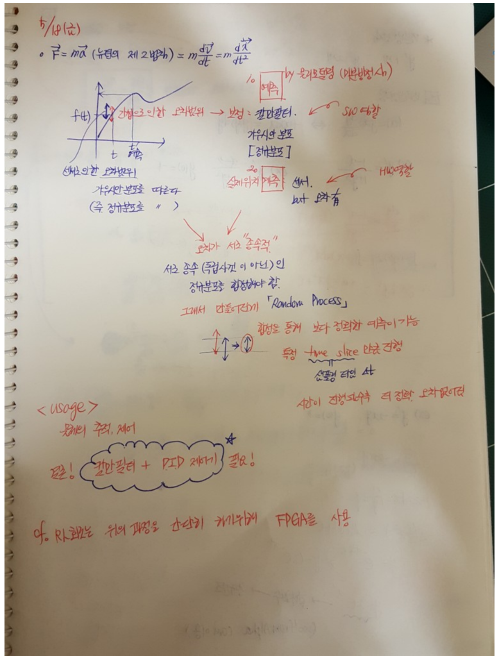
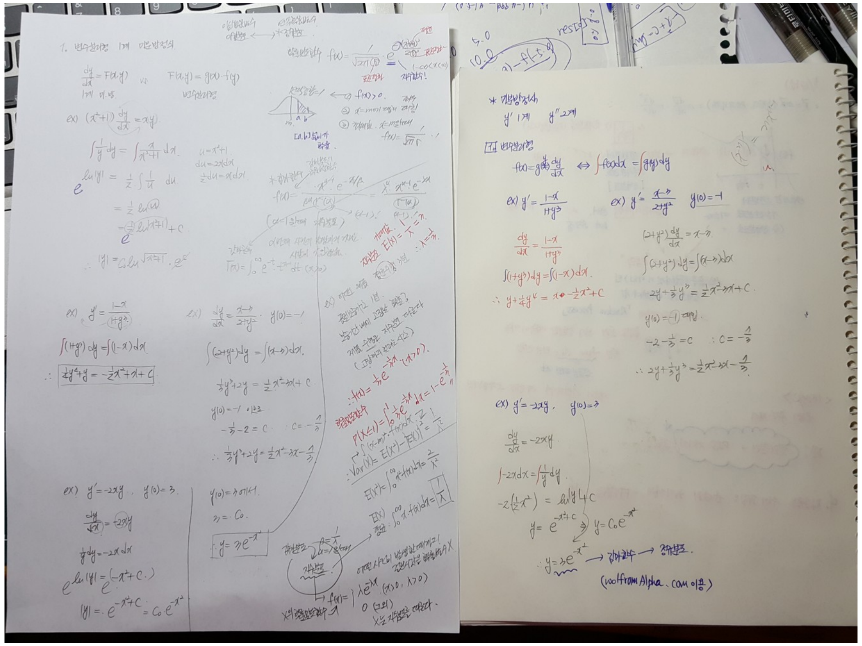
TI DSP, MCU 및 Xilinx Zynq FPGA
프로그래밍 전문가 과정

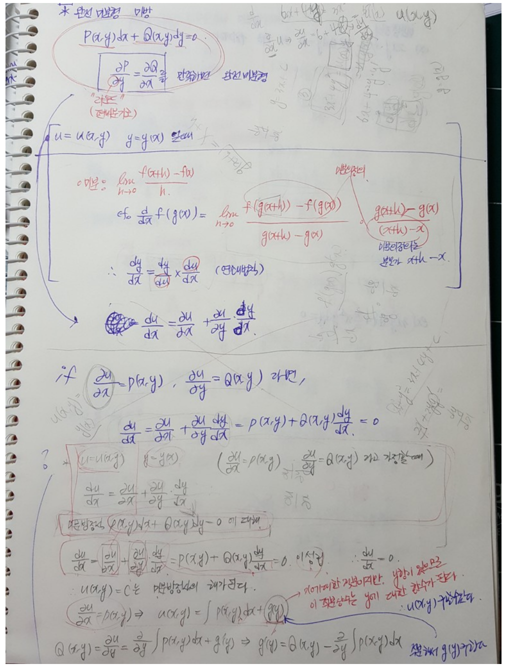
2018-05-18 (57 회차)

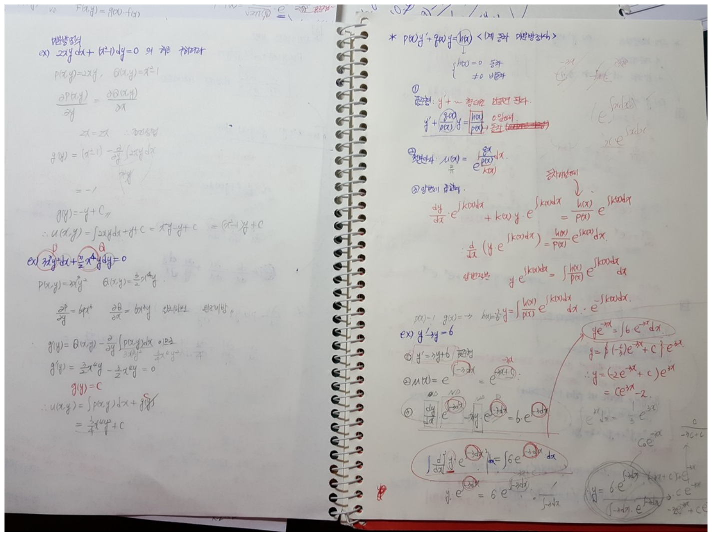
강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 정유경
ucong@naver.com

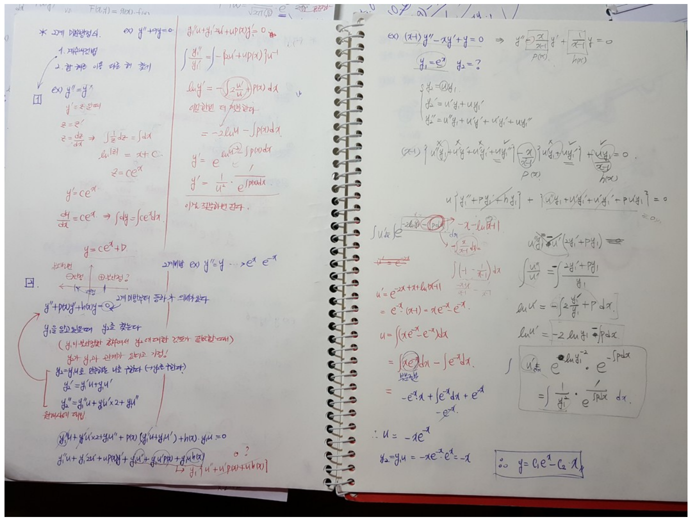
이론수업











1) 미분 방정식
2) 샘플링 이론
3) 신호 처리 방식으로 미분을 하는 방법
4) 신호 처리 레벨에서 적분 수행법

- 변수 분리형 미분 방정식 풀이법

- e^(-x^2) 은 지수함수 < 감마 함수< 정규분포함수 → 칼만필터와 연관 (필기참고)

- 완전 미분형 미분 방정식 풀이법

- 연쇄 법칙(Chain Rule) 설명

- 완전 미분형 미분 방정식 동작 메커니즘 해석(수치해석으로 프로그래밍 할 때 이 절차대로 프로그래밍)

- 완전 미분형 미분 방정식 문제 풀이

- 1 계 미분 방정식 풀이 방법(적분 인자)

- 치환을 활용한 2 계 미분 방정식 해석

- 2 계 미분 방정식의 해는 2 개가 존재하며 일반해는 이 2 개를 모두 표현할 수 있어야 한다.

  즉, 2 계 미분 방정식에서 1 개의 해를 알 때 나머지 해를 구해 일반해를 구하는 방법

<과제> 미분방정식의 해 구하는 프로그램 구현하기

컴퓨터는 미적분이 불가능하다. (수치해석+신호처리 기법으로 가능하다)
미분방정식을 프로그래밍으로 시뮬레이션 해본다.
x 값의 범위는 -5 ~5
delta_x = 0.001
delta_x 에 대한 각각의 y 값을 배열에 저장한다.
y’ = -2xy, y(0) = 3(인 미분방정식의 해는 y=3e^(-x^2))

#define PI
#define Euler 2.718182…..
(y[1] – y[0]) / delta_x = dy/dx = y’
y[1] 은 0.001 초의 신호값, y[0] 은 0.0 초의 신호 값

샘플링 간격이 0.001 초이므로 이 사이에서의 변화율(미분)은 두 신호의 차이를 샘플링 간격으로 나눈 값이 된다.
-5 ~ 5 를 집어넣으니 처음은 [y(-4.999) - y(-5.0)] / 0.001, 그 다음은 [y(-4.998) - y(4.999)] / 0.001 이 된다.

다시 구현해볼것
잘못 생각하고 있는것 같다

```
#include <stdio.h>
#include <math.h>

#define dx			0.1
#define e			2.71828  // EulerNumber

double f(double x);
double fprime(double x, double y);
double dydx(double y);

int main(void)
{
	int i=0;
//	double y;
	double x=-5.0;
	double res[100];
	/*y=3*e^(-x^2)을 이용, 수치해석으로 구한 y'값*/
	while( x < 5.0) // 10 번 돌겠다
	{

		x += dx;
		printf("%d\t",i);
		res[i] = (f(x)-f(x-dx))/dx;
		printf("수치해석: %lf\t",res[i]);

		y=f(x);
		// y'=-2xy 에 의한 y'값을 구한다
		printf("y'=-2xy: %lf\t",fprime(x,f(x)));

		printf("오차율:%lf\n", (res[i]-fprime(x,f(x)))/fprime(x,f(x)));
		i++;
	}
```

```
	return 0;
}

double f(double x)
{
	double y;
	y =  3*pow(e,-pow(x,2));
	return y;
}

double fprime(double x, double y)
{
	double res;
	res = -2*x*y;
}
```