

สัญญาณและระบบแบบเวลาไม่ต่อเนื่อง (Discrete-time Signal & System)

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

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

เราสามารถอธิบายระบบพลวัตแบบเวลาไม่ต่อเนื่องในรูปของสมการผลต่าง (**difference equation**) ที่จัดเป็นสมการปริภูมิสถานะ (**state-space**) โดยมีรูปแบบดังต่อไปนี้

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

หากระบบเป็น **LTI** เราสามารถเขียนสมการแสดงปริภูมิสถานะได้ดังต่อไปนี้

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

ในเชิงของการนำสมการไปเขียนในโปรแกรม เราจะไม่สามารถเขียนสมการเชิงอนุพันธ์ได้นี้เนื่องจากการทำงานของคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ทำงานเป็นทีละขั้นตอน ดังนั้นการคำนวณแบบเวลาต่อเนื่อง (อนุพันธ์) จึงไม่สามารถเขียนตรงๆได้ เราทำได้ที่สุดคือการประมาณโดยใช้ระเบียบเชิงตัวเลข (numerical method)

ในทางกลับกัน เราสามารถเขียนสมการเชิงผลต่างในโปรแกรมได้เนื่องจากการทำงานของคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์เป็นแบบเวลาไม่ต่อเนื่อง

ดังนั้น เราจะสังเคราะห์และเขียนโปรแกรมระบบต่างๆเป็นเวลาไม่ต่อเนื่อง ซึ่งหลายครั้งจะเกิดมาจากการประมาณระบบแบบเวลาต่อเนื่อง

ตัวอย่างที่ 1 : การเฉลี่ยเคลื่อนที่ (Moving average)

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

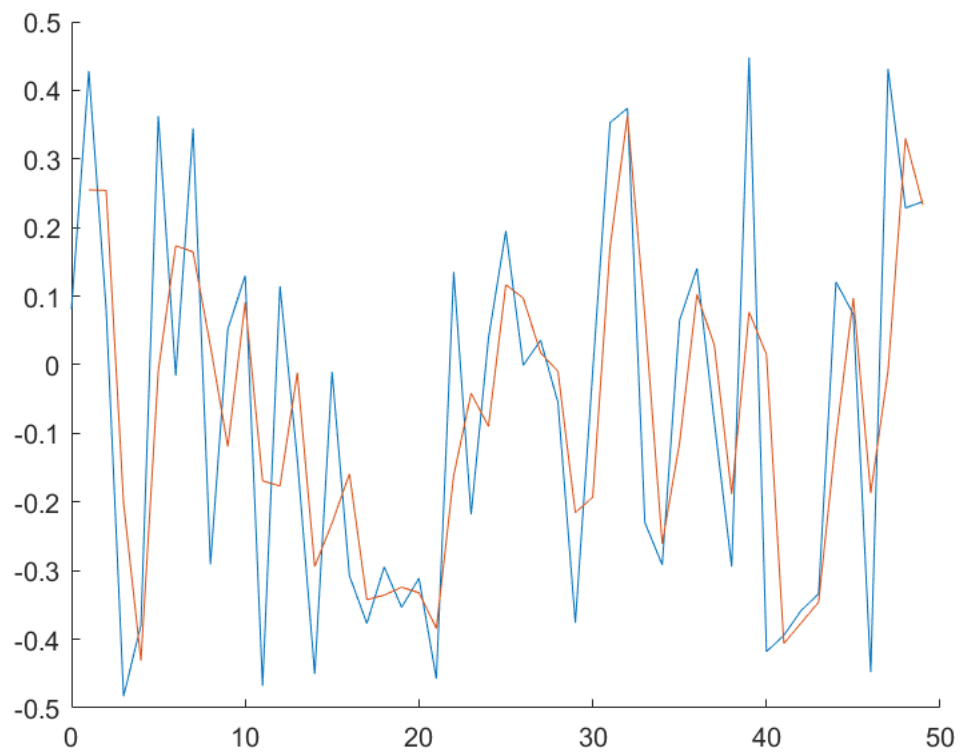
$$\begin{aligned} y[n] &= \frac{1}{2}(x[n] + u[n]) \\ x[n+1] &= u[n] \\ x[0] &= 0 \end{aligned}$$

โดยที่ $x[n]$ เป็นสถานะของระบบ

ความสัมพันธ์ที่ถูกกำหนดนับว่าเป็นระบบพลวัตแบบเวลาไม่ต่อเนื่องเพราะว่าความสัมพันธ์นั้นขึ้นอยู่กับค่าแปรสถานะด้วย

$y[n]$, $u[n]$, 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)

```
% เรียกครั้งแรก initialize
x = 0;

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