



Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 전문가 과정

날 짜 : 2018 . 5. 22

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – 정한별
hanbulkr@gmail.com

* 오일러 공식

$$1. e^{ix} = \cos x + i \sin x$$

$$x^2 + y^2 = r^2$$

$$\cos \theta \hat{i} + \sin \theta \hat{j} = \hat{r}$$

$$2. e^{-ix} = \cos x - i \sin x$$

$$(1) + (2) \Rightarrow e^{ix} + e^{-ix} = 2 \cos x$$

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}$$

$$(1) - (2) \Rightarrow e^{ix} - e^{-ix} = 2i \sin x$$

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

3. 복소근의 정

$$A) r_1 = \lambda + i\mu, r_2 = \lambda - i\mu$$

$$y_1 = e^{(\lambda + i\mu)x} = e^{\lambda x} e^{i\mu x} = e^{\lambda x} (\cos \mu x + i \sin \mu x)$$

$$y_2 = e^{(\lambda - i\mu)x} = e^{\lambda x} e^{-i\mu x} = e^{\lambda x} (\cos \mu x - i \sin \mu x)$$

$$y_1 + y_2 = e^{\lambda x} (\cos \mu x + i \sin \mu x) + e^{\lambda x} (\cos \mu x - i \sin \mu x)$$

$$\Rightarrow \begin{cases} e^{\lambda x} \cos \mu x = \frac{y_1 + y_2}{2} \\ e^{\lambda x} \sin \mu x = \frac{y_1 - y_2}{2i} \end{cases}$$

$$\Rightarrow \boxed{\therefore y = C_1 e^{\lambda x} \cos \mu x + C_2 e^{\lambda x} \sin \mu x}$$

$$1. y'' + 5y' + 6y = 0$$

$$y(0) = 1, y'(0) = 3$$

$$2. 3y'' + 12y' + 12y = 0$$

$$y(0) = 3, y'(0) = 1$$

$$3. y''' + 4y'' + 5y' = 0$$

$$y(0) = 1, y'(0) = 2$$

Cauchy - Euler 방정식

$$ax^2y'' + axy' + ay = 0$$

→ 특이 숫자가 있으면 배한다.

→ 제곱값을 없애기 위해

$$x^2y'' + \frac{a_1}{x}xy' + \frac{a_0}{x^2}y = 0$$

연립 미분방정식(0) 미분방정식이 된다.

→ 미분방정식이 가능하다.

→ 미분방정식

→ 미분방정식

→ 미분방정식

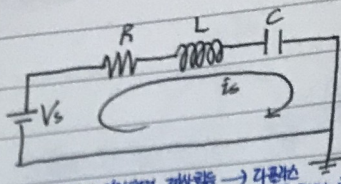
특정해

특정해

$$y'' + a(x)y' + b(x)y = g(x)$$

$$V_s = iR + L \frac{di}{dt} + \frac{1}{C} \int i dt$$

여기서 미분은 편미한다.
그러면 2개 미분 방정식



→ 이 방정식이 연립 방정식 → 다윈의 방정식

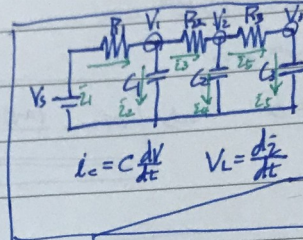
전압 전압과 같다.

$$L = C \frac{dV}{dt} \quad q = CV \quad \frac{dq}{dt} = i$$

* 전압으로에서는

→ 전압과 전압이 아주 쉽게 전압이 된다.

→ 전압은 전압이다



$$i_1 = C \frac{dV_1}{dt} \quad V_1 = \frac{dV_2}{dt}$$

→ 다윈의 방정식

$$\begin{aligned} 1. \quad \frac{V_s - V_1}{R_1} &= C_1 \frac{dV_1}{dt} + \frac{V_1 - V_2}{R_2} \\ 2. \quad \frac{V_1 - V_2}{R_2} &= C_2 \frac{dV_2}{dt} + \frac{V_2 - V_3}{R_3} \\ 3. \quad \frac{V_2 - V_3}{R_3} &= C_3 \frac{dV_3}{dt} \end{aligned}$$

$$y'' + a(x)y' + b(x)y = g(x)$$

$y_p = \text{특정해}$

$$y_g = \text{일반해} \quad y_g = y_p + y_h$$

$$y_h = \text{동차해} \quad y_h = y_g - y_p$$

$$\begin{aligned} y_g'' + a(x)y_g' + b(x)y_g &= g(x) \\ y_p'' + a(x)y_p' + b(x)y_p &= g(x) \end{aligned}$$

$$y_h'' + a(x)y_h' + b(x)y_h = g(x)$$

$$(y_g - y_p)'' + a(x)(y_g - y_p)' + b(x)(y_g - y_p) = g(x)$$

$$\frac{y_g'' + a(x)y_g' + b(x)y_g}{g(x)} - \frac{y_p'' + a(x)y_p' + b(x)y_p}{g(x)} = \frac{g(x)}{g(x)} \rightarrow \text{동차는 } 0 \dots$$

$y_p(\text{특정해})$ 조건

$$(y'' + 2y' + y = 2x^2 + 4)$$

1) $g(x)$ 가 다항식인 경우

$$A_n x^n + A_{n-1} x^{n-1} + \dots + A_0$$

$$y_h = C_1 e^{-x} + C_2 x e^{-x}$$

$$y_p = Ax^2 + Bx + C = 2x^2 - 8x + 16$$

2) $g(x)$ 가 지수함수인 경우

$$Ae^{mx}$$

$$y_p' = 2Ax + B$$

$$y_p'' = 2A$$

3) $g(x)$ 가 삼각함수인 경우

$$A \cos(mx) + B \sin(mx)$$

$$2A + 4Ax + 2B + Ax^2 + Bx + C$$

$$A^2x + (4A+B)x + (2A+2B+C) = 2x^2 + 4$$

$$A=2, B=-8, C=16$$

$$4A+B=0$$

$$2A+2B+C=4$$

선생님이 주지 않은 걸로 미방문제를 풀어 봤음.

```

#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <math.h>

#define delta_x 0.001
#define E 2.7182818284590452353602874

void original_solve(double *A)
{
    double x = delta_x;
    double ii = x*-5000;
    printf("first num = %lf \n", ii);
    for(int i=1;i <= 10001;i++){
        A[i-1] = 3*pow(E, -pow(-5.000+x*(double)(i-1),2.0));
        printf("%11.6lf", A[i-1]);
        if(i%10 == 0)
            printf("\n\r");
    }
}

void second_solve(double *B)
{
    double x = delta_x;
    double ii = x*-5000;
    printf("first num = %lf \n", ii);
    for(int i=1;i < 5000;i++){
        B[i] = (-2*x*(x*(double)(i-1))+1)*B[i-1];
        printf("%11.6lf", B[i-1]);
        if(i%10 == 0)
            printf("\n\r");
    }
}

void percentage(double *A, double *B)
{
    for(int i=0; i<4999; i++)
    {
        printf("%11.6lf%%", A[i+5000]/B[i+1]*100);
        if(i%10 == 0)
            printf("\n\r");
    }
}

int main(void)
{
    printf("original solve is  $y = 3e^{-x^2}$ .\n");
    printf("y' = -2xy , y(0) = 3. \n");

    printf("start 미방. \n");

    double first_matrix[10001] = {0};
    double second_matrix[10001] = {0};

```

```

second_matrix[0]=0;
second_matrix[0] = 3.0;

original_solve(first_matrix);
printf("\n0 일 때 : %.05lf \n", first_matrix[5000]);
printf("\n");

second_solve(second_matrix);
printf("\n0 일 때 : %.05lf \n", second_matrix[0]);

percentage(first_matrix, second_matrix);

return 0;
}

```

오차율은 x 가 증가함에 따라 크게 변화했다. 5에 가까워 질때 좀 9프로 까지 차이난다. 아마도 0에 가까워 질수록 자릿수가 너무 미세하여 컴퓨터가 표현하면서 오차가 커진것 같다.