*pi*t);
f_s = 1.2;
dt = 1/f_s;
n = 0:dt:t_max;
y_s = cos(2*pi*n);
y_c = cos(2*pi*0.2*t);
plot(ax,t,y,'k');
stairs(ax,n,y_s,'r');
plot(ax,t,y_c,'b');
legend({'$y(t)$','$y_s[n]$','$y_c(t)$'},'Interpreter',"latex")
```



เพื่อศึกษาช่วงของความถี่ที่เหมาะ เราเปลี่ยนแปลงความถี่การสุ่มและทำเป็นตารางออกมาคร่าวได้ดังนี้

```
f_s = 0.7;
idx = round(f_s/0.1);
f_c = [0 0 0.1 0.2 0 0.2 0.3 0.2 0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]';
clf;
ax = axes;
hold(ax,'on')
grid(ax,'on')
t_max = 10;
N = 1000;
t = linspace(0,t_max-t_max/N,N)';
y = cos(2*pi*t);
dt = 1/f_s;
n = 0:dt:t_max;
y_s = cos(2*pi*n);
y_c = cos(2*pi*f_c(idx)*t);
plot(ax,t,y,'k');
stairs(ax,n,y_s,'r');
plot(ax,t,y_c,'b');
legend({'$y(t)$','$y_s[n]$','$y_c(t)$'},'Interpreter',"latex")
```



เราจะสังเกตว่าเราไม่สามารถสังเคราะห์สัญญาณกลับมาที่ความถี่เดิมได้ถ้าความถี่การสุ่มมีค่าน้อยกว่า $2\ [\mathrm{Hz}]$ อีกความหมายหนึ่งคือการสุ่มตัวอย่างที่น้อยกว่า $2\ [\mathrm{Hz}]$ นั้นไม่สามารถใช้กับข้อมูลที่มีความถี่ $1\ [\mathrm{Hz}]$ ได้ **ปรากฏการดังกล่าวถูกเรียกว่า การซ้อนทับเชิงความถี่ (Aliasing)**

ทฤษฎีการสุ่มของไนควิสท์ (Nyquist Sampling Theorem) กล่าวว่า เพื่อที่จะอ่านสัญญาณ เราจำเป็นต้องสุ่มตัวอย่างโดยมีความถี่การสุ่มเป็นอย่างน้อย 2 เท่าของความถี่ของสัญญาณที่ต้องการจะสุ่ม

หรือในอีกความหมายหนึ่งคือ ความถี่การสุ่มจำเป็นต้องมากกว่าสองเท่าของ bandwidth ของสัญญาณ

$$f_s \geq f_B$$

ในทางปฏิบัติ เราสามารถเพิ่มความถี่การสุ่มเพื่อให้ครอบคลุม bandwidth ของระบบที่อาจจะไม่แน่นอนได้

เราสามารถทำการสุ่มตัวอย่างกับเซนเซอร์และระบบทางกายภาพได้ผ่านอุปกรณ์เช่น ไมโครคอนโทรลเลอร์ (microcontroller) ซึ่งการเขียนโปรแกรมของไมโครคอนโทรลเลอร์ที่ทำให้ทำการสุ่มตัวอย่างอย่างสม่ำเสมอและคงที่นั้นจำเป็นต้องใช้ timer ของไมโครคอนโทรลเลอร์ การสุ่มตัวอย่างนั้นไม่ควรถูกทำใน while loop ที่ไม่มีเวลาแน่นอน

สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง (Discrete-time Signal & System)

ตอนที่ 2 : สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง (Discrete-time Signal & System)

เมื่อเราทำการวิเคราะห์หรือสังเคราะห์สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง **เราจะกำหนดให้ตัวแปรอิสระ (indepedent variable) เป็นจำนวนเต็ม** n ที่บ่งบอกถึงรอบในการคำนวณ (iteration)

เราสามารถอธิบายระ**บบพลวัตแบบเวลาไม่ต่อเนื่องในรูปของสมการผลต่าง (difference equation)** ที่จัดเป็นสมการปริภูมิสถานะ (state-space) โดยมีรูปแบบดังต่อไปนี้

$$\mathbf{x}[n+1] = \mathbf{f}(\mathbf{x}[n], \mathbf{u}[n])$$
$$\mathbf{y}[n] = \mathbf{g}(\mathbf{x}[n], \mathbf{u}[n])$$

หากระบบเป็น LTI เราสามารถเขียนสมการแสดงปริภูมิสถานะได้ดังต่อไปนี้

$$\mathbf{x}[n+1] = \mathbf{A} \cdot \mathbf{x}[n] + \mathbf{B} \cdot \mathbf{u}[n]$$
$$\mathbf{y}[n] = \mathbf{C} \cdot \mathbf{x}[n] + \mathbf{D} \cdot \mathbf{u}[n]$$

ในเชิงของการนำสมการไปเขียนในโปรแกรม เราจะไม่สามารถเขียนสมการเชิงอนุพันธ์ได้เนื่องจากการทำงานของคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ทำงานเป็นทีละขั้นตอน ดังนั้น **การคำนวณแบบเวลาต่อเนื่อง (อนุพันธ์) จึงไม่สามารถเขียนตรงๆได้** เราทำได้ดีสุดคือการประมาณโดยใช้ระเบียบเชิงตัวเลข (numerical method)

ในทางกลับกัน **เราสามารถเขียนสมการเชิงผลต่างในโปรแกรมได้เนื่องจากการทำงานของคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์เป็นแบบเวลาไม่ต่อเนื่อง**

ดังนั้น เราจะสังเคราะห์และเขียนโปรแกรมระบบต่างๆเป็นเวลาไม่ต่อเนื่อง ซึ่งหลายครั้งจะเกิดมาจากการประมาณระบบแบบเวลาต่อเนื่อง

ตัวอย่างที่ 1 : การเฉลี่ยเคลื่อนที่ (Moving average)

กำหนดให้สัญญาณอินพุต $u[n]$ และสัญญาณเอ้าท์พุต $y[n]$ มีความสัมพันธ์ดังต่อไปนี้

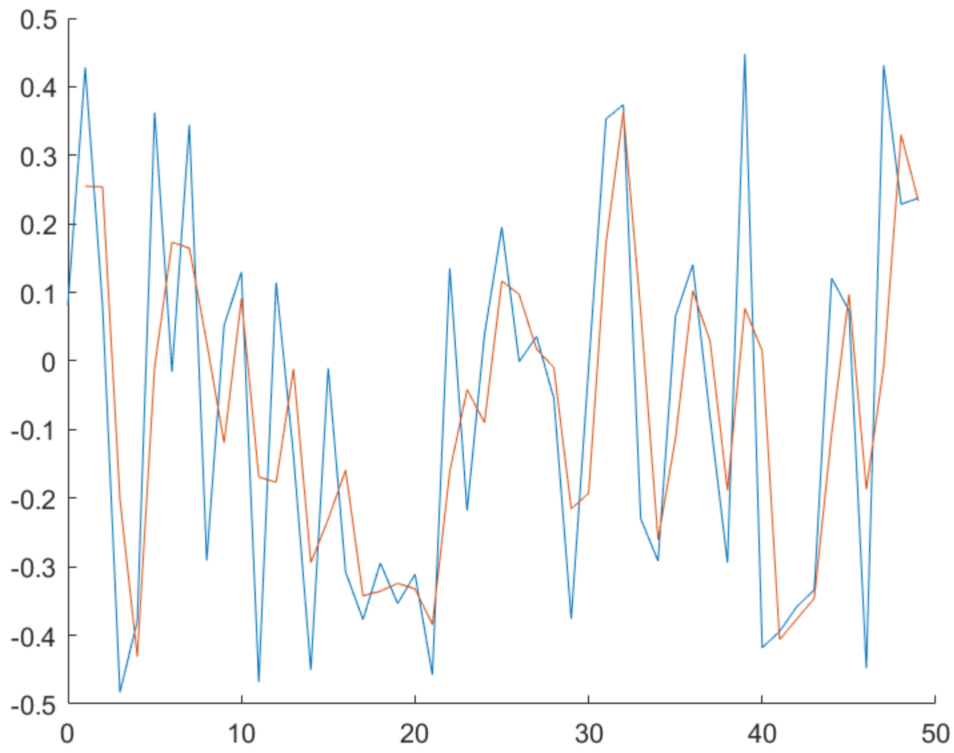
$$y[n] = \frac{1}{2}(x[n] + u[n])$$
$$x[n+1] = u[n]$$
$$x[0] = 0$$

โดยที่ $x[n]$ เป็นสถานะของระบบ

ความสัมพันธ์ที่ถูกกำหนดนับว่าเป็นระบบพลวัตแบบเวลาไม่ต่อเนื่องเพราะว่าความสัมพันธ์นั้นขึ้นอยู่อยู่กับตัวแปรสถานะด้วย

$y[n]$. $u[n]$, และ $x[n]$ นับว่าเป็นสัญญาณ แต่ความสัมพันธ์ทั้งหมดนับว่าเป็นระบบ

```
N = 50;
u = rand(N,1)-0.5;
y = 0.5*(u(1:end-1)+u(2:end));
n = linspace(0,N-1,N);
clf;
ax = axes;
hold(ax,'on')
plot(ax,n,u)
plot(ax,n(2:end),y)
```



เราสามารถนำความสัมพันธ์ดังกล่าวไปเขียนเป็นโปรแกรมของไมโครคอนโทรลเลอร์ได้ดังต่อไปนี้ (pseudo-code)

```
% เรียกแค่ครั้งแรก initialize
x = 0;

% เรียกทุกๆ dt วินาที call every dt seconds
u = read_sensor();
y = 0.5*(x+u);
x = u;
```

สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง (Discrete-time Signal & System)

ตอนที่ 3 : นิยามของการแปลง Z (Definition of Z-transform)

บางครั้งเราทำการออกแบบ วิเคราะห์ และ สังเคราะห์ระบบในโดเมนของความถี่ แต่กระบวนการเหล่านี้ล้วนแต่มีสมมุติฐานอย่างเดียวกันซึ่งคือระบบนั้นจะต้องเป็นระบบแบบเวลาต่อเนื่อง **หากเราต้องนำหลักการวิเคราะห์และสังเคราะห์มาใช้กับระบบแบบเวลาไม่ต่อเนื่อง เราจำเป็นต้องศึกษาการแปลง Z (Z-transformation)**

หากเราพิจารณาสัญญาณที่มีเวลาไม่ต่อเนื่อง $y_s[n]$ ในรูปของเวลาต่อเนื่อง $y(t)$ ที่ถูกสุ่มตัวอย่างด้วยคาบการสุ่ม Δt เราจะสามารถเขียนสมการได้ดังต่อไปนี้

$$y(t) = y[0]\delta(t) + y[1]\delta(t-\Delta t) + y[2]\delta(t-2\Delta t) + \cdots$$

เราสามารถแปลงสัญญาณให้อยู่ในโดเมนของลาปลาซได้ดังต่อไปนี้

$$Y(s) = y[0] + y[1]\mathrm{e}^{-s\cdot\Delta t} + y[2]\mathrm{e}^{-s\cdot 2\Delta t} + \cdots$$

เราจะกำหนดให้ ตัวแปรอิสระ $z = \mathrm{e}^{s\Delta t}$ เมื่อนำความสัมพันธ์นี้ไปแทนค่าในสัญญาณในโดเมนของลาปลาซ เราจะได้สมการดังต่อไปนี้

$$Y(z) = y[0] + y[1]z^{-1} + y[2]z^{-2} + \cdots$$

$$Y(z) = \sum_{n=0}^{\infty} y[n]z^{-n}, \;\; n \geq 0$$

ดังนั้นสัญญาณจะสามารถถูกเขียนอยู่ในรูปของตัวแปรอิสระ z การแปลงดังกล่าวถูกเรียกว่า การแปลง Z (Z-transform) เรานิยามการแปลง Z ได้ดังต่อไปนี้

$$\mathbf{Z}\{y[n]\} = Y(z) = \sum_{n=0}^{\infty} y[n]z^{-n}$$

การแปลง Z สามารถถูกมองเป็นการแปลงลาปลาซสำหรับระบบและสัญญาณแบบไม่ต่อเนื่องได้

ตัวอย่างที่ 1 : $y[n] = \theta[n]$

กำหนดให้สัญญาณ $y[n]$ เป็นฟังก์ชันขั้นเฮวิไซด์ (Heavidise's step function) เราสามารถแปลงฟังก์ชันดังกล่าวโดยใช้นิยามของการแปลง Z ได้ดังต่อไปนี้

$$\mathbf{Z}\{y[n]\} = Y(z) = \sum_{n=0}^{\infty} \theta[n]z^{-n}$$

$$\mathbf{Z}\{y[n]\} = \sum_{n=0}^{\infty} z^{-n} = \sum_{n=0}^{\infty} \left(\frac{1}{z}\right)^{n} = \frac{1}{1-\frac{1}{z}} = \frac{1}{1-z^{-1}}$$

สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง (Discrete-time Signal & System)

ตอนที่ 4 : สมบัติของการแปลง Z (Proterties of Z-transform)

หนึ่งในสมบัติของการแปลง Z คือ**สมบัติเชิงเส้น**ซึ่งสามารถเขียนเป็นสมการได้ดังต่อไปนี้

$$\boldsymbol{Z}\{\alpha \cdot x[n] + \beta \cdot y[n]\} = \alpha \cdot X(z) + \beta \cdot Y(z)$$

อีกหนึ่งสมบัติการวิเคราะห์ต้องใช้บ่อยครั้งคือ**การทำให้สัญญาณล่าช้า** หากเรากำหนดให้สัญญาณ $x[n]$ ถูกทำให้ล่าช้า (delay) ไป k รอบการคำนวณ $y[n-k]$ เราสามารถคำนวณการการแปลง Z ของสัญญาณได้ดังต่อไปนี้

$$\boldsymbol{Z}\{y[n]\} = Y(z) = \sum_{n=0}^{\infty} y[n]z^{-n}$$

$$\boldsymbol{Z}\{y[n-k]\} = \sum_{n=0}^{\infty} y[n-k]z^{-n} = \sum_{n=0}^{k-1} y[n-k]z^{-n} + \sum_{n=k}^{\infty} y[n-k]z^{-n} = \sum_{n=k}^{\infty} y[n-k]z^{-n}$$

$$\boldsymbol{Z}\{y[n-k]\} = \left(\sum_{n-k=0}^{\infty} y[n-k]z^{-(n-k)}\right)z^{-k} = z^{-k}Y(z)$$

เราสรุปได้ว่า ไม่ว่าสัญญาณจะเป็นลักษณะไหน **หากสัญญาณนั้นโดนทำให้ล่าช้าไป k รอบการคำนวณ เราสามารถหาการแปลง Z โดยนำ z^{-k} ไปคูณกับการแปลง Z ของสัญญาณที่ยังไม่ได้โดนทำให้ล่าช้า**

ตัวอย่างที่ 1

กำหนดให้ระบบคำนวณค่าเฉลี่ยเคลื่อนที่เป็นดังต่อไปนี้

$$y[n] = \frac{1}{2}(x[n] + u[n])$$
$$x[n+1] = u[n]$$
$$x[0] = 0$$

เราสามารถหาการแปลง Z ได้ดังต่อไปนี้

$$\boldsymbol{Z}(y[n]) = \boldsymbol{Z}\left(\frac{1}{2}(x[n] + u[n])\right) = \frac{1}{2}\boldsymbol{Z}(x[n]) + \frac{1}{2}\boldsymbol{Z}(u[n])$$
$$\boldsymbol{Z}(x[n+1]) = z^{k} \cdot \boldsymbol{Z}(x[n]) = \boldsymbol{Z}(u[n])$$
$$\boldsymbol{Z}(x[n]) = z^{-k}\boldsymbol{Z}(u[n])$$
$$Y(z) = \frac{1}{2}z^{-k}\boldsymbol{Z}(u[n]) + \frac{1}{2}\boldsymbol{Z}(u[n]) = \frac{1}{2}(z^{-k} + 1) \cdot U(z)$$

เราสรุปได้ว่า การแปลง Z ของสัญญาณเอ้าท์พุต $Y(z)$ ขึ้นอยู่กับการแปลง Z ของสัญญาณอินพุตในรูปแบบดังต่อไปนี้

$$Y(z) = \frac{1}{2}(z^{-k} + 1) \cdot U(z)$$

นอกจากนี้ เราสามารถหาฟังก์ชันถ่ายโอนในโดเมนของ Z ได้ดังต่อไปนี้อีกด้วย

$$G(z) = \frac{Y(z)}{U(z)} = \frac{1}{2}(z^{-k} + 1)$$

สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง (Discrete-time Signal & System)

ตอนที่ 5 : การทำให้เป็นเวลาไม่ต่อเนื่อง (Discretization)

ระบบทางกายภาพส่วนใหญ่เป็นระบบแบบเวลาต่อเนื่อง เราได้ทำการจำลอง วิเคราะห์ และออกแบบระบบที่มีเวลาแบบต่อเนื่อง แต่**ในโลกความเป็นจริง แพลตฟอร์มที่ใช้ในการคำนวณนั้นไม่สามารถทำให้เป็นแบบเวลาต่อเนื่องได้ เราจึงต้องทำการประมาณให้เป็นเวลาไม่ต่อเนื่อง (discretization)**

หากเราออกแบบระบบควบคุมแบบ LTI ที่อยู่ในโดเมนของเวลาดังต่อไปนี้

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(i(t)) = e(t)$$
$$c(t) = K_p \cdot e(t) + K_i \cdot i(t)$$

โดยที่

e คืออินพุต

c คือเอ้าท์พุต

K_p กับ K_i คือค่าคงที่

เราสามารถประมาณระบบนี้ได้โดยการประมาณอนุพันธ์ของ i

เราจะกำหนด $\Delta\, t$ เป็นระยะเวลาในแต่ละช่วงการคำนวณ ยิ่งช่วงการคำนวณน้อย การคำนวณก็จะละเอียดและแม่นยำมากขึ้น แต่การคำนวณที่มากขึ้นจะใช้พลังในการคำนวณมากขึ้นด้วยเช่นกัน

วิธีการแรกที่สามารถใช้ประมาณอนุพันธ์คือการหาผลต่างข้างหน้า (Forward Difference/ Euler's method) ซึ่งเขียนได้ดังต่อไปนี้

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(i(t_n)) \approx \frac{i(t_{n+1}) - i(t_n)}{\Delta t} = \frac{i[n+1] - i[n]}{\Delta t}$$

วิธีการที่สองคือการหาผลต่างข้างหลัง (Backward Difference) ซึ่งเขียนได้ดังต่อไปนี้

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(i(t_n)) \approx \frac{i(t_n) - i(t_{n-1})}{\Delta t} = \frac{i[n] - i[n-1]}{\Delta t}$$

วิธีการสุดท้ายคือการหาผลต่างกึ่งกลาง (Central Difference) ซึ่งเขียนได้ดังต่อไปนี้

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(i(t_n)) \approx \frac{i[n+1] - i[n-1]}{\Delta t}$$

หากเราใช้วิธีผลต่างข้างหน้า เราจะสามารถประมาณระบบควบคุมได้ดังต่อไปนี้

$$i[n+1] = i[n] + \Delta t \cdot e[n]$$
$$c[n] = K_p \cdot e[n] + K_i \cdot i[n]$$

นอกจากการประมาณในโดเมนของเวลา เราสามารถประมาณฟังก์ชันถ่ายโอนในโดเมนของลาปลาซได้เช่นกัน **การทำให้เป็นเวลาไม่ต่อเนื่องในโดเมนของลาปลาซคือการแปลงฟังก์ชันถ่ายโอนจากโดเมนของลาปลาซให้เป็นโดเมนของ Z** ซึ่งมีวิธีหลักทั้ง 3 วิธีดังนี้

วิธีแรกในการประมาณคือ การหาผลต่างข้างหน้า ซึ่งเราจะประมาณตัวแปร s ให้เป็นค่าดังต่อไปนี้

$$s \approx \frac{z-1}{\Delta t}$$

วิธีที่สองในการประเมาณคือ การหาผลต่างข้างหลัง ซึ่งเขียนความสัมพันธ์ได้ดังต่อไปนี้

$$s \approx \frac{1-z^{-1}}{\Delta t}$$

วิธีทีสุดท้ายคือ การแปลงแบบสองเส้น (bilinear transformation/Tustin' method) หาผลต่างข้างหลัง ซึ่งเขียนความสัมพันธ์ได้ดังต่อไปนี้

$$s \approx \frac{2}{\Delta t}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

ตัวอย่างที่ 1: ตัวกรอง low-pass อันดับที่หนึ่ง

กำหนดให้ตัวกรองถูกออกแบบดังต่อไปนี้

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_c}{s+\omega_c}$$

เราประมาณโดยใช้วิธีแปลงแบบสองเส้น $s \approx \frac{z-1}{\Delta t}$ ซึ่งทำให้เราได้ฟังก์ชันถ่ายโอนในโดเมนของ Z

$$G(z) = \frac{\omega_c}{\left(\frac{z-1}{\Delta t}\right)+\omega_c}$$

$$G(z) = \frac{\omega_c \Delta t}{z+(\omega_c \Delta t - 1)}$$

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\omega_c \Delta t}{z+(\omega_c \Delta t - 1)}$$

เมื่อเราได้ฟังก์ชันถ่ายโอนในโดเมนของ Z เราสามารถแปลงฟังก์ชันถ่ายโอนกลับไปยังโดเมนของเวลาแบบไม่ต่อเนื่องได้

เนื่องจากระบบเป็นอันดับที่หนึ่ง เราสามารถเขียนตัวกรองของเราในรูปแบบปริภูมิสถานะดังต่อไปนี้

$$x[n+1] = a \cdot x[n] + b \cdot u[n]$$
$$y[n] = c \cdot x[n] + d \cdot u[n]$$

โดยที่

$x \in \mathfrak{R}$ เป็นสถานะ

$u \in \mathfrak{R}$ เป็นสัญญาณอินพุตที่โดนกรอง

$y \in \mathfrak{R}$ เป็นสัญญาณเอ้าท์พุตที่ได้มาจากการกรอง

ส่วน a, b, c, และ d เป็นค่าคงที่ที่ต้องหา หรืออีกความหมายหนึ่งก็คือ เราจำเป็นต้องหาว่าระบบแบบเวลาไม่ต่อเนื่องใดที่ทำให้ฟังก์ชันถ่ายโอนมีลักษณะเดียวกันกับตัวกรองที่ออกแบบมา

$$
\begin{aligned}
x[n+1] &= a \cdot x[n] + b \cdot u[n] \\
y[n] &= c \cdot x[n] + d \cdot u[n]
\end{aligned}
\quad \Longrightarrow \quad
\frac{Y(z)}{U(z)} = \frac{\omega_c \Delta t}{z + (\omega_c \Delta t - 1)}
$$

เราจะสามารถแก้โจทย์นี้ได้หลายวิธี โดยปกติการคำนวณนี้สามารถใช้การแปลง Z ผกผัน (Inverse Z-transform) แต่การทำในรูปแบบนั้นอาจจะทำให้เขียนในรูปปริภูมิสถานะได้ลำบาก เราจึงใช้การแปลง Z จากปริภูมิสถานะไปเป็นฟังก์ชันถ่ายโอนแทนซึ่งเราจะสามารถเขียนเป็นสมการได้ดังต่อไปนี้

$$
\begin{aligned}
&\mathbf{Z}\{x[n+1]\} = \mathbf{Z}\{a \cdot x[n] + b \cdot u[n]\} \\
&z^1 \cdot X(z) = a \cdot X(z) + b \cdot U(z) \\
&X(z) = \frac{b}{z-a} \cdot U(z) \\
&\mathbf{Z}\{y[n]\} = \mathbf{Z}\{c \cdot x[n] + d \cdot u[n]\} \\
&Y(z) = c \cdot X(z) + d \cdot U(z) = \left(\frac{b \cdot c}{z-a} + d\right) \cdot U(z)
\end{aligned}
$$

จากโจทย์การแปลง เราได้เปลี่ยนรูปให้การเป็นโจทย์การหาสัมประสิทธิ์ของเศษส่วนดังต่อไปนี้

$$
\begin{aligned}
&\frac{Y(z)}{U(z)} = \frac{b \cdot c}{z-a} + d = \frac{\omega_c \Delta t}{z + (\omega_c \Delta t - 1)} \\
&a = 1 - \omega_c \Delta t \\
&b = \omega_c \Delta t \\
&c = 1 \\
&d = 0
\end{aligned}
$$

เราสามารถกำหนดให้

$$
\begin{aligned}
&a = 1 - \omega_c \Delta t \\
&b = \omega_c \Delta t \\
&c = 1 \\
&d = 0
\end{aligned}
$$

ดังนั้นเราจะได้รูปแบบปริภูมิสถานะดังต่อไปนี้

$$
\begin{aligned}
&x[n+1] = (1 - \omega_c \Delta t) \cdot x[n] + (\omega_c \Delta t) \cdot u[n] \\
&y[n] = x[n]
\end{aligned}
$$

ระบบที่ได้เป็นระบบแบบเวลาไม่ต่อเนื่องซึ่งมาจากระบบแบบเวลาต่อเนื่องดังต่อไปนี้

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\,t}(x(t)) = -\omega_c \cdot x(t) + \omega_c \cdot u(t) \\
&y(t) = x(t)
\end{aligned}
$$

เราจะเทียบระบบแบบเวลาต่อเนื่องกับระบบที่ถูกประมาณให้เวลาไม่ต่อเนื่อง ณ ช่วงการคำนวณที่แตกต่างกันไป

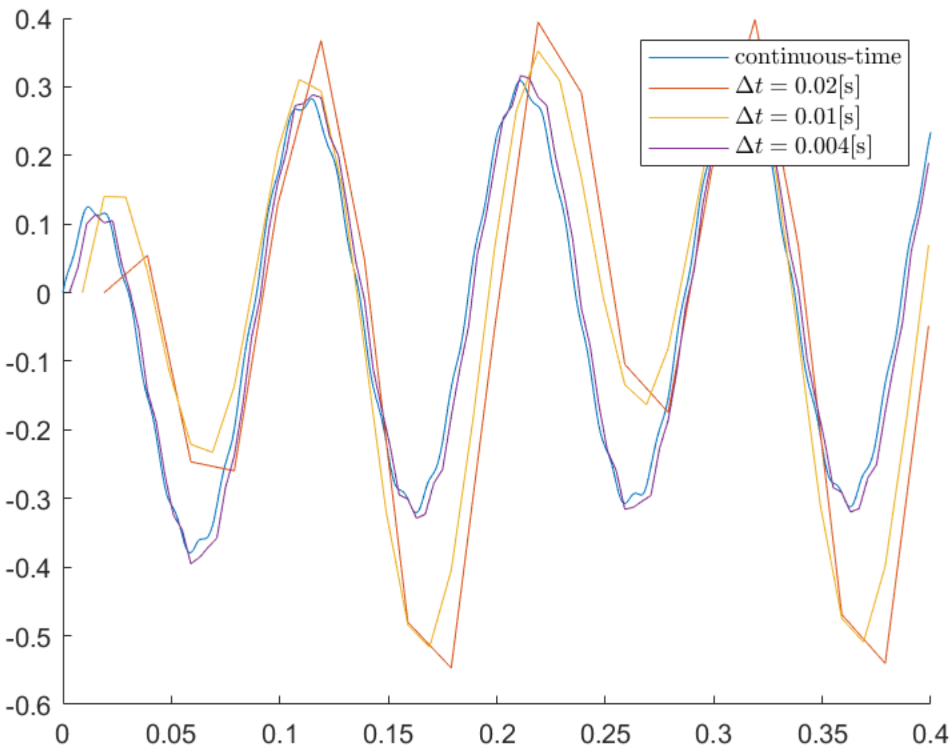
```
% generate signal with noise
R_in = 1;
R_noise = 0.3;
f_in = 10;
f_noise = 95;
```

```
w_in = 2*pi*f_in;
w_noise = 2*pi*f_noise;
phi_in = pi/6;
phi_noise = pi/2;
f = @(t)R_in*cos(w_in*t+phi_in);
w = @(t)R_noise*cos(w_noise*t+phi_noise);

w_c = 20;
t_max = 0.4;
y0 = 0;
[t,y] = ode45(@(t,y)w_c*(-y+f(t)+w(t)),[0:(1/1000):t_max],y0);

clf
ax = axes;
hold(ax,'on')
plot(ax,t,y)

%
time_step = 1/1000;
for f_s = [50 100 250]
    dt = 1/f_s;
    N = 400;
    t_u = [];
    t_y = [];
    u_out = [];
    y_out = [];
    t = 0;
    x = 0;
    for n = 1:N
        u = f(t)+w(t);
        t_u = [t_u;t];
        u_out = [u_out;u];
        if ~mod(n,dt/time_step)
            y = x;
            x = (1-w_c*dt)*x+w_c*dt*u;
            y_out = [y_out;y];
            t_y = [t_y;t];
        end
        t = t+time_step;
    end
    plot(ax,t_y,y_out)
end
legend({'continuous-time','$\Delta t=0.02$[s]','$\Delta t=0.01$[s]','$\Delta t=0.004$[s]'},'Int
```



เราจะสังเกตได้ว่า ยิ่งช่วงเวลาคำนวณน้อยมากเท่าไหร่ ระบบที่ถูกประมาณจะใกล้เคียงกับระบบแบบเวลาต่อเนื่องมากขึ้นเท่านั้น ดังนั้น การจำกัดช่วงการคำนวณที่คงที่และมีค่าน้อยเป็นส่วนสำคัญในการประยุกต์ใช้ **digital filter/controller**