

Linear & Non-linear Simulation

HuStar

2020.07.

이성윤

모델링 선형화

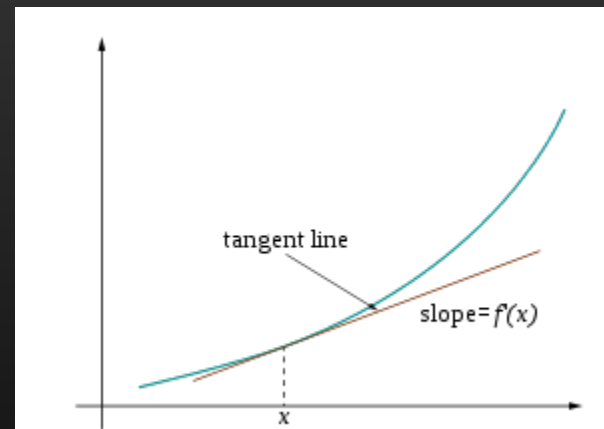
- 시스템 모델링이 LTI일 경우, 모델을 해석하고 파악하며 다룰 수 있는 방법이 굉장히 많다.
- 실제 물리 환경에서 시스템이 완전히 선형일 수는 없으며, 일반적으로 특정한 가동 범위 안에서 선형을 만족하도록 한다.

$$u_i(t) \rightarrow y_i(t)$$

$$u_1(t) + u_2(t) \rightarrow y_1(t) + y_2(t), \quad \text{additivity}$$

$$au_i(t) \rightarrow ay_i(t), \quad \text{homogeneity}$$

$$a_1u_1(t) + a_2u_2(t) \rightarrow a_1y_1(t) + a_2y_2(t), \quad \text{superposition}$$



- 예상 가동 범위가 넓어 한 번에 선형화하기 힘들다고 판단될 경우, 여러 구간으로 나누어 따로 선형화를 수행하며, 구간과 구간 사이의 연결을 고려해주어야 한다.
- 특정 가동 범위 안에서 선형화를 할 때는 테일러 급수를 활용한다.

테일러 급수

- 테일러 급수는 임의의 함수를 무한한 차수의 다항식으로 표현할 수 있다는 가정에서 출발하여, 각 차수의 계수를 찾아낸다.

$$\begin{aligned} f(x) \Big|_{x=a} &= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n \\ &= f(a) + \frac{f'(a)}{1!} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \frac{f'''(a)}{3!} (x-a)^3 + \dots \\ &= f(a) + \frac{f'(a)}{1!} (x-a) + O(x^2) \end{aligned}$$

Example 1

- 테일러 급수를 활용하여 $\sin(x)$ 과 $\cos(x)$ 를 다항식으로 전개하고 선형화하여, 각 차수별로 다항식으로 근사한 함수를 함께 그려보자.
 - 차수는 1, 2, 3, 4, 5까지 그려보자.

비선형 모델의 시뮬레이션

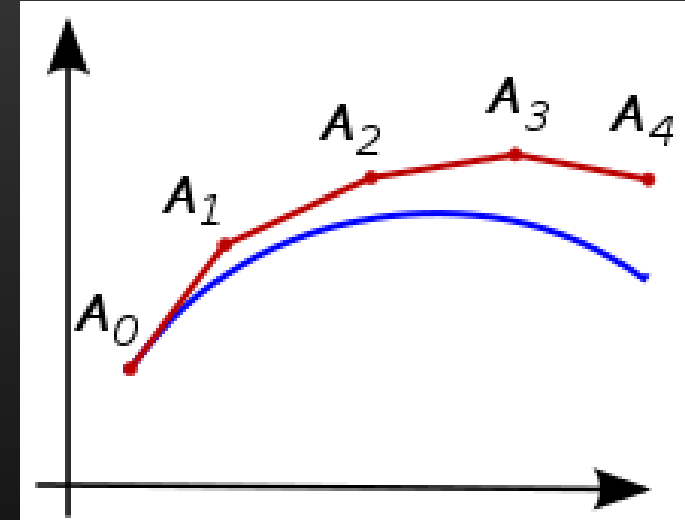
- 시스템을 선형화하여 분석한다고 해도, 실제 시뮬레이션은 가급적 획득한 정보를 최대한 반영하여 사실적으로 수행하고 싶은 것이다.
- 선형 시스템의 시뮬레이션은 전달함수와 Z-transform을 사용하여 비교적 쉽게 할 수 있다.
- 비선형 시스템까지 시뮬레이션을 수행하기 위해 일반화된 미분방정식의 수치적 해법이 있다. → Euler Method, Runge Kutta Method
- 특정 가동 범위에서 모델링을 선형화 했다고 해도, 선형화가 올바르게 이루어졌는지 확인하기 위해 비선형 시스템과 응답이 일치하는지 확인해볼 필요가 있다.

Euler Method - 기본형

$$\frac{dy(t)}{dt} = f(t, y(t)), \quad y(t_0) = y_0$$

$$\left. \frac{dy(t)}{dt} \right|_{t=t_0} \approx \frac{y(t_0 + h) - y(t_0)}{h}$$

$$y_{n+1} = y_n + h f(t_n, y_n)$$



- 시간의 단위 h (또는 dt)가 충분히 작지 않을 경우 결과는 발산함.

Example 2

- 다음 식의 해석적 해, 디지털 필터를 이용한 해, 그리고 오일러 방법을 사용한 해를 모두 구하여 비교해보자.

$$\frac{dy(t)}{dt} = y, \quad y(0) = 1$$

Euler Method - 행렬형

- 2차 이상의 미분방정식의 경우, 오일러 방법을 그대로 적용할 수 없다.
← 1계도함수에 대한 함수로 표현하기 어려움
- 따라서, 일차연립미분방정식으로 변환하여 해결한다.

$$y^{(n)}(t) = f\left(t, y(t), y'(t), \dots, y^{(n-1)}(t)\right)$$

$$z_1(t) = y(t), \quad z_2 = y'(t), \quad \dots, \quad z_n = y^{(n)}(t)$$

$$\mathbf{z}'(t) = \begin{pmatrix} z_1'(t) \\ \vdots \\ z_n'(t) \end{pmatrix} = \begin{pmatrix} y'(t) \\ \vdots \\ y^{(n)}(t) \end{pmatrix} = \begin{pmatrix} z_2(t) \\ \vdots \\ f(t, z_1(t), \dots, z_n(t)) \end{pmatrix}$$

Example 3

- 다음 식의 해석적 해, 디지털 필터를 이용한 해, 그리고 오일러 방법을 사용한 해를 모두 구하여 비교해보자.

$$\frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} + y(t) = 1, \quad y(0) = 1$$

Runge-Kutta Method

- RK는 가장 널리 쓰이는 미분방정식 수치 해법으로, Euler보다 훨씬 정확하다.
- 현재 시점에서의 추정값과 미래 시점에서의 추정값을 바탕으로 가장 정확할 것으로 예상하는 기울기를 구하는 방법이다.

$$\frac{dy(t)}{dt} = f(t, y(t)), \quad y(t_0) = y_0$$

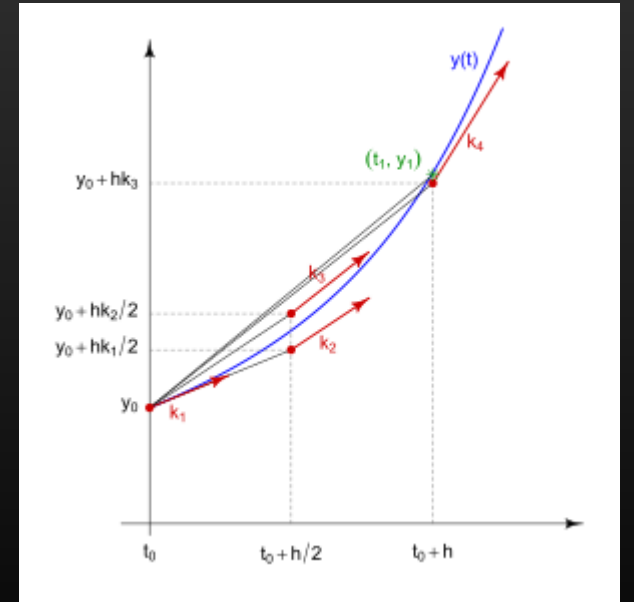
$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad t_{n+1} = t_n + h$$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$



Example 4

- Example 2와 3에 나온 미분방정식에 RK를 적용하여 Euler와 비교해보자.

$$\frac{dy(t)}{dt} = y, \quad y(0) = 1$$

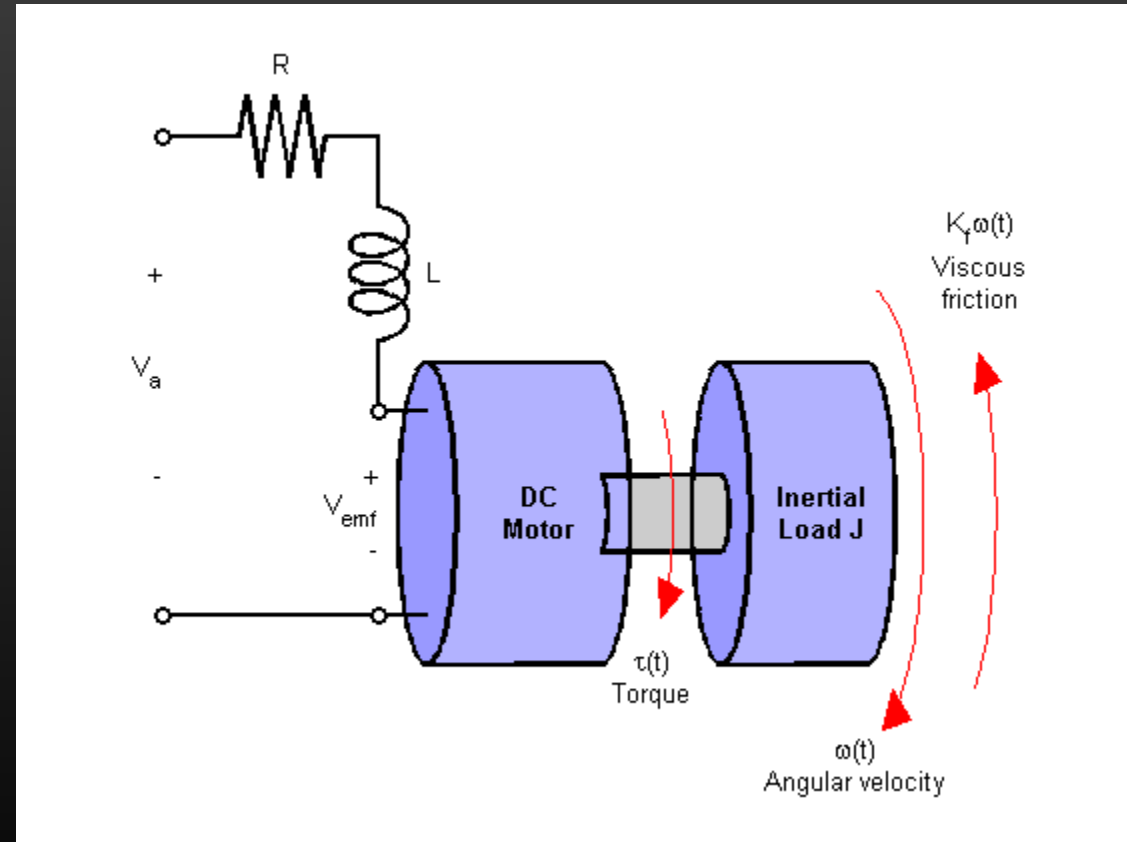
$$\frac{d^2y(t)}{dt^2} + \frac{dy(t)}{dt} + y(t) = 1, \quad y(0) = 1$$

Example 5

- 다음 DC모터를 모델링하여 블록다이어그램으로 나타내고, 스펙시트에 있는 수치와 비교해보자.
(참고식)

$$\tau = K_m i$$

$$V_{emf} = K_m \omega$$



Example 6

- 다음 카트형 역진자를 모델링하여 RK를 통해 진자가 15° 에서 출발했을 때의 응답을 계산해보자.
 - u : 외부에서 입력된 힘 (단위 N)
 - B_c : 카트에서 속력에 비례하는 마찰력
 - B_p : 진자의 각속도에 비례하는 마찰력

