HAFTA 4: SİSTEMLER

İçindekiler

BÖLÜM 2: SİSTEMLER

Elektrik-Elektronik Mühendisliği'nin ilgi alanına giren sistemlere ya da bu sistemlere ait karmaşık matematiksel analizlere girmeden önce sistem tanımını yapmakta fayda vardır.

2.1 Sistem Nedir?

Sistem; Türk Dil Kurumu (TDK) sözlüğüne göre muhtelif anlamlar taşımaktadır. Mühendisler için belki de en uygun olanı: "Bir aracı oluşturan düzen, düzenek, tertibat" olarak tanımlanmaktadır (www.tdk.gov.tr). Başka bir deyişle, bir bütünü teşkil edecek şekilde birbiri ile ilişki içinde ya da bağımsız olarak bulunan bir dizi bileşen ve düzenek sistem olarak adlandırılmaktadır. Ben bu tanımların tamamında bir sıkıntı görüyorum. Örnek olarak arabayı sistem olarak ele aldığımızda, bu arabaya ateşleme, aktarma organlarının oluşturduğu bir bütün gözü ile mi bakmak daha geçerli ve faydalıdır? Yoksa bizi A noktasından B noktasına ulaştıran bir araç gözü ile mi bakmak daha geçerli ve faydalıdır? (Daha doğrudur diye sormuyorum çünkü bu tanımların hepsi bir şekilde doğru olmalı, aksi takdirde aklımıza araba örneği gelmezdi).

TDK sözlüğünde verilen bir diğer tanım ise "Bir sonuç elde etmeye yarayan yöntemler düzeni" olarak geçmektedir (www.tdk.gov.tr). Bu tanım bana daha yakın geldi. Ancak hala bir eksik görüyorum. Sonuç elde etmek için düzen kurma, düzenek… Çok bencilce. Acaba bu sistemin bir girdisi yok mudur? Sistem istediğimiz sonucu bize vermek için hiçbir girdiye ihtiyaç duymaz mı? Bu çıktılar ya da sonuçlar yoktan mı elde edilir acaba?

Şimdi tanımlar hakkında bilgi sahibi olduğumuza göre fikir sahibi de olabiliriz. Benim aklıma yatan, içime sinen ve en önemlisi bir mühendis gözü ile baktığımızda "Budur" diyebileceğimiz tanım: Verilen bir girişe göre istenilen çıkışı elde etmek üzere tasarlanan, birbiri ile ilişki içinde ya da bağımsız olarak çalışan düzenek bütününe sistem denir. Burada sisteme tasarımcı gözü ile baktığımıza dikkat ediniz. Sistem bize tasarlanmış olarak da verilebilirdi. Bu durumda biz sadece verdiğimiz giriş için bir çıkış alır ve sistemi kullanıcı gözü ile de irdeleyebilirdik. Şekil 2.1'de bize teslim edilmiş olan bir sürekli zaman sistemine ait blok diyagram verilmiştir.

$$x(t) \longrightarrow \boxed{\text{Sistem}} \longrightarrow y(t)$$

Şekil 2.1 Sürekli zaman sistem blok diyagramı

Sinyaller ve Sistemler kapsamında belki de en uygun tanım: Verilen bir giriş sinyali için istenilen çıkış sinyalini elde etmek üzere tasarlanan, birbiri ile ilişki içinde ya da bağımsız olarak çalışan düzenek bütününe sistem denir, diyebiliriz.

Gelelim sistemin detaylarına. Araba örneğine geri dönersek, arabanın girişi benzin, dizel, elektrik olabileceği gibi, bu girişi ne derece ekonomik tükettiği ise bir sistem özelliği olacaktır. Sistem özelliklerine göre de tercihlerimiz değişecektir. Bu durumda sistem kadar önemli olan bir diğer husus, sistem özellikleridir.

2.2 Sistem Özellikleri

Bu bölümde sürekli zaman sistemlerine ait özellikler açıklanacak, bu sistemlere ait matematiksel tanımlar verilecek ve en önemlisi bu modeller ile bu modellerin gerçek dünyada ifade ettiği sistemler arasında bir köprü kurulacaktır.

2.2.1 Bellekli ve belleksiz sistemler

İngilizce literatürde "systems with and without memory (memoryless)" olarak adlandırılan bu sistem özellikleri için tanım son derece basittir. Bir sistemin çıkışı, girişin SADECE o andaki değerlerine bağlı ise bu sisteme belleksiz sistem denir.

$$y(t) = ax(t) + bx^2(t) \tag{2.1}$$

Eşitlik 2.1'de verilen sistem belleksiz sisteme güzel bir örnektir. Sistem çıkış sağlamak için girişin sadece o andaki değerine ihtiyaç duyar, o değerin karesini alır, gerekli işlemleri yaparak bize sonucu sunar. Sistemin çıkışı, girişin SADECE o andaki değerlerine bağlı olduğundan sistem belleksizdir. Bellekli sisteme ait en güzel örnek ise Eşitlik 2.2'de matematiksel olarak ifade edilen akümülatördür (biriktirici, toplayıcı). Girişin $x[n] = \delta[n]$, çıkışın $y[n] = u[n]$ olduğu bu sistemde çıkışın o anki değerini sunabilmek için girişin o ana kadarki bütün değerlerine ihtiyaç vardır. Bu durumda sistemin bir bellek ihtiyacı duyduğu açıktır.

$$u[n] = \sum_{m=-\infty}^{n} \delta[m] \tag{2.2}$$

Dolayısıyla bir sistemin çıkışı, girişin önceki ve/veya sonraki değerlerine bağlı ise bu sisteme bellekli sistem denir.

Bu bölümü kapatmadan evvel biraz dimağımızı açmakta fayda vardır. Acaba Eşitlik 2.3 ve 2.4'te sunulan sistemler bellekli midir?

$$y(t) = x^2(t+1) \tag{2.3}$$

$$y(t+1) = x^2(t+1) \tag{2.4}$$

Eşitlik 2.3'te sistemin çıkış sinyali $y(t)$, giriş sinyali $x(t)$'nin sonraki değerlerine bağlı olarak değiştiği için, bu sisteme bellekli sistem diyebiliriz. Öte yandan Eşitlik 2.4'te, sistemin çıkış sinyali $y(t)$, giriş sinyali $x(t)$'nin sadece o anki değerine bağlı olarak değişmektedir (Örneğin, $y(1)$, $y(2)$ ve $y(3)$ değerlerini elde edebilmek için; sırasıyla $t = 0$, $t = 1$ ve $t = 2$ zamanlarında $x(1)$, $x(2)$ ve $x(3)$ değerlerine ihtiyaç duyulmaktadır.). Dolayısıyla Eşitlik 2.4'te ifade edilen sistem belleksiz bir sistemdir.

2.2.2 Doğrusal sistemler

Bu ve bir sonraki bölümde anlatılacak olan sistemlerin Elektrik-Elektronik Mühendisliği bölümünde çok özel bir yeri vardır. Gerçek dünyada karşılaştığımız birçok sistem doğrusal ve zamanla değişmez sistemler olarak modellenebilir.

Doğrusal (linear) bir sistem çok basit iki özelliğe sahiptir. Toplanırlık (additivity) ve Homojenite (homogeneity). Özellikler basit olmakla birlikte nedense öğrencilerin %80'i verilen örnek bir sistemin doğrusal olup olmadığını ispatlayamazlar ya da yanlış ispatlarlar.

a) <u>Toplanırlık</u>: Sisteme $x_1(t)$ sinyali girdiğimizde $y_1(t)$ sinyalini ve $x_2(t)$ sinyali girdiğimizde $y_2(t)$ sinyalini elde ediyorken; $x_1(t) + x_2(t)$ sinyalini girdiğimizde $y_1(t) + y_2(t)$ sinyalini elde ediyor isek sistem doğrusallığın ilk basamağını geçmiştir.

b) <u>Homojenite</u>: Sisteme $x_1(t)$ sinyali girdiğimizde $y_1(t)$ sinyalini elde ediyorken, $ax_1(t)$ sinyalini girdiğimizde $ay_1(t)$ sinyalini elde ediyor isek (a herhangi bir sabit olmak üzere) sistem artık doğrusaldır diyebiliriz.

Bu durumda; a ve b şıklarını birleştirerek, süperpozisyon (superposition) özelliğini sağlayan sistemler doğrusal sistemlerdir. Başka bir deyişle; $ax_1(t) + bx_2(t)$ sinyalini girdiğimizde $ay_1(t) + by_2(t)$, (a ve b herhangi iki sabit olmak üzere) çıkış sinyalini elde ettiğimiz sistemlere doğrusal sistemler diyebiliriz.

Doğrusallık (linearity) son derece önemli bir sistem özelliğidir. Bunun nedeni büyük ve zor bir problemi $y(t) = a_1y_1(t) + a_2y_2(t) + a_3y_3(t) + \cdots + a_Ny_N(t)$ N adet küçük ve basit probleme indirgeyerek bu problemlerin her birini ayrı ayrı çözmemiz durumunda, N bağımsız problem çözümünü toplayarak büyük problemin çözümünü elde edebilmemizdir. Bu ancak doğrusal bir problem, doğrusal bir sistem için geçerli olacaktır.

Tüm sistem özellikleri tamamlandığında geniş kapsamlı örnekler çözülecektir.

2.2.3 Zamanda değişmez sistemler

Bir sistemden beklenen belki de en önemli ve gerekli özellik zamanda değişmezliktir. Zamanda değişmezlik tanımı son derece kolaydır. Bir sistemin girişinde t_0 kadar gecikme oluyorsa, çıkışında da t_0 kadar gecikme olması beklenir. Bu durumda sistem zamanda değişmezdir (time invariant).

Bir sisteme $x(t)$ sinyalini girdiğimizde $y(t)$ sinyalini elde ediyorken, $x(t - t_0)$ sinyalini girdiğimizde çıkış aynı miktarda gecikerek $y(t - t_0)$ çıkış sinyalini veriyorsa sistem zamanda değişmez (time invariant) bir sistemdir.

Bir önceki bölümde açıklandığı üzere bir sistem hem doğrusal, hem de zamanda değişmezlik özelliğini sağlıyorsa çok özel bir sınıf kategorisine sahiptir [LTI (linear, time invariant) veya DZD (Doğrusal, zamanda değişmez)]. DZD sistemlerin matematiksel olarak modellenmesi son derece kolaydır ve birçok sistem DZD sistem özelliği gösterdiğinden DZD olarak modellenebilir.

DZD bir sistemin çıkışı ile girişi arasında katlama (convolution) ilişkisi bulunmaktadır. Katlama kelimesinin İngilizce karşılığının "folding" yada "origami" olması ifadede hata olduğunu düşündürmemelidir. "Convolution" işleminin sinyal işlemede karşılığı Türkçe sinyal işleme literatürüne "katlama" olarak yerleşmiştir.

Katlamaya ait matematiksel ilişki ispatlanarak sunulacaktır. Lütfen en sade şekilde verilen ispat bölümlerini atlamayın, özümsemeye çalışın.

2.2.4 Nedensel sistemler

Eğer bir sistemin çıkışı, girişin o an dâhil olmak üzere önceki değerlerine bağlı ise sisteme nedensel (casual) sistem denir. İlk bakışta karışık gibi gelen bu tanım aslında çok basittir. Eğer sistemimiz nedensel ise çıkış verebilmesi için bulunduğumuz andan sonraki hiçbir girdiye ihtiyaç duymamalıdır. Doğaldır ki ileri zamanda elde edebileceğim bir girdi ile bir sistemden çıktı almam nedensel bir sistem için mümkün değildir. Ancak nedensel olmayan (non-casual) bir sistem için böyle bir çıktı almak mümkün olabilir.

Geleneksel çoktan seçmeli sorulardan farklı olan aşağıdaki ifadeler için bir miktar düşünmekte fayda vardır.

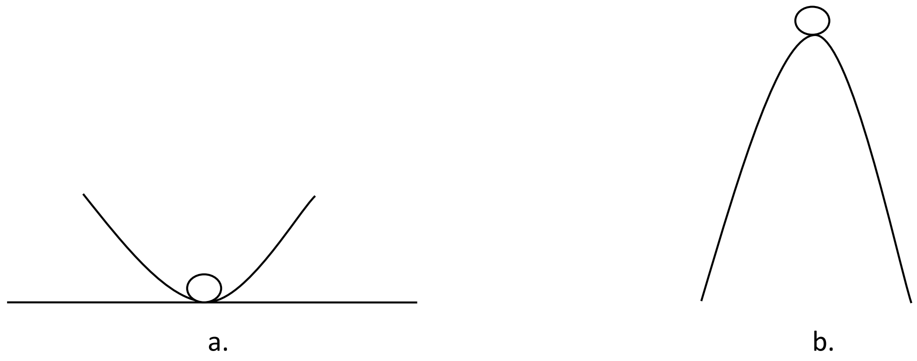
a) Tüm nedensel sistemler belleklidir.
b) Tüm nedensel sistemler belleksizdir.
c) Tüm bellekli sistemler nedenseldir.
d) Tüm bellekli sistemler nedensel değildir.

Yukarıdaki ifadelerin doğruluğu tartışılmayacaktır, okuyucuya bırakılmıştır.

Düşünmeden öğrenmek, vakit kaybetmektir. KONFÜÇYÜS

2.2.5 Kararlı sistemler

Sınırlı bir girişe sınırlı bir çıkış veren sistemlere kararlı sistemler denir. Diğer bir deyişle kararlı bir sistem sonlu bir girişe ıraksayan bir tepki vermemelidir. Kararlı (stable) ve kararsız (unstable) sistemlere en güzel örnek Şekil 2.2'de sunulmuştur.



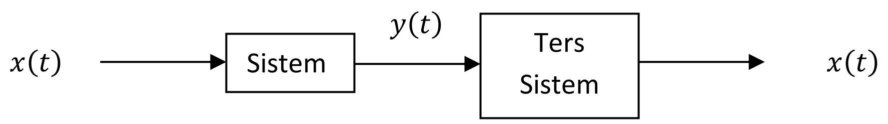
Şekil 2.2 a. Kararlı ve b. Kararsız sistemler.

Şekil 2.2.a'da kararlı sistem için metal topa verilecek sınırlı bir darbe topu çukur içinde belli bir miktar hareket ettirecektir. Oysa Şekil 2.2.b'deki kararsız bir sistem için aynı miktarda sınırlı bir darbe topun tepeden aşağı yuvarlanmasına ve hızının giderek artmasına neden olacaktır. Doğal olarak bu genel bir sistem özelliği tanımıdır. Elektrik-elektronik mühendislerinin ilgi alanına giren sistemlere ait kararlılık farklı şekillerde tezahür edebilir. Örneğin, 12 V'luk bir bataryanın bir direnç ile seri bağlı lambayı belirli parlaklıkta yakması kararlı bir sistem örneği olarak verilebilir. Bu örnekte, sınırlı gerilim ile sınırlı güçte (dolayısıyla parlaklıkta) lambanın yanması söz konusudur.

2.2.6 Tersinir sistemler

Bir sistemin tersinin olması özellikle elektronik ve haberleşme mühendislerinin özel ilgi alanına girmektedir. İlk cümlede üstü kapalı olarak ifade edildiği üzere tüm sistemlerin tersi yoktur. En güzel örnek yazılı bir kâğıdı silen sistem örneğidir. İz bırakmadan mükemmel silme işlemi yapan bir sistemin tersinir sistemi (inverse systems and invertibility) bulunamaz. Benzer şekilde alçak geçirgen bir süzgeç için tersinir sistem tasarlamak mümkün değildir.

Şekil 2.3'te tersinir sistem gösterilmektedir.



Şekil 2.3 Tersinir sistem.

İlk bakışta bir sistemin tersinir olması ve ters sistemin ne olduğu önemli görünmeyebilir. Ancak A noktasındaki $x(t)$ sinyalinin B noktasında tekrar geri çatılabilmesi (elde edilebilmesi), A ile B noktaları arasında iletişim kurulabildiğinin ispatıdır. $x(t)$ sinyaline bağlı olarak sesli ve/veya görüntülü iletişim sağlayabilmek mümkün olacaktır.

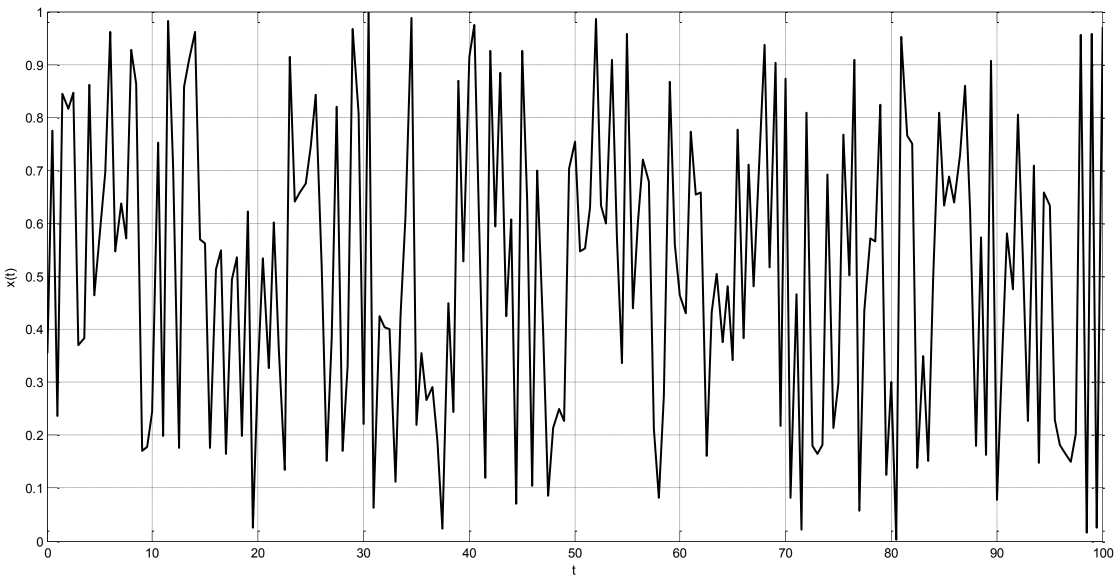
2.3 Sistem özellikleri: Örnek çalışma

Eşitlik 2.5 ile verilen sisteme (Oppenheim, 1996) ait tüm sistem özelliklerini inceleyiniz (τ, gerçel bir sabit olmak üzere).

$$y(t) = x(t) + cos(t + \tau) \tag{2.5}$$

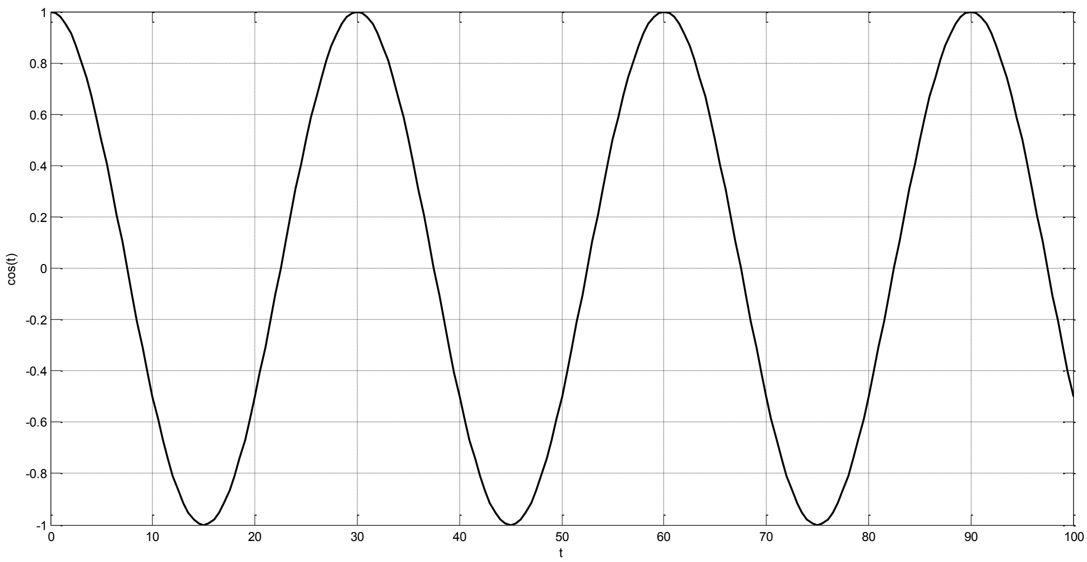
Eşitlik 2.5'te verilen örnek başlangıç olarak birçok öğrenci tarafından zor bulunabilir. Bu örneği kesinlikle öncelikle kendiniz yapmaya çalışmalısınız. Eğer örneği çözemiyor, daha da kötüsü örneğin çözümünü anlamıyor ya da çözüm, Sizin çözümünüz ile herhangi bir biçimde uyuşmuyorsa çok daha basit örnekler ile başlamalısınız. Uygulamaları kendiniz çözerek anlamadan lütfen bir sonraki bölüme geçmeyiniz. Bu örnek çalışma son derece basit ancak bir o kadar temel noktaların anlaşılıp anlaşılmadığını ölçmeye çalışmaktadır. Bu nedenle özel bir hali sadece nedensellik açısından incelenen $(Oppenheim\ ,1996, \tau = 1)$ bu sistem genel hali ile tüm sistem özellikleri açısından detaylı bir biçimde irdelenecektir.

Öncelikle bu sistemin ne işe yaradığını inceleyelim. Sistem kendisine verilen bir giriş sinyaline τ kadar gecikmiş bir kosinüs sinyalini ekleyerek çıkış sinyali üretmektedir. Sistemin nasıl çalıştığını Şekil 2.4'te görsel olarak incelememiz mümkün olacaktır.



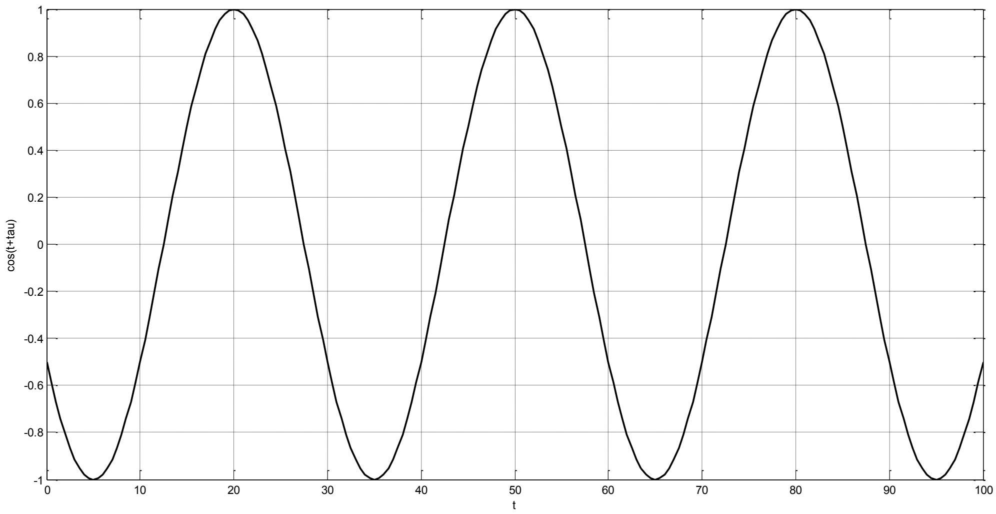
x(t)
t
0
10
20
30
40
50
60
70
80
90
100
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

a.

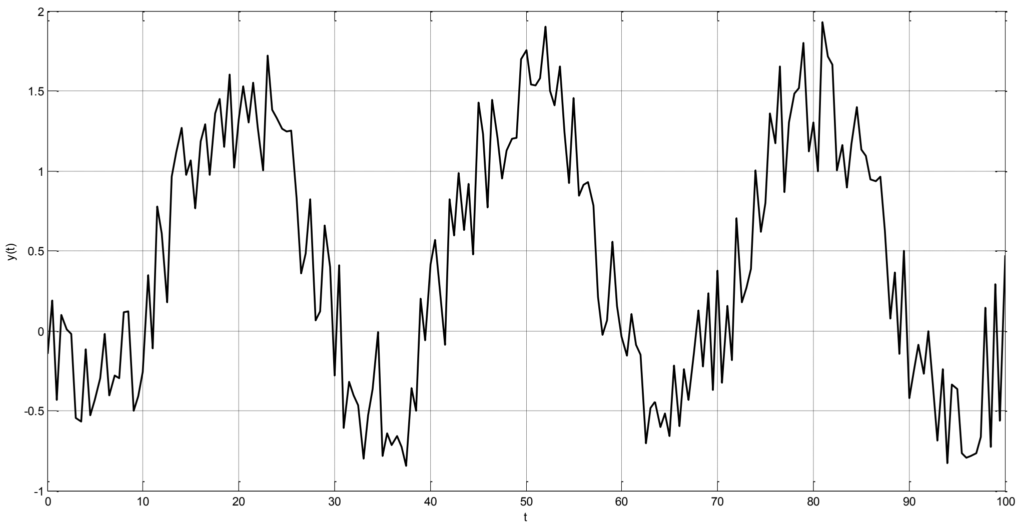


cos(t)
t
0
10
20
30
40
50
60
70
80
90
100
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

b.



c.



d.

Şekil 2.4 $y(t) = x(t) + cos(t + \tau)$ sisteminin sinyallerine ait gösterim. a. Herhangi bir $x(t)$ rastgele sinyali, b. $cos(t)$, c. $cos(t + \tau)$ ve d. $y(t) = x(t) + cos(t + \tau)$.

2.3.1 Bellek kontrolü

Eşitlik 2.5'te, sistemin çıkışı, sistemin girişinin sadece o anki değerine bağlı olarak değiştiği için, söz konusu sistem belleksiz sistemdir. Burada kosinüs terimindeki τ kadar gecikme, giriş sinyali ile ilgili olmadığı için sistemin bellek kontrolünde dikkate alınmamaktadır. Zamanda geciken giriş sinyali değildir. $y(t) = x(t) + cos(t + \tau)$

2.3.2 Doğrusallık kontrolü

Doğrusallık tanımına geri dönersek,

$x(t) = ax_1(t) + bx_2(t)$ sinyalini girdiğimizde $y(t) = ay_1(t) + by_2(t)$, a ve b herhangi iki sabit olmak üzere, çıkış sinyalini elde ettiğimiz sistemlere doğrusal sistemler diyebiliriz.

Bu durumda $x_1(t)$ giriş sinyali uygulandığında Eşitlik 2.6'daki $y_1(t)$ sinyali ve $x_2(t)$ giriş sinyali uygulandığında Eşitlik 2.7'deki $y_2(t)$ sinyali elde edilir.

$$y_1(t) = x_1(t) + \cos(t + \tau) \tag{2.6}$$

$$y_2(t) = x_2(t) + \cos(t + \tau) \tag{2.7}$$

Şimdi sisteme $x(t) = ax_1(t) + bx_2(t)$ girişini uyguladığımızı varsayalım. Çıkış, Eşitlik 2.5 kullanılarak; $y(t) = x(t) + cos(t + \tau) = y(t) = (ax_1(t) + bx_2(t)) + \cos(t + \tau)$ olarak elde edilir. Sadece basit bir biçimde sistemin yeni girişini eski iki girişin süperpozisyonu olarak verdik. Oysa incelediğimiz sistemin doğrusal olma koşulu, $x(t)$ giriş sinyaline $y'(t) = ay_1(t) + by_2(t) = ax_1(t) + bx_2(t) + a\cos(t + \tau) + b\cos(t + \tau)$ çıkış sinyalini vermesidir. Bu durumda sistemin hesaplanan çıkış sinyali $y(t)$ ile sistemin doğrusal olduğu varsayımı altında hesaplanan $y'(t), y(t) \neq y'(t)$ nedeniyle sistemin doğrusal olduğunu söyleyemeyiz. Her ne kadar bir kosinüs terimi doğrusallığı bozsa da (bu sabit bir terim de olabilirdi – incrementally linear), Eşitlik 2.5'te sunulan sistem doğrusal değildir.

2.3.3 Zamanda değişmezlik kontrolü

Eşitlik 2.5'te verilen sistemin zamanda değişmez olduğunu varsayarsak, bu sistemin girişine $x(t - t_0)$ sinyalini uyguladığımızda, sistemin çıkışında $y(t - t_0) = x(t - t_0) + cos(t - t_0 + \tau)$ çıkış sinyalini elde etmeyi bekleriz. Ancak, söz konusu sistemin girişine $x_1(t) = x(t - t_0)$ sinyalini uyguladığımızda, çıkışta $y_1(t) = x_1(t) + cos(t + \tau) = x(t - t_0) + cos(t + \tau)$ elde edildiği için ve $y_1(t) \neq y(t - t_0)$ eşitsizliği söz konusu olduğu için, örnekteki sistem zamanda değişmez bir sistem değildir.

2.3.4 Nedensellik kontrolü

Eşitlik 2.5'teki sistemin çıkışı, girişin sadece o anki değerlerine bağlı olduğu için sistem nedensel bir sistemdir. İlgili tanımlamalara geri dönersek*; tüm belleksiz sistemlerin nedensel sistem olduklarını* söyleyebiliriz. Soruda; kosinüs terimindeki gecikme kasıtlı olarak sabit bir sayı olarak verilmemiştir ($\tau = 2\ gibi$). Bunun nedeni birçok öğrencinin zamanda geciken sinyalin giriş sinyali olduğu yanılgısına düşüp, τ sabitinin negatif ve pozitif değerleri için sırasıyla sistemi nedensel (causal) ve nedensel olmayan (non-causal) olarak değerlendirmeleridir. Oysa zamanda geciken giriş sinyali değildir, bu nedenle τ sabitinin tüm değerleri için sistem nedenseldir.

2.3.5 Kararlılık kontrolü

Sınırlı bir girişe sınırlı bir çıkış veren sistemlere kararlı sistemler denir. Sınırlı bir $x(t)$ girişine genlik değerleri $-1 \leq cos(t + \tau) \leq 1$ arasında değişen kosinüs fonksiyonu eklendiğinde yine sınırlı bir çıkış elde edileceğinden adı geçen sistem kararlıdır.

Tekrar tekrar ifade etmekte büyük fayda görülmektedir ki: Birçoğu tek bir satır ile ispatlanan sistem özelliklerini kendiniz çözerek anlamadan lütfen bir sonraki bölüme geçmeyiniz. Önemli olan bu tek satırı kitabın yazarlarının değil, Sizin anlayarak yazabilmeniz ve böylece farklı sistem örnekleri için benzer ispatları kolayca ve doğruca yapabilmenizdir.