11:20 AM 10/18/2015

Sunday Meeting 4 (11AM – 2 PM)

To do:

1. **Puzzle**

   Given a number n, return the nth Fibonacci sequence number. Note: 1,1,2,3,5…

```
//using recursion

int fibonacci(unsigned int n)
{
        if(n==0)
                return 0;
        else if(n==1)||(n==2))
                return 1;
        return fibonacci(n-1) + fibonacci(n-2)
}

//using LUT

int main()
{
        int arr[size];
        printf("%d", fibonacci(10, arr);
        int fibonacci(unsigned int n, int *LUT)
}

{
        if(n==0)
                return 0;
        else if((n==1)(n==2)
                return 1;
        if(LUT[n]!=0)
                return LUT[n];
        LUT[n] = fibonacci(n-2, LUT) + fibonacci(n-1, LUT);
}
```

2. **pid.c**

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de}{dt}$$

- pidInit – initialize pid sruct
- pidUpdate – does the calculations

- pidSetIntegralLimit – sets limit of the integral in the formula to avoid overflow
- pidReset – resets the PID
- pidSetError – sets the error

3. stabilizer.c
    - stabilizerInit – initialize stabilizer struct, calls controllerInit
    - controllerInit – initializes all the PIDs(roll, pitch, yaw, altitude)
    - stabilizerTask – calls controllerCorrect/AltitudePID
    - Cascading PIDs