BLM2041 Signals and Systems

Syllabus

The Instructors:

Doç. Dr. Ali Can Karaca

ackaraca@yildiz.edu.tr

Dr. Ahmet Elbir

aelbir@yildiz.edu.tr

Systems Properties

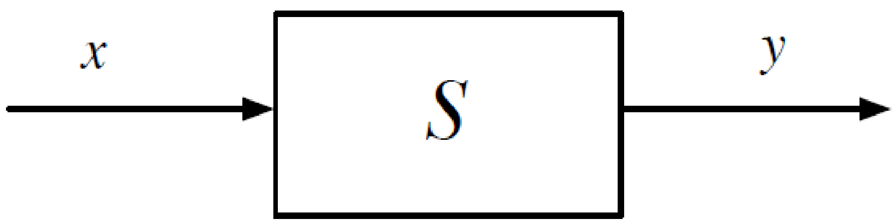
Systems

- A system transforms *input signals* into *output signals*.
- A system is a *function* mapping input signals into output signals.
- We will concentrate on systems with one input and one output *i.e.* *single-input, single-output* (SISO) systems.
- Notation:
  - $y = Sx$ or $y = S(x)$, meaning the system S acts on an input signal x to produce output signal y.
  - $y = Sx$ does not (in general) mean multiplication!

Systems Properties

Block diagrams

Systems often denoted by *block diagram*:



- Lines with arrows denote signals (*not* wires).
- Boxes denote systems; arrows show inputs & outputs.
- Special symbols for some systems.

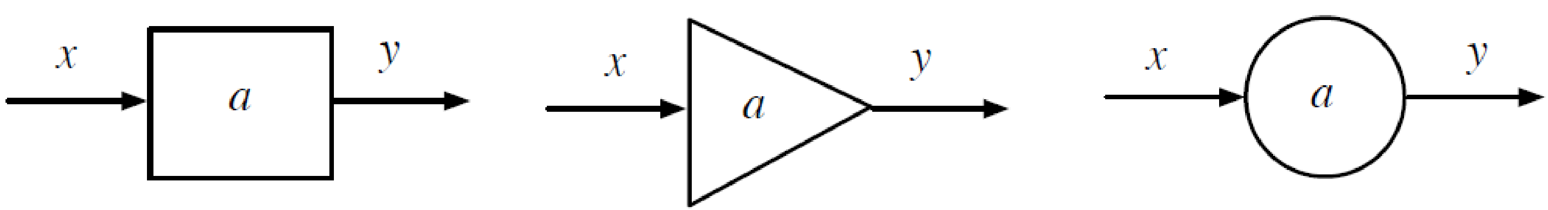
Systems Properties

Examples

(with input signal x and output signal y)

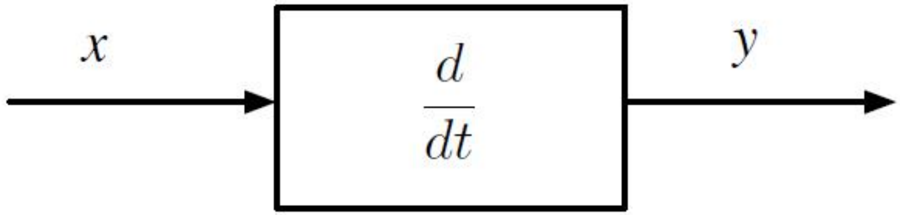
Scaling system: $y(t) = ax(t)$

- Called an *amplifier* if $|a| > 1$.
- Called an *attenuator* if $|a| < 1$.
- Called *inverting* if $a < 0$.
- a is called the *gain* or *scale factor*.
- Sometimes denoted by triangle or circle in block diagram:



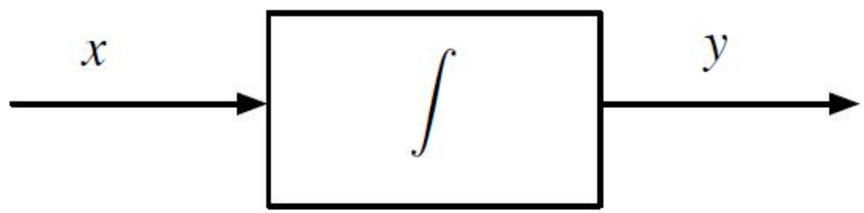
Systems Properties

Differentiator: $y(t) = x'(t)$



Integrator: $y(t) = \int_a^t x(\tau)\, d\tau$ (a is often 0 or $-\infty$)

Common notation for integrator:



Systems Properties

time shift system: $y(t) = x(t - T)$

- called a *delay system* if $T > 0$
- called a *predictor system* if $T < 0$

Systems Properties

convolution system:

$$y(t) = \int x(t-\tau)h(\tau)\, d\tau,$$

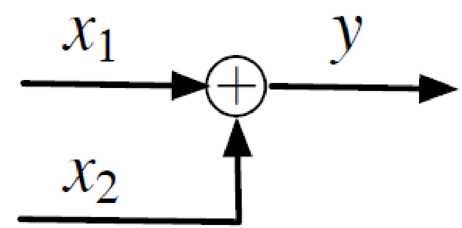
where h is a given function (you'll be hearing much more about this!)

Systems Properties

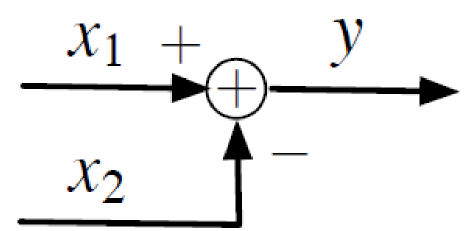
Examples with multiple inputs

Inputs $x_1(t)$, $x_2(t)$, and Output $y(t)$)

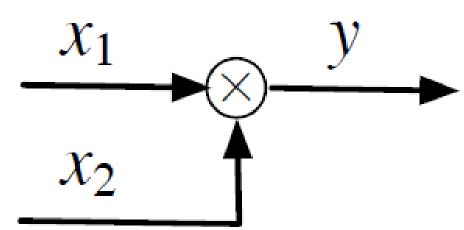
- **summing system:** $y(t) = x_1(t) + x_2(t)$



- **difference system:** $y(t) = x_1(t) - x_2(t)$



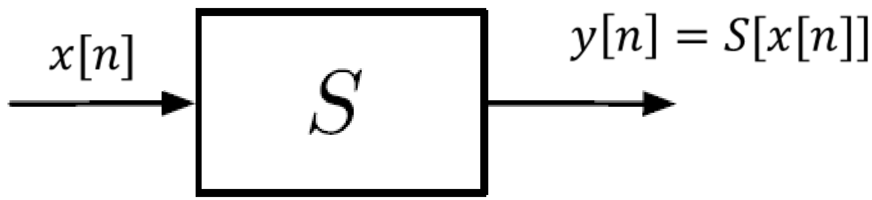
- **multiplier system:** $y(t) = x_1(t)x_2(t)$



Ayrık Zamanlı Sistemler

S giriş fonksiyonunu çıkış fonksiyonuna dönüştüren bir operatördür.

Ayrık zamanda sistem:



ÖRNEK: Faiz sonrası ay sonunda banka hesabındaki para miktarını ele alalım.

x[n] ay boyunca net para girişi (yatırılan-çekilen) ve y[n] ay sonunda hesaptaki para olmak üzere, y[n]'nin aşağıda verilen fark denklemiyle belirlendiğini varsayalım:

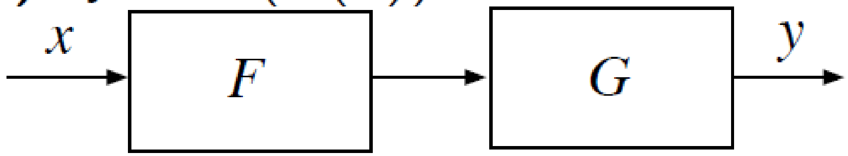
$$y[n] = 1.01y[n\text{-}1] + x[n]$$

Systems Properties

Interconnection of Systems

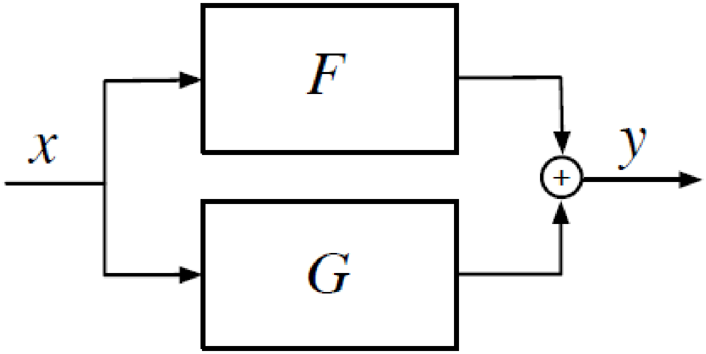
We can interconnect systems to form new systems,

- **cascade (or series):** $y = G(F(x)) = GFx$



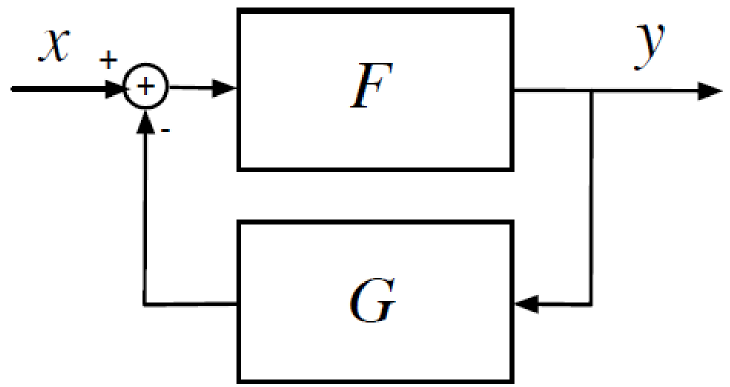
(note that block diagrams and algebra are *reversed*)

- **sum (or parallel):** $y = Fx + Gx$



Systems Properties

- **feedback:** $y = F(x - Gy)$

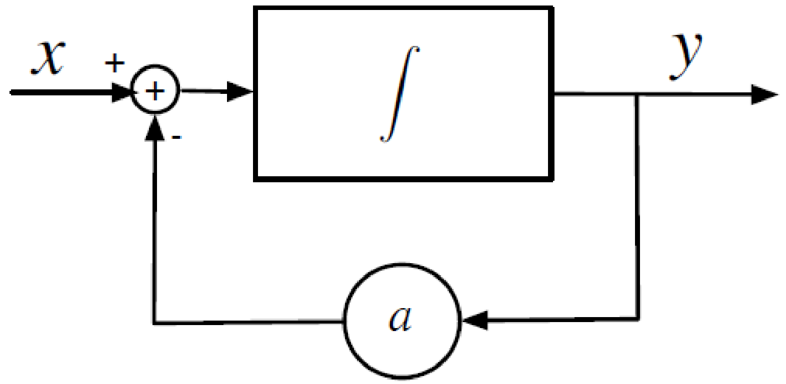


In general,

- Block diagrams are a symbolic way to describe a connection of systems.
- We can just as well write out the equations relating the signals.
- We can go back and forth between the system block diagram and the system equations.

Systems Properties

Example: Integrator with feedback



Input to integrator is $x - ay$, so

$$\int^{t} (x(\tau) - ay(\tau))\, d\tau = y(t)$$

Another useful method: the *input* to an integrator is the derivative of its output, so we have

$$x - ay = y'$$

Systems Properties

Linearity

A system F is **linear** if the following two properties hold:

1. **homogeneity:** if x is any signal and a is any scalar,

$$F(ax) = aF(x)$$

2. **superposition:** if x and $\tilde{x}$ are any two signals,

$$F(x + \tilde{x}) = F(x) + F(\tilde{x})$$

In words, linearity means:

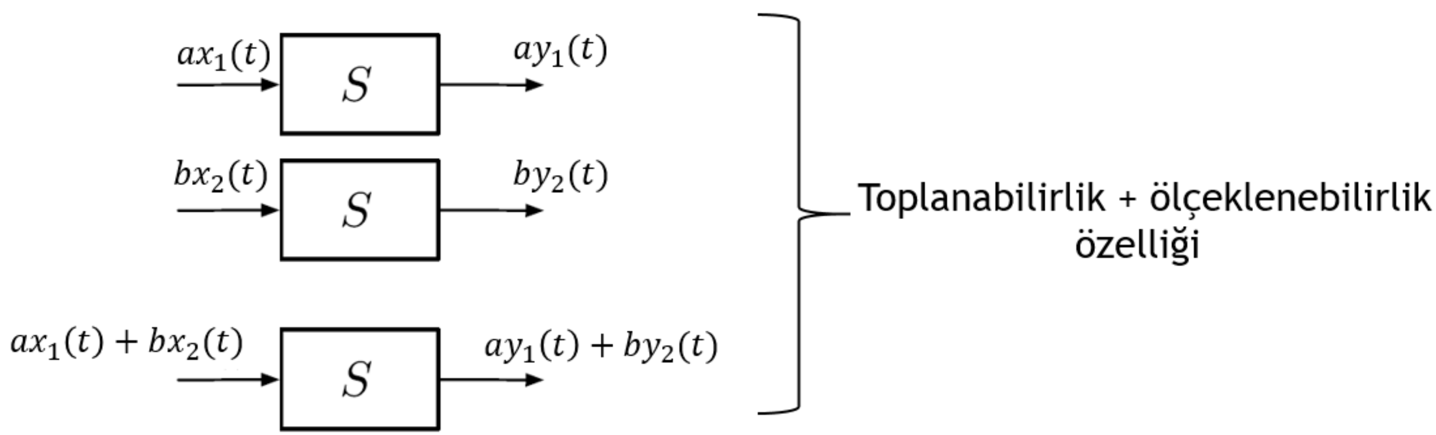
- Scaling before or after the system is the same.
- Summing before or after the system is the same.

Doğrusal Sistemler

İki veya daha fazla işaretin toplamından oluşan bir girişe olan yanıtı, giriş işaretini oluşturan bileşenlere yanıtlarının toplamına eşit olan sistemlere *doğrusal sistemler* denir.

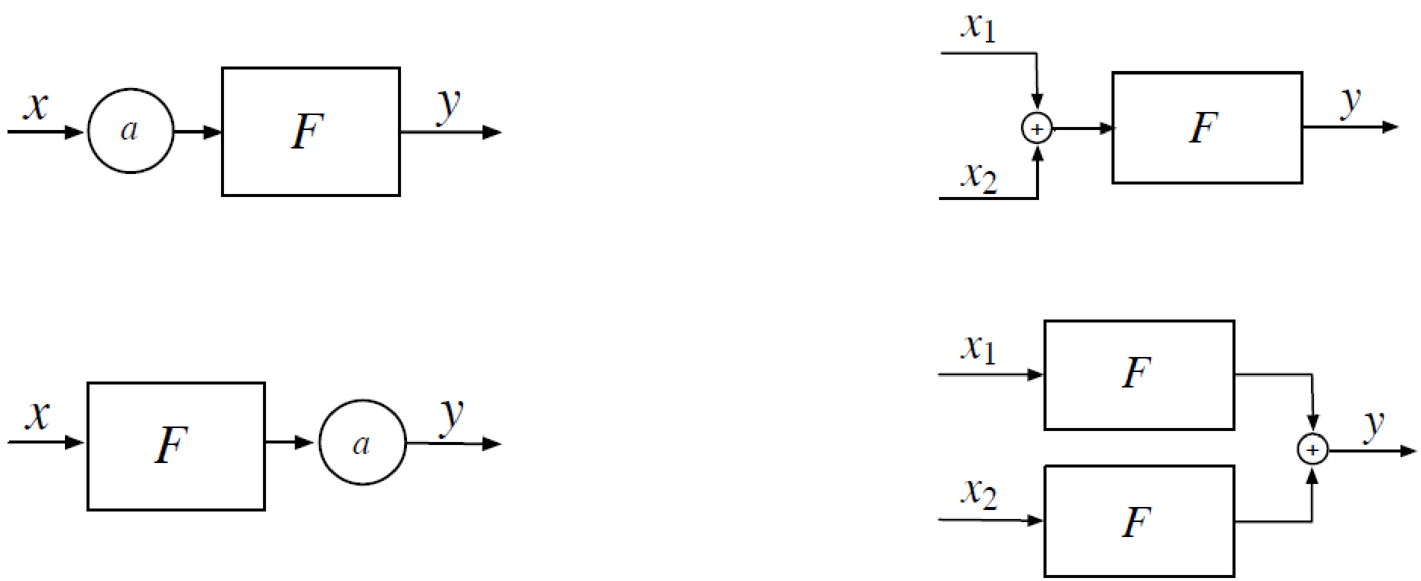
Doğrusal sistemler aşağıdaki özelliklerinin ikisini de sağlamalıdır!!

1) Ölçeklenebilirlik
2) Toplanabilirlik



Systems Properties

Linearity means the following pairs of block diagrams are equivalent, *i.e.*, have the same output for any input(s)

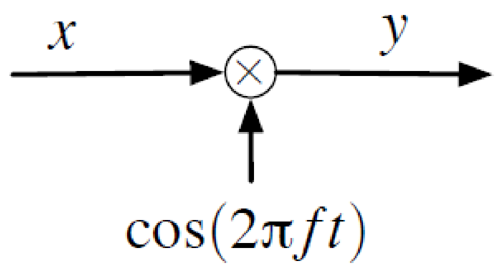


Examples of linear systems: scaling system, differentiator, integrator, running average, time shift, convolution, modulator, sampler.

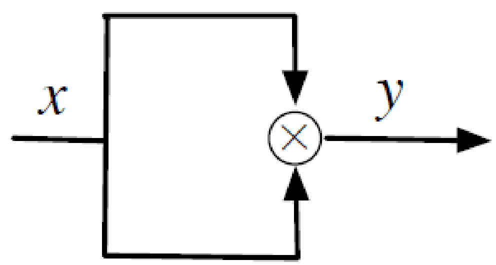
Examples of nonlinear systems: sign detector, multiplier (sometimes), comparator, quantizer, adaptive filter

Systems Properties

- Multiplier as a modulator, $y(t) = x(t)\cos(2\pi ft)$, is *linear*.



- Multiplier as a squaring system, $y(t) = x^2(t)$ is *nonlinear*.



Örnekler

$$x(t) = a\,x_1(t) + b\,x_2(t) \Rightarrow y(t) = a\,y_1(t) + b\,y_2(t)$$

Örnek: $y(t) = tx(t)$ sistemini doğrusallık açısından inceleyiniz.

1) Girişe ölçeklenmiş ve toplanmış bir işaret verilir.

$$x_3(t) = ax_1(t) + bx_2(t)$$

$$x_1(t) \rightarrow y_1(t) = tx_1(t)$$
$$x_2(t) \rightarrow y_2(t) = tx_2(t)$$

2) Elde edilen çıkış hesaplanır.

$$\begin{aligned} y_3(t) &= tx_3(t) \\ &= t(ax_1(t) + bx_2(t)) \\ &= atx_1(t) + btx_2(t) \\ &= ay_1(t) + by_2(t) \end{aligned}$$

Çıkış girişteki ölçek ve toplam kombinasyonu ile aynı olduğundan sistem doğrusaldır.

Örnekler

Örnek: $y(t) = x^2(\mathrm{t})$ sistemini doğrusallık açısından inceleyiniz.

1) Giriş tanımlandı.

$$x_3(t) = ax_1(t) + bx_2(t) \qquad \begin{aligned} x_1(t) &\rightarrow y_1(t) = x_1^{\,2}(t) \\ x_2(t) &\rightarrow y_2(t) = x_2^{\,2}(t) \end{aligned}$$

2) Giriş fonksiyonda yerine koyuldu.

$$\begin{aligned} y_3(t) &= x_3^{\,2}(t) \\ &= (ax_1(t) + bx_2(t))^2 \\ &= a^2x_1^{\,2}(t) + b^2x_2^{\,2}(t) + 2abx_1(t)x_2(t) \\ &= a^2y_1(t) + b^2y_2(t) + 2abx_1(t)x_2(t) \end{aligned}$$

$y_3(t) = ay_1(t) + by_2(t) = ax_1^2(t) + bx_2^2(t)$ olması gerekirken sonuçlar uyuşmuyor.

Doğrusal bir sistem değildir.

Systems Properties

System Memory

- A system is *memoryless* if the output depends only on the present input.
  - Ideal amplifier
  - Ideal gear, transmission, or lever in a mechanical system
- A *system with memory* has an output signal that depends on inputs in the past or future.
  - Energy storage circuit elements such as capacitors and inductors
  - Springs or moving masses in mechanical systems
- A *causal* system has an output that depends only on past or present inputs.
  - Any real physical circuit, or mechanical system.

Hafızalı Sistemler

Eğer sistemin herhangi bir *t* anındaki çıkış değeri, girişin önceki veya sonraki değerlerine bağlıysa bu sistemlere *hafızalı sistemler* denir.

Örn:

$$y(t) = 2x(t) + 5x(t-3) - x(t+2)$$

Örn:

$$y(t) = 6x(t)$$

Hafızalı sistemlerde, girişi çıkışın hesaplandığı an dışındaki zamanlarda saklayan mekanizmalar olmalıdır. Çoğu fiziksel sistemde, hafıza enerjinin depolanması ile doğrudan ilişkilidir. Örneğin, kondansatör elektriksel yük biriktirerek enerji saklar.

Nedensel Sistemler

Herhangi bir *t* anındaki çıkışı, girişin geçmişteki veya o andaki değerlerine bağlı olan sistemlere *nedensel sistemler* denir.

Örn:

$$y(t) = 2x(t) + 5x(t-3) - x(t+2)$$

Örn:

$$y[n] = 3x[n] - x[n-7]$$

$$y(t) = 6x(t)$$

$$y(t) = 2x(t-4)\cos(t+5)$$

Systems Properties

Time-Invariance

- A system is time-invariant if a time shift in the input produces the same time shift in the output.
- For a system F,

$$y(t) = Fx(t)$$

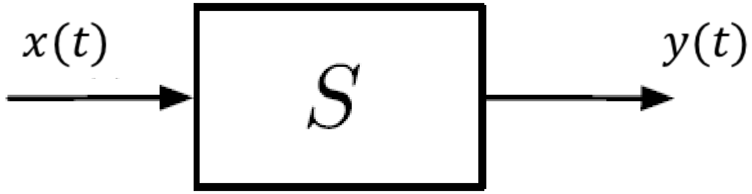
implies that

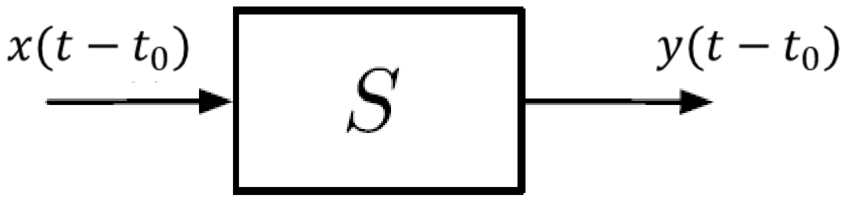
$$y(t-\tau) = Fx(t-\tau)$$

for any time shift τ.

Zamanla Değişmeyen Sistemler

Bir sistemde, giriş işaretine uygulanan bir öteleme çıkış işaretinde de aynı miktarda ötelemeye neden oluyorsa sisteme *zamanla değişmeyen* sistemler denir.





Örnek

$y[n] = nx[n]$ işaretini zamanla değişmeyen özelliği açısından inceleyiniz.

1) Girişe x_1 işareti uygulandığında y_1 çıkışı şu şekilde olur. $x_1[n] \rightarrow y_1[n] = nx_1[n]$

2) x_1 girişi bir miktar geciktirildiğinde oluşan x_2 işareti sisteme uygulanırsa:

$$x_2[n] = x_1[n - n_0] \longrightarrow y_2[n] = nx_1[n - n_0]$$

3) Çıkışı aynı miktar geciktirdiğimizde sonuçların uyuşması gerekir.

$$y_1[n - n_0] = (n - n_0)x_1[n - n_0] \neq y_2[n]$$

Bundan dolayı, zamanla değişen bir sistemdir.

Systems Properties

System Stability

- Stability important for most engineering applications.
- Many definitions
- If a bounded input

$$|x(t)| \leq M_x < \infty$$

always results in a bounded output

$$|y(t)| \leq M_y < \infty,$$

where M_x and M_y are finite positive numbers, the system is *Bounded Input Bounded Output (BIBO) stable.*

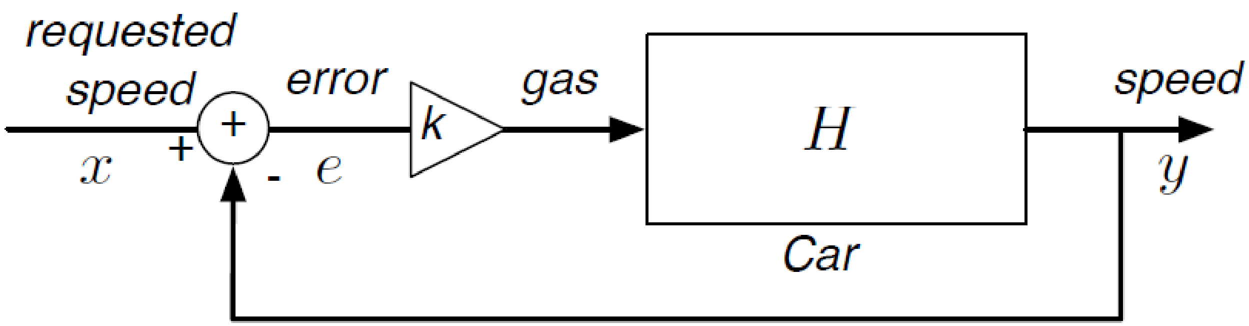
Örnek

$$y(t) = e^{x(t)}$$

$$y(t) = te^{x(t)}$$

Systems Properties

Example: Cruise control, from introduction,



The output y is

$$y = H(k(x - y))$$

We'll see later that this system can become unstable if k is too large (depending on H)

- Positive error adds gas
- Delay car velocity change, speed overshoots
- Negative error cuts gas off
- Delay in velocity change, speed undershoots
- Repeat!

Systems Properties

System Invertibility

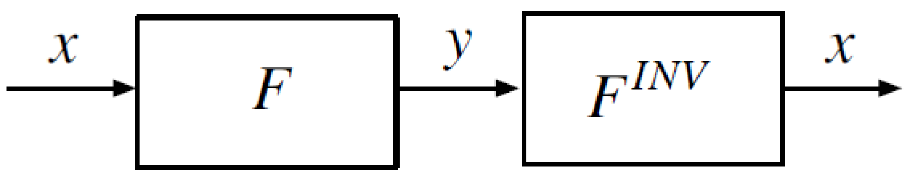
- A system is invertible if the input signal can be recovered from the output signal.
- If F is an invertible system, and

$$y = Fx$$

then there is an inverse system F^{INV} such that

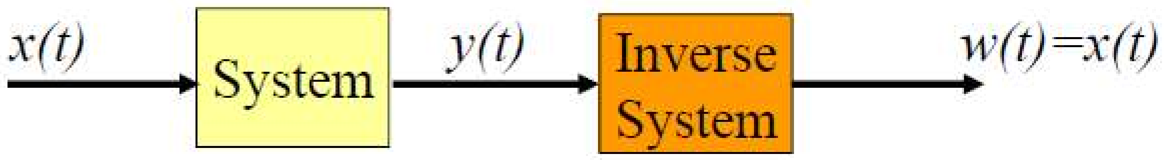
$$x = F^{INV} y = F^{INV} F x$$

so $F^{INV} F = I$, the identity operator.



Örnek

Eğer sistemin çıktısı başka bir sistem ile giriş değerlerine dönüştürülebiliyorsa o sistem tersinebilirdir.



Sistem tanımı $y(t) = 4x(t)$ olsun.

$w(t) = \frac{1}{4} y(t)$ bu sistemin ters sistemidir.

Her sistemin bir ters sistemi yoktur.

System	Linear	Time invariant	Memoryless	Causal	BIBO stable
Constant offset $y[n] = x[n] + C, C \neq 0$					
Time shift $y[n] = x[n - n_d]$					
Squaring $y[n] = x^2[n]$					
Accumulator $y[n] = \sum_{k=-\infty}^{n} x[k]$					
Compressor $y[n] = x[Mn], M > 1$					
Differentiator $y[n] = x[n] - x[n-1]$					
A difference equation $y[n] = x[n] + y[n-1]$					

System	Linear	Time invariant	Memoryless	Causal	BIBO stable
Constant offset $y[n] = x[n] + C, C \neq 0$	N	Y	Y	Y	Y
Time shift $y[n] = x[n - n_d]$	Y	Y	N, if $n_d \neq 0$	Y, if $n_d > 0$	Y
Squaring $y[n] = x^2[n]$	N	Y	Y	Y	Y
Accumulator $y[n] = \sum_{k=-\infty}^{n} x[k]$	Y	Y	N	Y	N
Compressor $y[n] = x[Mn], M > 1$	Y	N	N	N	Y
Differentiator $y[n] = x[n] - x[n-1]$	Y	Y	N	Y	Y
A difference equation $y[n] = x[n] + y[n-1]$	Y	Y	N	Y	N

Systems Properties

Systems Described by Differential Equations

Many systems are described by a *linear constant coefficient ordinary differential equation* (LCCODE):

$$a_n y^{(n)}(t) + \cdots + a_1 y'(t) + a_0 y(t) = b_m x^{(m)}(t) + \cdots + b_1 x'(t) + b_0 x(t)$$

with given *initial conditions*

$$y^{(n-1)}(0), \quad \ldots \quad , y'(0), \quad y(0)$$

(which fixes $y(t)$, given $x(t)$)

- n is called the *order* of the system
- $b_0, \ldots, b_m, a_0, \ldots, a_n$ are the *coefficients* of the system

Systems Properties

This is important because LCCODE systems are **linear** when initial conditions are all zero.

- Many systems can be described this way
- If we can describe a system this way, we know it is linear

Note that an LCCODE gives an *implicit* description of a system.

- It describes how $x(t)$, $y(t)$, and their derivatives interrelate
- It doesn't give you an explicit solution for $y(t)$ in terms of $x(t)$

Soon we'll be able to *explicitly* express $y(t)$ in terms of $x(t)$

Systems Properties

Examples

Simple examples

- scaling system ($a_0 = 1,\ b_0 = a$)

$$y = ax$$

- integrator ($a_1 = 1,\ b_0 = 1$)

$$y' = x$$

- differentiator ($a_0 = 1,\ b_1 = 1$)

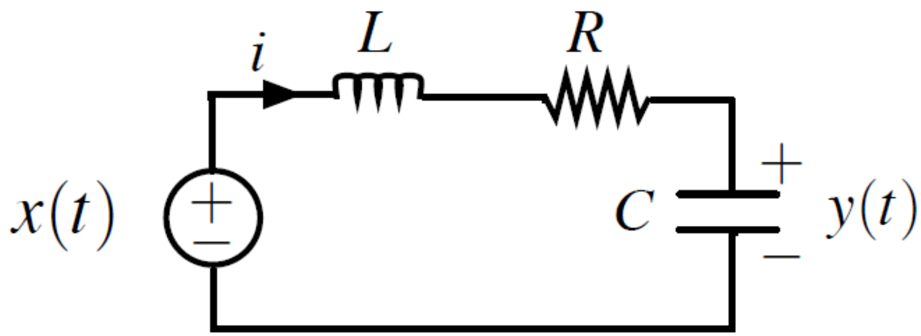
$$y = x'$$

- integrator with feedback (a few slides back, $a_1 = 1, a_0 = a, b_0 = 1$)

$$y' + ay = x$$

Systems Properties

2nd Order Circuit Example



By Kirchoff's voltage law

$$x - Li' - Ri - y = 0$$

Using $i = Cy'$,

$$x - LCy'' - RCy' - y = 0$$

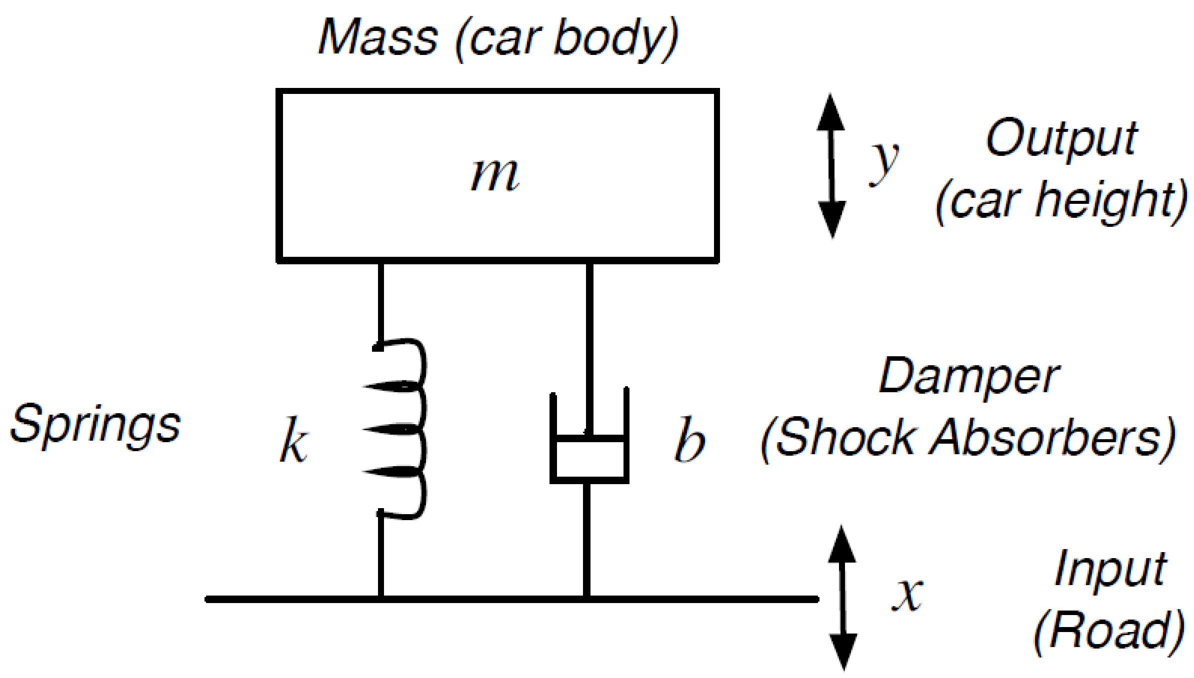
or

$$LCy'' + RCy' + y = x$$

which is an LCCODE. This is a linear system.

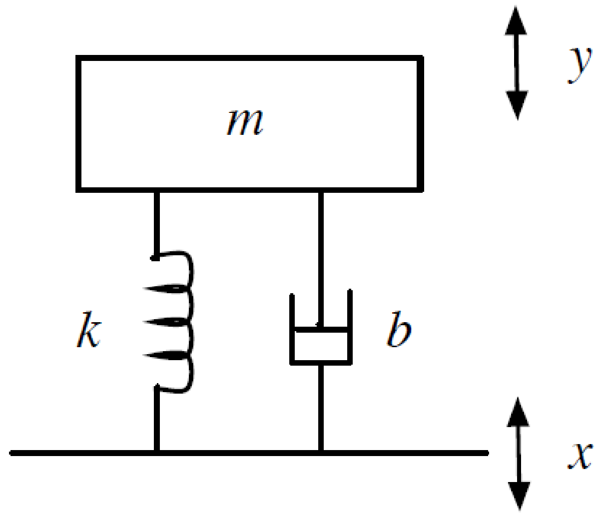
Systems Properties

Mechanical System



This can represent suspension system, or building during earthquake, . . .

Systems Properties



- $x(t)$ is displacement of base; $y(t)$ is displacement of mass
- spring force is $k(x - y)$; damping force is $b(x - y)'$
- Newton's equation is $my'' = b(x - y)' + k(x - y)$

Rewrite as second-order LCCODE

$$my'' + by' + ky = bx' + kx$$

Systems Properties

Discrete-Time Systems

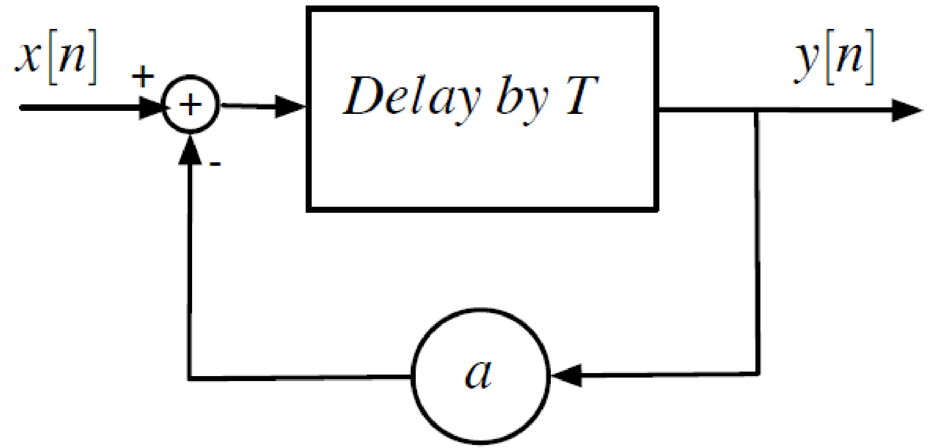
- Many of the same block diagram elements
- Scaling and delay blocks common
- The system equations are *difference equations*

$$a_0 y[n] + a_1 y[n-1] + \ldots = b_0 x[n] + b_1 x[n-1] + \ldots$$

where $x[n]$ is the input, and $y[n]$ is the output.

Systems Properties

Discrete-Time System Example



- The input into the delay is

$$e[n] = x[n] - ay[n]$$

- The output is $y[n] = e[n-1]$, so

$$y[n] = x[n-1] - ay[n-1].$$