



T.C
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

EET-306 SAYISAL KONTROL LABORATUVARI
DENEY FÖYÜ

Arş. Gör. Deniz KORKMAZ

Arş. Gör Ahmet TOP

DENEY:1 SAYISAL SİNYALLERİN İŞLENMESİ

Amaç:

Sayısal türev ve integrasyon yöntemlerinin MATLAB kullanarak incelenmesi.

I. Ön Bilgi ve deneye (uygulamaya) hazırlık:

1-) Bir $f(x)$ fonksiyonunun türevi, x bağımsız değişkenine göre $f(x)$ fonksiyonunun değişim oranı olarak tanımlanır :

$$f'(x) = \frac{df(x)}{d(x)}$$

Sayısal türev alma tekniği ile bir fonksiyonun x_k noktasındaki türevinin değeri yaklaşık olarak hesaplanabilir. Bu hesaplamada x_k noktaları yakınındaki fonksiyon değerlerini kullanmak sureti ile x_k daki teğetin eğimi yaklaşık olarak bulunmuş olur. x_k noktasındaki teğet çizgisinin eğiminin yaklaşık sayısal hesaplaması, fark alma yöntemi olarak bilinir.

a) Geriye Fark Alma Kuralı: $f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$

b) İleri Fark Alma Kuralı: $f'(x_k) = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}$

c) Merkezi Fark Alma Kuralı: $f'(x_k) = \frac{f(x_{k+1}) - f(x_{k-1}))}{x_{k+1} - x_{k-1}}$

Geriye Fark Alma Kuralı ile $f(x)=x^5-4x^4-9x^3+32x^2+28x-48$ fonksiyonunun $[-3, 5]$ aralığındaki türevini bulan bir MATLAB programı yazarsak;

```
x(1)=-3; T=0.1;
f(1)=x(1)^5-4*x(1)^4-9*x(1)^3+32*x(1)^2+28*x(1)-48;
df(1)=468; // df(1)=468 değeri sizce nasıl belirlenmiştir ?
for i=2:80;
    x(i)=x(i-1)+T;
    f(i)=x(i)^5-4*x(i)^4-9*x(i)^3+32*x(i)^2+28*x(i)-48;
    df(i)=(f(i)-f(i-1))/T;
end
plot(x,df)
```

Bu işlem MATLAB da hazır bir fonksiyon ile de yapılabilir:

diff (Fark Alma) fonksiyonu: Türev alma işlemlerinde MATLAB içinde yer alan **diff** fonksiyonundan yararlanılabilir. **diff** fonksiyonu bir vektör (sıra veya sütun matrisi) içindeki birbirini izleyen değerler arasındaki farkları hesaplar. Bu hesaplamadan, eleman sayısı orijinal vektör elemanlarından bir eksik olan bir vektör elde edilir. Genel kullanım biçimleri aşağıda verilmiştir.

y=diff(x) (x vektörü içerisindeki değerlerin birinci dereceden farkını alır).

x=[x(1) x(2) x(3)...x(n)]

y=[x(2)-x(1) x(3)-x(2) ... x(n)-x(n-1)]

y=diff(x,n) (x vektörü içerisindeki değerlerin n. dereceden farkını alır,yani **diff(x,n)=diff(diff(x,n-1))**).

x=[1 3 6 8 10];

diff(x) komutunun sonucu

2 3 2 2 ve

diff(x,2) komutunun sonucu

1 -1 0

olur. Bu açıklamalara göre diff fonksiyonu kullanılarak **y=f(x)** biçiminde tanımlanan bir fonksiyonun türev değerinin sayısal yaklaşığı:

diff(y)./diff(x) şeklinde hesaplanır.

Not: Yukarıdaki açıklamalardan anlaşılacağı üzere **diff** fonksiyonu ileri farkları hesaplamaktadır.

f(x)=x⁵-4x⁴-9x³+32x²+28x-48 fonksiyonunun türevini hesaplatırıp, türevin x değişkenine göre grafiğini çizdirecek bir MATLAB programı yazalım.

Önceki örnekte olduğu gibi türev aralığını **[-3 5]** seçelim.

Bu aralığa ait sayısal türev değerleri aşağıdaki bildirimler yardımıyla hesaplanır:

x=3:0.1:5;

f=x⁵-4*x⁴-9*x³+32*x²+28*x-48;

df=diff(f)./diff(x);

w=3:0.1:4.9;

plot(w,df)

2-) Herhangi bir **f(x)** fonksiyonunun **[a b]** aralığındaki belirli integrali, a ve b arasında **f(x)** eğrisinin altında kalan alan olarak tanımlanır. İntegral değeri K ise ;

$$K = \int_a^b f(x)dx \text{ biçiminde verilir.}$$

Sayısal integrasyon teknikleri arasında en sık kullanılanları yamuk ve parabolik (Simpson) yöntemleridir. Yamuk yönteminde; $f(x)$ fonksiyonu eğrisi altında kalan alan küçük yamuk alanlarının toplamı olarak hesaplanır. Burada eğri, n adet parçaya ayrılır ve eğri üzerinde yer alan her iki nokta bir düz çizgi ile birleştirilerek eğri altındaki alan yamuk alanlarına bölünmüş olur. Parabolik veya Simpson yönteminde; $f(x)$ fonksiyonu altında kalan alan parabolik (quadratic) fonksiyonların alanları toplamı olarak hesaplanır ve yamuk yönteminden daha doğru sonuç verir.

Yamuk kuralı ile $f(t)=20\sin 2t$ fonksiyonunun $[0 \ 0.5]$ aralığında integralini bulan bir MATLAB programı yazalım;

```
t(1)=0; T=0.01; top(1)=0;
f(1)=20*sin(2*t(1));
for i=2:50;
    t(i)=t(i-1)+T;
    f(i)=20*sin(2*t(i));
    top(i)=top(i-1)+(f(i)+f(i-1))*T/2;
end
plot(t,top)
```

MATLAB içinde sayısal integrasyon hesaplamasında hazır fonksiyon olarak **quad** fonksiyonu kullanılır. **quad** fonksiyonu Simpson kuralının adaptif (uyarlamalı) biçimini kullanır.

Kullanım biçimleri:

```
a=quad('fname',a,b)
a=quad('fname',a,b,tol)
```

Fonksiyon fonksiyonu olan bu fonksiyonlar ile hesaplama yapılabilmesi için ayrıca bir fonksiyon dosyası hazırlanması gerekir. **fname**, sayısal integrasyonu yapılacak fonksiyona verilen ismi, a integrasyon işleminin başlangıç, b son değerini, tol ise integrasyon işlemindeki tolerans değerini belirler. Tolerans değeri tanımlanmadığı takdirde otomatik olarak 0.001 olarak kabul edilir.

$f(t)=20\sin(2t)$ fonksiyonunun integralini $[0 \ 0.5]$ aralığında hesaplayan program parçası aşağıda verilmiştir. .

Intgrl.m

```
function q=intgrl(t)
q=20*sin(2*t);

>>quad('intgrl',0,0.5)
q=4.5970
```

II. Uygulamada yapılacak işlemler:

1-) $y=10e^{-2t}$ fonksiyonunun $[0\ 5]$ aralığında $T=0.01$ için (T : adım aralığı)

a) türevini matematiksel olarak bulunuz ve MATLAB da çizdiriniz.

b) İleri Fark, Geri Fark veya Merkezi Fark yöntemlerinden birini kullanarak türevini alan bir MATLAB programı yazınız ve grafiğini çizdiriniz.

c) a ve b seçeneklerinde elde ettiğiniz sonuçları karşılaştırınız

d) Aynı fonksiyonun türevini farklı adım aralıkları için tekrar bulunuz. Adım aralıklarını değiştirmenin türev sonucundaki etkisini görünüz.

2-) $y=10e^{-2t}$ fonksiyonunun $[0\ 5]$ aralığında $T=0.01$ için

a) integralini matematiksel olarak bulunuz ve grafiğini MATLAB da çizdiriniz.

b) Yamuk Kuralı ile integralini alan bir MATLAB programı yazınız ve grafiğini çizdiriniz.

c) a ve b seçeneklerinde elde ettiğiniz sonuçları karşılaştırınız.

III. Sorular ve değerlendirme:

1-) Sayısal türev ve integral için matematiksel çözüm ile MATLAB da bir program yardımıyla çözüm arasında sonuç olarak ne fark vardır? Varsa bu farklılığın sebebi nedir? İki sonucu birbirine yaklaştırmak için ne yapılmalıdır?

2-) **quad** fonksiyonunu kullanarak $y=10e^{-2t}$ nin $[0\ 5]$ aralığında $T=0.01$ için integralini bulan ve bulunan integral fonksiyonun grafiğini çizdiren bir MATLAB programı yazınız.

DENEY:2 MATLAB KONTROL SİSTEMLERİ TOOLBOX'IN AYRIK ZAMAN FONKSİYONLARI VE KULLANIMI

Amaç:

MATLAB kontrol sistemleri toolbox'ının açık ve kapalı çevrim sistem cevaplarının bulunması ile ilgili fonksiyonların incelenmesi.

I. Ön Bilgi ve deneye (uygulamaya) hazırlık:

MATLAB da ayrık zaman sistemlerinin incelenmesinde kullanılan çeşitli ayrık zaman analiz ve tasarım fonksiyonları mevcuttur. Kontrol sistem toolbox'ta yer alan bu fonksiyonlar genellikle, karşılık gelen sürekli zaman fonksiyonlarının önüne “d” harfi koymak suretiyle adlandırılırlar.

dimpulse: Bu fonksiyon sistemlerin ayrık zaman ani darbe cevabını bulur.

Transfer fonksiyonu veya durum uzay modeli verilmiş ise;

$[y \ x \ k]=\text{dimpulse}(\text{num},\text{den},k)$

$[y \ x \ k]=\text{dimpulse}(a,b,c,d,iu,k)$ (y:çıkış matrisi, x:durum vektörü değerleri, k:adım sayısı, num: transfer fonksiyonu payının katsayıları, den: transfer fonksiyonu paydasının katsayıları, iu:giriş sayısı) fonksiyonları yardımı ile cevap belirlenir.

dstep: Ayrık zaman sistemlerin basamak cevabını bulur.

dstep in kullanım biçimleri **dimpulse** ile aynıdır.

dlsim: Ayrık zaman sistemlerin tanımlanan bir giriş için cevabını bulur.

Transfer fonksiyonu verilmiş ise;

$[y,x]=\text{dlsim}(\text{num},\text{den},u)$ (u:giriş sinyali)

durum uzay modeli verilmiş ise;

$[y,x]=\text{dlsim}(a,b,c,d,u,x0)$ (x0:başlangıç koşulları)

yardımı ile çıkış belirlenir.

stairs: Bu komut sistemlerin eşit biçimde artan değerler içeren bir giriş vektörüne karşılık gelen çıkış grafiğini basamak şeklinde çizdirir.

stairs(x,y) (x'e bağlı y değerlerinin basamak grafiğini çizdirir)

stairs(y) (y değerlerinin basamak grafiğini çizdirir)

plot fonksiyonuyla doğrusal grafik çizdirilirken, **stairs** fonksiyonuyla basamak şeklinde grafik çizdirilir.

ÖRNEK-1:

$$\begin{bmatrix} x1(k+1) \\ x2(k+1) \end{bmatrix} = \begin{bmatrix} 1.955 & -0.96 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x1(k) \\ x2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} U(k) \quad y(k) = \begin{bmatrix} 0.01 & 0.02 \end{bmatrix} \begin{bmatrix} x1(k) \\ x2(k) \end{bmatrix}$$

Verilen ayrık zaman durum denkleminin $k=1,2,3,\dots,100$ için,

a-) birim basamak cevabını,

b-) birim impulse cevabını ve,

c-) $u(k)=e^{-2k}\sin(3k/20)$ giriş sinyali için çıkış cevabını bulan ve bu üç cevabı 3×1 ekranda çizen bir MATLAB programı yazınız.

```
a=[1.955 -0.96; 1 0]; b=[1;0]; c=[0.01 0.02]; d=0; x0=[0;0];
k=1:1:100;
y1=dstep(a,b,c,d,1,k);
subplot(3,1,1)
stairs(k,y1)
y2=dimpulse(a,b,c,d,1,k);
subplot(3,1,2)
stairs(k,y2)
u(k)=exp(-2.*k).*sin(3.*k/20);
y3=dlsim(a,b,c,d,u(k),x0);
subplot(3,1,3)
stairs(k,y3)
```

ÖRNEK-2:

$G(z) = \frac{0.3404z + 0.2415}{z^2 - 0.4455z + 0.6094}$ olarak verilen sistemin basamak cevabını bulan bir MATLAB program parçası aşağıda verilmiştir.

```
num=[0.3404 0.2415]; den=[1 -0.4455 0.6094];
dstep(num,den)
```

Bunların dışında sürekli zaman sistemlerde kullanılan; model dönüştürme fonksiyonları, **ss2tf**, **tf2ss**, **series**, **paralel**, **cloop**, **feedback** fonksiyonları ayrık zaman sistemlerde de aynı biçimde kullanılabilir.

Örneğin, **ss2tf** fonksiyonu durum uzay modelini transfer fonksiyonu modeline dönüştürür. Genel kullanım biçimi:

[num den] = ss2tf(a, b, c, d, iu) şeklindedir.

ÖRNEK-3:

$$\begin{bmatrix} x1(k+1) \\ x2(k+2) \end{bmatrix} = \begin{bmatrix} 0.3 & 0.04 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x1(k) \\ x2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} U(k) \quad y = \begin{bmatrix} 0 & 2 \end{bmatrix} \begin{bmatrix} x1(k) \\ x2(k) \end{bmatrix} \quad \text{Ayrık zaman durum uzay}$$

modeline karşılık gelen transfer fonksiyonunu bulan bir MATLAB program parçası aşağıda verilmiştir.

```
A=[0.3 0.04 ; 1 0]; B=[1; 0];
C=[0 2]; D=0;
[num den] = ss2tf(A, B, C, D, 1)
```

tf2ss: Transfer fonksiyonu modelinden durum uzay modeline dönüşümü sağlar. Genel kullanım biçimi:
[a, b, c, d] = tf2ss(num,den) şeklindedir.

ÖRNEK-4:

$G(z) = \frac{0.064z^2 + 0.128z + 0.064}{z^2 - 1.168z + 0.424}$ transfer fonksiyonuna karşılık gelen durum uzay matrislerini bulan bir

MATLAB program parçası aşağıda verilmiştir.

```
num=[0.064 0.128 0.064];  
den=[1 -1.168 0.424];  
[a, b, c, d] = tf2ss(num,den)
```

series: Transfer fonksiyonu veya durum uzay modeli ile tanımlanmış seri bağlı iki sistemin indirgenmesini sağlar. Genel kullanım biçimleri:

```
[a b c d] = series (a1,b1,c1,d1,a2,b2,c2,d2)  
[a b c d] = series (a1,b1,c1,d1,a2,b2,c2,d2,outputs1,inputs2)  
[num den]= series(num1,den1,num2,den2) şeklindedir.
```

İlk iki kullanım biçiminde, durum uzay modeli ile tanımlanan iki sistemden durum uzay modeli ile tanımlanan bir sistem elde edilir. Birinci komutta tek girişli- tek çıkışlı sistemlerin seri bağlantısı sağlanmakta, buna karşılık ikinci komutta çok girişli çok çıkışlı sistemlerin belirtilen çıkıştan girişe seri indirgenmesi gerçekleştirilmektedir.

ÖRNEK-5:

Transfer fonksiyonları $G_1(z) = \frac{10z + 5}{z - 1}$ ve $G_2(z) = \frac{1}{z - 0.2}$ olarak verilen seri bağlı iki sistemi

indirgeyen bir MATLAB program parçası aşağıda verilmiştir.

```
num1=[10 5]; den1=[1 -1];  
num2=[1]; den2=[1 -0.2];  
[num den]= series(num1,den1,num2,den2)
```

paralel: Paralel bağlı iki blok veya sistemin indirgenmesini gerçekleştirir. Kullanım biçimi aynı **series** fonksiyonunda olduğu gibidir.

feedback: Herhangi bir geribesleme döngüsünü indirger. Bu fonksiyonun genel kullanım biçimi aşağıda verilmiştir:

feedback(G,H) veya **feedback(G,H,-1)** negatif geri besleme için

feedback(G,H,1) ise pozitif geri besleme için kullanılır.

```
[num den]=feedback(num1,den1,num2,den2)
```

ÖRNEKLER:

feedback(G,1) (pozitif birim geri besleme) (**G:** İleri yol transfer fonksiyonu)

feedback(G) (negatif geri besleme)

feedback(G,-1) (negatif geri besleme) şeklindedir.

ÖRNEK-6:

$G(z) = \frac{5z}{z^2 - 1.1z + 0.1}$ olan birim geri beslemeli bir sistem için kapalı çevrim sistemin birim basamak

cevabını bulan ve grafiğini çizen bir program yazınız.

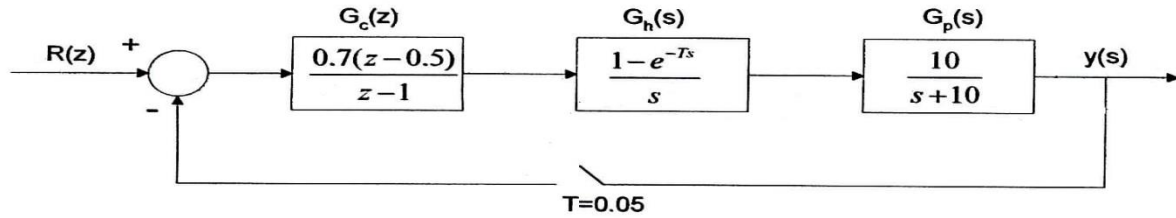
```
num=[5 0];  
den=[1 -1.1 0.1];  
G=tf(num,den);  
s=feedback(G)  
step(s)
```

II. Uygulamada yapılacak işlemler:

$$1-) \begin{bmatrix} x1(k+1) \\ x2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & -0.0952 \\ 0 & 0.905 \end{bmatrix} \begin{bmatrix} x1(k) \\ x2(k) \end{bmatrix} + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} U(k) \quad y(k) = \begin{bmatrix} 10 & 0 \end{bmatrix} \begin{bmatrix} x1(k) \\ x2(k) \end{bmatrix}$$

$k=1,2,3,4,\dots,100$ olmak üzere $u(k)=\sin(5k)$ giriş sinyali için sistem cevabını bulan bir **MATLAB** programı yazınız.

2-) Aşağıda verilen sistemin birim basamak cevabını bulan bir program yazınız.



III. Sorular ve değerlendirme:

1-)Verilen bir transfer fonksiyonunun ayrık zamanda birim basamak cevabını bulan (**dstep** fonksiyonunun görevini yapan) bir **MATLAB** fonksiyonu yazınız (Temel **MATLAB** operatörleri kullanılabilir ancak kontrol **toolbox** fonksiyonları kullanılmayacaktır).

DENEY:3 ÖRNEKLEME PERİYODU SEÇİMİ VE GECİKMELERİN ETKİSİ

Amaç:

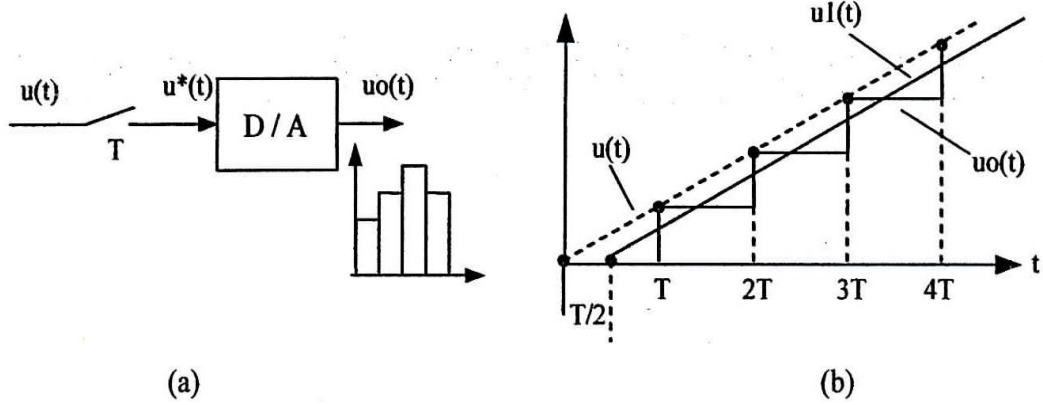
Ayrık zaman sistemlerde sistem performansını etkileyen gecikmelerin incelenmesi ve örnekleme periyodu seçimi.

I. Ön Bilgi ve deneye (uygulamaya) hazırlık:

Ayrık zaman sistemlerde gecikmelerin başlıca iki nedeni vardır:

1-) D/A Dönüştürücüden Kaynaklanan Gecikme ve Kontrol Performansına Etkisi:

Bir ayrıklaştırma yöntemiyle elde edilen ayrık zaman sistemin , orijinal sürekli zaman sistemle aynı kapalı çevrim performansını sağlayıp sağlamayacağı Şekil 5.1 'de verilen D/A 'nın çalışması incelenerek belirlenebilir.



Şekil 5.1

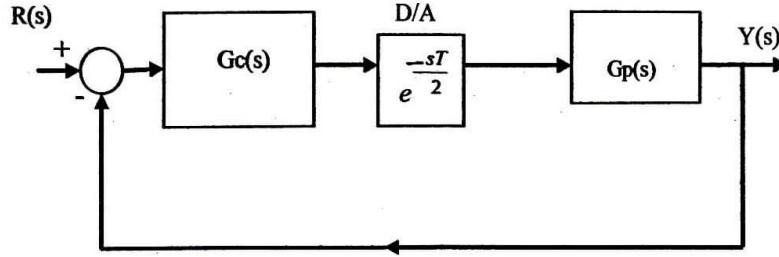
- (a) D/A' nın blok şeması
(b) D/A 'da işlenen sinyaller

Bir D/A çeviricinin yaptığı iş temel olarak $u(t)$ 'den örneklenmiş $u(k)$ değerini bir örnekleme süresince yani $u(k+1)$ gelene kadar sabit tutarak çıkışa göndermektir. Yukarıdaki şekilde görüldüğü gibi, z.o.h. için $u_o(t)$ merdiven şeklinde oluşmaktadır. Buna göre $u_o(t)$ 'nin ortalama değeri olan $u_1(t)$, $u^*(t)$ 'nin tanımladığı zarftan (orijinal analog $u(t)$ sinyalinden) $T/2$ kadar gecikmiş durumdadır. Bu gecikme özellikle T 'nin büyük değerlerinde, sistemin performansını oldukça kötüleştirir ve hatta kararsızlığa bile neden olabilir.

Şimdi T 'nin değerinin sistemin kapalı çevrim performansına etkisini görebilmek için Şekil 5.2 'de görüldüğü gibi $T/2$ kadar gecikme eklenmiş analog kontrol sistemini ele alalım. Laplace dönüşümünün gecikme teoremini kullanırsak:

$$L\{\delta(t-T)\} = \int \delta(t-T)e^{-st} dt = e^{-sT}$$

D/A dönüştürücüden kaynaklanan gecikmeyi $e^{-s\frac{T}{2}}$ olarak ifade edebiliriz.



Şekil 5.2 D/A dönüştürücüden kaynaklanan gecikmenin eklendiği analog kontrol sistemi

Şekil 5.2 'deki sistemin değişik $T/2$ gecikmeleri için simülasyonu yapılırsa, bu gecikme süresi arttıkça kontrol performansının kötüleşeceği rahatlıkla görülebilir.

Şimdide analog olarak tasarlanmış denetleyicilerin ayrık zaman eşdeğerlerini bulurken örnekleme periyodu seçiminin nasıl yapılması gerektiğini inceleyelim. Örnekleme frekansı $\omega_s = \frac{2\pi}{T}$ dir. Örnekleme

Frekans Oranı yaklaşık olarak, $\zeta = 0.707$ civarında, $\dot{O}FO = \frac{\omega_s}{\omega_n}$ olarak tanımlanır. **Şu unutulmamalıdır**

ki, sistemin kapalı çevrim performansındaki bozulmaların derecesi T ' nin sayısal değerinden ziyade örnekleme frekansının doğal frekansa oranına yani $\dot{O}FO$ 'ya bağlıdır.

Eğer, kapalı çevrim sistemin doğal frekansı ω_n olacak şekilde sürekli zamanda bir $Gc(s)$ denetleyicisi tasarlanır ve bu denetleyici çift yönlü doğrusal dönüşüm yöntemi ile ayrıklaştırılırsa $\omega_s \geq 20\omega_n$ seçilmelidir (Yani $\dot{O}FO \geq 20$ olmalıdır). Ancak uygulamada, yüksek $\dot{O}FO$ değerlerinin seçilmesine en büyük engel; A/D dönüştürücüye gelen sinyallerdeki gürültü problemidir. Bu nedenle eğer sürekli zamanda tasarladığınız $Gc(s)$ 'i mikroişlemci gerçeklemesinde kullanmak istiyorsanız $\dot{O}FO$ 'nun;

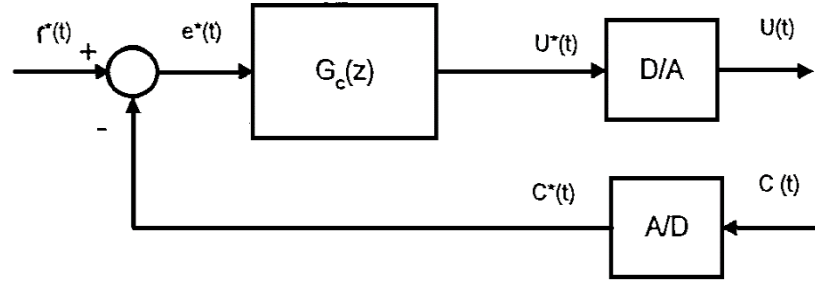
$$20 \leq \dot{O}FO \leq 40$$

olarak belirlenmesi makul bir yaklaşım olacaktır.

Eğer $\dot{O}FO$ yukarıda verilen aralıkta seçilemiyorsa, yani çeşitli nedenlerden dolayı düşük seçilmek zorunluluğu varsa (Bu nedenler yavaş bir mikroişlemci, düşük bit sayılı bir A/D veya gürültü olabilir), bu durumda $Gc(s)$ kullanılamaz ve $Gc(z)$ direk olarak z-düzleminde tasarlanmalıdır. z-düzleminde de tasarım yapılırsa, $\dot{O}FO$ ' nun 5'den küçük alınması tavsiye edilmez. Çok düşük $\dot{O}FO$ 'ların sistemi kararsız yapacağı unutulmamalıdır.

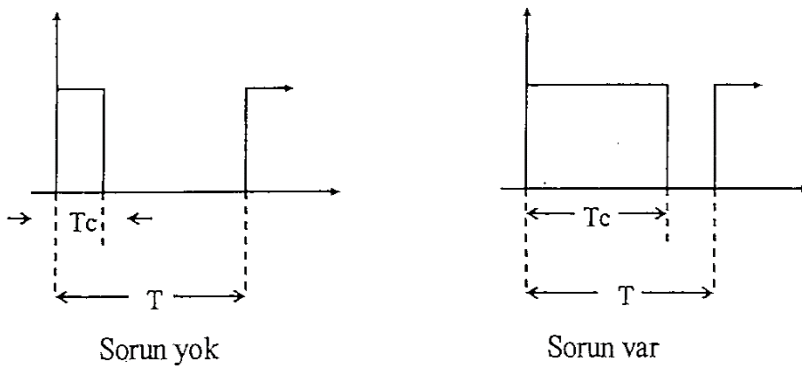
$\dot{O}FO$ 'nun yüksek olması durumunda ($\dot{O}FO \geq 20$), z-düzleminde yapılan analiz ve tasarımın hiçbir sakıncası yoktur. Ancak bu tasarım, s-düzleminde yapılan tasarımla yaklaşık aynı sonucu verecektir (yani yaklaşık aynı kontrol performansını verecektir).

2-) Hesaplama Gecikmesi:



Şekil 5.4 D/A , A/D dönüştürmeleri ve mikroişlemci hesaplama işlemleri

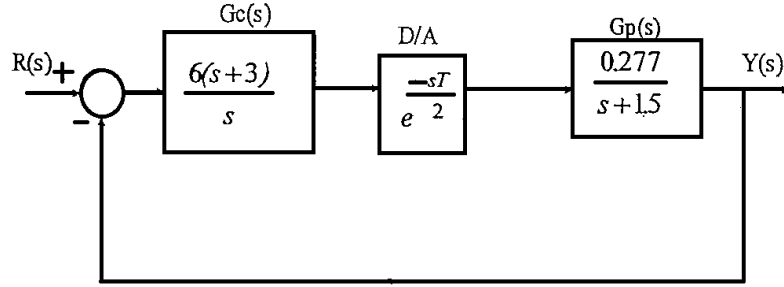
Giriş referans sinyali $r^*(t)$ klavyeden gelen sayısal (dijital) bir sinyaldir. Mikroişlemci her kT saniye ($k=0,1,2,3,\dots$) anında $c^*(t)$ 'yi alır ve $u^*(t)$ 'yi çıkış olarak verir. İdeal olarak, $c(k)$ çıkış sinyalinin ölçülmesi ve $u(k)$ denetleyici çıkışının D/A dönüştürücüye gönderilmesinin $t=kT$ anında olması gerekir. Ancak bu fiziksel olarak mümkün değildir. Eğer denetleyici çıkışının hesaplama süresi T_c , örnekleme periyodu T 'den oldukça küçükse ($T_c \ll T$), T_c ile ifade edilen gecikme ihmal edilebilir. Fakat T_c T 'ye yakın bir değerse (ki bunun anlamı mikroişlemci yavaştır veya T çok küçük seçilmiştir veya denetim kuralı karmaşık bir hesaplama dayalıdır), bu durumda sistemin kapalı çevrim cevabında T_c 'den kaynaklanan gecikmeden dolayı bozulmalar görülür.



Bu problemin çözümü, denetlenecek sistemin z-transfer fonksiyonunu z^{-1} ile çarpmak ve yeniden bir denetleyici tasarlamaktır. Denetlenecek sistemin transfer fonksiyonunu z^{-1} ile çarpmak, T süresi kadar gecikmeyi dikkate almak demektir.

II. Uygulamada yapılacak işlemler:

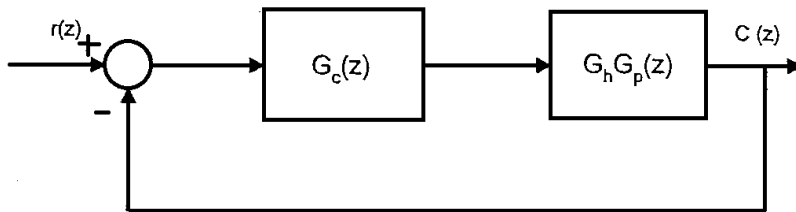
1-)



Yukarıdaki D/A yokken ($T=0$), $G_c(s)$ denetleyicisi $\omega_n = 2.23 \text{ rad/s}$, $\zeta = 0.707$ olacak şekilde tasarlanmıştır. D/A'lı ve D/A'sız olmak üzere iki SIMULINK modeli oluşturunuz. $R(s)$ giriş sinyalini, oluşturulan iki SIMULINK modelinin girişine verip devrelerin çıkışlarını aynı anda görüntüleyip karşılaştırınız. Bu işlem D/A'nın farklı gecikme değerleri için (yani farklı T değerleri için, örneğin $T=0.05$ den 1'e kadar farklı birkaç değer kullanınız) gerçekleştirilip, gecikmenin olumsuz etkisini gözlemleyiniz..

2-) Yukarıda verilen sistemde ÖFO = 4, 10 ve 20 olacak şekilde denetleyiciyi bilineer dönüşümle ayrıştırarak üç sisteminde çıkışını aynı anda izleyebilecek şekilde bir SIMULINK blok diyagramı oluşturunuz (D/A yerine z.o.h. bloğunu kullanınız).

3-)



$$G_c(z) = \frac{0.5(z-0.8)}{z-1}, \quad G_h G_p(z) = \frac{0.2}{z-0.9} \quad \text{ve referans giriş birim basamak fonksiyonudur. Bir}$$

periyotluk hesaplama gecikmesinin etkisini görebilmek için gecikmeli ve gecikmesiz sistem çıkışlarını aynı anda izleyebileceğiniz bir SIMULINK blok diyagramı kurunuz ($T = 0.5$ alınız).

III. Sorular ve değerlendirme:

1-) Yukarıda bahsedilen gecikme türlerine ilave olarak uygulamada bilgisayar kontrollü bir sistemde sizce başka ne tür gecikmeler olabilir?