

ELE 515
MATLAB İLE BİLGİSAYARLI KONTROL
2022-23/II, ÖDEV-2 RAPOR

Adı	Doruk
Soyadı	Bilgi
Öğrenci No	221211041
Tarih	22.03.2023

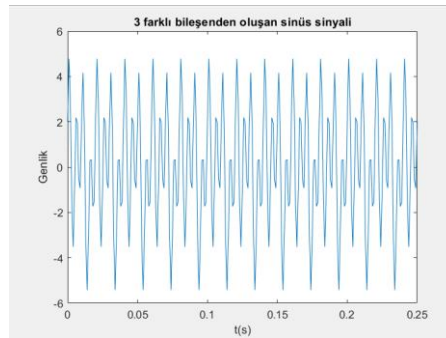
İçindekiler

1. SORU 1.....	3
2. SORU 2.....	12
3. SORU 3.....	25
4. Referanslar	29

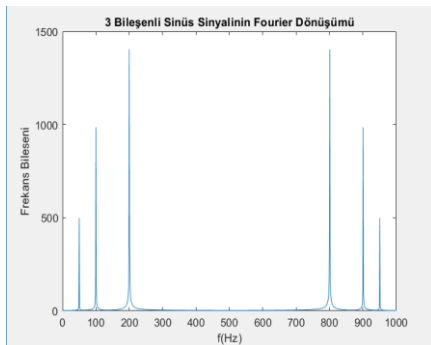
1. SORU 1

1. Soruda $[b,a] = \text{filtre01}(fs, \text{gecirmeFrekanslari}, \text{'durtuTepkisiCizdir'}, \text{true|false}, \text{'bodeCizdir'}, \text{true|false}, \text{'ornekVer'}, \text{true|false}, \text{'ornekSayisi'}, N)$ genel formu ile kullanılacak olan “filtre01” fonksiyonu tasarlanmıştır. Bu fonksiyon ile verilen giriş parametrelerine karşılık tasarlanması istenilen filtrenin pay ve payda katsayıları sırasıyla “b” ve “a” değişkenlerine kaydedilmektedir. Katsayılar elde edildikten sonra filtrelenmiş çıkış sinyali elde etmek için “filter” komutuna giriş sinyali ile birlikte bu katsayılar argüman olarak verilmektedir.

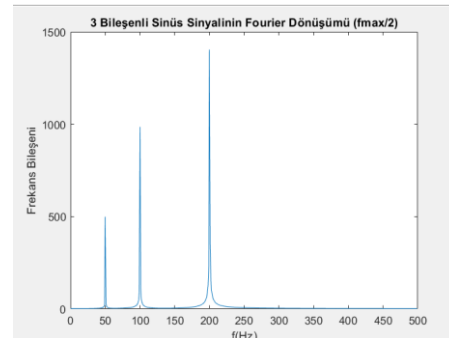
“filtre01” fonksiyonu ile filtre tasarımı öncesinde 1000 Hz örnekleme frekansında üç bileşenli sinüs sinyali $u = \sin(2\pi f_1 t) + 2\cos(2\pi f_2 t) + 3\sin(2\pi f_3 t)$ ve “fu1”, “fu2”, “fu3” değerleri sırasıyla 50 Hz, 100 Hz, 200 Hz frekans değerlerinde oluşturulmuştur ardından “t” zaman vektörü ($t = 0:Ts:50/fu1$) tanımlanmıştır. Filtre giriş sinyali olarak oluşturulmuş olan üç bileşenli sinüs sinyali “t” zaman vektörüne karşılık çizdirilmiştir. Üç bileşenli sinüs sinyali Şekil-1’de gösterilmektedir. Tasarlanmış olan filtre “u” giriş sinyaline uygulandıktan sonra, filtreye verilen frekans değerlerinde istenilen frekans aralıklarının geçirilip geri kalan frekans aralıklarının bastırıldığı durumu gözlemleyebilmek adına “u” giriş sinyalinin Fourier Dönüşümü “fft” komutu ile alınarak “ufft” değerine kaydedilmiştir. “ufft” değeri bulunduktan sonra bu değerlere karşılık frekans bileşeni “f” ($f = \text{linspace}(0,fs,\text{length}(\text{ufft}))$) oluşturulmuştur. “plot(f,abs(ufft))” komutu ile giriş sinyalinin Fourier Dönüşümü Şekil-2’de gösterildiği gibi çizdirilmiştir. Giriş sinyalinin filtrelenmeden önceki Fourier Dönüşümü ile filtrelendikten sonraki halinin (çıkış sinyalinin) rahatlıkla gözlemlenebilmesi, verilen frekans aralıklarının geçirildiği geri kalan frekans aralıklarının bastırıldığı durumun daha rahat gözlemlenebilmesi adına “fmax = fs/2” için ilgili sinyalin Fourier Dönüşümü tekrardan Şekil-3’te çizdirilmiştir. Şekil-3’e bakarak istenilen frekans aralıklarında çıkış sinyali elde etmek için “filtre01” fonksiyonu ile kullanılacak olan filtrenin katsayıları elde edilmektedir.



Şekil 1: Üç Bileşenli Sinüs Giriş Sinyali

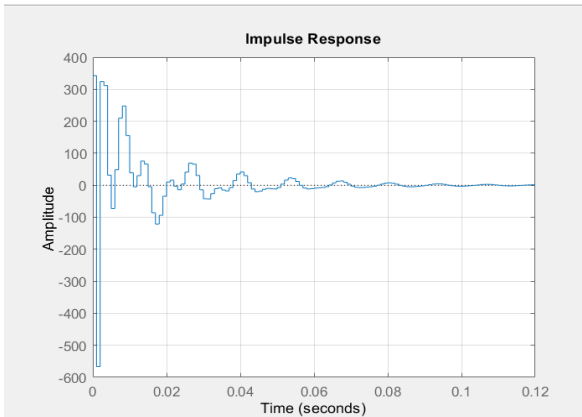


Şekil 2: Üç Bileşenli Sinüs Sinyalinin Fourier Dönüşümü

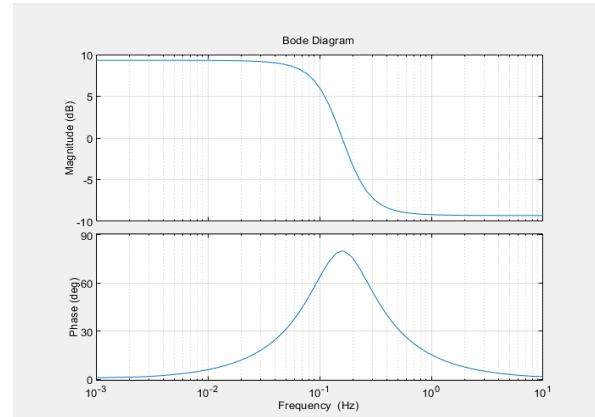


Şekil 3: Üç Bileşenli Sinüs Sinyalinin Fourier Dönüşümü ($fs = f_{\max}/2$)

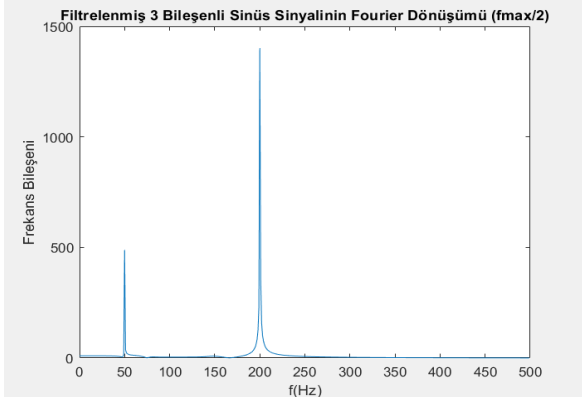
Şekil-3'ten yola çıkılarak istenilen frekans aralıklarında filtrelenmiş çıkış sinyalini elde etmek için "filtre01" fonksiyonu kullanılarak istenilen filtre tasarlanmaktadır. ("filtre01" fonksiyonunun tasarım aşamaları raporun devamında tarif edilmektedir) Tasarlanmış olan filtrenin katsayıları ve üç bileşenli giriş sinyali, "filter" komutuna "y=filter(b,a,u)" olarak girildikten sonra "y" çıkış sinyali elde edilmektedir. "y" çıkış sinyali elde edildikten sonra giriş-çıkış sinyalleri üst üste çizdirilmektedir. Grafik başlığında kullanılmak üzere filtrenin "fu2 = 100 Hz" frekansında "faz" ve "açı" değerleri sırasıyla "mag" ve "phase" değişkenlerine, filtrenin transfer fonksiyonu "tf_filtre = tf(b,a)" şeklinde elde edildikten sonra "bode" fonksiyonu ile "[mag,phase] = bode(tf_filtre,fu2)" tarafından kaydedilmektedir. (https1) Giriş-çıkış sinyallerinin bulunduğu grafikte kullanılacak olan başlık "title([num2str(fu2),'Hz için filtre genliği',num2str(20*log10(mag)), 'dB ve fazı ',num2str(phase), char(176)])" şeklinde oluşturulmaktadır. Şekil-3'ten yola çıkılarak ilk olarak üç bileşenli sinüs sinyalinin 20 Hz - 75 Hz ve 150 Hz - inf aralığında geçirildiği durum, ikinci olarak 1 Hz - 40 Hz ve 150 Hz - inf aralığında geçirildiği durum ve üçüncü olarak 20 Hz - 60 Hz ve 400 Hz - inf frekans aralığında geçirildiği durum için "filtre01" fonksiyonu kullanılarak ilgili filtre katsayıları elde edilmiştir. İlk aşamada belirtilen frekans aralıkları için tasarlanmış olan fonksiyon "filtre01(1000, [20 75; 150 Inf], 'bodeCizdir', true, 'durtuTepkisiCizdir', true)" şeklinde, ikinci ve üçüncü aşamada ise belirtilen frekans aralıkları için tasarlanmış olan fonksiyon sırasıyla "filtre01(1000, [1 40; 150 Inf], 'durtuTepkisiCizdir', true, 'bodeCizdir', true)" ve "filtre01(1000, [20 60; 400 Inf], 'durtuTepkisiCizdir', true, 'bodeCizdir', true)" şeklinde kullanılmaktadır. Opsiyonel 'bodeCizdir' ve 'durtuTepkisi' argümanları farklı sırayla verilmektedir. Bu durum karşısında "filtre01" fonksiyonu argüman sırasından bağımsız bir şekilde filtrenin bode çizgesini ve filtrenin dürtü tepkisini çizdirmektedir. (https2) Filtrenin, "filtre01" fonksiyonuna ilgili aşamalarda verilmiş olan aralıklarda istenilen şekilde çalışıp çalışmadığı durumlar çıkış sinyalinin Fourier Dönüşümünün çizdirilmesiyle gözlemlenebilmektedir. Birinci aşamada elde edilmiş olan filtrenin dürtü tepkisi, bode diyagramı, filtrelenmiş çıkış sinyalinin Fourier Dönüşümü ve giriş-çıkış sinyallerinin üst üste çizimi sırasıyla Şekil-4, Şekil-5, Şekil-6 ve Şekil-7'de gösterilmektedir.



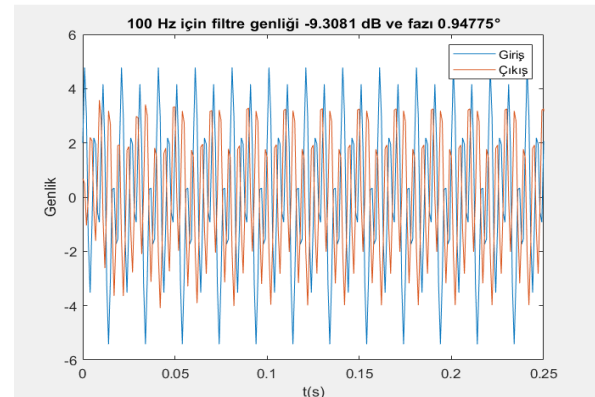
Şekil 4: Birinci aşama [20 75;150 inf] Dürtü Tepkisi



Şekil 5: Birinci aşama [20 75;150 inf] Bode Çizgesi

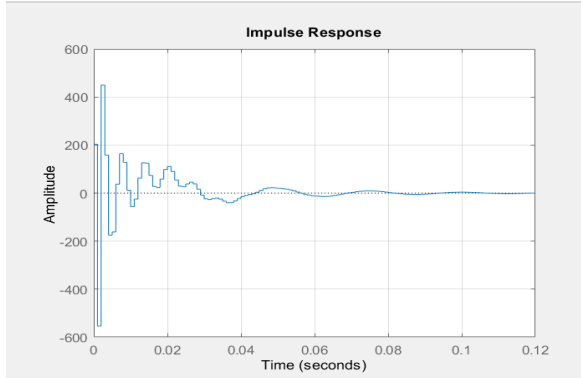


Şekil 6: Birinci aşama [20 75;150 inf] Fourier Dönüşümü

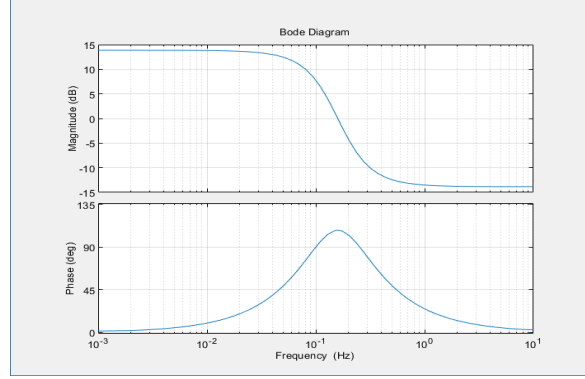


Şekil 7: Birinci aşama [20 75;150 inf] Giriş-Çıkış Sinyalleri

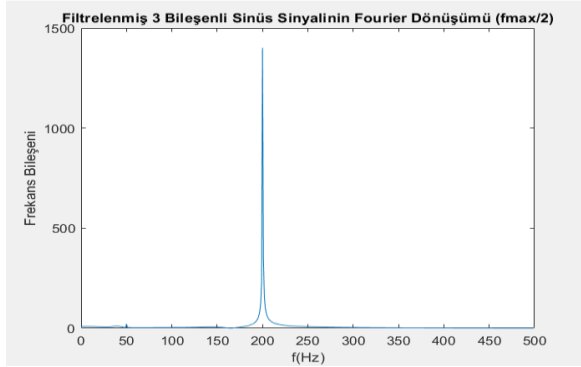
İkinci aşamada elde edilmiş olan filtrenin dürtü tepkisi, bode diyagramı, filtrelenmiş çıkış sinyalinin Fourier Dönüşümü ve giriş-çıkış sinyallerinin üst üste çizimi sırasıyla Şekil-8, Şekil-9, Şekil-10 ve Şekil-11’de gösterilmektedir.



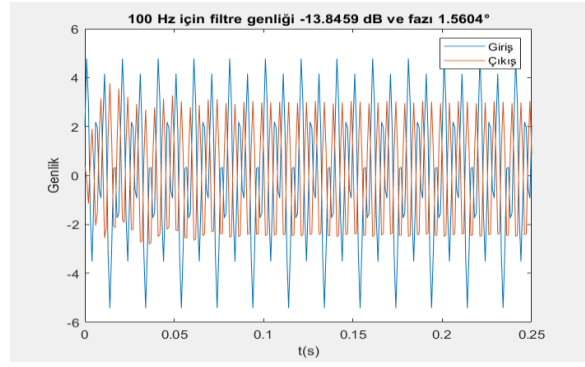
Şekil 8: İkinci aşama [1 40;150 inf] Dürtü Tepkisi



Şekil 9: İkinci aşama [1 40;150 inf] Bode Çizgesi

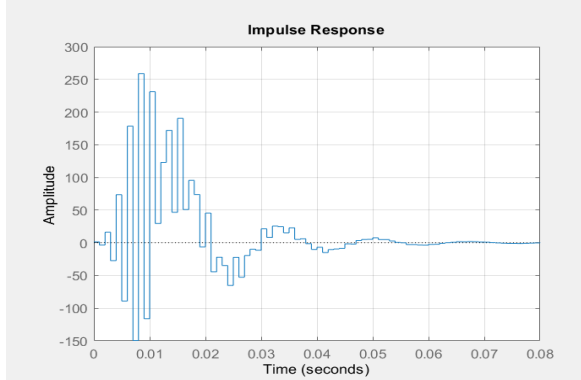


Şekil 10: İkinci aşama [1 40;150 inf] Fourier Dönüşümü

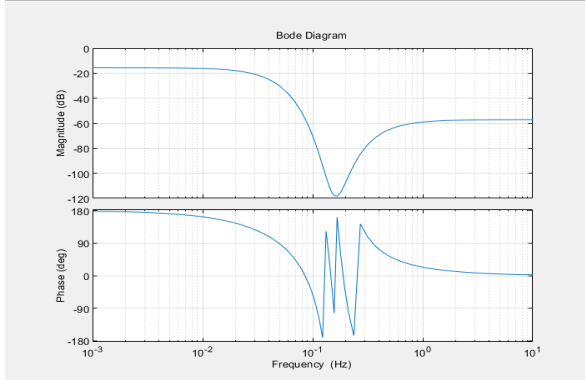


Şekil 11: İkinci aşama [1 40;150 inf] Giriş-Çıkış Sinyalleri

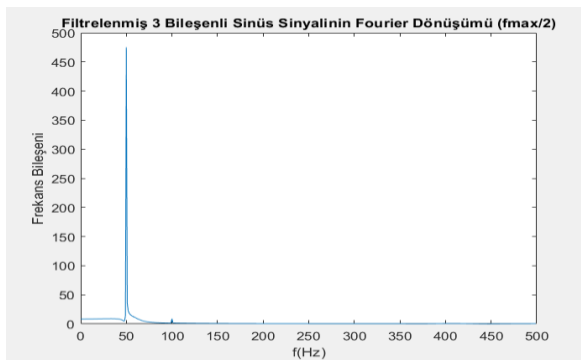
Üçüncü aşamada tasarlanan filtrenin dürtü tepkisi, bode diyagramı, filtrelenmiş çıkış sinyalinin Fourier Dönüşümü ve giriş-çıkış sinyalleri Şekil-12, Şekil-13, Şekil-14 ve Şekil-15’te gösterilmektedir.



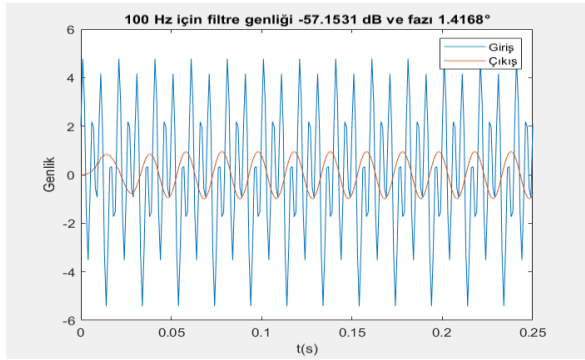
Şekil 12: Üçüncü aşama [20 60;400 inf] Dürtü Tepkisi



Şekil 13: Üçüncü aşama [20 60;400 inf] Bode Çizgesi



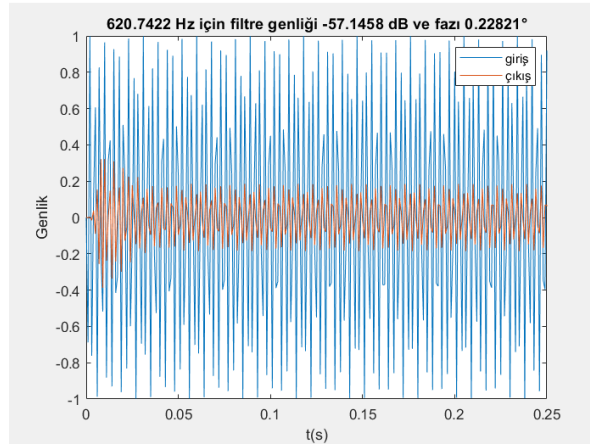
Şekil 14: Üçüncü aşama [20 60;400 inf] Fourier Dönüşümü



Şekil 15: Üçüncü aşama [20 60;400 inf] Giriş-Çıkış sinyalleri

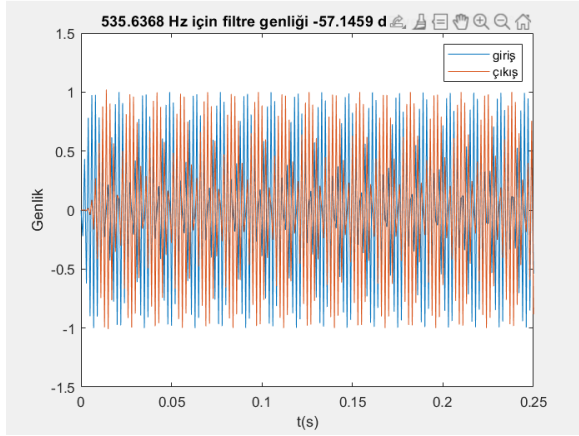
Önceki üç aşamada tarif edildiği gibi “filtre01” fonksiyonu ile tasarlanan filtreler ile üç bileşenli sinüs sinyalinin bileşenleri üzerine birinci aşamada ikinci frekans bileşeninin filtrelendiği, ikinci aşamada ilk iki frekans bileşeninin filtrelendiği ve üçüncü aşamada sadece ilk frekans bileşeninin geçirildiği durumlarda tasarlanan filtreler üzerine bir takım yorumlar yapılabilmektedir. Tasarlanan filtrelerin üçünün de birim basamak cevapları incelendiğinde tasarlanmış olan filtrelerin kararlı olduğu çıkarımı yapılmaktadır. Giriş-çıkış sinyalleri incelendiğinde birinci aşamada ikinci bileşenin çıkarılması sonucu filtrelenmiş sinyalin orijinal sinyale göre genliğinin azaldığı, ikinci aşamada birinci ve ikinci bileşenin çıkarılması sonucu genliğinin azaldığı fakat frekansının orijinal sinyale göre arttığı, üçüncü aşamada ise ikinci ve üçüncü bileşenin çıkarılması sonucu genliğinin ve frekansının azaldığı çıkarımı yapılabilmektedir.

Önceki aşamalarda “fs” ve “gecirmeFrekanslari” gerekli argümanlarına ek olarak “bodeCizdir” ve “durtuTepkisiCizdir” opsiyonel argümanları kullanılmıştı. Bu aşamada gerekli argümanlar yine üçüncü aşamadaki gibi “fs=1000”, “gecirmeFrekanslari=[20 60;400 inf]” değerlerinde olacak şekilde kullanılmıştır ancak “bodeCizdir” ve “durtuTepkisiCizdir” opsiyonel argümanları bu aşamada “false” ve “ornekVer” argümanı “true” girilmiştir ([b,a] = filtre01(1000, [20 60;400 inf], ‘durtuTepkisiCizdir’, false, ‘bodeCizdir’, false, ‘ornekVer’, true)). Sonuç gözlemlendiğinde Şekil-14 ve Şekil-15’e ek olarak 1 genlikli rastgele seçilmiş frekansta sinüs sinyaline tasarlanmış olan filtre “filtre01” fonksiyonu altında uygulanmaktadır. Giriş-Çıkış sinyalleri yine “filtre01” fonksiyonu altında çizdirilmektedir. (Önceki aşamalarda filtre katsayıları “filtre01” fonksiyonu altında elde edildikten sonra SORU1.m altında çizdirilmekteydi, bu aşamada “filtre01” altında filtre tasarlandıktan sonra yine “filtre01” fonksiyonu altında filtrelenip giriş-çıkış sinyalleri çizdirilmektedir ve ilgili başlık grafiğe eklenmektedir.) Çizdirilen örnek giriş-çıkış sinyalleri Şekil-16’da gösterilmektedir. Rastgele 1 ve 1000 aralığında üretilmiş olan 620.7422 Hz frekans değeri için, tasarlanmış olan filtrenin bu frekanstaki genlik değeri -57.1458 dB ve fazı 0.22821 derece olarak başlığa eklenmiştir.

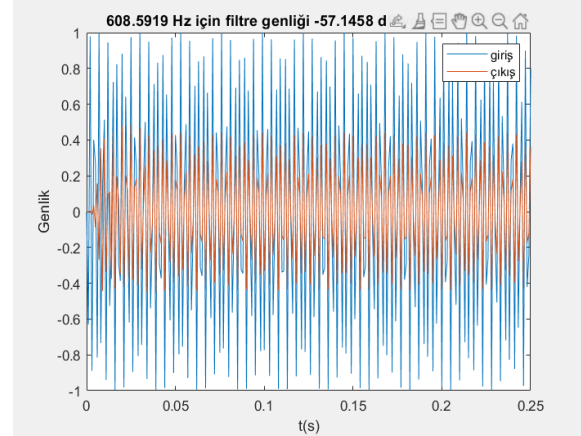


Şekil 16: “ornekVer” argümanı ile Giriş-Çıkış sinyallerinin çizdirilmesi.

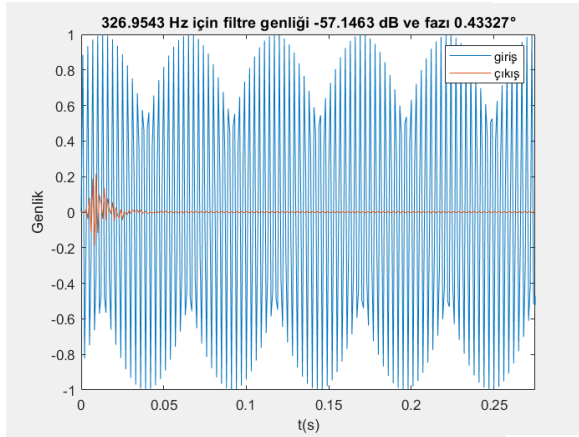
Bir önceki aşamada kullanılmış olan “filtre01” argümanlarına ek olarak bu aşamada opsiyonel “ornekSayisi” argümanı kullanılmaktadır. Bu argümanın değeri N, diğer argümanlardan farklı olarak pozitif bir tamsayı olarak girilmektedir. “N” değerinin girilmesiyle “filtre01” fonksiyonu altında bir önceki aşamada rastgele üretilmiş olan frekans değeri bu kez N adet üretilmekte ve üretilen N adet frekans için N adet giriş-çıkış sinyali üst üste çizdirilmektedir. Önceki aşamadan farklı olarak bu aşamada kullanıcı N farklı örnek alabilmektedir. “N=4”, [b,a] = filtre01(1000, [20 60;400 inf], ‘durtuTepkisiCizdir’, false, ‘bodeCizdir’, false, ‘ornekVer’, true, ‘ornekSayisi’,4); şeklinde fonksiyon argümanı olarak verildikten sonra ilgili grafikler çizdirilmiştir. N=4 adet farklı giriş-çıkış sinyalleri Şekil-17, Şekil-18, Şekil-19 ve Şekil-20’de gösterilmektedir. “ornekVer” parametresi “false” değerinde ise “ornekSayisi” parametresi için “N” değeri girilememektedir. Bu opsiyonel argüman ancak “ornekVer” parametresi “true” olduğu durumda kullanılabilmektedir.



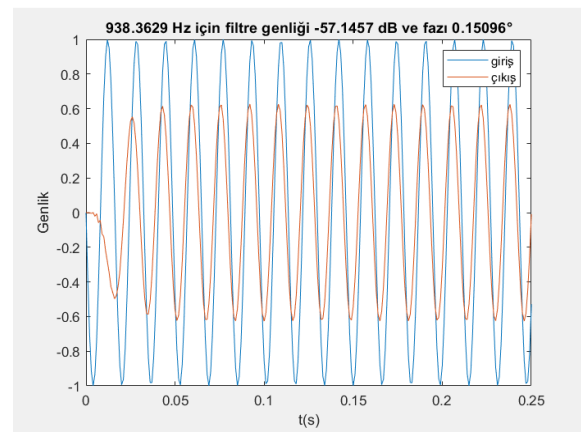
Şekil 17: "ornekSayisi" N=1



Şekil 18: "ornekVer" N=2

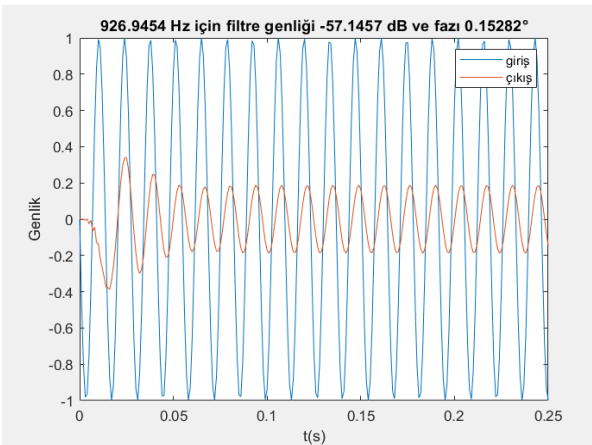


Şekil 19: "ornekSayisi" N=3

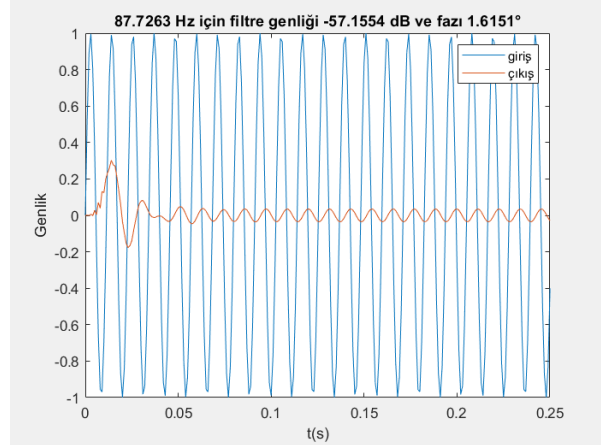


Şekil 20: "ornekSayisi" N=4

"ornekSayisi" argümanının "ornekVer" argümanı parametresinin "true" girildikten sonra "filtre01" fonksiyonu altında önceki aşamadaki argüman sırasından farklı verilmesine karşılık fonksiyon yine N adet farklı frekansta N adet farklı giriş-çıkış sinyali için gerekli işlemleri yapabilmektedir. "filtre01" fonksiyonuna ilgili argümanlar istenilen sırada verilebilmektedir. Opsiyonel argümanlar filtre fonksiyonuna verilmek zorunda değildir, kullanılmayan durumlarda verilmeyebilir. Zorunlu argümanlar filtreye verilmek zorundadır. Örneğin, [b,a] = filtre01(1000, [20 60; 400 Inf], 'ornekVer', true, 'ornekSayisi', 2, 'durtuTepkisiCizdir', false) şeklinde fonksiyon argümanları verildikten sonra 'ornekSayisi' argümanına ait ilgili grafikler Şekil-21 ve Şekil-22'de çizdirilmiştir.



Şekil 21: "ornekSayisi" N=1



Şekil 22: "ornekSayisi" N=2

Tasarlanmış olan “filtre01” fonksiyonu hatalı kullanımlara karşı anlamlı hata mesajları içermektedir. Girilmesi zorunlu olan “fs” örnekleme frekansı argümanının “double” ya da “scalar” girilmediği durumlarda (Örneğin [1 2 3] vektörü) fonksiyon “Örnekleme frekansı hatalı” hatası vermektedir. Bu durum “filtre01” fonksiyonu altında “if-else” blokları tarafından sağlanmaktadır. “~isa(fs,’double’)” ve “~isscalar(fs)” mantık değerleri tarafından bu durum kontrol edilmektedir. (https3) Girilen değer “double” veya “scalar” bir değer değilse, mantık değerleri lojik-1 dönmekte ve hata mesajı komut ekranı altında görülmektedir. Örneğin, “[b,a] = filtre01([1 2 3], [20 50; 200 Inf]);” girilmesi durumunda komut ekranında ilgili hata mesajı gösterilmektedir. Hata mesajı Şekil-23’te gösterilmektedir.

```
Command Window
Error using filtre01 (line 7)
Örnekleme frekansı hatalı.
```

Şekil 23: “Örnekleme frekansı hatalı”

Aynı şekilde zorunlu argüman olan “gecirmeFrekanslari” için de benzer hata mesajları “filtre01” altında oluşturulmuştur. Bu argüman için girilmekte olan 2x2’lik geçirilecek frekans bantlarını belirten matrisin her bir elemanının değeri “if-else” yapısı altında kontrol edilmektedir. Bu matrisin “gecirmeFrekanslari(2,2)” hariç diğer elemanlarının “inf” girildiği durumlarda “Geçirme frekansı vektörü hatalı” hata mesajı komut ekranı altında görülmektedir. Örneğin, “[b,a]=filtre01(1000, [40 -10; Inf 5; 200 Inf]);” için hata mesajı Şekil-24’te gösterilmektedir. Aynı şekilde “fmax=fs/2” değerine eşit veya büyük girilme durumlarında “Maksimum frekanstan (%d Hz) büyük frekans olmamalı.” Hata mesajı komut ekranında gösterilmektedir. Örneğin, [b,a] = filtre01(300, [40 10; 200 Inf]) için hata mesajı Şekil-25’te gösterilmektedir.

```
Command Window
Error using filtre01 (line 19)
Geçirme frekansı vektörü hatalı.
```

```
Command Window
Error using filtre01 (line 21)
Maksimum frekanstan (150 Hz) büyük frekans olmamalı.
```

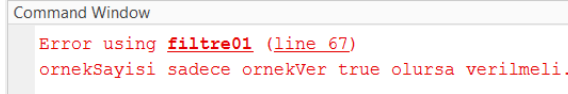
Şekil 24: “Gecirme frerkansı vektörü hatalı” Şekil 25: “Maksimum frekanstan %d Hz büyük frekans olmamalı”

“bodeCizdir”, “ornekVer” ve “durtuTepkisiCizdir” opsiyonel argümanları için de benzer şekilde hata mesajları “filtre01” altında oluşturulmuştur ancak önceki hata mesajlarından farklı olarak bu aşamada opsiyonel argümanlar için hata analizi yapılacağından öncelikle “inputparser” komutu ile argümanların atanma işlemi yapılmalıdır. (https4) “addRequired” ve “addOptional” fonksiyonları ile “filtre01” fonksiyonuna verilecek olan argümanlar başlangıç (default) değerleri ile tanımlanmıştır. Örneğin, “addRequired(p,’gecirmeFrekanslari’,validScalarMatrix);” komut satırı ile girilmesi zorunlu argüman olan “gecirmeFrekanslari”, “p = inputparse” değerinde tutulmaktadır. Aynı şekilde girilmesi opsiyonel olan diğer argümanlar için örneğin “addoptional(p,’durtuTepkisiCizdir’,defaultdurtuTepkisiCizdir);” komut satırı ile “durtuTepkisiCizdir”, “p=inputparse” değerinde tutulmaktadır. Fonksiyona argümanlar girildikten sonra bu argümanların değerlerine “p.Results” altından ulaşılabilir. Örneğin “bodeCizdir” argümanının parametresinin doğru girilip girilmediği “if-else” yapısı altında “~islogical(p.Results.bodeCizdir)” mantık değeri kontrolü ile sağlanmaktadır. (https5) Bu değer lojik-0 olduğu durumda “bodeCizdir parametresi true ya da false olmalı.” Hata mesajı komut ekranında gösterilmektedir. Aynı kontrol diğer opsiyonel argümanlar için de yapılmaktadır. Örneğin [b,a] = filtre01(1000, [20 50; 200 Inf], 'bodeCizdir',[43; 56; -22]); girilmesi durumunda hata mesajı Şekil-26’da gösterilmektedir.

```
Command Window
Error using filtre01 (line 61)
bodeCizdir parametresi true ya da false olmalı.
```

Şekil 26: “bodeCizdir parametresi true ya da false olmalıdır.”

Bir diğer hata ise “ornekSayisi” argümanının ve bu argümanın parametresinin “ornekVer” argümanının parametresinden önce veya “ornekVer” “false” olarak girildikten sonra verilmesidir. Bu durumun kontrolü yine “if-else” yapısı ile “filtre01” fonksiyonu altında yapılmaktadır. “p.Results.ornekVer == defaultornekVer && p.Results.ornekSayisi ~= defaultornekSayisi” lojik-0 değerini aldığı durumda yani eğer “ornekVer” argümanının parametresi “true” girilmez ve “ornekSayisi” argümanının parametresi “N” sıfır (default) değerinden farklıysa hata mesajı gösterilmektedir. Örneğin, [b,a] = filtre01(1000, [20 50; 200 Inf], 'ornekVer', false, 'ornekSayisi', 7) için hata Şekil-27’de ki gibi komut ekranında gösterilmektedir.



```
Command Window
Error using filtre01 (line 67)
ornekSayisi sadece ornekVer true olursa verilmeli.
```

Sekil 27: “ornekSayisi sadece ornekVer true olursa verilmeli”

Hatasız yapılan girişler sonrasında “filtre01” fonksiyonunda kullanılan argümanlar ve parametreleri “p.Results” altında tutulmaktadır. Örneğin, “durtuTepkisiCizdir” argümanının parametresi “true” girildiyse “p.Results.durtuTepkisiCizdir” değeri lojik-1 olmaktadır ve “if-else” bloğu altında “impulse(tf(num,den,Ts))” komutu ile elde edilmiş olan filtre katsayıları ile bu filtrenin transfer fonksiyonun dürtü tepkisi çizdirilmektedir.

Aynı şekilde “p.Results.bodeCizdir” lojik-1 ise “bodeoptions”, “opts” değerine kaydedilmektedir. “opts” değişkeni ile “opts.FreqUnits = ‘Hz’” ayarlaması yapılarak frekans eksenine “Hz” birimi yazılmaktadır. “opts.PhaseWrapping = ‘on’” ayarlaması yapılarak filtrenin faz değerlerinin {-180,180} aralığında gösterilmesi sağlanmaktadır. (https6) Gerekli ayarlamalar sağlandıktan sonra “bodeplot(tf_filtre,opts)” ile filtrenin transfer fonksiyonunun bode çizgesi çizdirilmektedir. (https7)

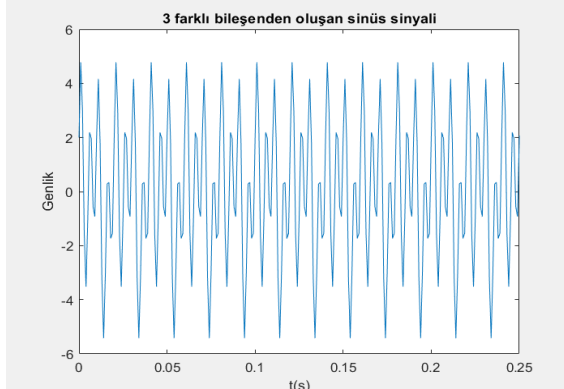
Benzer şekilde “p.Results.ornekVer” değeri lojik-1 olursa yani “ornekVer” argümanının parametresi “true” girildiği durumda öncelikle “ornekSayisi” argümanının parametresi “p.Results.ornekSayisi ~= defaultornekSayisi” şeklinde kontrol edilmektedir. Eğer “ornekSayisi” için N parametresine giriş sağlandı ise “ornekVer” argümanı yerine “ornekSayisi” argümanı çalışacaktır ve sonuç olarak N farklı giriş-çıkış grafiği çizdirilmiş olacaktır. Bu durum “for” döngüsü altında “p.Results.ornekSayisi” kadar farklı frekans üretilerek üretilen farklı frekanslar için 1 genlik değerinde sinüs giriş sinyali oluşturularak bu sinyali katsayıları elde edilmiş olan filtreden “filter” komutu ile filtrelendikten sonra filtrelenmiş sinyali ve giriş sinyalini üst üste çizdirilmesi ile sağlanmaktadır. Aynı zamanda filtrenin transfer fonksiyonu kullanılarak üretilen her frekansta filtrenin faz ve genlik değerleri [mag,phase]=bode(tf_filtre,fu(i)) komut satırı ile bulunmaktadır. İlgili başlık N adet grafiğe “title([num2str(fu(i)),’ Hz için filtre genliği ',num2str(20*log10(mag)),’ dB ve fazı ',num2str(phase),char(176)])” olacak şekilde eklenmektedir. “num2str(fu(i))” ile her aşamada rastgele üretilmiş olan frekans değerleri başlığa eklenmektedir. (https8) “num2str(20*log10(mag))” ile bulunmuş olan genlik değeri “dB” cinsinden hesaplanarak başlığa eklenmektedir. “char(176)” ile derece sembolü faz değerinden hemen sonra grafiğin başlığında yazdırılmaktadır. (https9) Eğer “ornekSayisi” için N parametresine giriş sağlanmadı ise “ornekVer” argümanı için if-else bloğu çalışacaktır ve yukarıda tarif edilen adımlar “N” defa yerine sadece bir kez yapılacaktır. Rastgele üretilmiş frekans değerinde, katsayıları hesaplanmış olan filtre ile bu frekanstaki 1 genlikli sinüs giriş sinyalinin “filter” komutu ile filtrelenerek “y” çıkışının bulunması ve giriş-çıkış sinyallerinin üst üste çizdirilme işlemi yapılmaktadır. (https10)

Filtre katsayılarının bulunması için öncelikle filtre fonksiyonuna verilmesi zorunlu olan birinci argüman olan “fs” değerinden tasarlanacak olan filtrenin örnekleme periyodu ($T_s = 1/fs$) hesaplanmaktadır. Sonrasında “filtre01” fonksiyonuna verilmesi zorunlu olan ikinci argüman olan “gecirmeFrekanslari” parametrelerinden 2x2 double tipinde “gecirmeFrekanslari” parametresinin birinci satırındaki ikinci elemanı ve ikinci satırındaki birinci elemanı “fc3” değerine “fc3 =

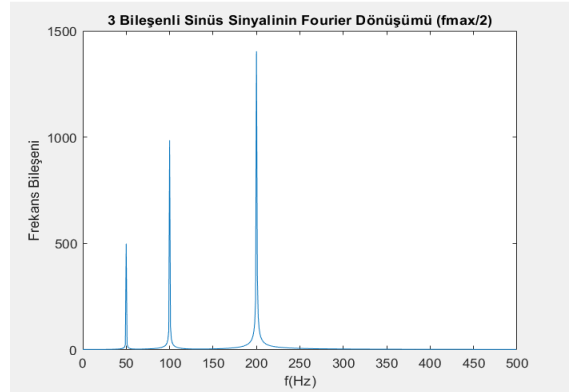
[gecirmeFrekanslari(1,2), gecirmeFrekanslari(2,1)]" olarak kaydedilmektedir. 1x2'lik "fc3" vektörü tarif edildiği gibi oluşturulduktan sonra radyan cinsinden kesim frekansı " $wc3=2*\pi*fc3$ " olarak hesaplanmaktadır. Aynı şekilde örnekleme frekansı " $ws = 2*\pi*fs$ " olarak hesaplandıktan sonra maksimum frekans " $wmax = ws/2$ " şeklinde bulunmaktadır. "butter" fonksiyonuna verilmesi gereken normalize kesim frekansı yine benzer şekilde " $wn3 = wc3/wmax$ " olarak hesaplandıktan sonra radyan cinsinden doğal frekans ve filtrenin derecesi argümanları "butter" fonksiyonuna "[num,den] = butter(N,wn3,'stop');" şeklinde girilerek verilen frekans aralıklarının çıkartıldığı (bastırıldığı) bir filtre tasarlanmaktadır. (https11) "num" ve "den" değerleri sırasıyla "b" ve "a" değerlerine kaydedilerek filtrenin katsayıları elde edilmektedir. "filter" fonksiyonuna giriş sinyali "u" girilerek bu değerler altında çıkış sinyali "y" elde edilebilmektedir. Sonraki aşamalarda kullanılmak üzere tasarlanmış olan bu filtrenin transfer fonksiyonu " $tf_filtre = tf(num,den)$ " tarafından oluşturularak " tf_filtre " değişkenine kaydedilmektedir.

Bu tasarımdaki önemli noktalardan biri "gecirmeFrekanslari" argümanına verilen frekans aralıkları için hangi frekans aralıklarının bastırılacağına hesaplanmasıdır. Filtre tasarımında "butter" komutunda filtre tipi bu açıklamalar çerçevesinde "stop" olarak girilmektedir. Diğer yaklaşımlar da denenmiştir. Örneğin "fc3" kesim frekansı yerine "gecirmeFrekanslari" matrisinin her satırı için iki farklı kesim frekansı "fc1" ve "fc2" oluşturularak iki farklı filtre katsayıları "butter" fonksiyonu ile filtre tipi "bandpass" girilerek hesaplanmıştır. Sonrasında hesaplanmış olan iki filtre katsayısı için iki farklı transfer fonksiyonu elde edilmiştir ve bu iki transfer fonksiyonu kullanılarak üçüncü bir transfer fonksiyonu " $tf3=tf1*tf2$ " olarak hesaplanmıştır. Daha sonra "tf3" transfer fonksiyonundan "tfdata(tf3,'v')" komutuyla kullanılması istenen filtrenin katsayıları elde edilmiştir. (https12) Bu katsayılar ile filtre test edilmiştir fakat filtre istenilen performansı gösterememiştir. Bunun gibi farklı yaklaşımlar örneğin "cascade" gibi yaklaşımlar da denenmiştir fakat en iyi tasarım "gecirmeFrekanslari" argümanı ile verilen frekans aralıkları haricindeki bölgelerin bastırılmasıyla elde edilmiştir. (https13)

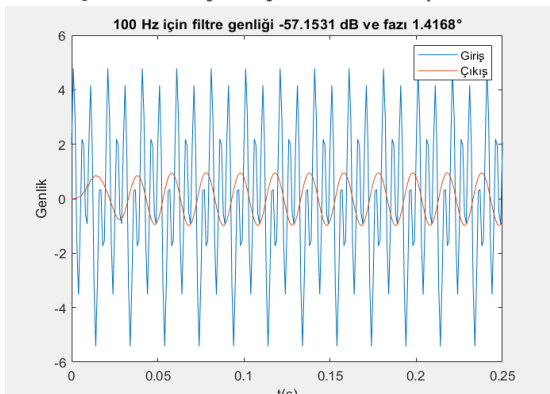
"filtre01" fonksiyonunun bütün argümanlarını içeren örnek bir kullanımı [b,a] = filtre01(1000, [20 60;400 inf], 'durtuTepkisiCizdir', true, 'bodeCizdir', true, 'ornekVer', true, 'ornekSayisi',2); için sonuçlar Şekil 28, Şekil 29, Şekil 30, Şekil 31, Şekil 32, Şekil 33, Şekil 34, Şekil 35'te çizdirilmiştir.



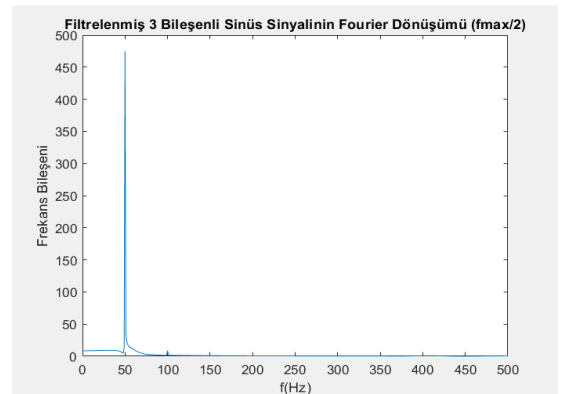
Şekil 28: Üç bileşenli sinüs sinyali



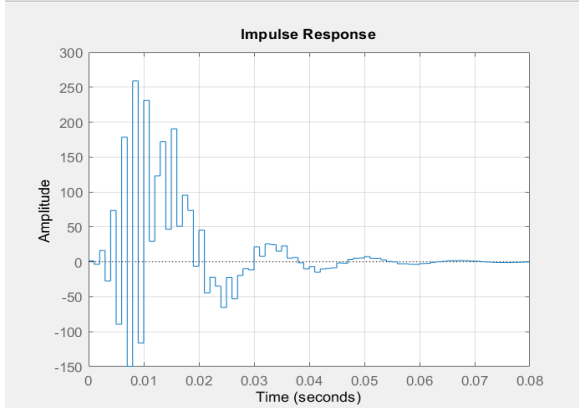
Şekil 29: Üç bileşenli sinüs sinyali Fourier Dönüşümü



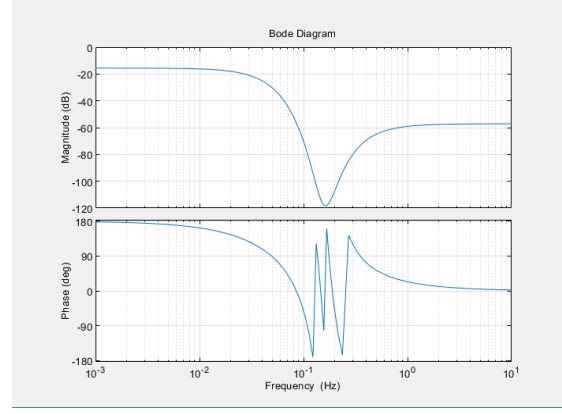
Şekil 30: Giriş-Çıkış Sinyalleri



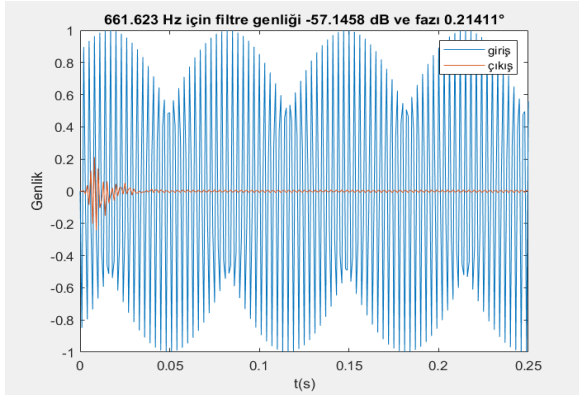
Şekil 31: Çıkış Sinyali Fourier Dönüşümü



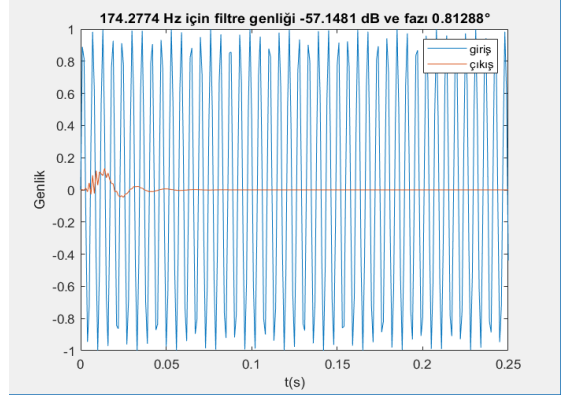
Şekil 32: Dürtü Tepkisi



Şekil 33: Bode Çizgesi



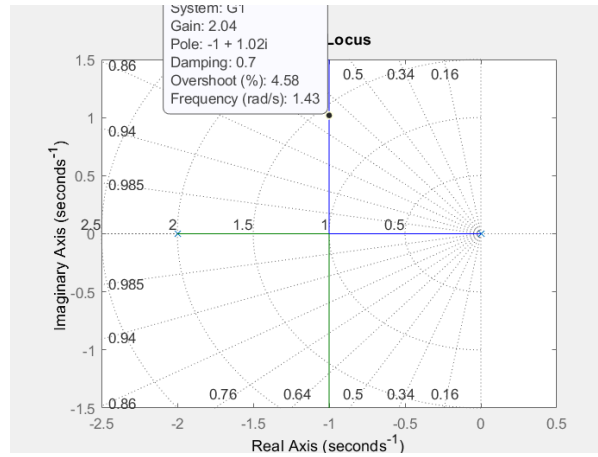
Şekil 34: "ornekVer" N=1



Şekil 35: "ornekVer" N=2

2. SORU 2

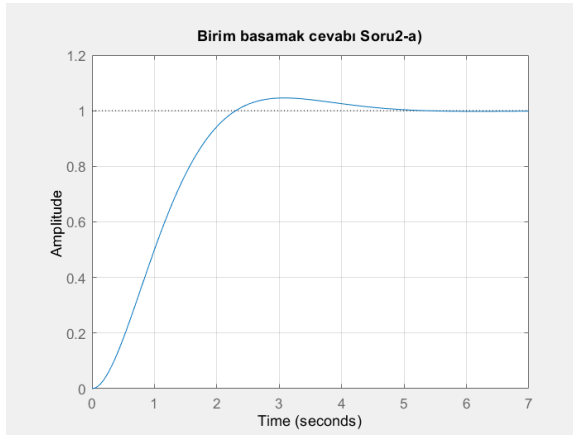
2. Sorunun A şıkında $G(s) = 1/(s*(s+2))$ ile gösterilen sistem için öncelikle sürekli zaman transfer fonksiyonu ifadesi “s”, “s = tf(‘s’)” komutu ile oluşturulmuştur. Ardından $G(s) = 1/(s*(s+2))$ sistemi “G1=1/(s*(s+2))” şeklinde oluşturulmuştur. Bu sistemin kararlılığı “isstable” komutu ile incelendiğinde sonuç “0” olarak bulunmaktadır. (https14) Verilen “G1” sistemi kararsızdır. “G1” sisteminin “rlocus” komutu ile kök yer eğrisi çizdirilmiştir. Kök yer eğrisi kullanılarak kapalı çevrim sisteminin kararlı yapacak ve sönüm oranı 0.7 olacak şekilde bir P-kontrol tasarımı sağlanabilmesi için sistemi kararlı yapacak olan “K=2.04” değeri kök yer eğrisi üzerinden bulunmuştur. “G1” sisteminin kök yer eğrisi Şekil-36’da gösterilmektedir.



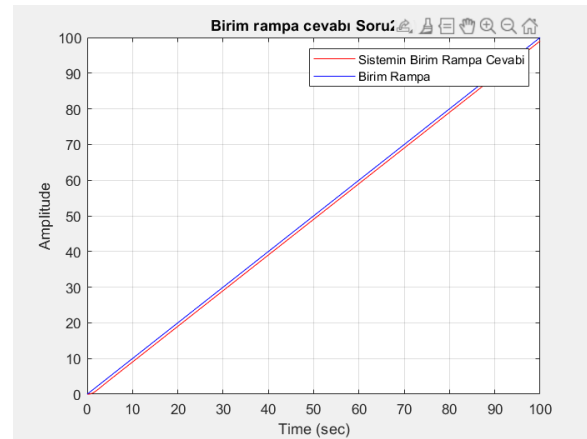
Şekil 36: “G1” sistemi kök yer eğrisi.

P-Kontrolcü “K=2.04” kazanç değeri “C_a” değerine kaydedildikten sonra kapalı çevrim sistemi “feedback” komutu ile “feedback(G1*C_a,1)” şeklinde oluşturulmuş olup “Gcl_a” değerine kaydedilmiştir. Kapalı çevrim sisteminin kararlılığı “isstable(Gcl_a)” komutu ile incelenmiş olup sonuç “1” olarak bulunmaktadır. “G1” sistemi ve “K” değerindeki P-kontrolcü ile kapalı çevrim sistemi kararlı hale getirilmiştir. “Gcl_a” kapalı çevrim sistemi için doğal frekans “wn”, sönüm oranı “zeta” ve “kutuplar”, “damp” fonksiyonu ile “[wn_a, zeta_a, p_a] = damp(Gcl_a);” şeklinde bulunmuş olup bu değerler sırası ile “wn_a”, “zeta_a” ve “p_a” değerlerine kaydedilmiştir. (https15) Bu değerler sırası ile [1.4283, 1.4283], [0.7001, 0.7001] ve [-1.0000 + 1.0198i, -1.0000 - 1.0198i] bulunmuştur. Bu değerler “damp” fonksiyonu kullanılmadan direkt olarak Şekil-36’da verilen grafikte işaretlenen noktada verilen bilgilerle de bulunabilir. Sönümlü doğal frekans “wd” ve zayıflatma değeri “attenuation” bu değerler kullanılarak bulunmuştur. “wd_a” sönümlü doğal frekans değeri “sqrt(attenuation_a.^2-wn_a.^2)” komutu ile hesaplanmış olup [1.0198i; 1.0198i] değerinde bulunmuştur. “attenuation_a” zayıflatma değeri “zeta” ve “doğal frekansın” çarpılmasıyla “zeta_a.*wn_a” hesaplanmaktadır. Bu değer [1;1] olarak bulunmuştur.

“Gcl_a” kapalı çevrim sisteminin birim basamak cevabı “step(Gcl_a)” komutu ile çizdirilmiştir. Grafik Şekil-37’de gösterilmektedir. Birim rampa cevabı ise ilk olarak “u = t” rampa fonksiyonunun oluşturularak “[y,t] = lsim(Gcl_a,u,t);” komut satırıyla kapalı çevrim sistemi “Gcl_a” için birim rampa cevabı “y” değerine kaydedilmektedir. Kapalı çevrim sistemi birim rampa cevabı ve birim rampa sinyalleri üst üste çizdirilmiştir. Grafik Şekil-38’de gösterilmektedir. “Gcl_a” kapalı çevrim sistemi için birim basamak ve birim rampa cevapları incelendiğinde kapalı çevrim sisteminin kararlı olduğu çıkarımı yapılmaktadır. Rampa cevabı için ise “rapma” sinyalinin takip edildiği gözlemlenmektedir.

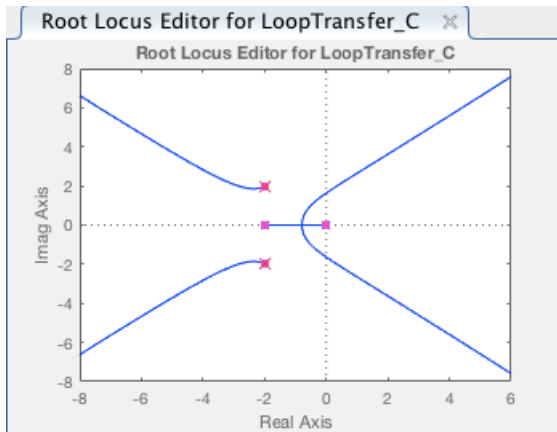


Şekil 37: "Gcl_a" sistemi birim basamak cevabı.

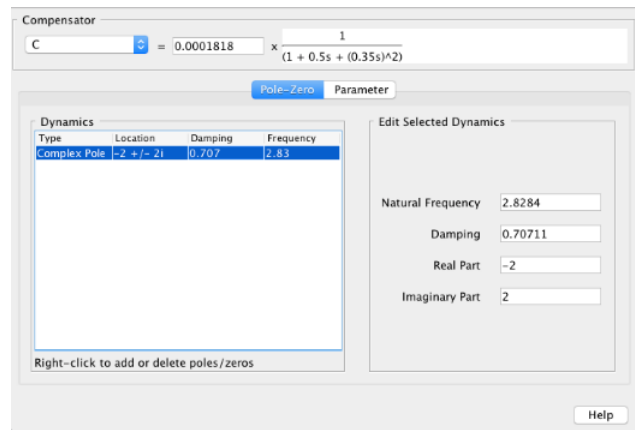


Şekil 38: "Gcl_a" sistemi birim rampa cevabı.

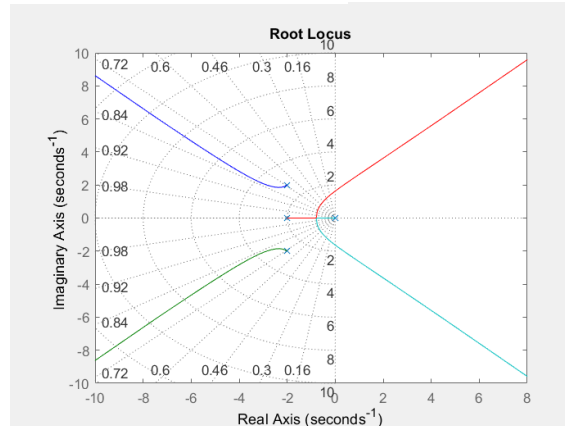
2. Sorunun B seçeneğinde "G1" ile verilen kapalı çevrim sistemi için kararlı olacak ve baskın kutuplar $-2 \pm 2j$ olacak şekilde bir P-kontrolcü ($C(s) = K$) tasarlanamaz, belirtilen kutuplar "G1" sisteminin kök yer eğrisi üzerinde değildir. Bu tasarım "faz ilerletici" (Lead Compensator) kontrol yöntemi kullanılarak sağlanmıştır. "sisotool" fonksiyonu ile Şekil-39, Şekil-40 ve Şekil-41'de gösterilen aşamalar tarafından "G1" sisteminin belirtilen noktalardan geçmesini sağlayacak olan faz ilerletici kontrolcü tasarımı bu noktalara kutup/sıfır eklenmesiyle sağlanmış olup bu kök yer eğrisine sahip bir "C" kontrolcünün transfer fonksiyonu elde edilmiştir ve bu "C" kontrolcü "export" edildikten sonra $0.0014544/(s^2 + 4s + 8)$ transfer fonksiyonu "C_b" değişkenine kaydedilmiştir. "C_b" kullanılarak "G1" sistemi için "feedback" komutu ile kapalı çevrim sistemi oluşturulmuş olup "Gcl_b" değerine kaydedilmiştir. Kapalı çevrim sistemi için kök yer eğrisi "rlocus(Gcl_b)" ile çizdirilmiş olup Şekil-39'da gösterilmektedir. Kapalı çevrim sistemi kararlı ve baskın kutuplar $-2 \pm 2j$ 'de olacak şekilde bir tasarım faz ilerletici kontrolcü ile yapılmıştır.



Şekil 39: "Gcl_b" sistemi kök yer eğrisi.



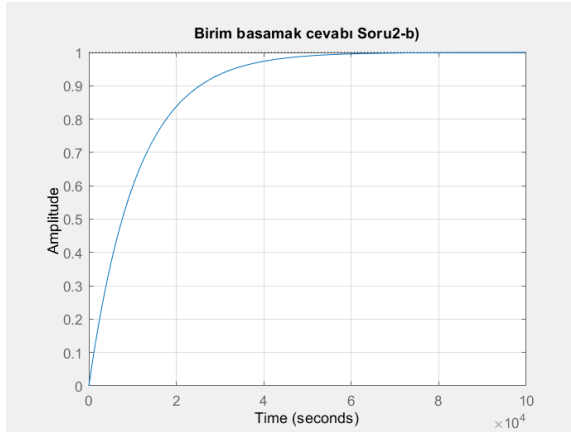
Şekil 40: "Gcl_b" sistemi kök yer eğrisi.



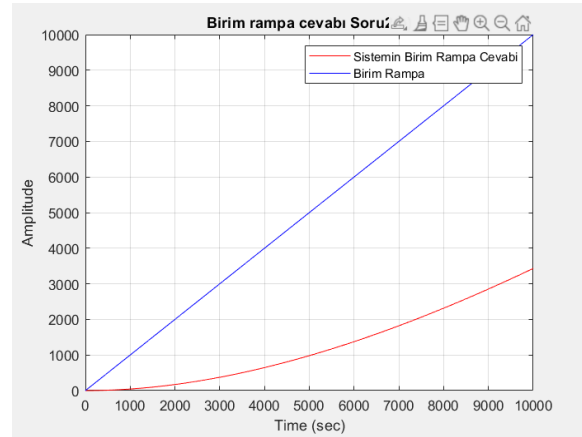
Şekil 41: "Gcl_b" sistemi kök yer eğrisi.

“Gcl_b” kapalı çevrim sistemi için doğal frekans “wn”, sönüm oranı “zeta” ve “kutuplar”, “damp” fonksiyonu ile “[wn_b zeta_b, p_b] = damp(Gcl_b);” şeklinde bulunmuş olup bu değerler sırası ile “wn_b”, “zeta_b” ve “p_b” değerlerine kaydedilmiştir. Bu değerler sırası ile [9.0908e-5, 1.9998; 2.8284; 2.82284], [1; 1; 0.7001, 0.7001] ve [-9.0908e-05; -1.9998; 2+2i; -2-2i] bulunmuştur. Sönümlü doğal frekans “wd” ve zayıflatma değeri “attenuation” bu değerler kullanılarak bulunmuştur. “wd_b” sönümlü doğal frekans değeri “sqrt(attenuation_b.^2-wn_b.^2)” komutu ile hesaplanmış olup [0; 0; 2i; 2i] değerinde bulunmuştur. “attenuation_b” zayıflatma değeri “zeta” ve “doğal frekansın” çarpılmasıyla “zeta_b.*wn_b” hesaplanmaktadır. Bu değer [-9.0908e-05; 1.9998; 2.0000; 2.0000] olarak bulunmuştur.

“Gcl_b” kapalı çevrim sisteminin birim basamak cevabı “step(Gcl_b)” komutu ile çizdirilmiştir. Grafik Şekil-38’de gösterilmektedir. Birim rampa cevabı ise ilk olarak “u = t” rampa fonksiyonunun oluşturularak “[y,t] = lsim(Gcl_b,u,t);” komut satırıyla kapalı çevrim sistemi “Gcl_b” için birim rampa cevabı “y” değerine kaydedilmektedir. Kapalı çevrim sistemi birim rampa cevabı ve birim rampa sinyalleri üst üste çizdirilmiştir. Grafik Şekil-39’da gösterilmektedir. “Gcl_b” kapalı çevrim sistemi için birim basamak ve birim rampa cevapları incelendiğinde kapalı çevrim sisteminin kararlı olduğu çıkarımı yapılmaktadır. Rampa cevabı için ise “rapma” sinyalinin sürekli hal hatası çok yüksek bir değerde olsa da rampa sinyalinin takip edildiği gözlemlenmektedir.



Şekil 42: “Gcl_b” sistemi birim basamak cevabı.

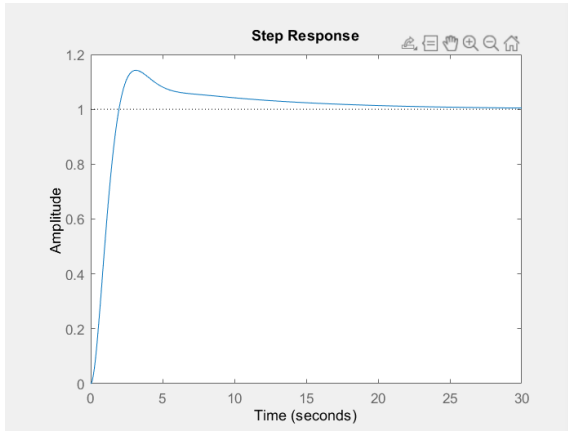


Şekil 43: “Gcl_b” sistemi birim rampa cevabı.

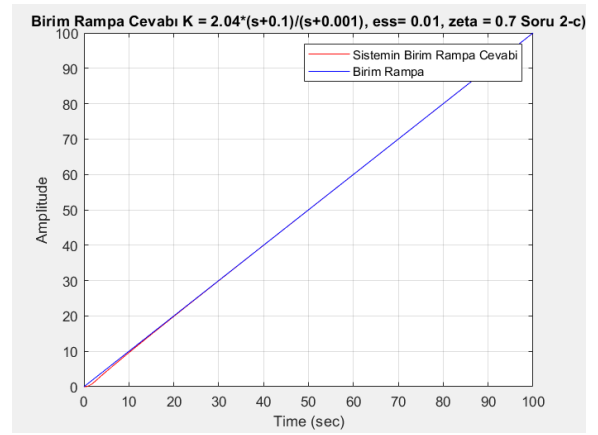
2. Sorunun C seçeneğinde “G1” ile verilen kapalı çevrim sistemi için kararlı olacak, sönüm oranı 0.7 olacak ve birim rampa referans takip hatası (sürekli halde) 0.1’den küçük olacak şekilde bir P-kontrolcü ($C(s) = K$) tasarlanamaz. $Ess = 1/K_v = 0.1$ isteniyor. $S*G(s)$ ’nin s, 0’a giderken limiti $K/2$ ’dir ve buradan da K yani P-kontrolcü 20 değerinde seçilecek olursa birim rampa referans takip hatası 0.1 olmaktadır. Fakat soruda bu değerin 0.1’den küçük olması ve zeta değerinin 0.7 olması istenmektedir. 2. Sorunun A seçeneğinde zeta = 0.7 için K değeri 2.04 olarak kök yer eğrisinden bulunmuştu fakat bu durumda sistemin birim rampa referans takip hatası “1” olarak hesaplanmaktadır. Bu değerin zeta = 0.7 değeri ile birlikte 0.1’den küçük seçilebilmesi P-kontrolcü ile mümkün değildir. Bu durumda faz geriletici” kontrolör kullanılmalıdır. “ $C(s) = K$ ” yerine “ $C(s) = K*(s+z)/(s+p)$ ” tipinde bir kontrolcü seçilerek istenilen tasarım sağlanabilir. zeta = 0.7’yi değiştirmeden $ess = 1$ değerini $ess = 0.01$ (0.1’den küçük) $1/100$ katına getirme işlemi yapılmalıdır. $Ess = 1/K_v = 0.01$ işleminden K_v ’nin 100 katına çıkması gerektiği hesaplanmıştır. $-p = -0.001$ ve $-z = 100 * -p = -0.1$ değerleri seçilmeli, $K = 2.04$ değerinin değiştirmeden zeta = 0.7’yi sağlayacak şekilde $C(s) = 2.04*(s+0.1)/(s+0.001)$ kontrolcü kullanabiliriz. “ $2.04*(s+0.1)/(s+0.001)$ ” kontrolcüsü “C_c” değişkenine kaydedildikten sonra “feedback” komutu ile “Gcl_c” değişkenine kapalı çevrim sistemi kaydedilmiştir. “isstable” komutuyla “Gcl_c” kapalı çevrim sisteminin kararlı olduğu kontrol edilmiştir ve sistemin kararlı olduğu tespit edilmiştir.

“Gcl_c” kapalı çevrim sistemi için doğal frekans “wn”, sönüm oranı “zeta” ve “kutuplar”, “damp” fonksiyonu ile “[wn_c, zeta_c, p_c] = damp(Gcl_c);” şeklinde bulunmuş olup bu değerler sırası ile “wn_c”, “zeta_c” ve “p_c” değerlerine kaydedilmiştir. Bu değerler sırası ile [0.1114;1.35333;1.3533], [1;0.6981;0.6981] ve [-0.1114;-0.9448 + 0.9690i;-0.9448-0.9690i] bulunmuştur. Sönümlü doğal frekans “wd” ve zayıflatma değeri “attenuation” bu değerler kullanılarak bulunmuştur. “wd_c” sönümlü doğal frekans değeri “sqrt(attenuation_c.^2-wn_c.^2)” komutu ile hesaplanmış olup [1.0198i;1.0198i] değerinde bulunmuştur. “attenuation_c” zayıflatma değeri “zeta” ve “doğal frekansın” çarpılmasıyla “zeta_c.*wn_c” hesaplanmaktadır. Bu değer [0.1114;0.9448;0.9448] olarak bulunmuştur.

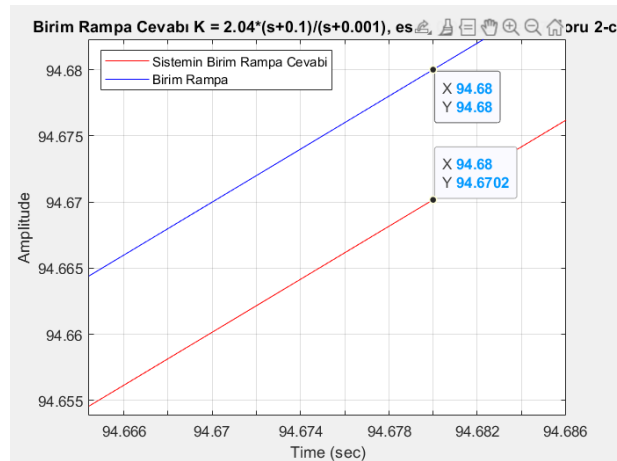
“Gcl_c” sisteminin birim basamak ve birim rampa cevabı önceki aşamalarda olduğu gibi çizdirilmiş olup Şekil-40 ve Şekil-41’de gösterilmektedir. Ek olarak birim rampa referans takip hatası Şekil-42’de gösterildiği gibi 0.1’den küçük bir değer olan 0.01 değerindedir. Sistemin kararlı olduğu grafiklerden anlaşılmaktadır. İstenilen tasarım “faz geriletici” kontrol ile yapılabilmektedir.



Şekil 40: “Gcl_c” sistemi birim basamak cevabı.

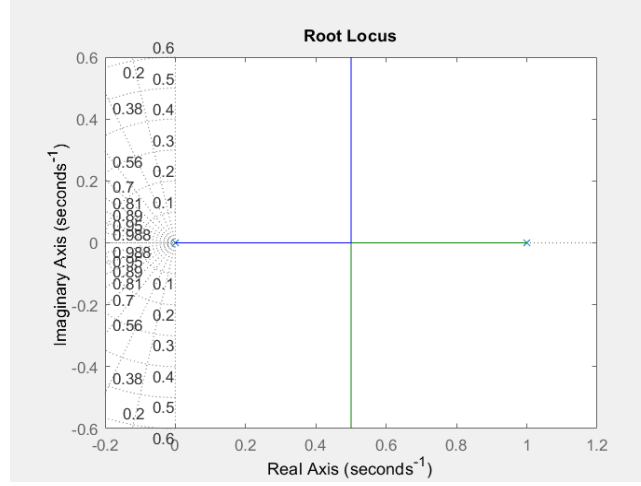


Şekil 41: “Gcl_c” sistemi birim rampa cevabı



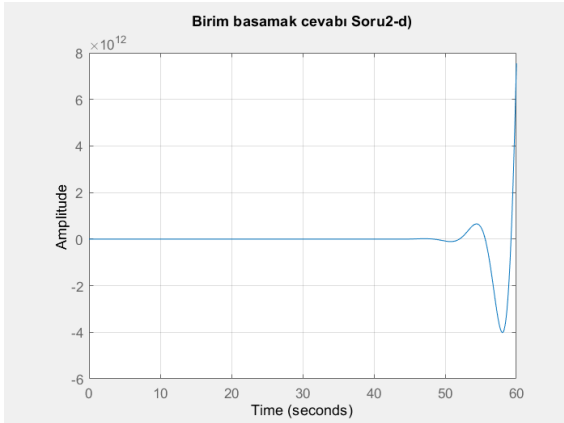
Şekil 42: “Gcl_c” sistemi rampa referans takip hatası analizi.

2. Sorunun D şıkında Denklem-2 ile gösterilen sistem “ $G2=1/(s*(s-1))$ ” oluşturulmuştur. Bu sistemin kararlılığı “isstable” komutu ile incelendiğinde sonuç “0” olarak bulunmaktadır. Verilen “G2” sistemi kararsızdır. “G2” sisteminin “rlocus” komutu ile kök yer eğrisi çizdirilmiştir. Kök yer eğrisi kullanılarak kapalı çevrim sistemini kararlı yapacak ve sönüm oranı 0.7 olacak şekilde bir P-kontrol tasarımı sağlanamaz. Sistemin kutupları sanal eksenin sağında bulunmaktadır. “G2” sisteminin kök yer eğrisi Şekil-43’te gösterilmektedir.

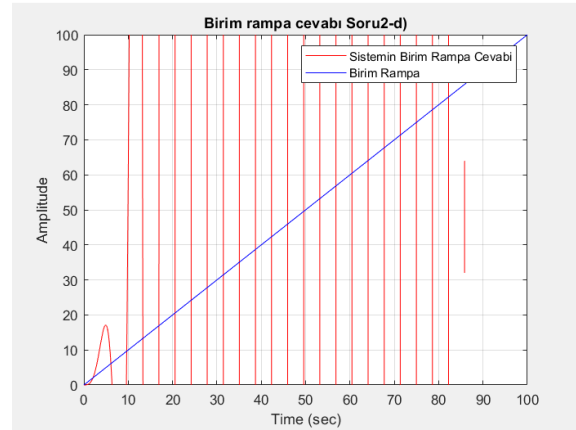


Şekil 43: “Gcl_d” sistemi kök yer eğrisi.

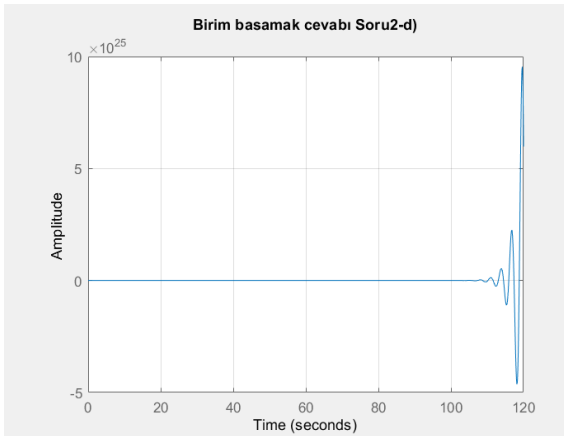
“K=1” ve “K=5” kazanç değeri “C_d” değerine kaydedilerek direkt olarak G(s)’nin kapalı çevrim sistemi “feedback” komutu ile oluşturulmuştur ve bu sistemin birim basamak/rampa cevabı çizdirilmiştir. Grafikler Şekil-44 ve Şekil-45’te gösterilmektedir. “K” değerinin artırılmasıyla kapalı çevrim sistemini kararlı yapacak bir “C(s)=K” P-Kontrolcü seçilememektedir. Farklı “K” değerleri için simülasyon tekrarlanmıştır fakat kapalı çevrim sistemi kararlı hale getirilememiştir.



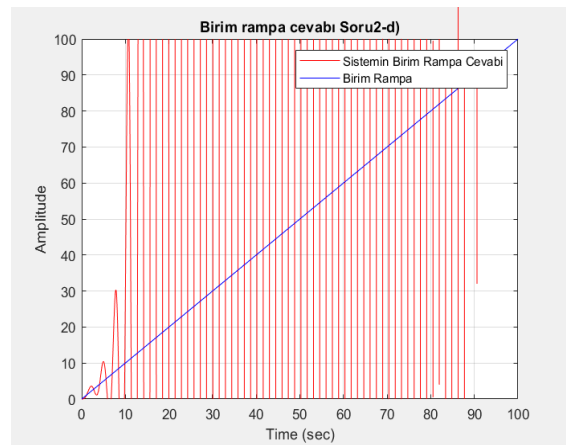
Şekil 44: “Gcl_d” sistemi birim basamak cevabı. (K=1)



Şekil 45: “Gcl_d” sistemi birim rampa cevabı. (K=1)

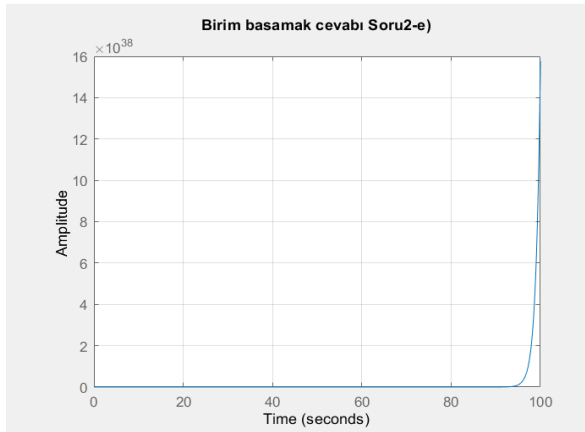


Şekil 46: “Gcl_d” sistemi birim basamak cevabı. (K=5)

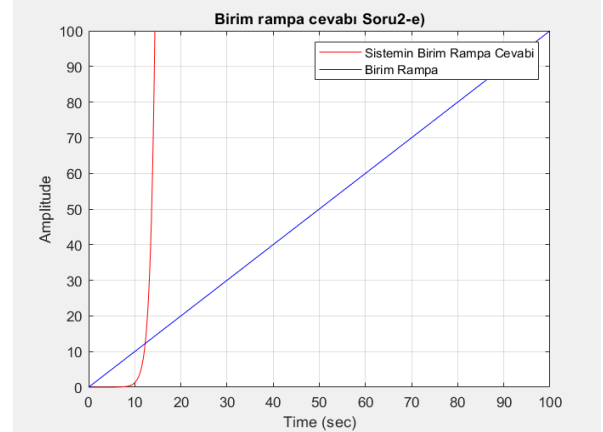


Şekil 47: “Gcl_d” sistemi birim rampa cevabı. (K=5)

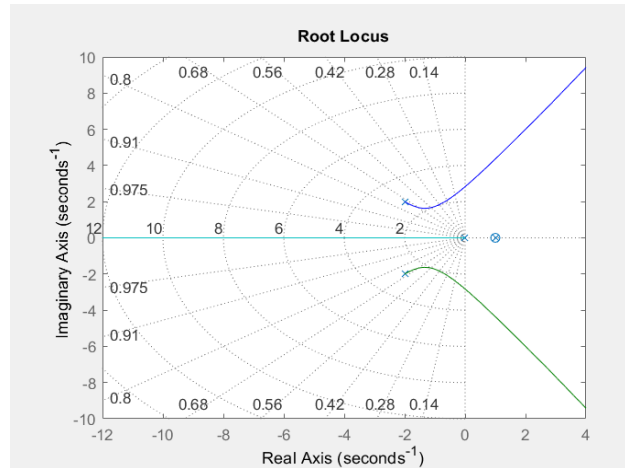
2. Sorunun E şıkında “G2” sistemi için bu sistemin kök yer eğrisi incelenmiştir. Kapalı çevrim sisteminin kutupları $-2 \pm 2j$ ’de olabilecek bir P-Kontrolcü ($C(s) = K$) tasarlanamaz, belirtilen kutuplar “G2” sisteminin kök yer eğrisi üzerinde değildir. “faz ilerletici” (Lead Compensator) kontrol yöntemi kullanılarak eğrinin belirtilen noktalardan geçme işlemi sağlanmış olsa da bu aşamada “G2”nin imajiner eksenin sağında ($s=1$) kutubu olması sebebiyle kapalı çevrim sistemi kararlı hale getirilememektedir. Sistemde belirtilen noktalar eğri üzerinde olsa bile sistem kararsızdır. “sisotool” ile kapalı çevrim sisteminin kök yer eğrisinin bu noktalardan geçecek şekilde ve sönüm oranı 0.7 olacak şekilde $C(s) = 0.0007569/(s^2 + 4s + 8)$ tasarlanmıştır fakat kapalı çevrim sistemi kararlı hale getirilememektedir. “Gcl_e” sistemi için kök yer eğrisi, birim basamak ve birim rampa cevapları Şekil-48, Şekil-49 ve Şekil-50’de gösterildiği gibi çizdirilmiştir. Kapalı çevrim sisteminin kararsız olduğu bu grafiklerden anlaşılmaktadır. Birim basamak cevabı zamanla artmaktadır. Birim rampa cevabında ise sürekli hal hatasından söz edilememektedir.



Şekil 48: “Gcl_e” sistemi birim basamak cevabı.

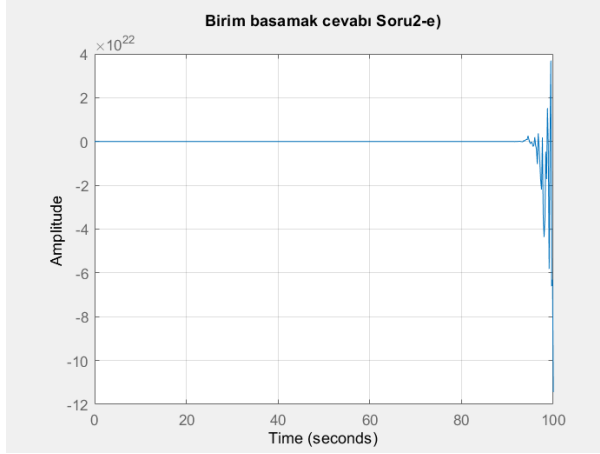


Şekil 49: “Gcl_e” sistemi birim basamak cevabı.

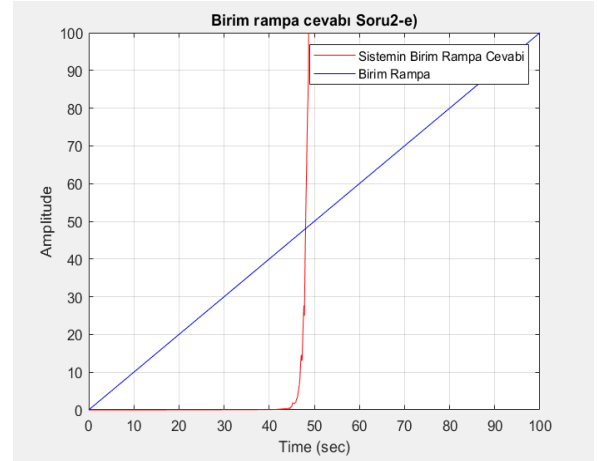


Şekil 50: “Gcl_e” sistemi kök yer eğrisi.

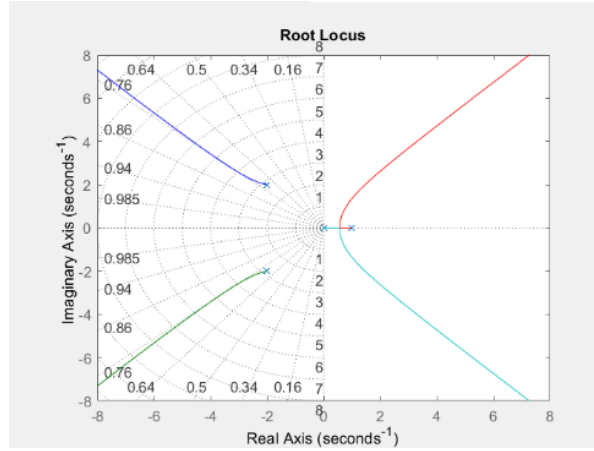
“Compensator” tasarımına Real-Zero eklenerek +1 noktasındaki kutup giderilmeye çalışılırken aynı zamanda eğrinin istenilen noktalardan geçmesi için $C(s) = 0.000841*(s-1)/(s^2+4s+8)$ kontrolcüsü tasarlanmıştır ve “export” edilmiştir fakat sistem yine kararlı hale getirilememiştir. “Gcl_e” sistemi için grafikler Şekil-51, Şekil-52 ve Şekil-53de gösterildiği gibi çizdirilmiştir.



Şekil 51: "Gcl_e" sistemi birim basamak cevabı.

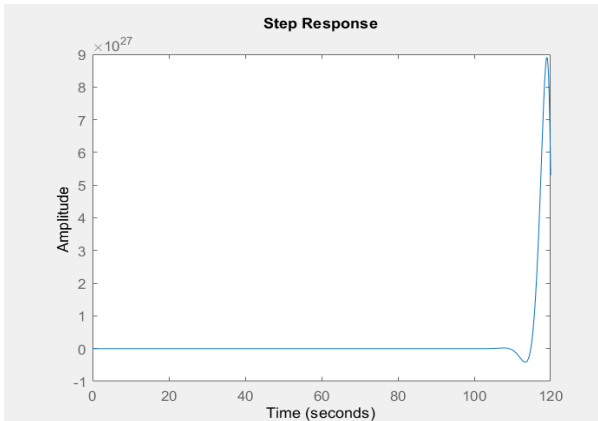


Şekil 52: "Gcl_e" sistemi birim rampa cevabı.

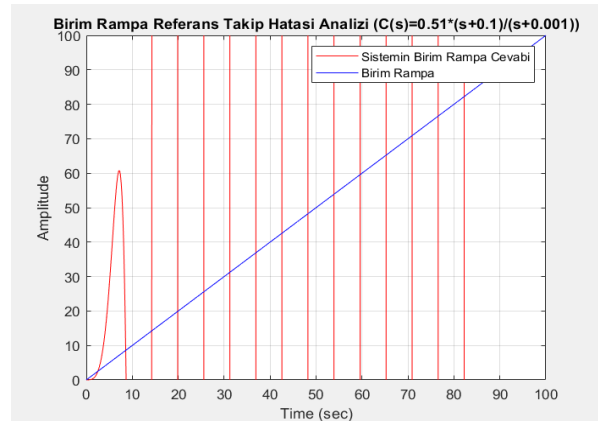


Şekil 53: "Gcl_e" sistemi kök yer eğrisi.

2. Sorunun F şıkında D şıkında da bahsedildiği üzere kapalı çevrim sistemini kararlı yapacak ve sönüm oranı 0.7 olacak şekilde bir P-kontrolcü $s=1$ 'de kutup olmasından dolayı tasarlanamaz. Sistem kararsızdır. Dolayısıyla zeta < 0 'dır. Kapalı çevrim sisteminin kök yer eğrisi incelendiğinde -0.7 sönüm oranı için -0.51 kazanç değeri P-kontrolcü olarak denenmiştir fakat elde edilen kapalı çevrim sistemi yine kararsızdır. 0.51 P-Kontrolcü ile 0.7 sönüm oranında kapalı çevrim sistemi kararlı yapılamamaktadır. Bu tasarım P-kontrolcü ile yapılamaz. Sorunun C şıkında kullanılan faz geriletici kontrolcüsüne benzer bir faz geriletici kontrolcü " $C(s) = 0.51 \cdot (s+0.1)/(s+0.001)$ " F şıkkı için denenmiş olup, kapalı çevrim sisteminin birim basamak/rampa cevapları incelenmiştir. Sonuç olarak P kontrolcü, faz ilerletici veya faz geriletici kontrolcüler ile bu kapalı çevrim sistemi kararlı yapılamaz. "Gcl_f" kapalı çevrim sistemi için birim basamak ve rampa cevapları sırasıyla Şekil-54 ve Şekil-55'te verilmiştir.



Şekil 54: "Gcl_f" sistemi birim basamak cevabı.



Şekil 55: "Gcl_f" sistemi rampa cevabı.

2. Sorunun G şıkında, B şıkında “sisotool” fonksiyonu ile yapılmış olan “faz ilerletici (Lead Compansator)” tasarımı bu aşamada “fazilerletici01” fonksiyonu ile sağlanmış olup tasarlanmış olan kontrolcü komut ekranında gösterilmiştir. Ardından tasarlanmış olan kontrolcü için verilen sistemin kapalı çevrim sistemi oluşturularak bu sistemin “rlocus” komutuyla kök yer eğrileri çizdirilmiştir. Fonksiyona argüman olarak verilen kapalı çevrim sistemi için bir kutup olan “p” girişin kök yer eğrileri üzerinde olup olmadığı çizdirilen grafikler üzerinden tespit edilmektedir.

“fazilerletici01” fonksiyonunun “G” ve “p” olmak üzere iki argümanı vardır. Fonksiyona kararsız bir “G” sistemi girilmesi durumunda “Verilen sistemin sanal eksenin sağında kutbu var, sistem kararsızdır.” hata mesajı komut ekranında Şekil-56’da gösterildiği gibi verilmektedir. Bu durum girilen “G” sisteminin “pole” komutuyla kutuplarının bulunması ve eğer bu kutup sıfırdan büyükse ilgili hata mesajının komut ekranına verilmesiyle sağlanmaktadır. Bir diğer hata mesajı ise “p” argümanı ile ilgilidir. Kapalı çevrim sistemi için istenen kutuplar sanal eksenin sağında ise yani sistem kararsız olacak ise “Kapalı çevrim sistemi kutupları sanal eksenin solunda olmalıdır.” hata mesajı Şekil-57’de gösterildiği gibi komut ekranında gösterilmektedir.

```
Command Window

Error using fazilerletici01 (line 8)
Verilen sistemin sanal eksenin sağında kutbu var, sistem kararsızdır.

Error in SORU2 (line 362)
C = fazilerletici01(G, p);fazilerletici01 fonksiyonu kullanımı.
```

Şekil 56: “G” argümanı için hata mesajı.

```
Command Window

Error using fazilerletici01 (line 10)
Kapalı çevrim sistemi kutupları sanal eksenin solunda olmalıdır.

Error in SORU2 (line 365)
C = fazilerletici01(G, p);fazilerletici01 fonksiyonu kullanımı.
```

Şekil 57: “p” argümanı için hata mesajı.

Hatalı giriş yoksa “fazilerletici01” fonksiyonu verilen giriş argümanlarına göre “C” kontrolcüsünü tasarlayıp tasarımın sonunda “disp” komutu ile bu kontrolcünün transfer fonksiyonunu komut ekranında göstermektedir. Tasarıma verilen “G” sisteminin kutupları “polesG” değişkenine “pole(G)” komutu ile kaydedilmektedir ve bu kutuplar “b” ve “a” değişkenlerinde tutulmaktadır. Sonrasında “z_lead” değişkenine “a” kutbu kaydedilmektedir. Kontrolcü sıfırı olan “z_lead”, “G” sisteminin sıfırdan farklı kutbu üzerine yerleştirilmiştir. “z_lead” tanımlandıktan sonra açılı koşulunda için kullanılacak olan “x” ve “y” değişkenleri oluşturulmuştur. “x” “p” kutbunun gerçek kısmı, “y” ise “p” kutbunun sanal kısmıdır. Bu değerler bulunduktan sonra açılı koşulu ile “h” açısı $h = 180 + (180 + \text{atand}(y/(b-x))) + \text{atand}(y/(-a-x)) - 360$ şeklinde hesaplanmaktadır. (https16) “h” açısı açılı eksiğini ifade etmektedir. Verilen “p” noktasının “G” sisteminin ve tasarlanmakta olan “C” kontrolcüsünün kutupları/sıfırlarıyla birleştirilmesi sonucu oluşan çizgilerin yatay eksenle yaptığı açılar “atand” fonksiyonu yardımıyla bulunmuştur. Açılı eksiği hesaplandıktan sonra “C” kontrolcüsünün kutbu “p_lead”, $y \cdot \text{tand}(h) + x$ şeklinde bulunmaktadır. Kutupların ve sıfırların yeri bulunduktan sonra genlik koşulu ile kazanç değeri “K” bulunmaktadır. “K” değeri çizdirilmiş olan eğriler baz alınarak hesaplanmaktadır. Kutuplar için çizilen eğrilerin uzunlukları çarpımının sıfırlar için çizilen eğrilerin uzaklığına bölünmesi ile “K” değeri hesaplanmaktadır. Kullanıcının “G” sisteminin kazancını 1’den farklı girdiği durumlar için “[z,g] = zero(G)” ifadesinden “G” sisteminin kazancı “g” bulunmaktadır. (https17) Sonrasında hesaplanmış olan “K” değeri bu “g” değerine bölünerek kontrolcünün kazanç değeri bulunmaktadır. İşlemler tamamlandıktan sonra faz ilerletici kontrolcü $C = K \cdot (s + z_lead) / (s + p_lead);$ şeklinde oluşturularak “disp(C)” ile komut ekranına verilmektedir. (https18)

2. Sorunun B şıkkı için tasarlanmış olan “faz ilerletici” kontrolcü bu aşamada tasarlanmış olan “fazillerletici01” kontrolcüsü ile sağlanmaktadır. “ $G = 1/(s*(s+2))$ ” sistemi için “p” değeri “-2+2j” olarak tanımlandıktan sonra “C = fazillerletici01(G, p);” satırı ile kontrolcü oluşturulmaktadır. Oluşturulan kontrolcünün transfer fonksiyonu ile ilgili bilgiler Şekil-58’de gösterildiği gibi komut ekranına verilmektedir. Oluşturulan kontrolcü “ $8*(s+2)/(s+4)$ ” şeklindedir.

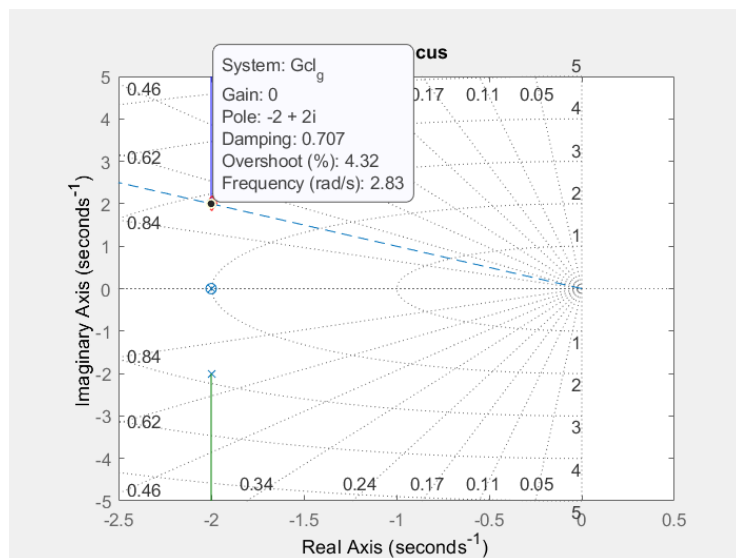
```
Command Window

tf with properties:

    Numerator: {[8.0000 16.0000]}
    Denominator: {[1 4]}
    Variable: 's'
    IODelay: 0
    InputDelay: 0
    OutputDelay: 0
    Ts: 0
    TimeUnit: 'seconds'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
    Notes: [0x1 string]
    UserData: []
    Name: ''
    SamplingGrid: [1x1 struct]
```

Şekil 57: “C” faz ilerletici kontrolcü parametreleri.

Kontrolcü tasarlandıktan sonra kapalı çevrim sisteminin istenilen noktalardan geçip geçmediği ve istenilen sönüm oranında olup olmadığı kapalı çevrim sistemi oluşturulduktan sonra kök yer eğrisi çizdirilerek kontrol edilmektedir. “zeta” sönüm oranı “ $p_{1,2} = -zeta*wn \pm j*wn*(1-zeta^2)^{1/2}$ ” formülünden hesaplanmaktadır. “x” ve “y” değişkenlerine sırası ile “p” kutbunun gerçekte ve sanal kısımları kaydedilmektedir. “wn” doğal frekans “ $wn = \sqrt{y^2 + x^2}$ ” şeklinde bulunarak “zeta” değeri “ $zeta = x/wn$ ” şeklinde hesaplanmaktadır. Kök yer eğrisi üzerine “p” noktası ve “zeta” sönüm oranı çizdirilmektedir. B şıkkı için tasarlanmış olan faz ilerletici kontrolcü ile elde edilen kapalı çevrim sisteminin kök yer eğrisi Şekil-58’de gösterilmektedir. “p” noktaları kök yer eğrisi üzerindedir ve sönüm oranı 0.7’dir.



Şekil 58: $G = 1/s(s+2)$ ve $p = -2 + 2j$ için kapalı çevrim sistemi kök yer eğrisi.

" $G = 4/s(s+2)$ " sistemi ve " $p=-2+2\sqrt{3}j$ " kapalı çevrim kutbu için "fazillerletici01" fonksiyonu kullanılarak "C" kontrolcü Şekil-59'da gösterildiği gibi elde edilmiştir. Oluşturulan kontrolcü " $4*(s+2)/(s+4)$ " şeklindedir. Oluşturulan kontrolcü ile elde edilen kapalı çevrim sistemi kök yer eğrisi Şekil-60'da gösterilmektedir. Tasarlanmış olan kontrolcü ile "p" noktasının eğri üzerinde olması sağlanmıştır.

```

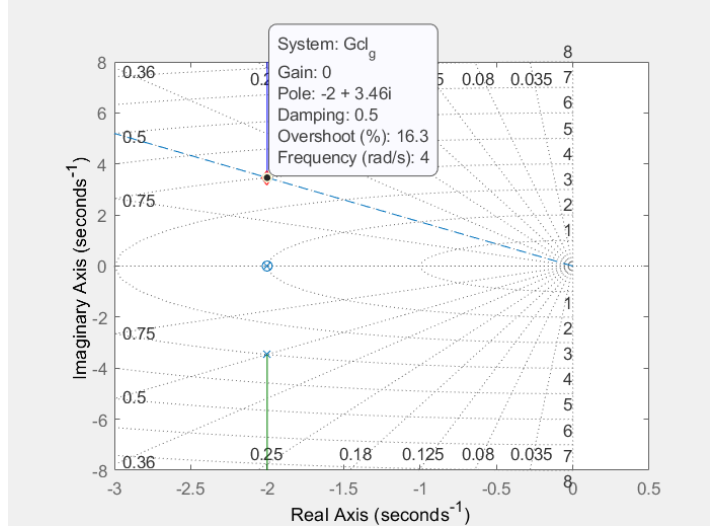
Command Window

tf with properties:

    Numerator: {[4.0000 8.0000]}
    Denominator: {[1 4]}
    Variable: 's'
    IODelay: 0
    InputDelay: 0
    OutputDelay: 0
    Ts: 0
    TimeUnit: 'seconds'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
    Notes: [0x1 string]
    UserData: []
    Name: ''
    SamplingGrid: [1x1 struct]

```

Şekil 59: "C" faz ilerletici kontrolcü parametreleri.



Şekil 60: $G = 1/s(s+2)$ ve $p = -2 + 2j$ için kapalı çevrim sistemi kök yer eğrisi.

" $G = 7/s(s+1)$ " sistemi ve " $p=-1+4.3j$ " kapalı çevrim kutbu için "fazillerletici01" fonksiyonu kullanılarak "C" kontrolcü Şekil-61'de gösterildiği gibi elde edilmiştir. Oluşturulan kontrolcü " $2.7843*(s+1)/(s+2)$ " şeklindedir. Oluşturulan kontrolcü ile elde edilen kapalı çevrim sistemi kök yer eğrisi Şekil-61'd1 gösterilmektedir. Tasarlanmış olan kontrolcü ile "p" noktasının eğri üzerinde olması sağlanmıştır.

```

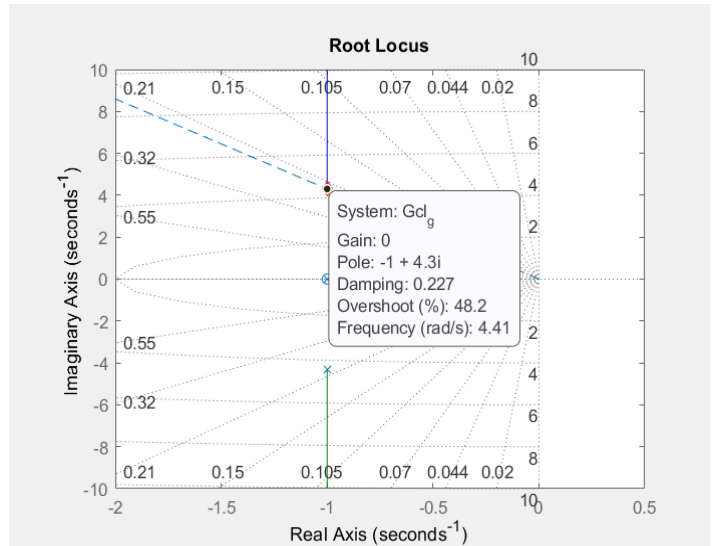
Command Window

tf with properties:

    Numerator: {[2.7843 2.7843]}
    Denominator: {[1 2.0000]}
    Variable: 's'
    IODelay: 0
    InputDelay: 0
    OutputDelay: 0
    Ts: 0
    TimeUnit: 'seconds'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
    Notes: [0x1 string]
    UserData: []
    Name: ''
    SamplingGrid: [1x1 struct]

```

Şekil 61: "C" faz ilerletici kontrolcü parametreleri.



Şekil 62: $G = 7/s(s+1)$ ve $p = -1 + 4.3j$ için kapalı çevrim sistemi kök yer eğrisi.

2. Sorunun H şıkında C şıkında tasarlanmış olan faz geriletici kontrolcü (Lag Compansator) “fazGeriletici01” fonksiyonu tarafından otomatik olarak tasarlanmaktadır. “fazGeriletici01” fonksiyonunun “fazIlerletici01” fonksiyonundan farklı olarak “ess_d” isimli bir argümanı daha bulunmaktadır. “ess_d” parametresi sürekli hal birim rampa takip hatasının istenildiği değerdir. Bu değer kullanıcı tarafından fonksiyona verilmektedir. Fonksiyon istenilen “G”, “p” ve “ess_d” değerleri için “faz geriletici” kontrolcüyü tasarlayıp “C” değişkenine kaydetmektedir. Tasarlanmış olan “C” kontrolcüsü komut ekranında gösterilmektedir. “fazGeriletici01” fonksiyonu “fazIlerletici01” fonksiyonu gibi bazı hata mesajları içermektedir. Fonksiyona kararsız bir “G” sistemi girilmesi durumunda “Verilen sistemin sanal eksenin sağında kutbu var, sistem kararsızdır.” hata mesajı komut ekranında Şekil-63’te gösterildiği gibi verilmektedir. Bu durum girilen “G” sisteminin “pole” komutuyla kutuplarının bulunması ve eğer bu kutup sıfırdan büyükse ilgili hata mesajının komut ekranına verilmesiyle sağlanmaktadır. Bir diğer hata mesajı ise “p” argümanı ile ilgilidir. Kapalı çevrim sistemi için istenen kutuplar sanal eksenin sağında ise yani sistem kararsız olacak ise “Kapalı çevrim sistemi kutupları sanal eksenin solunda olmalıdır.” hata mesajı Şekil-64’te gösterildiği gibi komut ekranında gösterilmektedir.

```
Command Window
Error using fazGeriletici01 (line 8)
Verilen sistemin sanal eksenin sağında kutbu var, sistem kararsızdır.

Error in SORU2 (line 392)
C = fazGeriletici01(G, p, ess_d);%fazIlerletici01 fonksiyonu kullanımı.
```

Şekil 63: “G” argümanı için hata mesajı.

```
Command Window
Error using fazGeriletici01 (line 10)
Kapalı çevrim sistemi kutupları sanal eksenin solunda olmalıdır.

Error in SORU2 (line 396)
C = fazGeriletici01(G, p, ess_d);%fazIlerletici01 fonksiyonu kullanımı.
```

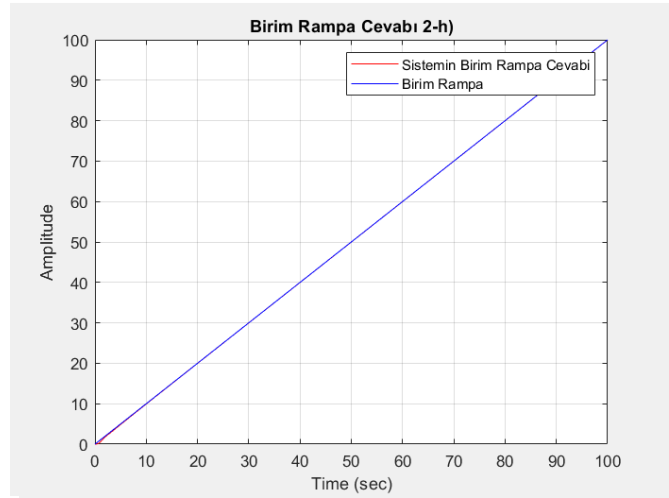
Şekil 64: “p” argümanı için hata mesajı.

Hatalı giriş yoksa “fazGeriletici01” fonksiyonu verilen giriş argümanlarına göre “C” kontrolcüsünü tasarlayıp tasarımın sonunda “disp” komutu ile bu kontrolcünün transfer fonksiyonunu komut ekranında göstermektedir. Tasarıma verilen “G” sisteminin kutupları “polesG” değişkenine “pole(G)” komutu ile kaydedilmektedir ve bu kutuplar “b” ve “a” değişkenlerinde tutulmaktadır. Girilen “p” kutbunun gerçek kısmı “x”, sanal kısmı “y” değişkeninde tutulmaktadır. “a”, “b”, “x”, “y” değişkenleri bulunduktan sonra genlik koşulu ile “K” kazanç değeri $K = (\sqrt{(-x-a)^2+y^2}) \cdot \sqrt{(-x+b)^2+y^2})$ tarafından hesaplanmaktadır. Kontrolcü kazancı “Kc” değerinin “G” sistemi kazancından ayrılabilmesi için “[z,g]= zero(G)” komutu ile girilmiş olan “G” sisteminin kazanç değeri “g” değişkenine kaydedilmektedir. “Kc = K/g;” ile kontrolcü kazancı “G” sisteminin kazancından ayrılmaktadır. Limit ile “Kv” kazanç değerinin hesaplanabilmesi için “sys_syms” değişkenine “G” sistemi transfer fonksiyonu sembolik olarak kaydedilmektedir. Sonrasında “limit(u*Kc*sys_syms,u,0);” komut satırı ile “s*AÇTF” yani “s” sürekli zaman transfer fonksiyonu ile açık çevrim sisteminin $(C(s).G(s))$ çarpımının sıfıra giderken ki limiti hesaplanmaktadır. (https19, https20) Hesaplanmış olan değer “Kv” kazanç değerine sembolik olarak kaydedilmektedir. “sym2poly(Kv)” komutu ile kazanç değeri sembolik değerden sayısal değere çevrilmektedir. (https21) Fonksiyon argümanlarından olan “ess_d” değeri ile yeni kazanç değeri “Kv_new”, “Kv_new = 1/ess_d” şeklinde oluşturulmaktadır. “Kv_new” değerinin “Kv” değerine bölünmesi ile “beta” değeri elde edilmektedir. Sıfıra çok yakın bir kutup “p_lag = 0.005;” şeklinde oluşturulmaktadır. Seçilen kutup ve bulunan “beta” değeri ile “z_lag” faz geriletici kontrolcü kutbu “z_lag = beta * p_lag;” tarafından elde edilmektedir. Hesaplanmış olan “Kc” ve kutupların “C = Kc*((s+z_lag)/(s+p_lag));” formatında yerleştirilmesiyle faz geriletici kontrolcü “C” oluşturulmaktadır ve “disp(C)” komutu ile bu kontrolcünün parametreleri komut ekranında gösterilmektedir. (https22)

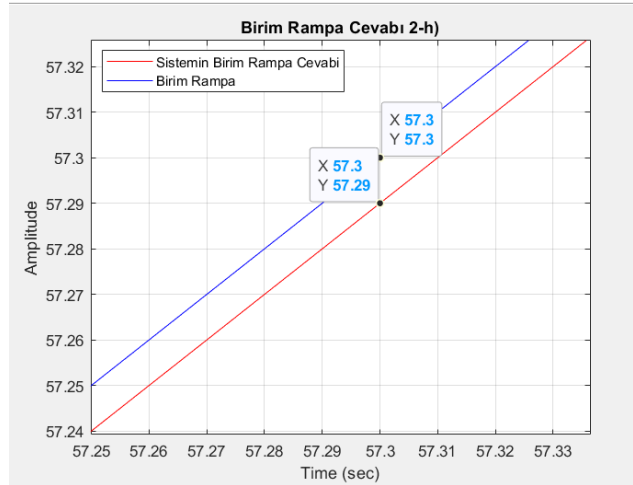
2. Sorunun C şıkkı için tasarlanmış olan “faz geriletici” kontrolcü bu aşamada tasarlanmış olan “fazGeriletici01” kontrolcüsü ile sağlanmaktadır. “ $G = 1/(s*(s+2))$ ” sistemi için “p” değeri “-1+1.02j” ve “ess_d” değeri “0.01” olarak tanımlandıktan sonra “C = fazGeriletici01(G, p, ess_d);” satırı ile kontrolcü oluşturulmaktadır. Sönüm oranı 0.7 olacak şekilde “p” kutup değişkeni “G” sisteminin kök yer eğrisi üzerinden seçilmiştir. Sorunun C şıkkında elde edilen tasarım elde edilen kontrolcü ile bu aşamada elde edilen kontrolcüler eşittir. Oluşturulan kontrolcünün transfer fonksiyonu ile ilgili bilgiler Şekil-65’te gösterildiği gibi komut ekranına verilmektedir. Oluşturulan kontrolcü “ $2.04*(s+0.5)/(s+0.005)$ ” şeklindedir. Kontrolcü “C” oluşturulduktan sonra “Gcl_h” değerine kapalı çevrim sistemi transfer fonksiyonu oluşturulmaktadır. “Gcl_h” kapalı çevrim sisteminin birim rampa cevabının ve birim rampa fonksiyonunun üst üste çizdirilmiş hali Şekil-66’da gösterilmektedir. Şekil-67’de birim rampa referans takip hatasının fonksiyona verilen “ess_d” değeri ile tutarlı olduğu gösterilmektedir.

```
Command Window
tf with properties:

    Numerator: {[2.0404 1]}
    Denominator: {[1 0.0050]}
    Variable: 's'
    IODelay: 0
    InputDelay: 0
    OutputDelay: 0
    Ts: 0
    TimeUnit: 'seconds'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
    Notes: [0x1 string]
    UserData: []
    Name: ''
    SamplingGrid: [1x1 struct]
```



Şekil 65: “C” faz geriletici kontrolcü parametreleri. Şekil 66: Birim rampa referans takip hatası analizi 1



Şekil 67: Birim rampa referans takip hatası analizi 2

“ $G = 8/s*(s+10)$ ”, “ $p=-10+4j$ ” ve “ess_d = 0.2” değerleri için “fazGeriletici01” fonksiyonu kullanılarak “C” kontrolcü Şekil-68’de gösterildiği gibi elde edilmiştir. Oluşturulan kontrolcü “ $C(s) = 5.3852(s + 0.00581)/(s+0.005)$ ” şeklindedir. Kapalı çevrim sistemi “Gcl_h” oluşturulduktan sonra bu sistemin birim rampa cevabı ve birim rampa fonksiyonu üst üste Şekil-69’da gösterildiği gibi çizdirilmektedir. Birim rampa referans takip hatası analizi yapıldığında “G” sistemi için birim rampa sürekli hal hatası “ess_d” fonksiyona verildiği “0.23” değerinde gözlemlenmektedir.

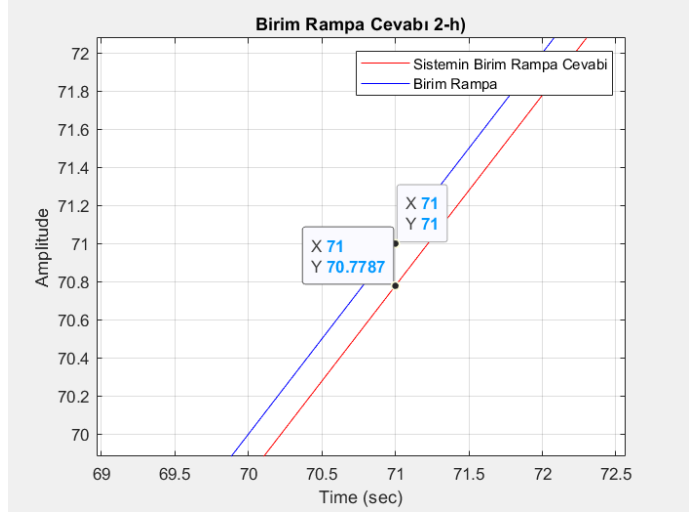
```

Command Window

tf with properties:

    Numerator: {[5.3852 0.0313]}
    Denominator: {[1 0.0050]}
    Variable: 's'
    IODelay: 0
    InputDelay: 0
    OutputDelay: 0
    Ts: 0
    TimeUnit: 'seconds'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
    Notes: [0x1 string]
    UserData: []
    Name: ''
    SamplingGrid: [1x1 struct]

```



Şekil 68: “C” faz geriletici kontrolcü parametreleri. Şekil 69: Birim rampa referans takip hatası analizi

“ $G = 1/s*(s+2)$ ”, “ $p=-1+j$ ” ve “ $ess_d = 0.89$ ” değerleri için “fazGeriletici01” fonksiyonu kullanılarak “C” kontrolcü Şekil-70’de gösterildiği gibi elde edilmiştir. Oluşturulan kontrolcü “ $C(s) = 2(s + 0.0056)/(s+0.005)$ ” şeklindedir. Kapalı çevrim sistemi “Gcl_h” oluşturulduktan sonra bu sistemin birim rampa cevabı ve birim rampa fonksiyonu üst üste Şekil-71’de gösterildiği gibi çizdirilmektedir. Birim rampa referans takip hatası analizi yapıldığında “G” sistemi için birim rampa sürekli hal hatası “ ess_d ” “0.96” değerinde gözlemlenmektedir.

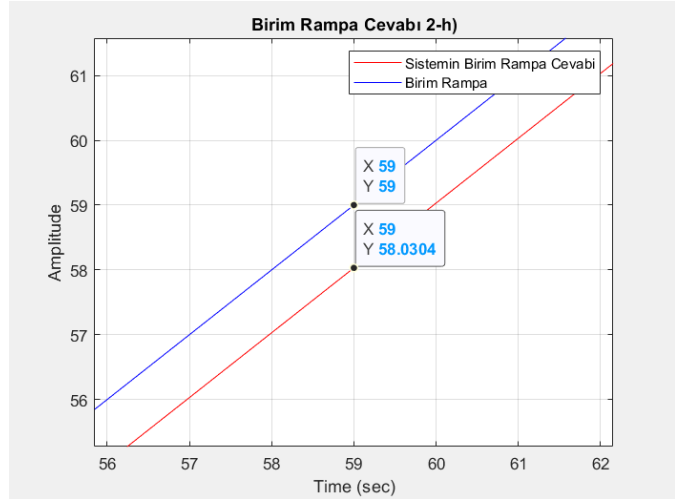
```

Command Window

tf with properties:

    Numerator: {[2.0000 0.0112]}
    Denominator: {[1 0.0050]}
    Variable: 's'
    IODelay: 0
    InputDelay: 0
    OutputDelay: 0
    Ts: 0
    TimeUnit: 'seconds'
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
    Notes: [0x1 string]
    UserData: []
    Name: ''
    SamplingGrid: [1x1 struct]

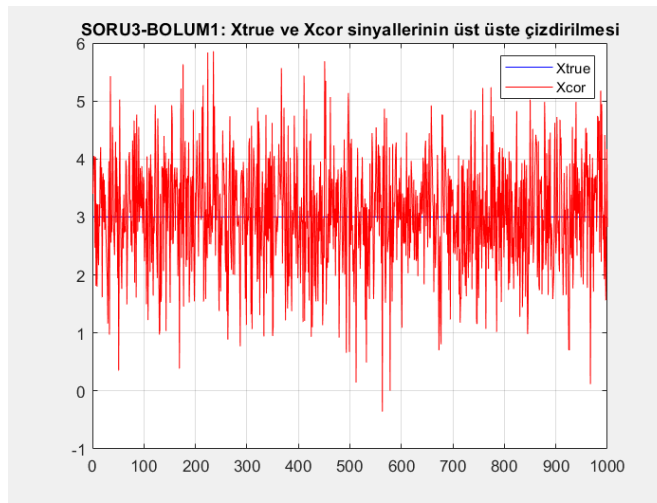
```



Şekil 70: “C” faz geriletici kontrolcü parametreleri. Şekil 71: Birim rampa referans takip hatası analizi

3. SORU 3

3. Sorunun 1. Bölümünde tüm elemanları 3 olan 1000 elemanlı bir vektör "linspace(3,3,1000)" komutu ile üretilmiştir ve "x_true" değerine bu vektör kaydedilmiştir. Sonrasında "randn(1,1000)" fonksiyonu kullanılarak "x_true" ile aynı boyutta bir "v" vektörü daha tanımlanmıştır. Bu "v" vektörü için ortalama ve varyans değerleri sırasıyla "mean_v = mean(v)" ve "var_v = var(v)" şeklinde bulunarak "mean_v" ve "var_v" değişkenlerine kaydedilmektedir. (https23) "v" vektörü için örnek bir ortalama ve varyans değeri sırasıyla "-0.0386", "1.0820" olarak bulunmuştur. "v" vektörü için ortalama değeri her zaman 0 değerine, varyans değeri ise her zaman 1 değerine yakın bulunmaktadır. Oluşturulmuş olan "v" vektörünün "x_true" sinyaline eklenmesiyle yani "x_cor = x_true + v" işlemiyle 1000 elemanlı "x_cor" sinyali elde edilmektedir. "x_cor" sinyali ile "x_true" sinyallerinin aynı grafikte çizdirilmiş hali Şekil-72'de gösterilmektedir. "x_true" sinyali mavi, "x_cor" sinyali ise kırmızı renk ile çizdirilmiştir.



Şekil 72: "x_true" ve "x_cor" sinyallerinin üst üste çizdirilmesi

3. Sorunun 2. Bölümünde bir önceki aşamada elde edilmiş olan "x_cor" sinyali kullanılarak gerçek sinyal "x_true" için Denklem-1 ile ifade edilen optimizasyon problemi oluşturularak bir tahmin elde edilmeye çalışılmıştır.

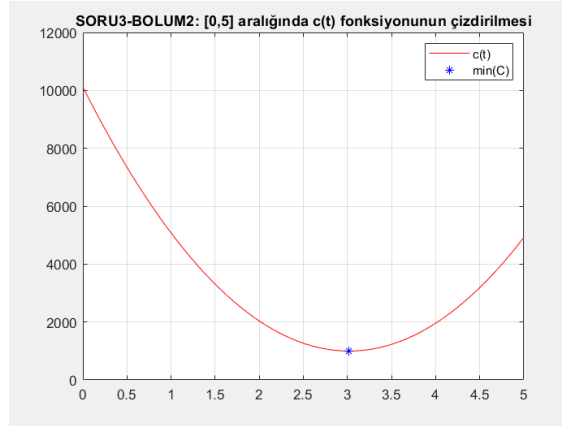
$$\min_{t \in R} \sum_{k=1}^{1000} (t - x_{cor}(k))^2 \quad (\text{Denklem-1})$$

Problemi çözme işlemine başlamadan önce [0,5] aralığında 1000 değerden oluşan "tvec" vektörü "linspace(0,5,1000)" komutu ile oluşturulmaktadır. "tvec" vektörünün her bir elemanı için Denklem-2 ile ifade edilen "c(t)" fonksiyonu oluşturulmaktadır.

$$c(t) = \sum_{k=1}^{1000} (t - x_{cor}(k))^2 \quad (\text{Denklem-2})$$

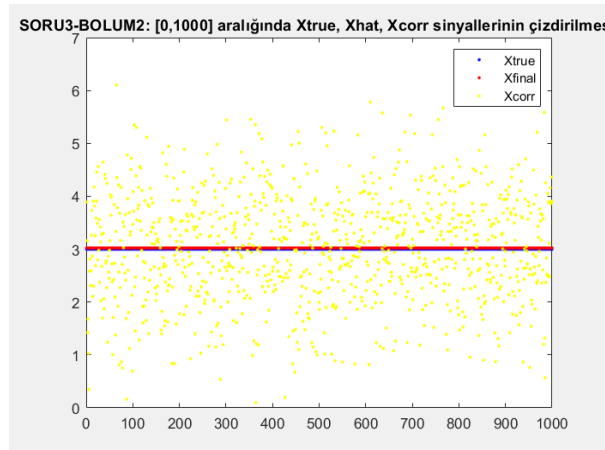
Bu işlem öncesinde 1000 elemanlı her elemanı sıfır olan bir "c" vektörü "linspace(0,0,1000)" komutu ile oluşturulmaktadır. "c" vektörü oluşturulduktan sonra iki adet "for" döngüsünün iç içe kullanılmasıyla "tvec" vektörünün her bir "t" elemanı için Denklem-2'deki optimizasyon problemi çözülürmektedir. Dıştaki "for" döngüsü ile "tvec" vektörünün her bir elemanı için gerekli işlem yapılmaktadır, içteki "for" döngüsü ile her "t" anındaki toplam serisi problemi "k" değişkeni için çözülürmektedir. İçteki "for" döngüsü seri toplamı için dıştaki "for" döngüsü ise her "t" anında işlemin tekrarlanması amacıyla kullanılmaktadır. (https24) Seri toplamı için ise içteki "for" döngüsü kullanılmaktadır. Örneğin "c(t=3)" anında "c(3)=c(3)+(tvec(3)-x_cor(k(1:1000))).^2" denklemi [1,1000]

aralığında her “k” değeri için toplanmaktadır. Bu işlemin yapıldığı komut satırı “ $c(t) = c(t) + (tvec(t) - x_cor(k)).^2$ ” şeklindedir. Döngüler tamamlandıktan sonra “c” sinyali, “tvec” eksenine karşılık çizdirilmektedir. “islocalmin(c)” fonksiyonu ile “min_c” değerine “c” sinyalinin minimum noktalarının koordinatları kaydedilmektedir. Çizdirilen grafik üzerine ek olarak “plot(tvec(min_c),c(min_c),‘*b’)” komut satırı ile “c(t)” fonksiyonunun minimum noktası işaretlenmektedir. (https25) [0,5] aralığında çizdirilen grafik Şekil-73’te gösterilmektedir. “c” sinyalinin minimum noktasının yaklaşık olarak “t=3” anında bulunması gerçek sinyal “x_true” ile alakalıdır. Denklem-2 ile verilen optimizasyon probleminde “tvec” değişkeninin “x_true” sinyali için bir tahmin olduğu çıkarımı yapılmaktadır.



Şekil 73: [0,5] aralığında c(t) sinyalinin ve minimum noktasının işaretlenmesi.

Bu aşamada Denklem-1 ve Denklem-2’de verilmiş olan optimizasyon problemi “fminunc” fonksiyonu ile yapılmıştır. İlk olarak “fminunc” fonksiyonu için verilecek olan “options” argümanı “options=optimoptions(@fminunc,'Algorithm','quasi-newton','OptimalityTolerance',1e-10,'MaxFunctionEvaluations',1e7,'Display','iter')” şeklinde tanımlanmaktadır. (https26) Sonrasında “fminunc” fonksiyonuna verilecek olan diğer bir argüman “t” değişkeni için başlangıç koşulu olan “t0”, “linspace(0,0,1000)” şeklinde oluşturulduktan sonra “fminunc” fonksiyonuna verilmektedir. “fun1” yani optimize edilmesi istenen fonksiyon “@(t) ((sum(t(1:1000)-x_cor(1:1000)).^2));” şeklinde oluşturulmaktadır. (https27) Burada önceki aşamadan farklı olarak Denklem-2’de verilen optimizasyon problemi “sum” fonksiyonu tarafından sağlanmaktadır. (https28) Önceki aşamada “for” döngüleri ile sağlanmış olan her 1000 eleman için yapılan işlemler bu aşamada “t(1:1000)” ve “x_cor(1:1000)” tarafından yapılmaktadır. İlgili fonksiyon “fun1”, başlangıç değeri “t0” ve “options” argümanları “fminunc” fonksiyonuna “[xfinal,fval,exitflag1] = fminunc(fun1,t0,options);” şeklinde verilerek “x_true” sinyali için bir tahmin “xfinal1” sinyali elde edilmektedir. “x_true”, “x_final1” ve “xcor” sinyalleri üst üste çizdirilmiştir. Grafik Şekil-74’te gösterilmektedir. Gerçek sinyal için “fminunc” fonksiyonu ile elde edilen tahmin sinyali “xfinal1” tutarlıdır. “fminunc” fonksiyonu ile gerçek sinyale çok yakın bir sinyal elde edilmiştir.



Şekil 74: “fminunc” fonksiyonu ile Denklem-2’nin optimizasyonu.

“fminunc” fonksiyonuna giriş olarak verilen “options” argümanı ile fonksiyon altında yapılan optimizasyonun iterasyon adımlarının komut ekranında gösterilme işlemi yapılabilmektedir. Şekil-75’te “fminunc” fonksiyonu ile bu aşamada yapılan optimizasyon adımları gösterilmektedir. Verilen argümanlar ile fonksiyonun “lokal minimum” değeri bulunmuştur.

Command Window				
Iteration	Func-count	f(x)	Step-size	First-order optimality
0	1001	9.13601e+06		6.05e+03
1	2002	4.09084e+06	0.000165423	4.05e+03
2	3003	0.000712142	1	0.0534
3	4004	7.67274e-15	1	1.3e-07

Local minimum found.

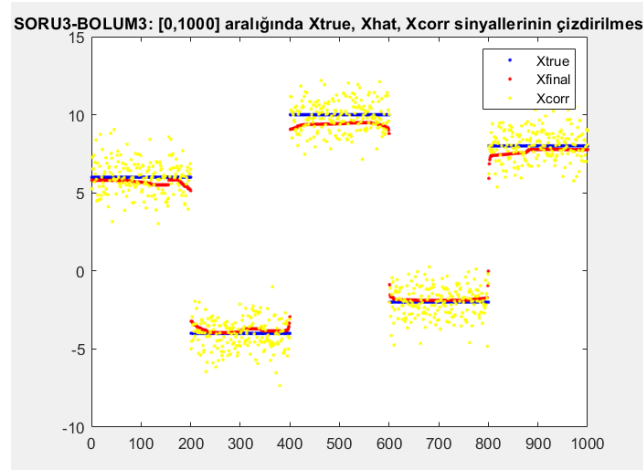
Optimization completed because the size of the gradient is less than

Şekil 75: “fminunc” optimizasyon adımları.

3. Sorunun 3. Bölümünde parçalı-sabit (piece-wise constant) “x_true” sinyali “[6*ones(1,200),-4*ones(1,200),10*ones(1,200),-2*ones(1,200),8*ones(1,200)];” şeklinde oluşturulmuştur. Bölüm-1’de olduğu gibi “v” sinyali “randn(1,1000)” komutu ile üretilmiştir. “x_cor” sinyali, “x_true” ve “v” sinyallerinin toplanmasıyla oluşturulmuştur. Bu durumda “x_true” sinyali için bir tahmin olan “x_hat” sinyalini elde etmek için Denklem-3’te verilen optimizasyon problemi oluşturularak bir tahmin elde edilmeye çalışılmıştır.

$$\min_{t \in R} \sum_{k=1}^{1000} (\hat{x}(k) - x_{cor}(k))^2 + \left(\sum_{k=1}^{999} |\hat{x}(k) - x_{cor}(k)| \right)^2 \quad (\text{Denklem-3})$$

Bir önceki aşamadan farklı olarak bu aşamada “fminunc” fonksiyonuna verilecek olan optimize edilmesi istenen fonksiyon “fun3 = @(xhat) (sum((xhat(1:1000) - x_cor(1:1000)).^2) + sum(norm(diff(xhat(1:1000)),1)).^2);” anonim fonksiyon şeklinde oluşturulmaktadır. Denklem-3’ün ikinci bileşenindeki ifadenin “diff” fonksiyonu ile türevi alınarak türevi alınmış ifadenin birinci normu için “norm” fonksiyonunun ikinci argümanının parametresi “1” olarak girildikten sonra “fun3” ile ifade edilen optimizasyon probleminin “fminunc” fonksiyonuna verilmesi sonucu “xfinal3” sinyali elde edilmiştir. (https29, https30, https31) “x_true” için elde edilen “xfinal3” tahmin sinyaline karşılık “x_true” ve “x_cor” sinyalleri üst üste çizdirilmiştir. Grafik Şekil-76’da gösterilmektedir. Optimizasyon Bölüm-2’de yapılan optimizasyona kıyasla daha uzun sürmektedir. Bu süre “fminunc” için giriş argümanı olan “options” parametresi ile ‘quasi-newton’ argümanı ‘trust-region’ olarak ayarlandıktan sonra ve ‘SpecifyObjectiveGradient’, true, ‘HessianFcn’, ‘objective’ parametreleri sırasıyla eklenerek gradient vektörü ve/veya Hessian matrisi kullanılarak azaltılabilmektedir. (https32, https33, https34, https35)



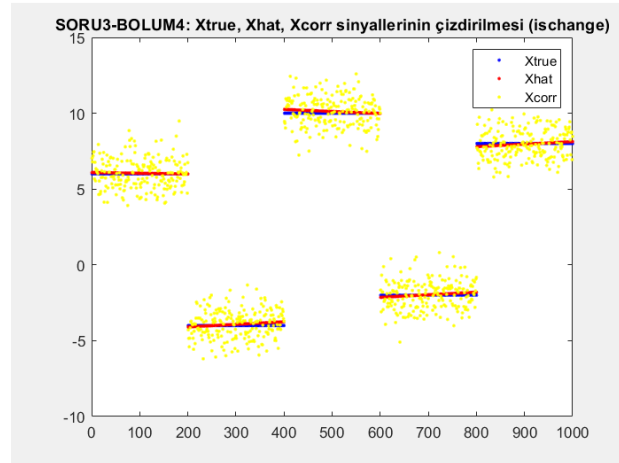
Şekil 76: “fminunc” fonksiyonu ile Denklem-3’ün optimizasyonu (fun3)

Denklem-3'te verilmiş olan problem ile yapılan optimizasyon sonucunda “x_true” sinyaline yakın “xhat” tahmin sinyali elde edilmektedir. “fun3” fonksiyonu için “fmincun” fonksiyonunun iterasyon adımları Şekil-77’de gösterilmektedir. Optimizasyon yaklaşık olarak 400 iterasyon sürmektedir. Daha iyi bir tahmin sinyali “xhat” elde edebilmek için “Options” argümanı “‘MaxFunctionEvaluations’,1e7” olarak ayarlanmıştır.

Command Window					
397	1218217	3128.11	0.00279679	187	
398	1222221	3127.8	0.00259135	187	
399	1226225	3127.5	0.00330768	187	
				First-order	
Iteration	Func-count	f(x)	Step-size	optimality	
400	1230229	3127.15	0.00261079	187	
401	1234233	3126.83	0.00283558	186	
<u>Solver stopped prematurely.</u>					
fminunc stopped because it exceeded the iteration limit, <u>options.MaxIterations</u> = 4.000000e+02.					
fx >>					

Şekil 77: “fminunc” optimizasyon adımları. (Fun3)

3. Sorunun 4. Bölümünde bir önceki aşamada yapılan optimizasyon işlemi sonucu elde edilen tahmin sinyalleri bu aşamada “ischange” komutunun kullanılmasıyla bulunmaktadır. (https36) “x_cor” sinyali “[xhat,S1,S2] = ischange(x_cor,'linear','Threshold',200);” şeklinde bu fonksiyona argüman olarak verilerek “S1” ve “S2” değerleri elde edilmektedir. “xcor” sinyalindeki değişimleri temsil eden bu parametreler “xhat_segline” sinyalinin elde edilmesinde “xhat_segline = S1.*(1:1000) + S2;” komut satırındaki gibi kullanılmaktadır. “xhat_segline” sinyali “xtrue” sinyali için bir tahmin sinyali. Bu sinyal bulunduktan sonra “xtrue”, “xcor” ve “xhat_segline” sinyalleri üst üste çizdirilmiştir. Grafik Şekil-78’de gösterilmektedir. 3. Bölümde “fminunc” fonksiyonu ile yapılan optimizasyonlarla elde edilmiş olan sonuçlara kıyasla “ischange” komutu ile “xtrue” sinyali için daha net sonuç tahmini sinyali “xhat” elde edilmektedir.



Şekil 78: “ischange” fonksiyonu ile optimizasyon.

Referanslar

1. <https://www.mathworks.com/help/ident/ref/dynamicsystem.bode.html#d124e14967> (https1)
2. <https://www.mathworks.com/help/ident/ref/dynamicsystem.impulse.html> (https2)
3. <https://www.mathworks.com/help/matlab/ref/isscalar.html> (https3)
4. <https://www.mathworks.com/help/matlab/ref/inputparser.html> (https4)
5. <https://www.mathworks.com/help/matlab/ref/islogical.html> (https5)
6. <https://www.mathworks.com/help/ident/ref/bodeoptions.html> (https6)
7. <https://www.mathworks.com/help/ident/ref/dynamicsystem.bodeplot.html> (https7)
8. <https://www.mathworks.com/help/matlab/ref/num2str.html> (https8)
9. <https://www.mathworks.com/matlabcentral/answers/32376-insert-degree-symbol-in-axes-title> (https9)
10. <https://www.mathworks.com/help/matlab/ref/filter.html> (https10)
11. <https://www.mathworks.com/help/signal/ref/butter.html> (https11)
12. <https://www.mathworks.com/help/ident/ref/dynamicsystem.tfdata.html> (https12)
13. <https://www.mathworks.com/help/dsp/ref/dfilt.cascadeallpass.html> (https13)
14. <https://www.mathworks.com/help/control/ref/dynamicsystem.isstable.html> (https14)
15. <https://www.mathworks.com/help/ident/ref/dynamicsystem.damp.html> (https15)
16. https://paulomarconi.github.io/blog/Phase-lead_phase-lag/ (https16)
17. <https://www.mathworks.com/help/ident/ref/dynamicsystem.zero.html> (https17)
18. <https://www.youtube.com/watch?v=ql-WOupopeY&list=PLXpRXgXiS1G8H8ies5GkHc4PIOTyvEpUt&index=23&t=3658s> (https18)
19. https://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_Ess (https19)
20. <https://www.mathworks.com/help/symbolic/limits.html> (https20)
21. <https://www.mathworks.com/help/symbolic/sym.sym2poly.html> (https21)
22. <https://www.youtube.com/watch?v=gjNZdUuEZPY&list=PLXpRXgXiS1G8H8ies5GkHc4PIOTyvEpUt&index=23&t=2739s> (https22)
23. <https://www.mathworks.com/help/matlab/ref/mean.html> (https23)
24. <https://www.mathworks.com/matlabcentral/answers/854200-how-to-make-a-sum-series-using-a-for-loop> (https24)
25. <https://www.mathworks.com/matlabcentral/answers/435628-minimum-point-of-graph> (https25)
26. https://www.mathworks.com/help/optim/ug/fminunc.html#butpb7p_sep_shared-x0 (https26)
27. <https://www.appsloveworld.com/bestanswer/matlab/115/how-to-use-fminunc-with-summation-in-matlab> (https27)
28. <https://www.mathworks.com/help/matlab/ref/sum.html#d124e1470540> (https28)
29. <https://www.mathworks.com/help/matlab/ref/norm.html> (https29)
30. <https://math.stackexchange.com/questions/289989/first-and-second-derivative-of-a-summation> (https30)
31. <https://www.mathworks.com/help/matlab/ref/diff.html> (https31)
32. <https://www.mathworks.com/help/optim/ug/hessian.html#bsapedg> (https32)
33. <https://www.mathworks.com/help/optim/ug/minimization-with-gradient-and-hessian.html> (https33)
34. <https://www.mathworks.com/help/optim/ug/optimize-large-nonlinear-problems.html> (https34)
35. <https://www.mathworks.com/help/optim/ug/writing-scalar-objective-functions.html#bu2xbs> (https35)
36. <https://www.mathworks.com/help/matlab/ref/ismatch.html#d124e800530> (https36)