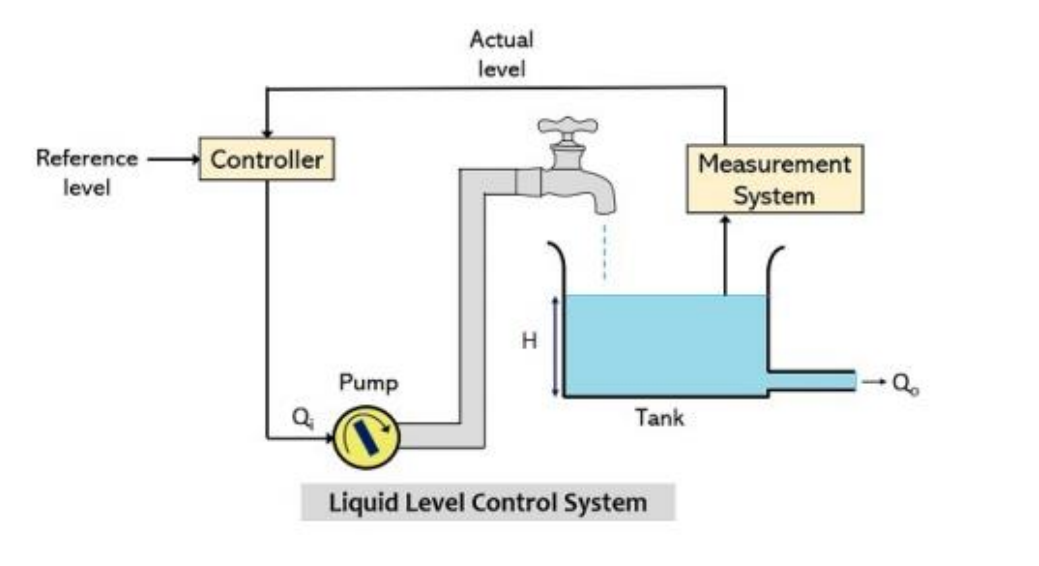
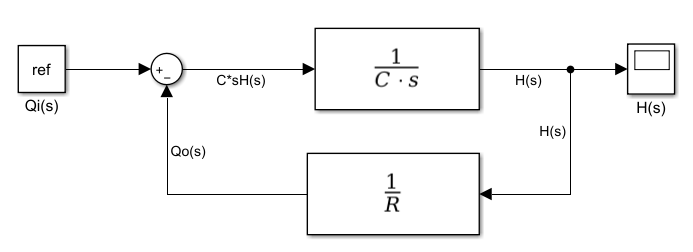
**SIVI SEVİYE KONTROLÜ**



Şekil 1 Sıvı Seviyesi Kontrol Sistemi

**Sistemin transfer fonksiyonu**

**Sistemin blok diyagramı**

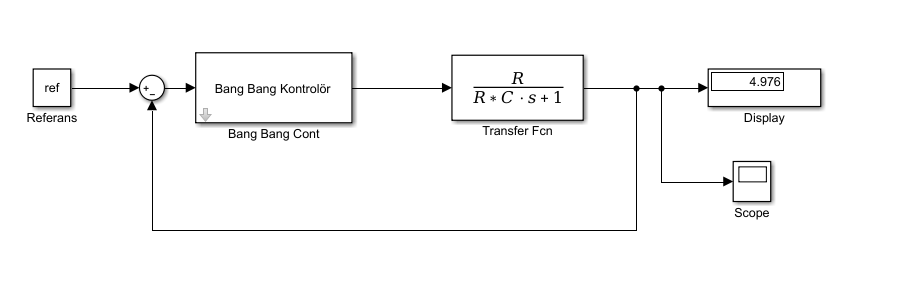


Şekil 2 Sıvı Seviyesi Kontrol Sistemi Blok Diyagramı

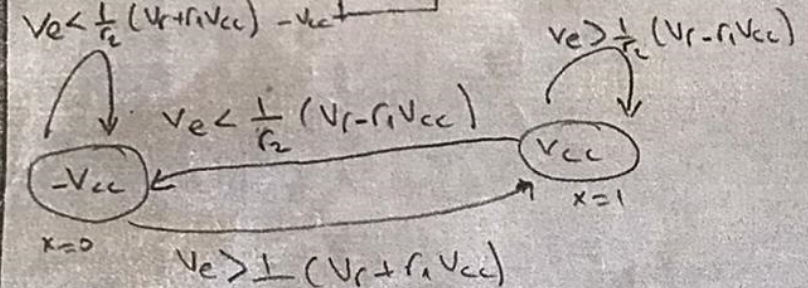
Sistemlerde çıktı alınması için R=2, C=1.5 alınmıştır. Ayrıca motorun ve musluğun fiziksel özelliklerinin transfer fonksiyonu hesaba katılmamıştır.

**Simulink'te bu sistem için iki konumlu bir kontrolör (bang-bang) geliştirimi**

Bang bang kontrölörün yaptığımız araştırmalar sonucunda on off kontorol olduğunu görüdük. Simulink ortamında hazır Relay bloğu ile yapabileceğimiz gibi bu kontrol işlemini kendi subsistemimizi yaparak da oluşturabiliriz. İki türlü de devresini gerçekleştirdik.

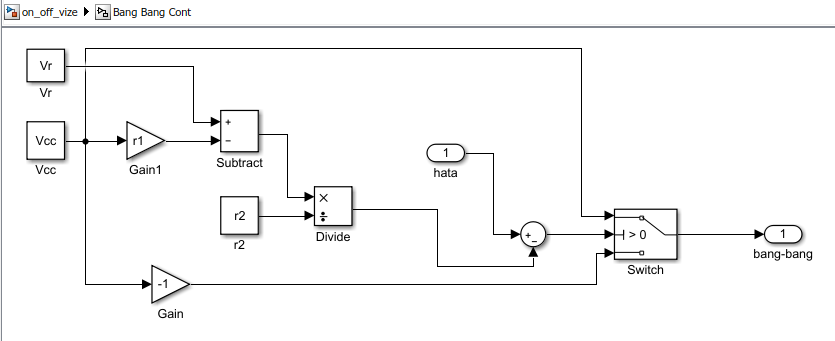


Şekil 3 Blok diyagramı

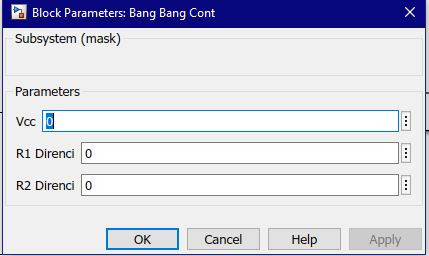
Subsistem yaptığımız zaman verilen formülü gerçekleştirdik. Parametreleri de mask uygulayarak giriyoruz.

Şekil 4 Bang Bang işlemleri

Yukarıdaki işlemler için gerekli olan Vr R1 R2 Vcc değerlerini subsisteme direkt girebiliyoruz. Aşağıdaki görselde mevcuttur. Yapmış olduğumuz işlemin çıktısını on off kontrol ile karşılaştırdığımızda aynı sonucu aldık.

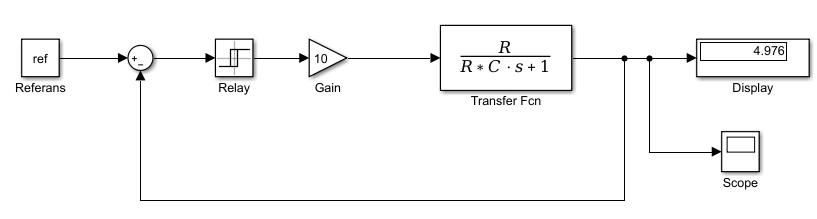


Şekil 5 Bang Bang kontrolör relaysız

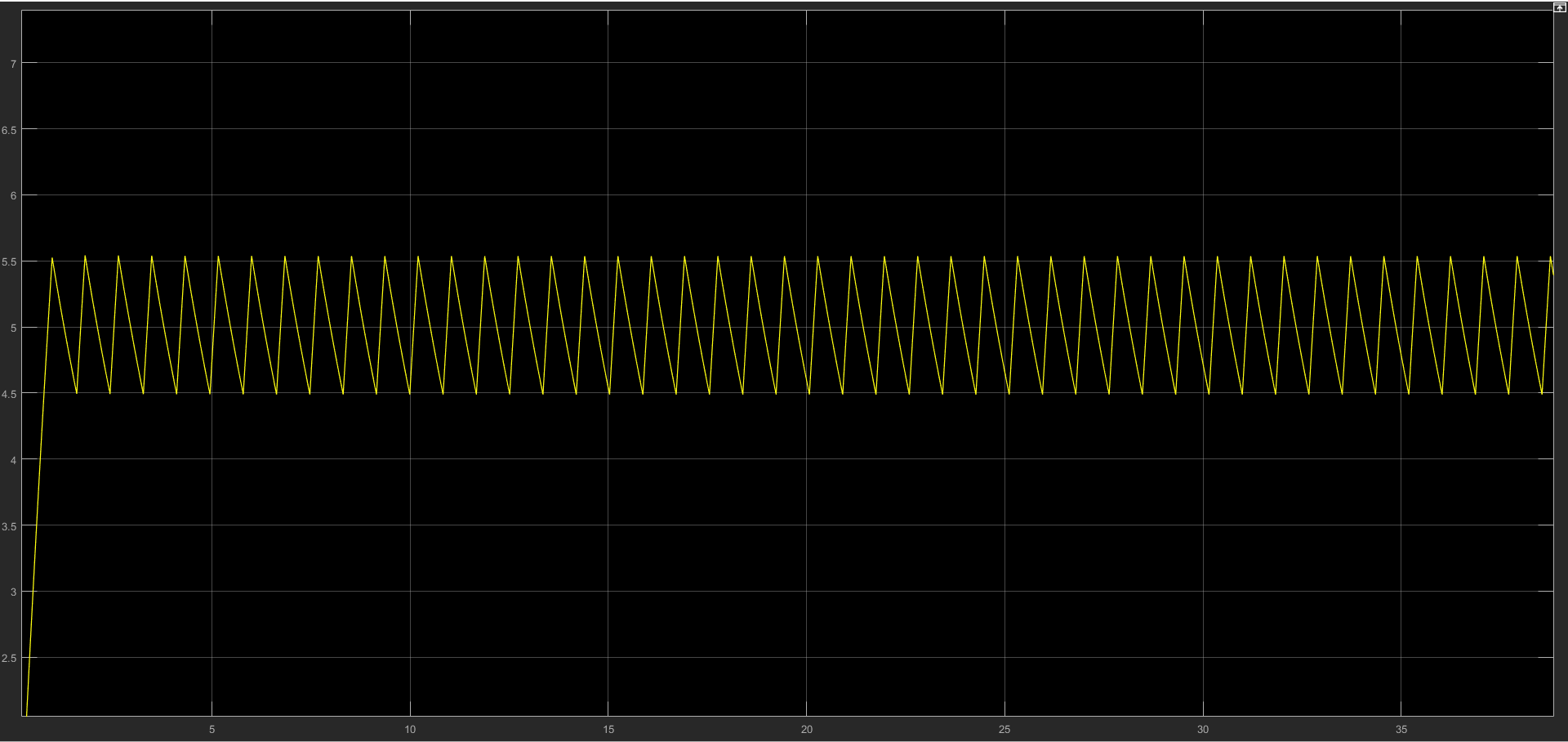


Şekil 6 Parametre giriş ekranı

Tasarladığımız kontrolörü relay kullanarak da yaptık aynı grafik çıkışını verdi. Relay verilerinde su yüksekliği referansın 0.5 fazlasına çıktığında Qi girişi kapanıyor 0.5 altına indiğinde Qi girişi açılıyor şeklinde on-off ile kontrol edilmektedir.



Şekil 7 Sıvı Seviyesi On-Off Kontrol Sistemi Blok Diyagramı

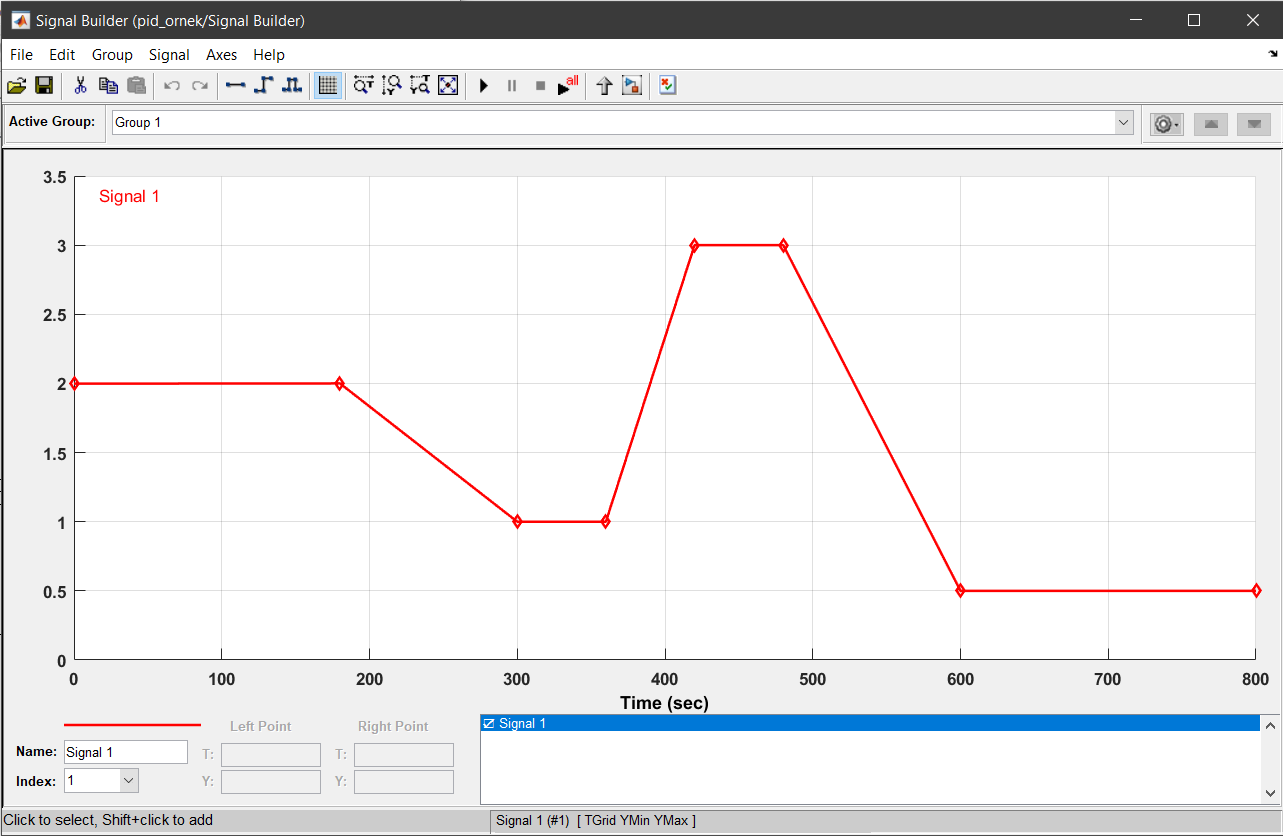


Şekil 8 Sıvı Seviyesi On-Off Kontrol Sistemi Çıktısı

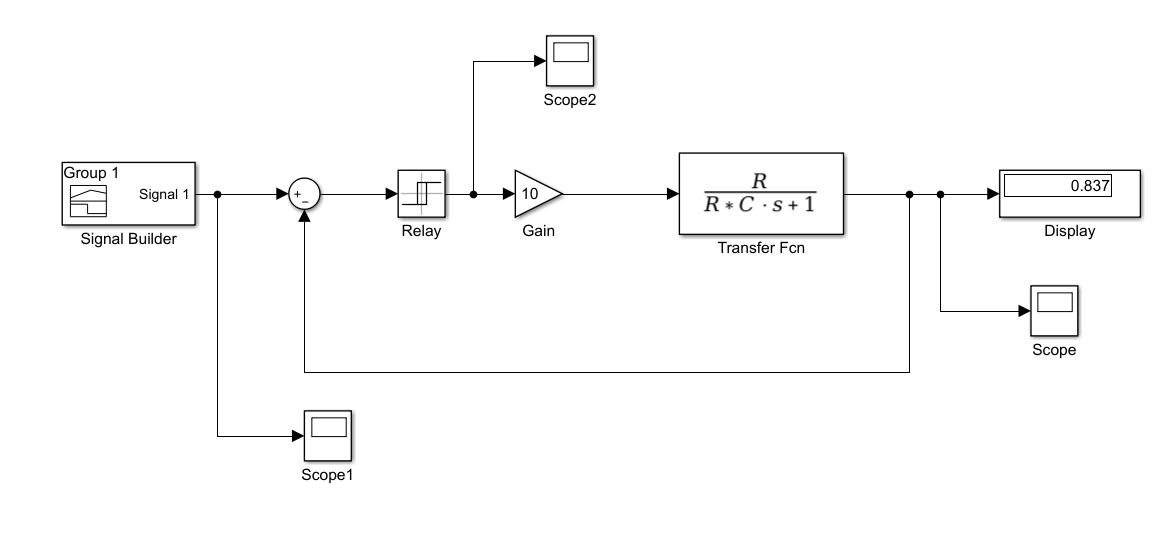
Şekil 3’de oluşturduğumuz blok diyagramında ref=5 verdikten sonra çalıştırdığımızda Şekil 4’de ki grafik elde edilmektedir.

**Ayar noktasının aşağıdaki şekilde verildiğini varsayın ve ayar noktası ve ayrı bir kontrolör grafiği ile sıvı seviyesi grafiğini elde edinimi**

Simulink kullanarak yaptığımız blok diyagramında (Şekil 5) verilen su seviye grafiğini signal builder ile hazırladık. Grafik dakika olarak verildiği için x ekseni grafiği tasarlarken hepsini saniye cinsinden yazarak çizdik.

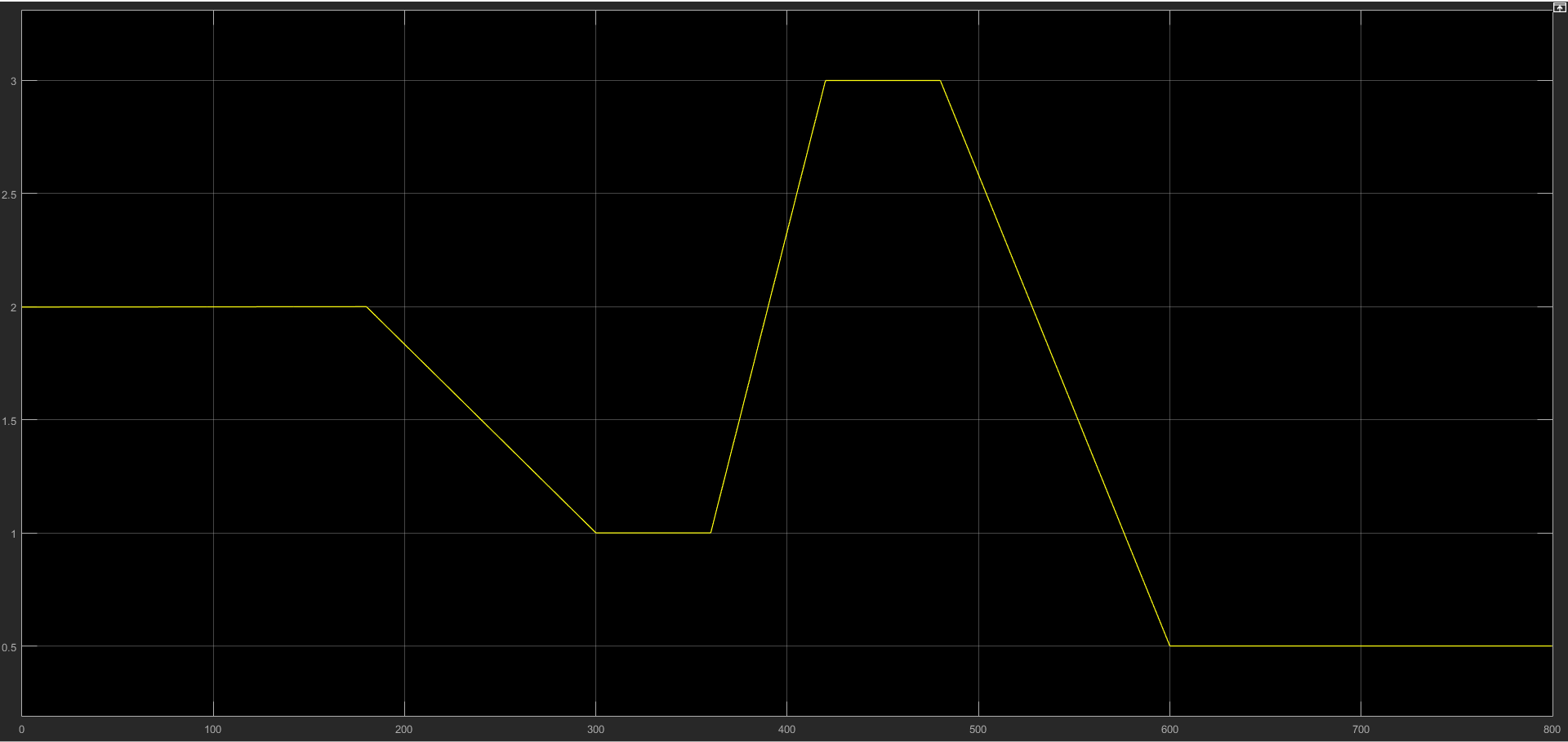


Şekil 9 Sıvı Seviyesi On-Off Kontrol Sistemi Referans Sıvı Yüksekliği



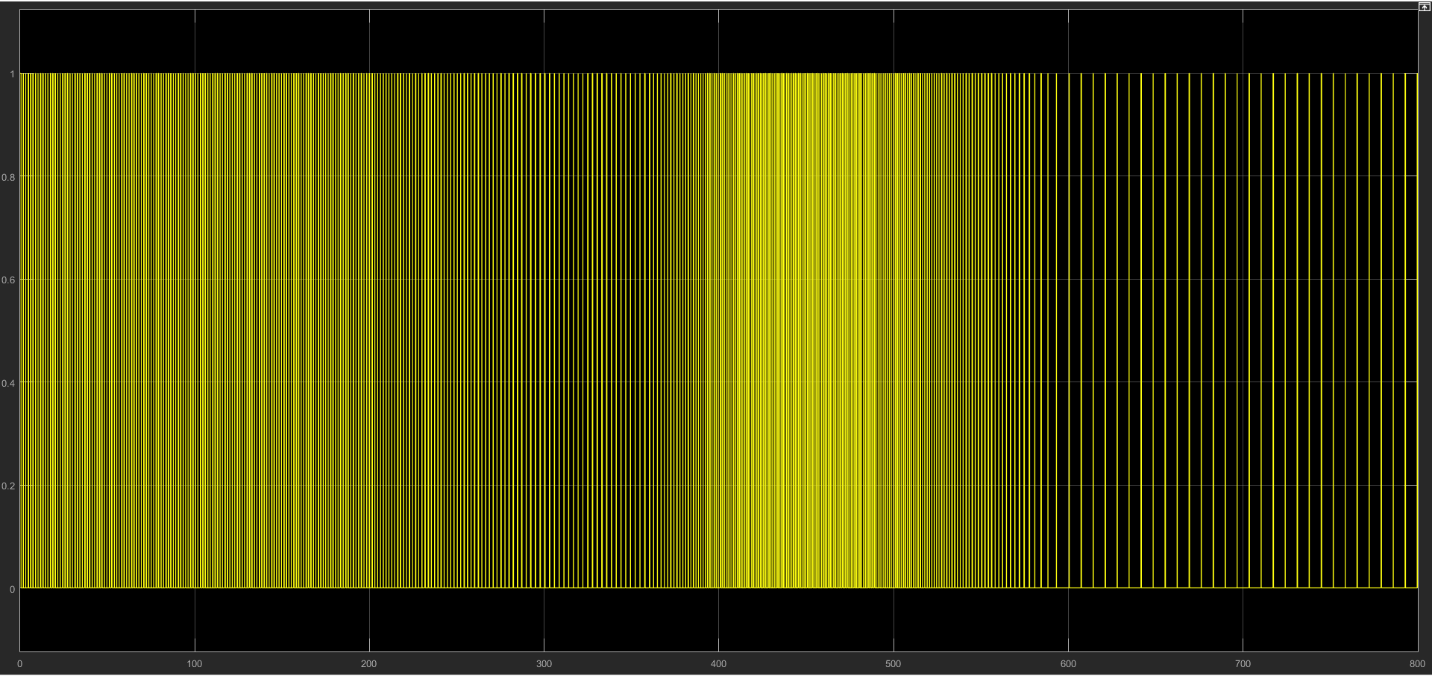
Şekil 10 Sıvı Seviyesi On-Off Kontrol Sistemi Referanslı Blok Diyagramı

Şekil 6’ daki blok diyagramı, verilen sinyale göre su yüksekliğini ayarlamaktadır. Relay giriş değerlerini ±0.4 yapılmasının nedeni istenilen su seviyesinde 0.5 gibi bir referans değeri olduğu için histerisiz alt sınırı mutlaka 0.5’den küçük olmalıdır aksi taktirde 600. saniyeden itibaren çıktılarda hata meydana gelecektir.



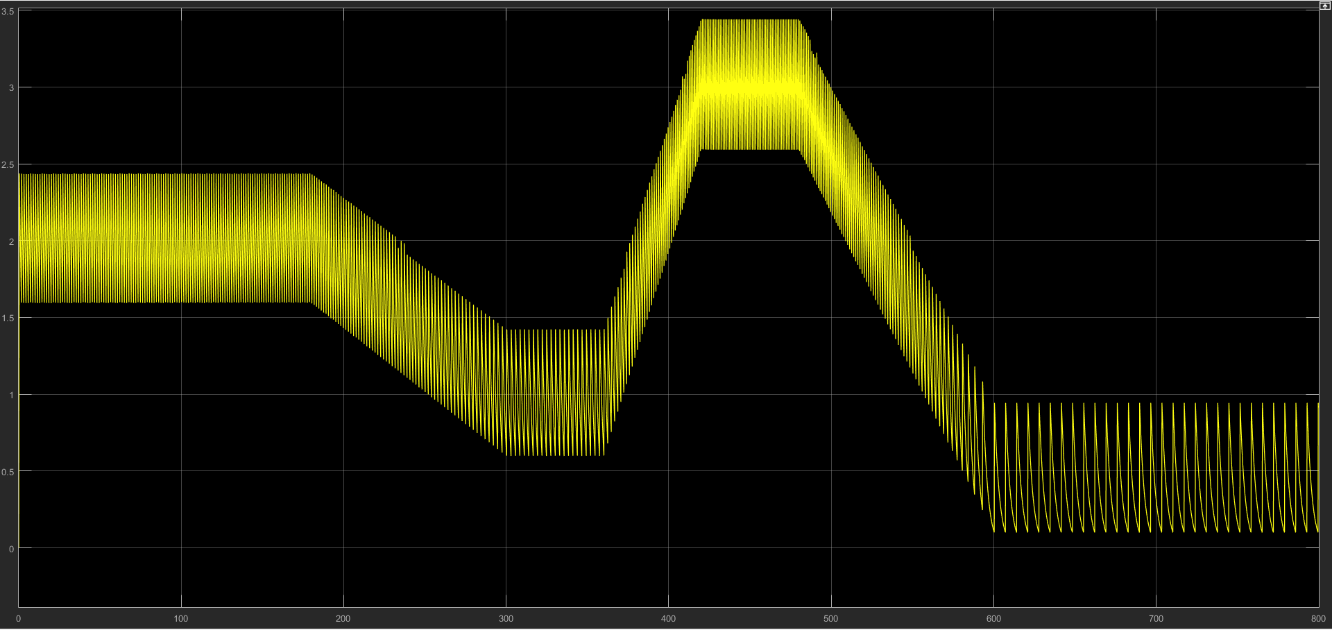
Şekil 11 Referanslı Sıvı Seviyesi On-Off Kontrol Sistemi Girişi

Scope1 den giriş sinyalimizi görmekteyiz (Şekil7).



Şekil 12 Referanslı Sıvı Seviyesi On-Off Kontrol Sisteminin Relay Çıkışı

(Şekil8) Scope2 den almış olduğumuz on off kontrolör çıkış sinyali scope1 deki sinyal ile incelerseniz normalde beklenmesi gereken grafiğin her diliminde aynı dalga sıklığında olmasıdır. Lakin doppler etkisinden dolayı referans sinyalinin değişimiyle birlikte histerisiz bölgede kayacağı için bu değişimlerde gözlemci sinyalin frekansının değişiyormuş gibi görecektir.



Şekil 13 Referanslı Sıvı Seviyesi On-Off Kontrol Sistemi Çıkışı

Çıkış sinyalimiz yukarıdaki grafikte yer almaktadır (Şekil9). İlk başta sistem açılıyor su dolmaya başlıyor. Su seviyesi 0.4’ den fazla olana kadar doluyor sonrasında 0.4’ den az olana kadar kapalı kalıyor bu şekilde girdiğimiz grafiğe göre su seviyesini ayarlamış oluyoruz.

**C dilini kullanarak aynı kontrol prosedürünü geliştirimi**

Sağdaki kodun main kısmında

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int simulate();

double on\_off(double e);

double plant(double c);

int main(void)

{ int n = simulate();

printf("%d adimda simule edildi\n", n);

return 0;

}

int simulate()

{ int BUF = 128; //line buffer

char line[BUF];

int n = 0;

FILE \*in = fopen("setpointvalues.txt", "r");

FILE \*out = fopen("on\_off\_out.txt", "w");

if (in == NULL)

{ printf("okunacak set degeri dosyasi yok\n");

fclose(out);

return 0; }

while (fgets(line, BUF, in))

{ double sp;

sscanf(line, "%lf", &sp);

static double y = 0;

double e = sp ;

y= on\_off(e);

fprintf(out, "%f\n", y);

n++;

}

printf("on\_off\_out.txt dosyasi olusturuldu.\n");

fclose(in);

fclose(out);

return n;

}

Simulate() fonksiyonunun içine giriyor. Öncelikle txt dosyasından olmış olduğumuz verileri okumamız gerekiyor. Bu değerleri okurken fgets komutu kullandık tüm verileri çektik. Çektiğimiz veriyi ‘e’ değişkenine atadık. Bu atadığımız ‘e’ değişkenini ise on\_off() fonksiyonunun içine atacağız.

Ölçüm yapmadığımızdan

double plant(double c)

{

const double k = 2.0;

const double t = 3.0;

const double h = 0.1;

static double cd = 0; //c(k-1)

static double yd = 0; //y(k-1)

double y = exp(-(1 / t) \* h) \* yd + (k - k \* exp(-(1 / t) \* h)) \* cd;

cd = c;

yd = y;

return y;

}

double on\_off(double e)

{

const double tol = 0.4; // tolerans degeri

static double sayi;

static double cikis;

if(sayi>=(e+tol))

cikis = 1;

else if(sayi<=(e-tol))

cikis = 0;

if(cikis==0)

sayi=sayi+0.2;

else if(cikis==1)

sayi=sayi-0.2;

return sayi;

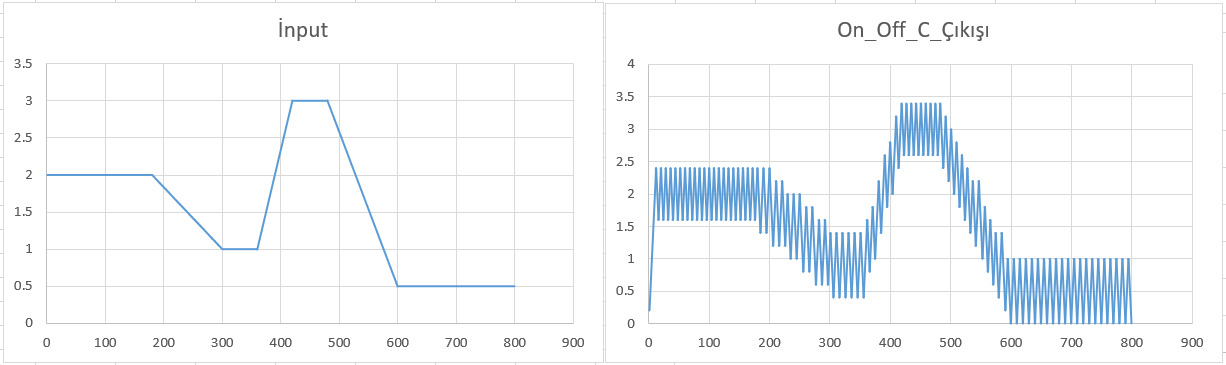
}

if(sayi>=(e+tol))cikis = 1;

else if(sayi<=(e-tol))cikis = 0;

şeklinde bir algoritma yazıldığında aslında gelen sayının direk giriş değerine eşitleyip histerisiz bölge oluşturduk. Giriş 0 dan başlayıp txt dosyasından okuduğumuz referans değerinin tolerans değeri kadar yükseliyor toleransı geçtiğinde sayı referans eksi tolerans kadar düşüyor tekrardan yükselmeye devam ediyor bu şekilde on off mantığını C de gerçekleştirmiş olduk.

Kodlar için verilen referans grafiğini Excell ortamında oluşturup setpointvalues.txt dosyasına kaydettik Çıkış grafiği aşağıda bulunmaktadır (Şekil 10).



Şekil 14 C Koduyla Yazılmış Referanslı Sıvı Seviyesi On-Off Kontrol Sistemi Giriş - Çıkışı Excell Görüntüsü

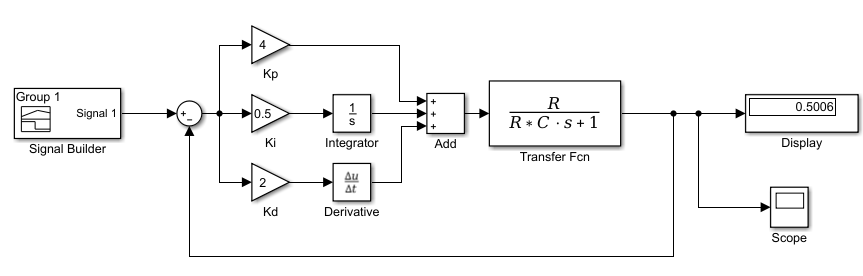
**PID kontrolörü için;**

**Sistemin transfer fonksiyonu.**

PID katsayıları Kp, Ki , Kd hesabını yaptık ve bunu sistemin transfer fonksiyonu ile çarptığımızda yeni bir blok tümleşiği elde ettik.

**Sistemin blok diyagramı**

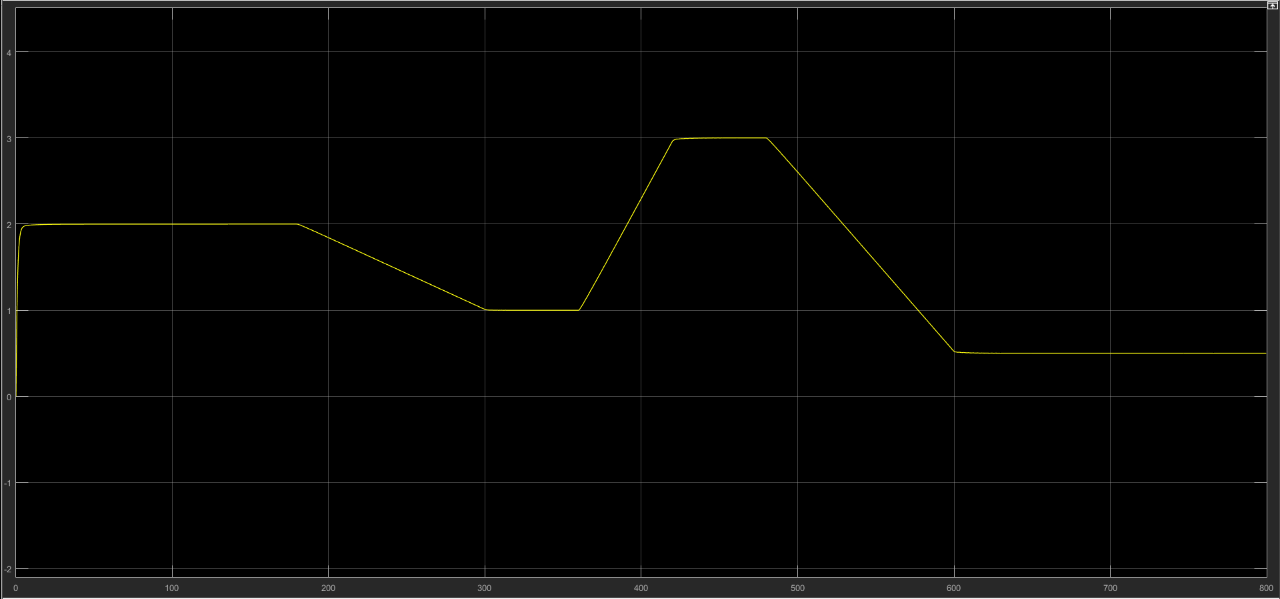
PID kullanmak istediğimizde karşımıza Şekil 13’ deki blok diyagramı çıkmaktadır. Kapalı çevrim transfer fonksiyonu ise yukarıda okun sağ tarafında hesaplanmıştır.



Şekil 17 Sıvı Seviyesi PID Kontrol Sistemi Referanslı Blok Diyagramı

**Ayar noktasınının verilen sinyale göre kontorlünü yapımı**

İstenilen su seviyesi giriş sinyali olarak ayarlandığında ve R=2, C=1.5; PID değerleri Kp = 4, Ki = 0.5, Kd = 2 şeklinde düzenlendiğinde çıkış olarak istenilen su seviyesine ulaşılmıştır. Hesapladığımız Kp, Ki , Kd değerleri aşımının olmadığı ve cevap süresinin çok geç olmayacağı şekilde ayarlanmıştır. Çıkış grafiği Şekil 14’ de görülmektedir.



Şekil 18 Referanslı Sıvı Seviyesi PID Kontrol Sistemi Çıkışı

**C dilini kullanarak aynı kontrol prosedürünü geliştirilmesi**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int simulate();

double pid(double e);

double plant(double c);

int main(void)

{

    int n = simulate();

    printf("simulated for %d steps\n", n);

    return 0;

}

int simulate()

{

    int BUF = 128;      //line buffer

    char line[BUF];

    int n = 0;

    FILE \*in = fopen("setpointvalues.txt", "r");

    FILE \*out = fopen("out.txt", "w");

    if (in == NULL) {

        printf("unable to read setpointvalues.txt\n");

        fclose(out);

        return 0;

    }

    while (fgets(line, BUF, in)) {

        double sp;

        sscanf(line, "%lf", &sp);

        static double y = 0;    //the plant output

        double e = sp - y;  //negative feedback

        double c = pid(e);  //compute control output

        y = plant(c);   //compute plant output

        fprintf(out, "%f\n", y);    //write to output file

        n++;

    }

    printf("plant output written to out.txt\n");

    fclose(in);

    fclose(out);

    return n;

}

Solda görüğünüz kod ilk başta main fonksiyonunda simulate() fonksiyonuna giriyor. Simulate fonksiyonu setpointvalues.txt dosyasından su seviye referans noktalarını çekiyor ve while işlemi yapıyor. Hata hesabı yaparken txt den gelen veri – y yani sonuç değerini çıkartarak hatayı hesaplıyor. Bu bulduğu hata değerini pid() fonksiyonuna veriyor. Pid fonksiyonu çıktısı ise plant yani sistemimizin bulunduğu fonksiyona giriş yapıyor. Kodlar hocamızın vermiş oladuğu örnek kodlardır ve kodları incelediğimizde bizim hesabını yapmış olduğumuz trasnfer fonksiyonu ile hemen hemen aynıdır.

Tam aşağıda gördüğünüz kod ise PID fonksiyonudur simulate fonksiyonundan e değerini alıyor ve işlem yapıyor. Bu fonksiyon çıkışında c değişkenini dönüdürüyor simulate() fonksiyonunda c değerini plant içerisine yerleştiriyor ve işlemi gerçekleştiriyor.

double pid(double e)

{

    const double kp = 4.0;

    const double kd = 2.0;

    const double ki = 0.5;

    const double tf = 4.0;

    const double h = 0.1;

    static double ed = 0;   //e(k-1)

    static double edd = 0;  //e(k-2)

    static double cd = 0;   //c(k-1)

    static double cdd = 0;  //c(k-2)

    double c =

        (e \*

         (4 \* kd + 4 \* kp \* tf + 2 \* kp \* h + 2 \* ki \* tf \* h +

          ki \* h \* h) + ed \* (-8 \* kd - 8 \* kp \* tf + 2 \* ki \* h \* h) +

         edd \* (4 \* kd + 4 \* kp \* tf - 2 \* kp \* h - 2 \* ki \* tf \* h +

            ki \* h \* h) - cdd \* (4 \* tf - 2 \* h) +

         cd \* (8 \* tf)) / (4 \* tf + 2 \* h);

    edd = ed;

    ed = e;

    cdd = cd;

    cd = c;

    return c;

}

double plant(double c)

{

const double k = 2.0;

const double t = 3.0;

const double h = 0.1;

static double cd = 0; //c(k-1)

static double yd = 0; //y(k-1)

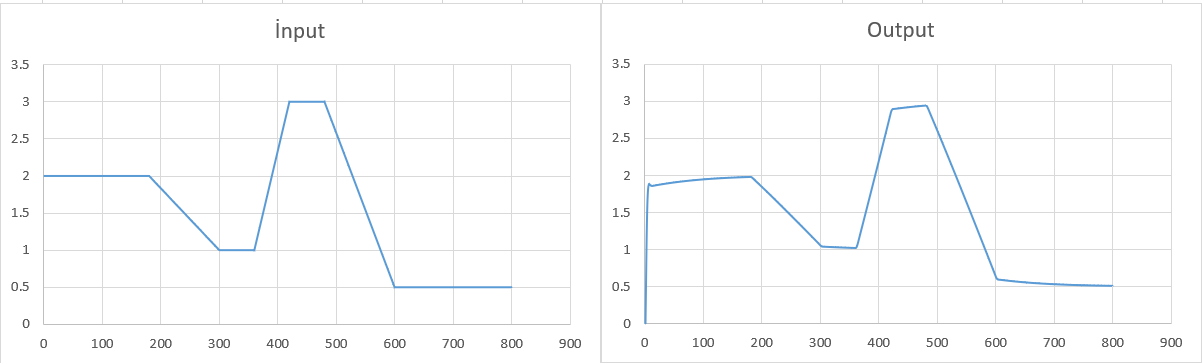
double y = exp(-(1 / t) \* h) \* yd + (k - k \* exp(-(1 / t) \* h)) \* cd;

cd = c;

yd = y;

return y;

}



Şekil 19 C Koduyla Yazılmış Referanslı Sıvı Seviyesi PID Kontrol Sistemi Giriş - Çıkışı Excell Görüntüsü

Setpoinvalue.txt değerlerini giriş olarak verdik ve çıkış olarak out.txt değerleri aldık. Daha sonra giriş ve çıkış değerleri excel ortamında grafik haline getirdik (Şekil15).