Control Slides with notes

Dr Mahmoud Barghash
(1st Semester 2023/2024)
Notes are written by Nada Ababneh

Laplace Transform

في ال Control سوف نتعلم التحكم بكمية الكهرباء الداخلة الى ال Heater مثلا

#The Start

Using Laplace Approach

Error = is the difference between the True value − measured value

الهدف التحويل من

Deffrential equations → Algebric equations

(معادلات مرتبطة بالزمن)

Consider the previous example again:

$$\ddot{x} + 3\dot{x} + 2x = 5\sin t$$

$$\underset{s^2}{\ddot{x}} \quad \underset{s}{\dot{x}} \quad \underset{1}{x}$$

$$x(0) = 1$$

$$\dot{x}(0) = 0$$

Apply the Laplace transform to given diff. eqn

$$[s^2X(s) - sx(0) - x'(0)] + 3[sX(s) - x(0)] + 2X(s) = \frac{5}{s^2+1}$$

Simplify it:

عملها موضوع القانون ←

$$\boxed{X(s)} = \underbrace{\frac{s+3}{(s^2+3s+2)}}_{X_1(s)} + \underbrace{\frac{5}{((s^2+1)(s^2+3s+2))}}_{X_2(s)}$$

$$\frac{K_1}{(S+2)} + \frac{K_2}{(S+1)}$$

← يهمني المقام اكثر من البسط

★ في حالة ال (Steady State) تكون $\boxed{S=0}$

Using Laplace Approach (cont.)

Partial fraction expansion:

$$X_1(s) = \frac{A}{s+1} + \frac{B}{s+2} = \frac{s+3}{(s+1)(s+2)}$$

and

$$X_2(s) = \frac{C}{s+1} + \frac{D}{s+2} + \frac{Es+F}{\underline{s^2+1}} = \frac{5}{(s+1)(s+2)(s^2+1)}$$

غير قابل للتحليل
(مجموع مربعين)

Determine the values for **A**,**B**,**C**,**D**,**E** & **F**

Then, $X(s) = X_1(s) + X_2(s)$

Using Laplace Approach (cont.)

Finally, x(t) can be found by applying the inverse Laplace transform of X(s)

$$\boxed{x(t) = L^{-1}\left[X(s)\right]}$$

Laplace Transforms

$$\begin{tikzcd} f(t) \arrow[r, bend left, "L"] & F(s) \arrow[l, bend left, "L^{-1}"] \end{tikzcd}$$

- Def: $$F(s) = L(f) = \int_0^\infty e^{-st} f(t)dt \quad \text{for} \quad f(t), t > 0$$
- Inverse: $$f(t) = L^{-1}(F)$$
- Linearity:

الثابت بيقدر يطلع لبرا وال Laplace بتتوزع

$$L\{af(t) + bg(t)\} = aL\{f(t)\} + bL\{g(t)\}$$

- Shifting Theorom:

→ Laplace → (s → s-a)

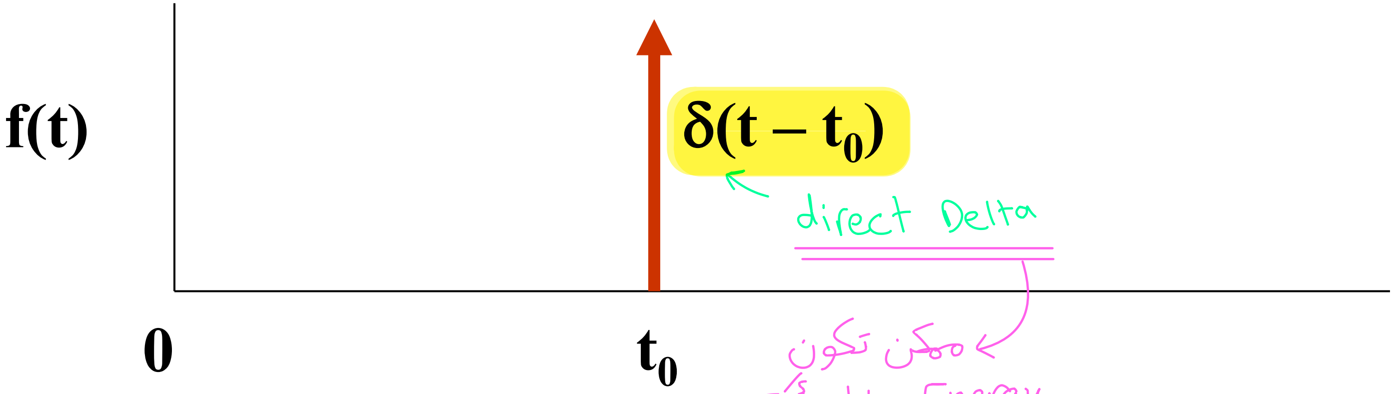
$$L\{e^{at} f(t)\} = F(s-a)$$

$$e^{at} f(t) = L^{-1}\{F(s-a)\}$$

The Laplace Transform

The Laplace transform of a unit impulse:

Pictorially, the unit impulse appears as follows:



direct Delta

ممكن تكون Energy مفاجأة مثلاً الارتطام

Mathematically:

$$\delta(t-t_0) = 0 \quad t \neq 0$$

$$\int_{t_0-\varepsilon}^{t_0+\varepsilon} \delta(t-t_0)\,dt = 1 \qquad \varepsilon > 0$$

فترة زمنية تقترب للصفر

*note لكن المساحة تحت المنحنى تساوي (١)

The Laplace Transform

Transform Pairs:

f(t)	F(s)
$\delta(t)$	1
$u(t)$ الثابت	$\frac{1}{s}$ الثابت
e^{-at}	$\frac{1}{s+a}$
اسمه الاخر Ramp ← t	$\frac{1}{s^2}$
t^n	$\frac{n!}{s^{n+1}}$

The Laplace Transform

Transform Pairs:

f(t)	F(s)
$\boxed{t}e^{-at}$	$\dfrac{1}{(s+a)^{\boxed{2}}}$
$\boxed{t^n}e^{-at}$	$\dfrac{n!}{(s+a)^{\boxed{n+1}}}$
يتذبذب ← $\sin(wt)$ sinusoidal	$\dfrac{w}{s^2+w^2}$
$\cos(wt)$	$\dfrac{s}{s^2+w^2}$

The Laplace Transform

Transform Pairs:

f(t)	F(s)
$e^{-at}\sin(wt)$	$\dfrac{w}{(s+a)^2+w^2}$
$e^{-at}\cos(wt)$	$\dfrac{s+a}{(s+a)^2+w^2}$
$\sin(wt+\theta)$	$\dfrac{s\sin\theta+w\cos\theta}{s^2+w^2}$
$\cos(wt+\theta)$	$\dfrac{s\cos\theta-w\sin\theta}{s^2+w^2}$

The Laplace Transform

Common Transform Properties:

f(t)	F(s)
$f(t-t_0)u(t-t_0), t_0 \geq 0$	$e^{-t_0 s}F(s)$
$f(t)u(t-t_0), t \geq 0$	$e^{-t_0 s}L[f(t+t_0)$
$e^{-at}f(t)$	$F(s+a)$
$\frac{d^n f(t)}{dt^n}$	$s^n F(s) - s^{n-1}f(0) - s^{n-2}f'(0) - \ldots - s^0 f^{n-1}f(0)$
$tf(t)$	$-\frac{dF(s)}{ds}$
$\int_0^t f(\lambda)d\lambda$	$\frac{1}{s}F(s)$

لحد ما
نوصل s^0 بنوقف
(مافي s قوة سالب)

The Laplace Transform

Using Matlab with Laplace transform:

Example Use Matlab to find the transform of te^{-4t}

The following is written in italic to indicate Matlab code

```
syms t s
laplace(t*exp(-4*t),t,s)
ans =
        1/(s+4)^2
```

$$\frac{1}{(s+4)^2}$$

The Laplace Transform

Using Matlab with Laplace transform:

Example

Use Matlab to find the inverse transform of

$$F(s)=\frac{s(s+6)}{(s+3)(\underbrace{s^2+6s+18}_{\text{complex}})} \qquad prob.12.19$$

```
syms s t

ilaplace(s*(s+6)/((s+3)*(s^2+6*s+18)))

ans =
      -exp(-3*t)+2*exp(-3*t)*cos(3*t)
```

for the inverse

The Laplace Transform

Theorem: **Initial Value Theorem:**

If the function f(t) and its first derivative are Laplace transformable and f(t) Has the Laplace transform F(s), and the $\lim_{s \to \infty} sF(s)$ exists, then

$$\lim_{s \to \infty} sF(s) = \lim_{t \to 0} f(t) = f(0)$$

Initial Value Theorem

The utility of this theorem lies in not having to take the inverse of F(s) in order to find out the initial condition in the time domain. This is particularly useful in circuits and systems.

The Laplace Transform

Example: **Initial Value Theorem:**

$$\lim_{S\to\infty} S\times \bigcirc$$

Given;

$$F(s)=\frac{(s+2)}{(s+1)^2+5^2}$$

Find f(0)

$$f(0)=\boxed{\lim_{s\to\infty} sF(s)}=\lim_{s\to\infty} s\frac{(s+2)}{(s+1)^2+5^2}=\lim_{s\to\infty}\left[\frac{s^2+2s}{s^2+2s+1+25}\right]$$

$$=\lim_{s\to\infty}\frac{s^2/s^2+2s/s^2}{s^2/s^2+2s/s^2+(26/s^2)}=1$$

قسمنا كل الاجزاء على أعلى أس

بهدف التبسيط وعشان نحل مشكلة ال ∞ في ال Limit

The Laplace Transform

Theorem: **Final Value Theorem:**

If the function f(t) and its first derivative are Laplace transformable and f(t) has the Laplace transform F(s), and the $\lim_{s \to \infty} sF(s)$ exists, then

$$\lim_{s \to 0} sF(s) = \lim_{t \to \infty} f(t) = f(\infty)$$

∞ في الهندسة
وليس بالأعداد

Final Value Theorem

Again, the utility of this theorem lies in not having to take the inverse of F(s) in order to find out the final value of f(t) in the time domain. This is particularly useful in circuits and systems.

The Laplace Transform

Example: **Final Value Theorem:**

$$\lim_{s \to 0} s \times \bigcirc$$

Given:

$$F(s) = \frac{(s+2)^2 - 3^2}{\left[(s+2)^2 + 3^2\right]} \qquad note\ \ F^{-1}(s) = te^{-2t}\cos 3t$$

Find $f(\infty)$.

Final value theorem

$0 \times$ (اي اشي) $= 0$

$$f(\infty) = \lim_{s \to 0} sF(s) = \lim_{s \to 0} s \frac{(s+2)^2 - 3^2}{\left[(s+2)^2 + 3^2\right]} = 0$$

Solution of Partial Fraction Expansion

- The solution of each distinct (non-multiple) root, real or complex uses a two step process.
  - The first step in evaluating the constant is to multiply both sides of the equation by the factor in the denominator of the constant you wish to find.
  - The second step is to replace *s* on both sides of the equation by the root of the factor by which you multiplied in step 1

$$X(s) = \frac{8(s+3)(s+8)}{s(s+2)(s+4)} = \frac{K_1}{s} + \frac{K_2}{s+2} + \frac{K_3}{s+4}$$

جذور المقام حقيقية
و غير متكررة

$$K_1 = \left.\frac{8(s+3)(s+8)}{(s+2)(s+4)}\right|_{s=0} = \frac{8(0+3)(0+8)}{(0+2)(0+4)} = 24$$

$$K_2 = \left.\frac{8(s+3)(s+8)}{s(s+4)}\right|_{s=-2} = \frac{8(-2+3)(-2+8)}{-2(-2+4)} = -12$$

$$K_3 = \left.\frac{8(s+3)(s+8)}{s(s+2)}\right|_{s=-4} = \frac{8(-4+3)(-4+8)}{-4(-4+4)} = -4$$

The partial fraction expansion is:

$$X(s) = \frac{24}{s} - \frac{12}{s+2} - \frac{4}{s+4}$$

* نفس الجواب بيطلع لو طبقنا ال inverse Laplace Transform أو initial value theorem أو final value theorem

- The inverse Laplace transform is found from the functional table pairs to be:

$$x(t) = \underbrace{\overset{e^{0}}{\underline{24}} - 12e^{-2t}} \; \underbrace{- 4e^{-4t}}$$

سوف يختفي بشكل أسرع لأنه أس أكبر

Repeated Roots

- Any unrepeated roots are found as before.
- The constants of the repeated roots (s-a)^m are found by first breaking the quotient into a partial fraction expansion with descending powers from *m* to *0:*

$$\frac{B_m}{(s-a)^{\textcircled{m}}}+\cdots+\frac{B_2}{(s-a)^2}+\frac{B_1}{(s-a)}$$

→ عدد ال fractions

وحدة من الحلول :-

- The constants are found using one of the following:

$$B_i = \frac{1}{(m-i)!} \frac{d^{m-i}}{ds^{m-i}} \left[\frac{P(s)}{Q(s)/(s-a_1)^m} \right]_{s=a_1}$$

$$B_m = \frac{P(a)}{\left[Q(s)/(s-a)^m \right]_{s=a}}$$

$$Y(s)=\frac{8(s+1)}{(s+2)^2}=\frac{K_1}{s+2}+\frac{K_2}{\underline{(s+2)}^{2}}$$

I can find it directly

أعلى قوة

$$K_2=\frac{8(s+1)\underline{(s+2)}^2}{(s+2)^2}\Bigg|_{s=-2}=8(s+1)\Big|_{s=-2}=-8$$

*

to find K_2 (صاحبة أعلى أس)

$$K_2\Big|_{s=-2}=8(s+1)=8(-2+1)=\boxed{-8=K_2}$$

now to find K_1

نستخدم طريقة التجميع و المعامل = المعامل
والثابت = الثابت

$$8(s+1)=K_1(s+2)+\underset{-8}{K_2}$$

$$8s+8=K_1 s+(2K_1-8)$$

معامل (s) = معامل (s)

$$\boxed{K_1=8}$$

يحتاج معادلة و تطبيق مباشر عليها → K_1

أسهل ↘ أو طريقة الـ تبسيط ثم معامل s = معامل s الثابت = الثابت

$$B_i = \frac{1}{(2-1)!}\frac{d}{ds}\left[\frac{8(s+1)}{(s+2)^2/(s+2)^2}\right]_{s=-2} = 8$$

The partial fraction expansion yields:

$$Y(s) = \frac{8}{s+2} - \frac{8}{(s+2)^2}$$

inverse Laplace transform

$$\longrightarrow y(t) = 8e^{-2t} - 8te^{-2t}$$

The inverse Laplace transform derived from the functional table pairs yields:

$$y(t) = 8e^{-2t} - 8te^{-2t}$$

A Second Method for Repeated Roots

$$Y(s) = \frac{8(s+1)}{(s+2)^2} = \frac{K_1}{s+2} + \frac{K_2}{(s+2)^2}$$

$$8(s+1) = K_1 \underline{(s+2)} + K_2$$

لتوحيد المقام

Method ①

ساوِ (معامل s = معامل s)

$$8s + 8 = K_1 s + 2K_1 + K_2$$

Method ②

نشتق اليمين و اليسار

Equating like terms:

Method ③

نفرض s=0 أي رقم
s=□
مرتين عشان عندي مجهولين ونحل تعويض وحذف

$$8 = K_1 \quad and \quad 8 = 2K_1 + K_2$$

$$8 = K_1 \quad and \quad 8 = 2K_1 + K_2$$

$$8 = 2 \times 8 + K_2$$

$$8 - 16 = -8 = K_2$$

Thus

$$\mathcal{L}^{-1} \quad Y(s) = \frac{8}{s+2} - \frac{8}{(s+2)^{2}}$$

Laplace inverse

$$y(t) = 8e^{-2t} - 8\underline{t}e^{-2t}$$

Another Method for Repeated Roots

$$Y(s) = \frac{8(s+1)}{(s+2)^2} = \frac{K_1}{s+2} + \frac{K_2}{(s+2)^2}$$

As before, we can solve for K_2 in the usual manner.

$$K_2 = \left.\frac{8(s+1)(s+2)^2}{(s+2)^2}\right|_{s=-2} = 8(s+1)\big|_{s=-2} = -8$$

$$(s+2)^2\frac{8(s+1)}{(s+2)^2}=(s+2)^2\frac{K_1}{s+2}-(s+2)^2\frac{8}{(s+2)^2}$$

$$\frac{d[8(s+1)]}{ds}=\frac{d[(s+2)K_1-8]}{ds}$$

$$8=K_1$$

$$Y(s)=\frac{8(s+1)}{(s+2)^2}=\frac{8}{s+2}-\frac{8}{(s+2)^2}$$

$$y(t)=8e^{-2t}-8te^{-2t}$$

Unrepeated Complex Roots

~~Reapeted complex Root~~

- Unrepeated complex roots are solved similar to the process for unrepeated real roots. That is you multiply by one of the denominator terms in the partial fraction and solve for the appropriate constant.
- Once you have found one of the constants, the other constant is simply the complex conjugate.

يعني عكس الاشارة

Complex Unrepeated Roots

ممكن استخدام طريقة المميز

يوجد طريقة بدل ايجاد المميز

① نقسم البسط والمقام على معامل (s²)

② بناخذ نص معامل (s) وبربعه

③ بقارنه مع ال constant

إذا كان ← أقل من الconstant لا يوجد حل حقيقي (يوجد حل complex)

أكبر من ال constant يوجد حل حقيقي

على ال calculator أسهل !!

$$\frac{5.2}{s^2+2s+5}=\frac{5.2}{s^2+2s+1+4}$$

÷2 = 1 → +1 -1 → (لاكمال المربع)

* باخذ نص معامل (S) = 1

بربعه 1^2 = 1

و بجمعه و بطرحه عشان اعمل مربع كامل

$$\frac{5.2}{(s+1)^2+2^2} \quad w=2 \;\; a=1$$

* لازم المقام = المقام

$e^{-at}\sin(wt)$	$\dfrac{w}{(s+a)^2+w^2}$
$e^{-at}\cos(wt)$	$\dfrac{s+a}{(s+a)^2+w^2}$
$\sin(wt+\theta)$	$\dfrac{s\sin\theta+w\cos\theta}{s^2+w^2}$
$\cos(wt+\theta)$	$\dfrac{s\cos\theta-w\sin\theta}{s^2+w^2}$

$$\frac{5.2}{2}e^{-t}\sin(2t)$$

أسهل

* بنحط المعادلة على ال calculator

s^2+2s+5

اعطاني complex

إذاً بفكر بإكمال المربع !!

و بالأغلب حوصل لشكل ال ⇐ sin أو cos

* General case → Non repeated pole (s+1)
→ complex (s^2+2s+5)
→ repeated pole $(s+3)^2$

can be found easyly at first

$$\frac{s+2}{(s+1)(s^2+2s+5)(s+3)^2}=\frac{K1}{s+3}+\frac{K2}{(s+3)^2}+\frac{K3}{s+1}+\frac{K4s+K5}{s^2+2s+5}$$

$$F(t)=K1e^{-3t}+K2te^{-3t}+K3e^{-t}+K4e^{-t}\cos(2t)+\frac{K5-K4}{2}e^{-t}\sin(2t)$$

$$s+2=k1(s+1)(s+3)(s^2+2s+5)+k2(s+1)(s^2+2s+5)$$

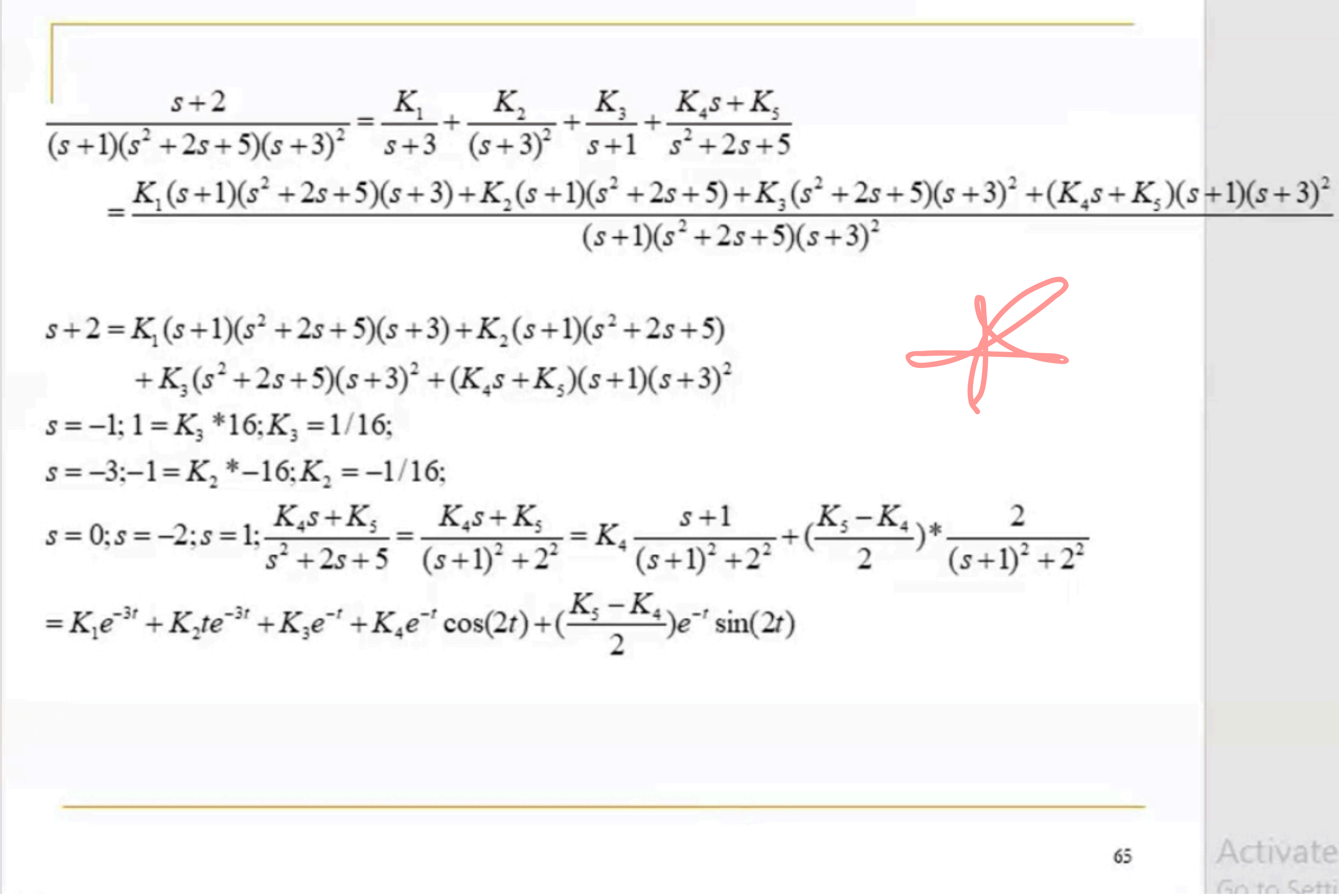
$$+k3(s+3)^2(s^2+2s+5)+(k4s+k5)(s+3)^2(s+1)$$

$$\frac{s+2}{(s+1)(s^2+2s+5)(s+3)2}=\frac{K_1}{s+1}+\frac{K_2s+K_3}{s^2+2s+}$$

$$K2=\frac{s+2}{(s+1)(s^2+2s+5)}\quad when\ s=-3$$

teams.microsoft.com is sharing your screen. Stop sharing Hide

محاضرة تاريخ (15/10)



$$\frac{s+2}{(s+1)(s^2+2s+5)(s+3)^2}=\frac{K_1}{s+3}+\frac{K_2}{(s+3)^2}+\frac{K_3}{s+1}+\frac{K_4s+K_5}{s^2+2s+5}$$

$$=\frac{K_1(s+1)(s^2+2s+5)(s+3)+K_2(s+1)(s^2+2s+5)+K_3(s^2+2s+5)(s+3)^2+(K_4s+K_5)(s+1)(s+3)^2}{(s+1)(s^2+2s+5)(s+3)^2}$$

$$s+2=K_1(s+1)(s^2+2s+5)(s+3)+K_2(s+1)(s^2+2s+5)$$
$$+K_3(s^2+2s+5)(s+3)^2+(K_4s+K_5)(s+1)(s+3)^2$$

$$s=-1;\ 1=K_3*16;K_3=1/16;$$

$$s=-3;-1=K_2*-16;K_2=-1/16;$$

$$s=0;s=-2;s=1;\frac{K_4s+K_5}{s^2+2s+5}=\frac{K_4s+K_5}{(s+1)^2+2^2}=K_4\frac{s+1}{(s+1)^2+2^2}+(\frac{K_5-K_4}{2})*\frac{2}{(s+1)^2+2^2}$$

$$=K_1e^{-3t}+K_2te^{-3t}+K_3e^{-t}+K_4e^{-t}\cos(2t)+(\frac{K_5-K_4}{2})e^{-t}\sin(2t)$$

65

Activate
Go to Setti

Electrical Systems
And
Mechanical Systems

Chapter 2: Mathematical Models of Systems

Objectives

We use quantitative mathematical models of physical systems to design and analyze control systems. The dynamic behavior is generally described by ordinary differential equations. We will consider a wide range of systems, including mechanical, hydraulic, and electrical. Since most physical systems are nonlinear, we will discuss linearization approximations, which allow us to use Laplace transform methods.

We will then proceed to obtain the input–output relationship for components and subsystems in the form of transfer functions. The transfer function blocks can be organized into block diagrams or signal-flow graphs to graphically depict the interconnections. Block diagrams (and signal-flow graphs) are very convenient and natural tools for designing and analyzing complicated control systems

© 2001 by Prentice Hall, Upper Saddle River, NJ.

$V = IR$

Basic Elements of Electrical Systems

two or more power supplies
active system
passive system
one power supply

Symbol →

V I R

- The time domain expression relating voltage and current for the resistor is given by Ohm's law i-e

$$\boxed{v_R(t) = i_R(t)R}$$

- The Laplace transform of the above equation is

$f(t) \longrightarrow F(s)$

$$V_R(s) = I_R(s)R$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

in the laplace
the resistor name is
Impedence

Basic Elements of Electrical Systems

$\frac{1}{C} \rightarrow$ صغير capacitor
بترفع ال Voltage كتير
(تعبر عن السعة التخزينية)

- The time domain expression relating voltage and current for the Capacitor is given as:

V I C

$\frac{V}{I}=R$

كل ما زادت الشحنات على ال
capacitor زاد ال Voltage

ال capacitor يلي مساحة سطحه أكبر
يحتاج كمية أكبر من الشحنات
عشان يبين نفس ال voltage

إذا بصير معناه تكامل
وسوف نحوله إلى
Laplace Transform

$$v_c(t) = \frac{1}{C}\int i_c(t)dt$$

- The Laplace transform of the above equation (assuming there is no charge stored in the capacitor) is

$$V_c(s) = \frac{1}{Cs} I_c(s)$$

Illustrations

© 2001 by Prentice Hall, Upper Saddle River, NJ.

V=IR
ohms Law

Basic Elements of Electrical Systems

ال Voltage لا علاقة
بمعدل التغير لل current

- The time domain expression relating voltage and current for the inductor is given as:

voltage
نتيجة مقاومة
الاسلاك الداخلية

$$v_L(t) = L\frac{di_L(t)}{dt}$$

- The Laplace transform of the above equation (assuming there is no energy stored in inductor) is

$$\boxed{V_L(s) = LsI_L(s)}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

V-I and I-V relations

Component	Symbol	V-I Relation	I-V Relation
Resistor		$v_R(t) = i_R(t)R$	$i_R(t) = \frac{v_R(t)}{R}$
Capacitor		$v_c(t) = \frac{1}{C}\int i_c(t)dt$	$i_c(t) = C\frac{dv_c(t)}{dt}$
Inductor		$v_L(t) = L\frac{di_L(t)}{dt}$	$i_L(t) = \frac{1}{L}\int v_L(t)dt$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

* integrated circuts
يكون في Power supply مخفي
it supplies the output

Assumption → initial conditions = zero

Example#1

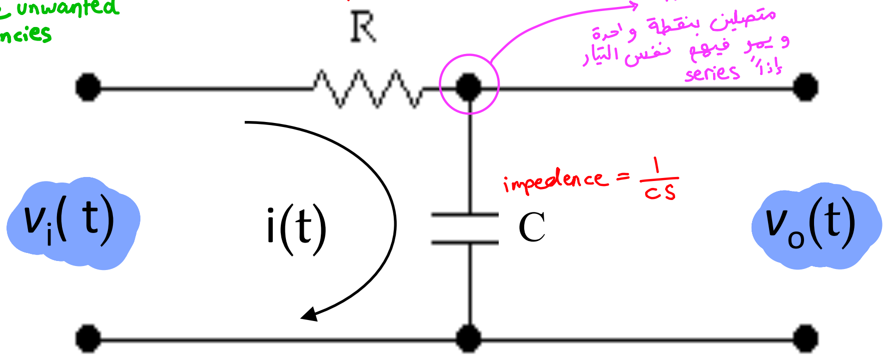
شبكة ذات منفذين

- The two-port network shown in the following figure has $v_i(t)$ as the input voltage and $v_o(t)$ as the output voltage. Find the transfer function $V_o(s)/V_i(s)$ of the network.

the circuit name is R-C Filter → to get rid of some unwanted frequancies

$\frac{1}{RCs+1} = \left(\frac{1}{1}\right)$ ← دائماً
ناقص *
s=0
initial = final

*(مبدأ علم الاتصالات)
at high frequency everything becomes a capacitor



impedence = R

R and c متصلين بنقطة واحدة ويمر فيهم نفس التيار إذاً series

impedence = $\frac{1}{cs}$

total impedence = $R + \frac{1}{cs}$

$$\frac{I = \frac{V_i}{R + \frac{1}{cs}}}{I = \frac{V_o}{\frac{1}{cs}}} \div$$

قسمة المعادلتين

$$\boxed{v_i(t) = i(t)R + \frac{1}{C}\int i(t)dt}$$

$$\boxed{v_o(t) = \frac{1}{C}\int i(t)dt}$$

المعادلات من الجدول

Transfer Function → $= \boxed{\frac{V_o}{V_i} = \frac{1}{RCs+1}}$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#1

$$v_i(t) = i(t)R + \frac{1}{C}\int i(t)dt \qquad v_o(t) = \frac{1}{C}\int i(t)dt$$

- Taking Laplace transform of both equations, considering initial conditions to zero.

$$V_i(s) = I(s)R + \frac{1}{Cs}I(s) \qquad V_o(s) = \frac{1}{Cs}I(s)$$

- Re-arrange both equations as:

$$V_i(s) = I(s)\left(R + \frac{1}{Cs}\right) \xrightarrow{} \text{total impedence} \qquad V_o(s) = \frac{I(s)}{Cs}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#1

$$V_i(s) = I(s)(R + \frac{1}{Cs}) \qquad CsV_o(s) = I(s)$$

- Substitute *I(s)* in equation on left

$$V_i(s) = CsV_o(s)(R + \frac{1}{Cs})$$

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{Cs(R + \frac{1}{Cs})}$$

Transfer Function =

$$\boxed{\frac{V_o(s)}{V_i(s)} = \frac{1}{1 + RCs}}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#1

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{1+RCs}$$

- The system has one pole at

$$1+RCs = 0 \qquad \Rightarrow \boxed{s = -\frac{1}{RC}}$$

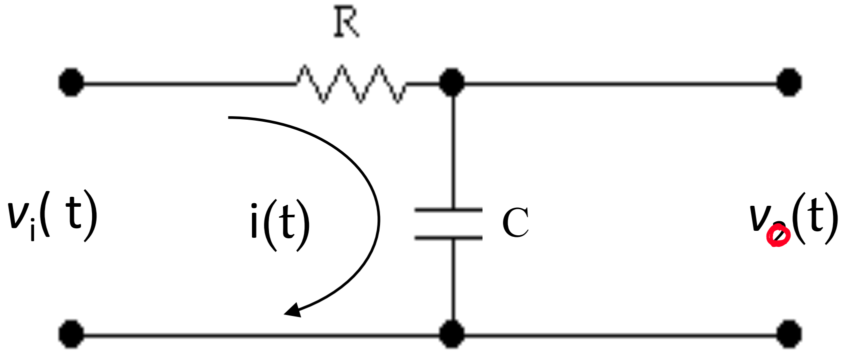
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#2

- Design an Electrical system that would place a pole at -3 if added to another system.

transfer function

$$\left(\frac{V_o(s)}{V_i(s)}\right) = \frac{1}{1+RCs}$$



- System has one pole at

$$s = -\frac{1}{RC}$$

- Therefore,

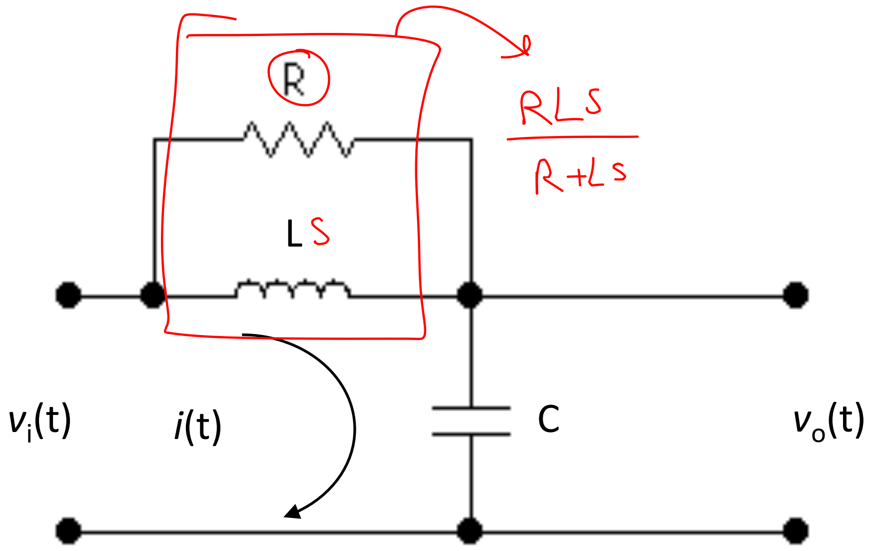
$$-\frac{1}{RC} = -3 \qquad \textit{if} \qquad R = 1\,M\Omega \qquad \textit{and} \qquad C = 333\,pF$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#3

- Find the transfer function G(S) of the following two port network.

$\frac{V_o}{V_i}$



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#3

* برضه هاي الدائرة = $\frac{1}{1}$

لكن معادلة المقام سوف تكون تربيعية (complex)

- Simplify network by replacing multiple components with their equivalent transform impedance.

$$\frac{1}{Z} = \frac{1}{R} + \frac{1}{LS}$$

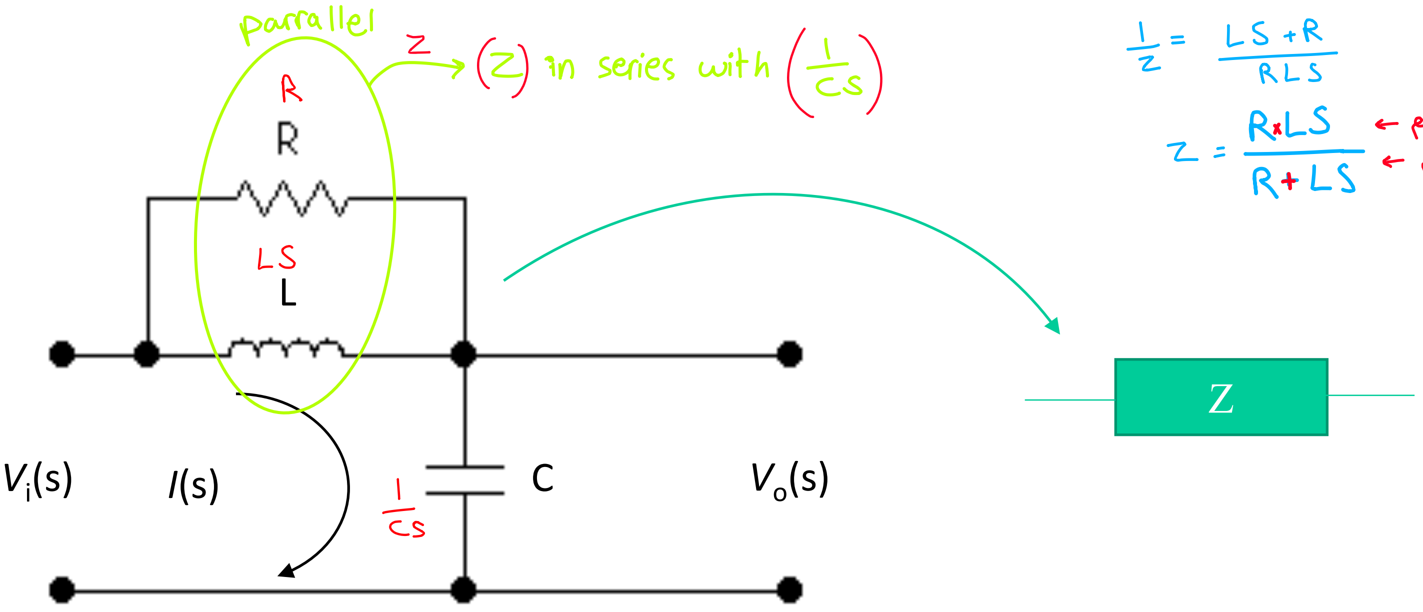
تبسيط

$$\frac{1}{Z} = \frac{LS + R}{RLS}$$

$$Z = \frac{R \times LS}{R + LS}$$

← بضربهم

← بجمعهم

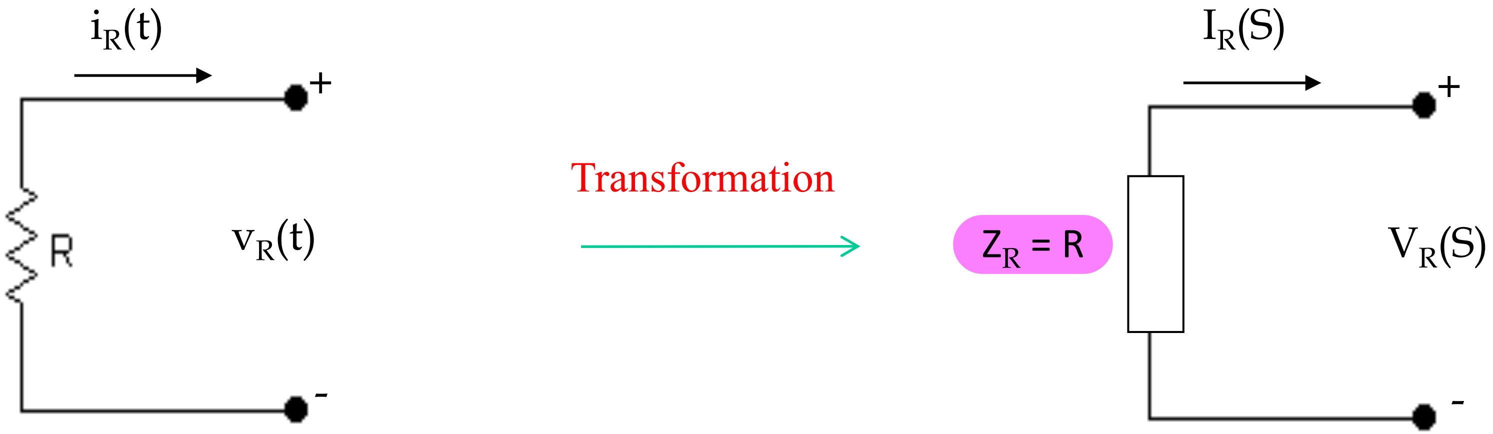


$$\frac{V_o}{V_i} = \frac{1}{ZCS + 1}$$

من الدرجة الثانية

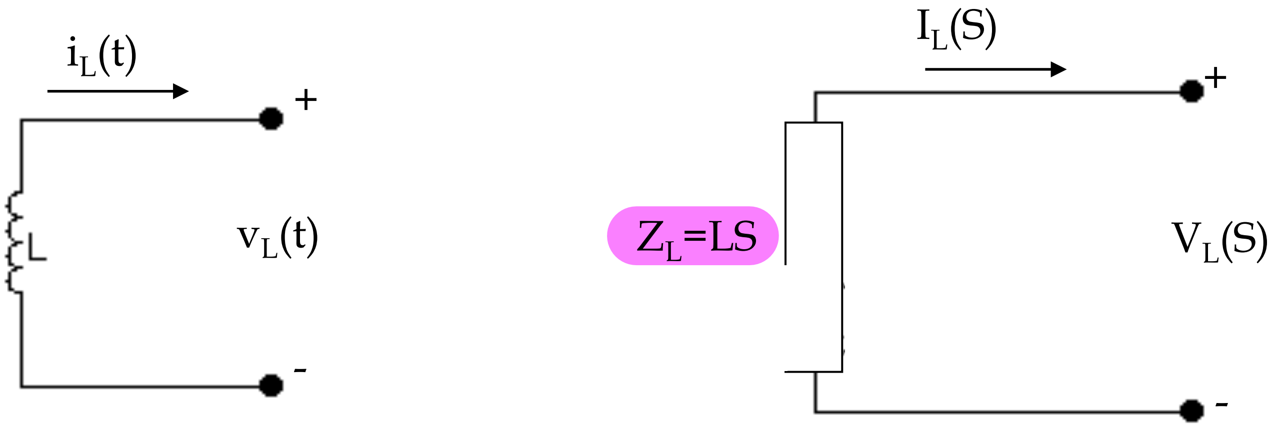
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Transform Impedance (Resistor)



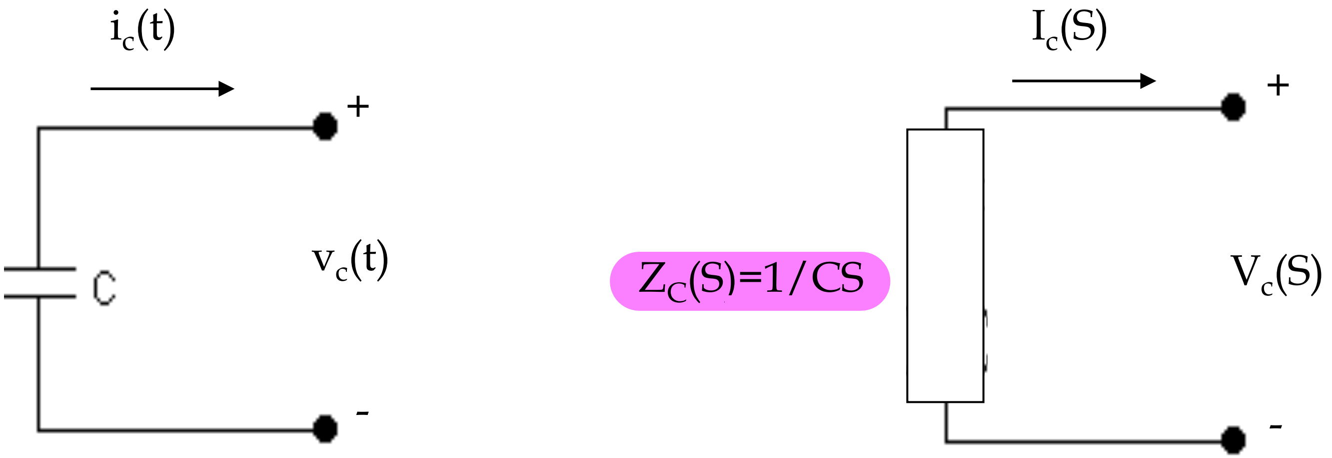
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Transform Impedance (Inductor)



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Transform Impedance (Capacitor)



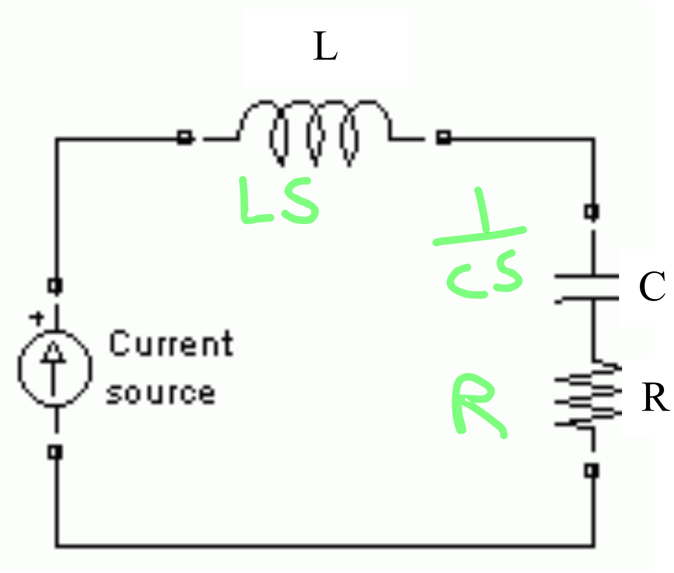
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Equivalent Transform Impedance (Series)

- Consider following arrangement, find out equivalent transform impedance.

$$Z_T = Z_R + Z_L + Z_C$$

$$\underline{Z_T = R + Ls + \frac{1}{Cs}}$$

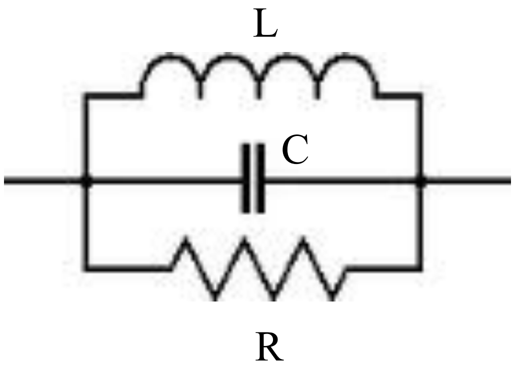


Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Equivalent Transform Impedance (Parallel)

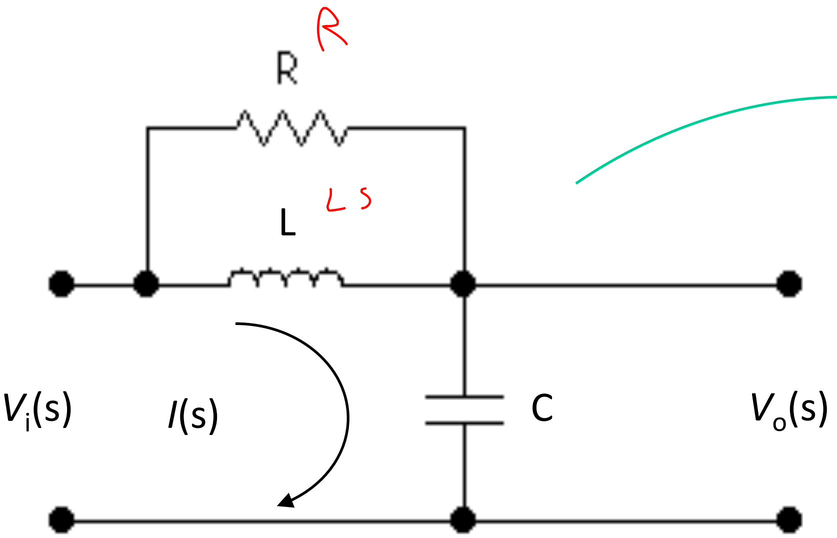
$$\frac{1}{Z_T} = \frac{1}{Z_R} + \frac{1}{Z_L} + \frac{1}{Z_C}$$

$$\frac{1}{Z_T} = \frac{1}{R} + \frac{1}{Ls} + \frac{1}{\frac{1}{Cs}}$$



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Back to Example#3



$$\frac{1}{Z} = \frac{1}{Z_R} + \frac{1}{Z_L}$$

$$\frac{1}{Z} = \frac{1}{R} + \frac{1}{Ls}$$

$$Z = \frac{RLs}{1 + RLs}$$

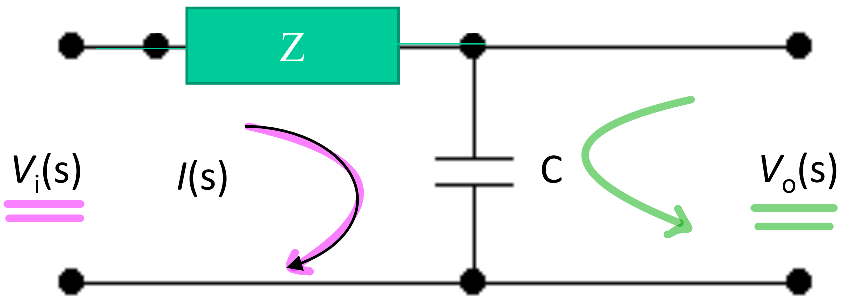
طلعت معي

$$Z = \frac{RLs}{R+Ls}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#3

$$Z = \frac{RLs}{1 + RLs}$$



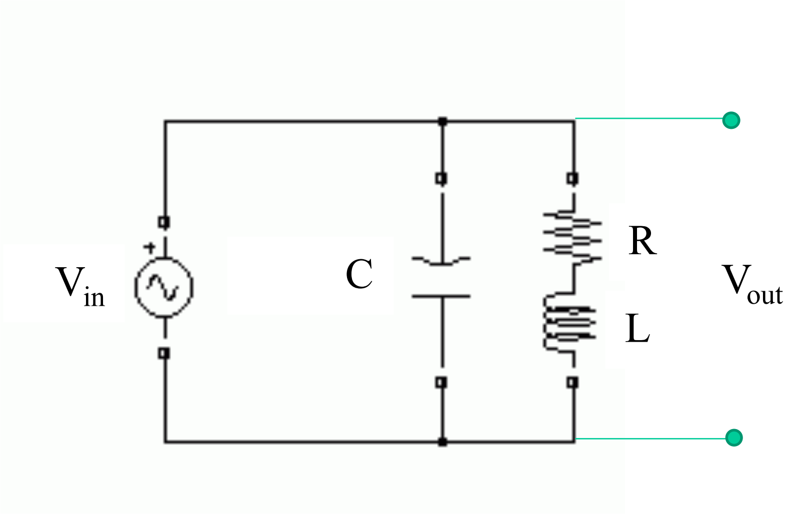
$$\underline{\underline{V_i(s)}} = I(s)Z + \frac{1}{Cs}I(s)$$

$$\underline{\underline{V_o(s)}} = \frac{1}{Cs}I(s)$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#4

- Find transfer function $V_{out}(s)/V_{in}(s)$ of the following electrical network



غير
محلول

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Electronic Systems

Part-II

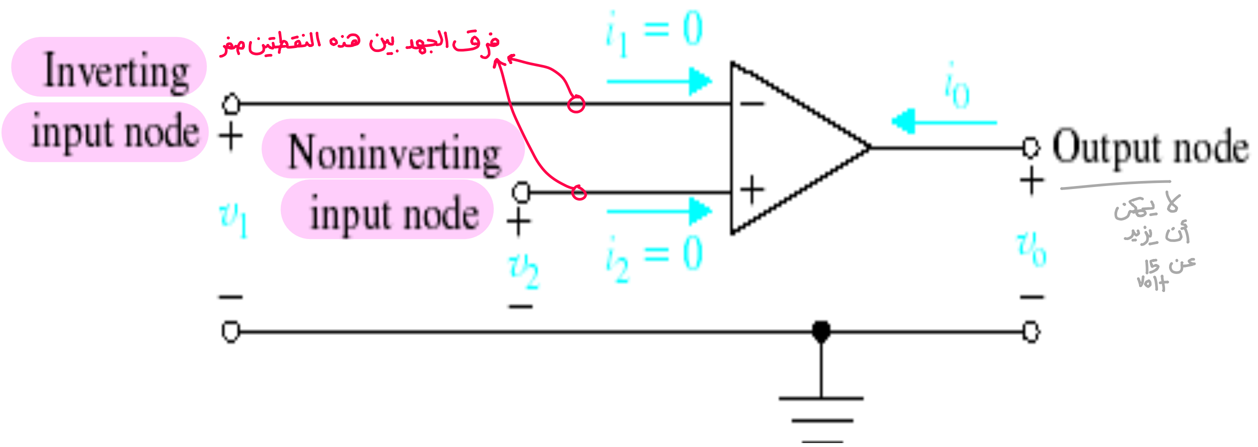
Amplifiers

video
لتسهيل
الفهم

https://youtu.be/yJkgsMQc8Gw?si=7dxogUJ0FpTITYxe

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

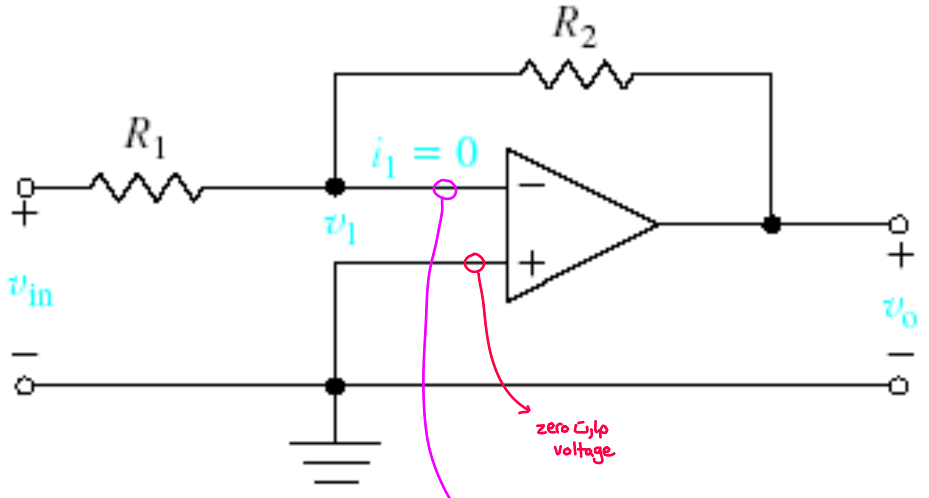
The Transfer Function of Linear Systems



فرق الجهد بين هذه النقطتين صفر

لا يمكن أن يزيد عن 15 volt

The ideal op-amp



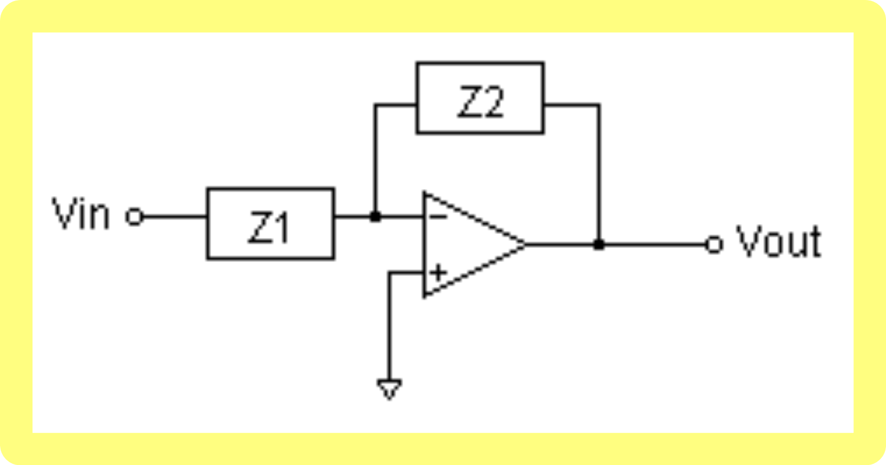
صارت zero voltage

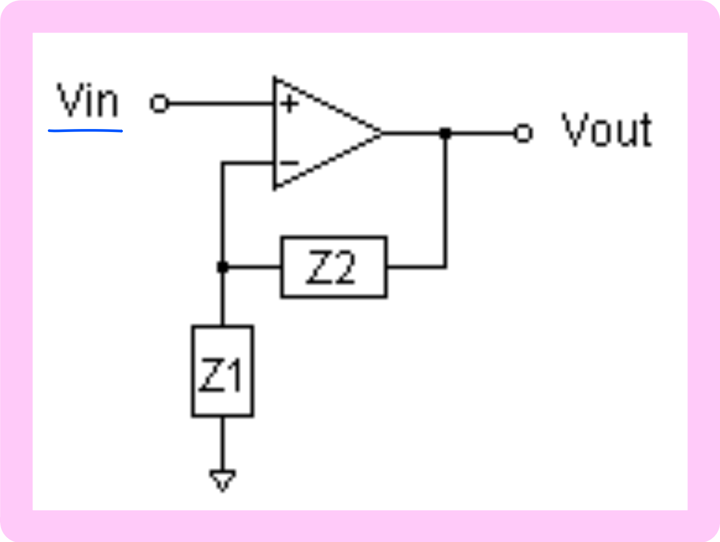
اذاً مباشرةً صارت هاي كمان zero voltage

An inverting amplifier operating with ideal conditions.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Operational Amplifiers





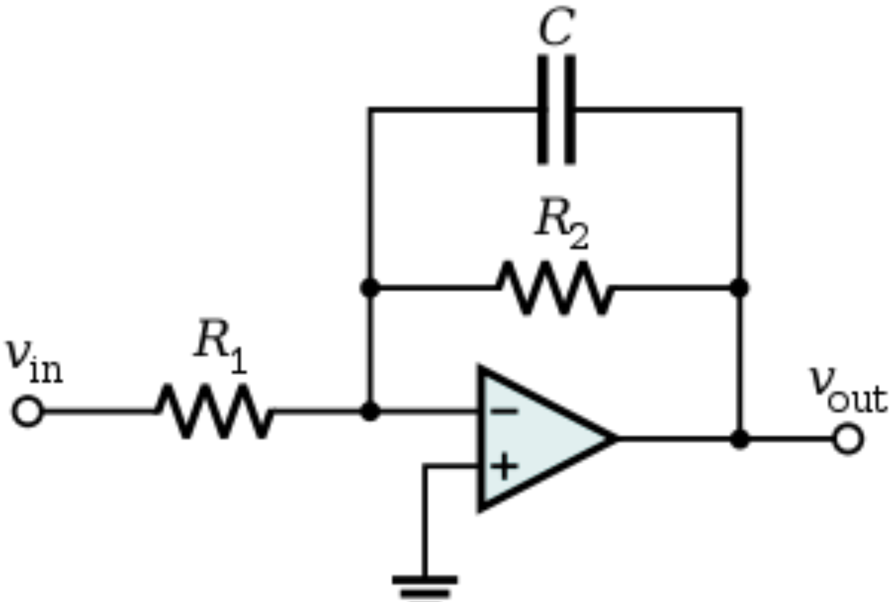
$$\frac{V_{out}}{V_{in}} = -\frac{Z_2}{Z_1}$$

$$\frac{V_{out}}{V_{in}} = 1 + \frac{Z_2}{Z_1}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#6

- Find out the transfer function of the following circuit.



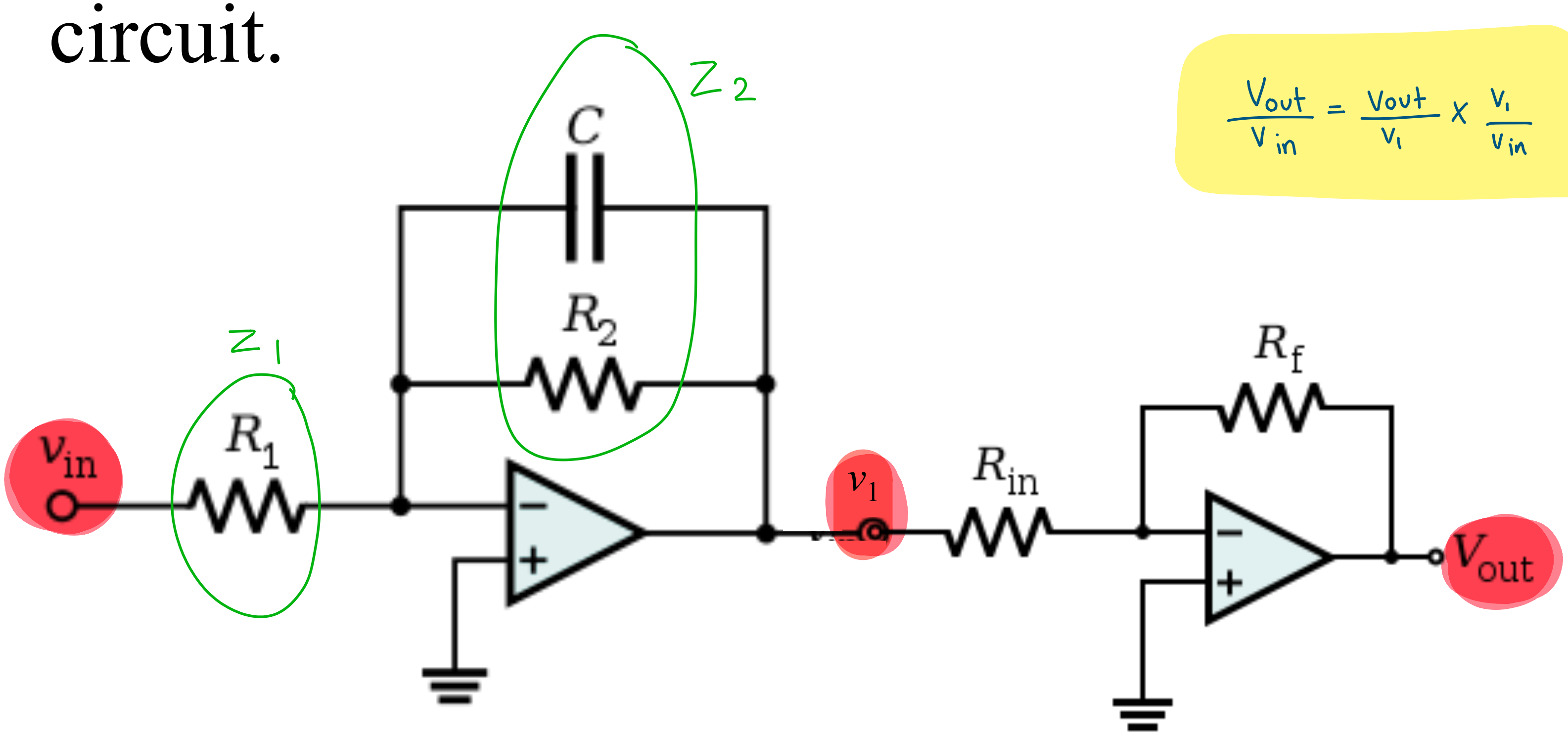
$$\frac{V_{out}}{V_{in}} = -\frac{Z_2}{Z_1}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#7

- Find out the transfer function of the following circuit.

$$\frac{V_{out}}{V_{in}} = \frac{V_{out}}{V_1} \times \frac{V_1}{V_{in}}$$

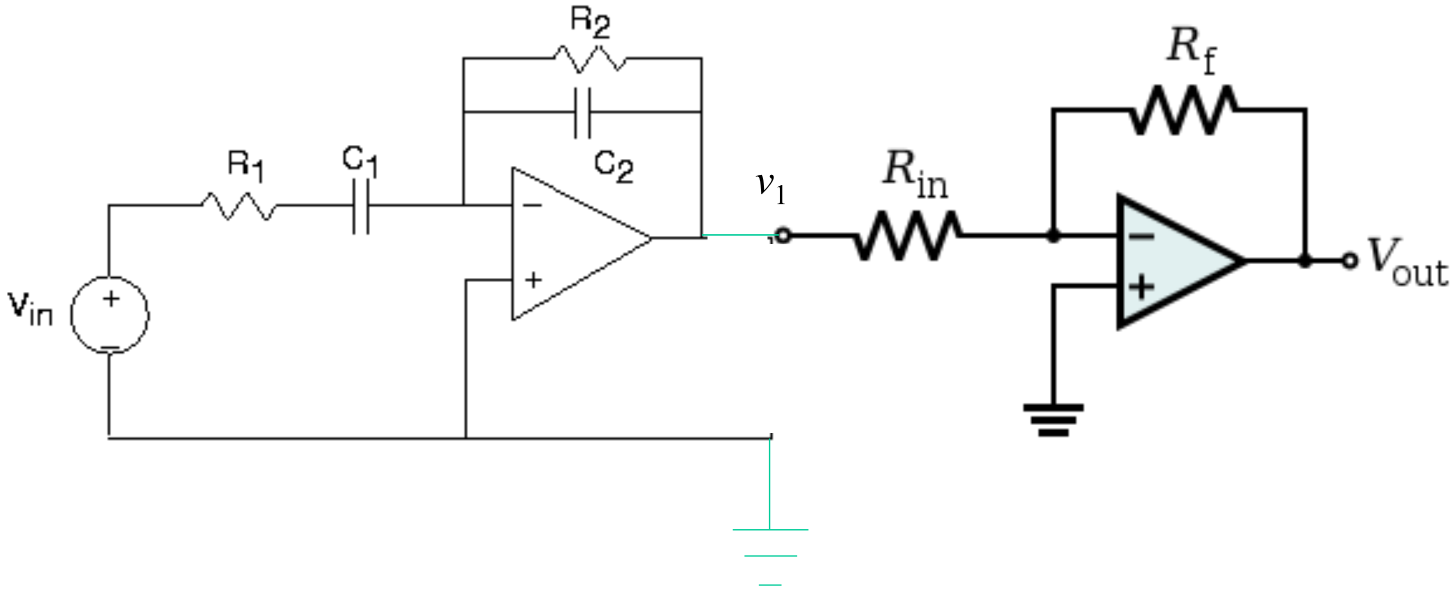


Illustrations

© 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#8

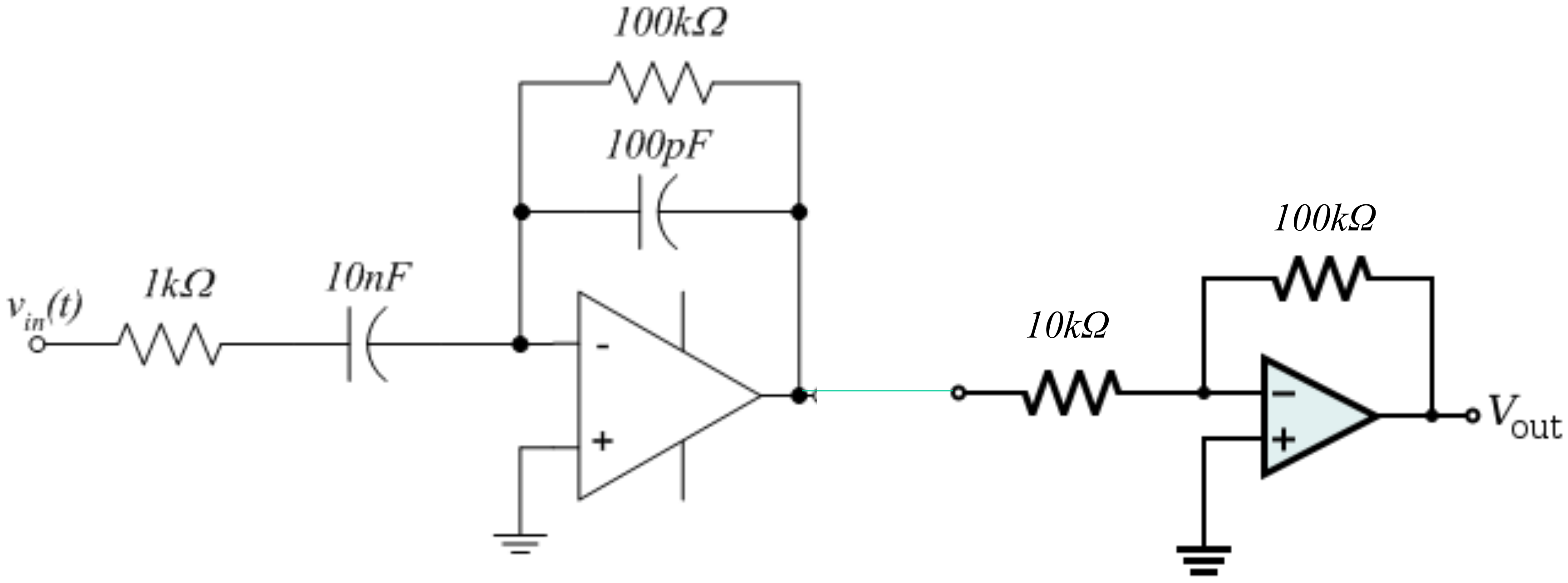
- Find out the transfer function of the following circuit.



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#9

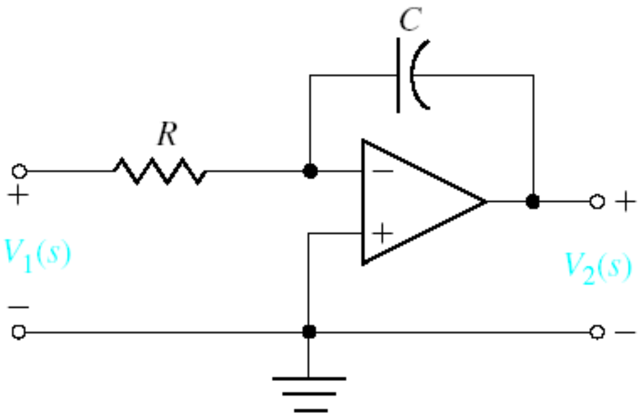
- Find out the transfer function of the following circuit and draw the pole zero map.



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

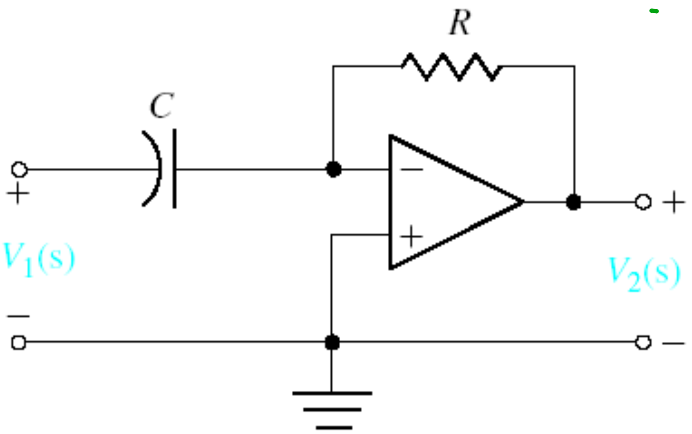
Examples write the transfer function for the following systems

✓



$$\frac{V_2(s)}{V_1(s)} = \frac{-1}{RCs}$$

$$= \frac{\frac{-1}{Cs}}{R}$$



$$\frac{V_2(s)}{V_1(s)} = -RCs$$

$$= \frac{-R}{\frac{1}{Cs}}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

* To See it on Matlab :-

⟶ in the (Command window)

≫ simulink

⟶ then in "New" tab ⟶ press "simscape" ⟶ choose "Electrical" ⟶ . . . . .

2. ☆

Mechanical systems

The modelling of mechanical systems are mainly based on *Newton's second law*

$$F = ma \tag{3.4}$$

F is the *force* acting on the *mass* m and a is the *acceleration* of the mass.

Example 3.3. An undamped pendulum.

Figure 3.4 shows an undamped swinging pendulum. The pendulum can only move in two directions in the plane of the figure. Its point of sus-pension is at a distance u and its center of mass (the round weight at the lower end of the pendulum) is at a distance y from the left-side vertical line.

How does the position y depend on u ?

Notation:

- ℓ = length of pendulum, m = weight of mass
- h = vertical position of the center of mass
- θ = angle of swing away from a vertical position
- F = force acting on the suspension point in the "negative direction" (upwards)

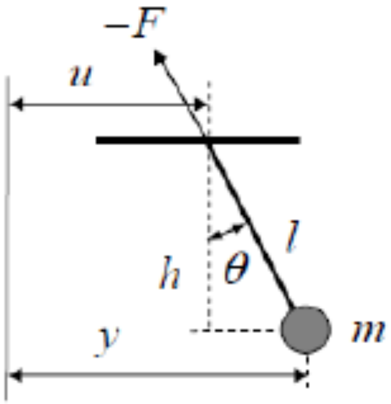


Fig. 3.4. Swinging pendulum.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

* مش عارفة
اذا داخلة

ة When the pendulum is affected by the suspension force F and the gravitational force mg, *Newton's second law* yields

- horizontal force components: $m\ddot{y} = -F \sin\theta$ (1)
- vertical force components: $m\ddot{h} = -F\cos\theta + mg$ (2)

Here $\ddot{y}$ and $\ddot{h}$ are second-order time derivatives of y and h, respectively, i.e. the *acceleration* in the respective directions.

Assume that the swing of the pendulum is moderate so that the angle θ is always small. The pendulum then moves very little in the vertical direction and we can assume that $\ddot{h} \approx 0$. Elimination of F then gives

$$\ddot{y} + g\tan\theta = 0 \tag{3}$$

The angle θ is given by the trigonometric identity

$$\tan\theta = \frac{y-u}{h} \approx \frac{y-u}{l} \tag{4}$$

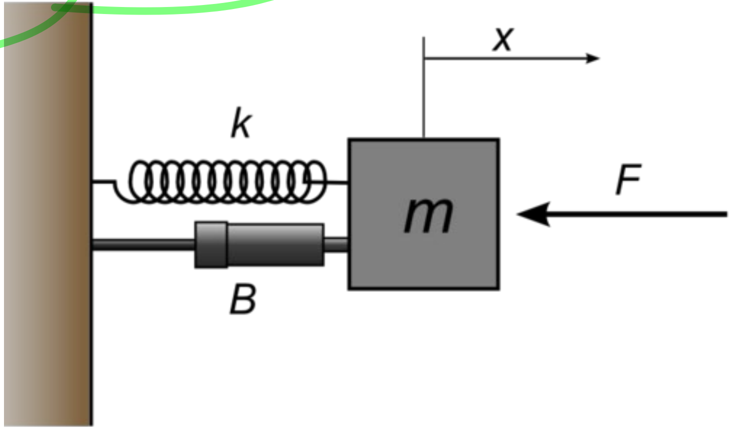
Combination of (3) and (4) yields the model

$$\ddot{y} + \left(\frac{g}{\ell}\right)y = \left(\frac{g}{\ell}\right)u \tag{5}$$

Notice that the approximations $\ddot{h} \approx 0$ and "θ small" ***limit the validity of the model***.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Basic Types of Mechanical Systems



- Translational
  - Linear Motion

*we have 3 components
- spring
- mass
- damper

- Rotational
  - Rotational Motion

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

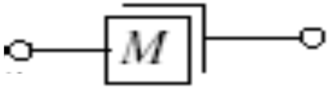
Basic Elements of Translational Mechanical Systems

Translational Spring

i)

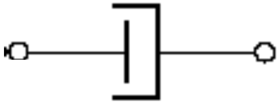
Translational Mass

ii)



Translational Damper

iii)



Illustrations

© 2001 by Prentice Hall, Upper Saddle River, NJ.

Elastic Strain
$< 2\%$

(المادة ممكن تطول 2%)

Translational Spring

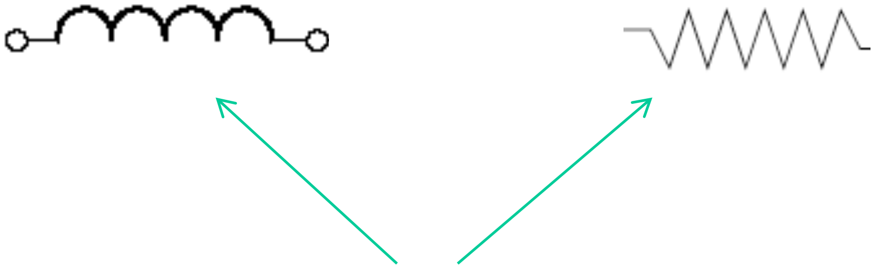
The Spring should Stay in the elastic region

- A translational spring is a mechanical element that can be deformed by an external force such that the deformation is directly proportional to the force applied to it.

Helical shape

Translational Spring

i)



Circuit Symbols

Translational Spring

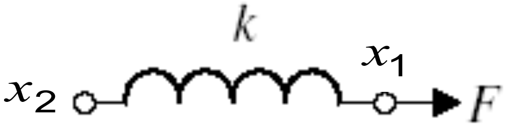
طول ما هي بترجع زي ما هي

إذاً we are in the elastic region

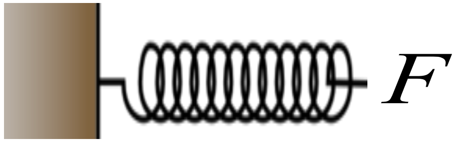
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Translational Spring

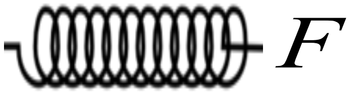
- If *F* is the applied force



- Then x_1 is the deformation if $x_2 = 0$



- Or $(x_1 - x_2)$ is the deformation.



- The equation of motion is given as

معادلة الزمبرك
هي معادلة مقاومة

$F = K(X_2 - X_1)$

يقاوم من جهة x_2

↔
in opposite
sides
لأن الزمبرك يقاوم

$$\boxed{F = k(\underline{x_1} - x_2)}$$

يقاوم من جهة x_1

- Where **k** is stiffness of spring expressed in N/m

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

The Force in general will move

Translational Mass

- Translational Mass is an inertia element. الكتلة الانتقالية هي عنصر القصور الذاتي.

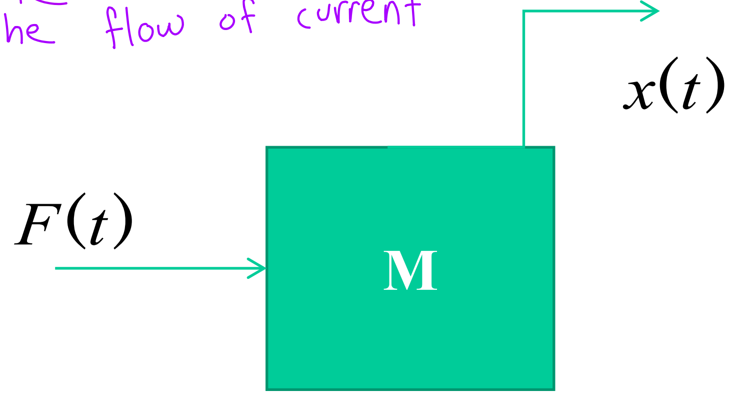
ii)

Translational Mass

- A mechanical system without mass does not exist.

spring resists the translation x
Mass resists the acceleration $\ddot{x}$
resistance resist the flow of current
capacitor resist the flow of current

- If a force *F* is applied to a mass and it is displaced to *x* meters then the relation b/w force and displacements is given by Newton's law.



higher mass resist more than lower mass

$$F = M\ddot{x}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Translational Damper

- Damper opposes the rate of change of motion.

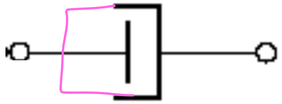
velocity

- All the materials exhibit the property of damping to some extent.

- If damping in the system is not enough then extra elements (e.g. Dashpot) are added to increase damping.

iii)

Translational Damper



the Damper is a cylinder filled with fluid or gas

إذا الDamper بدها تتحرك لليمين لازم ال fluid ينتقل لليسار و العكس

اثناء انتقال ال Fluid ممكن يمر بفتحات ضيقة و يتأثر بال Friction و هذا ينعكس على ال (Coefficient of friction μ)

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Common Uses of Dashpots

Door Stoppers

Vehicle Suspension

Bridge Suspension

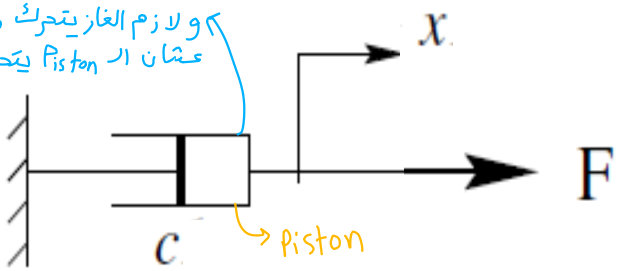
Flyover Suspension

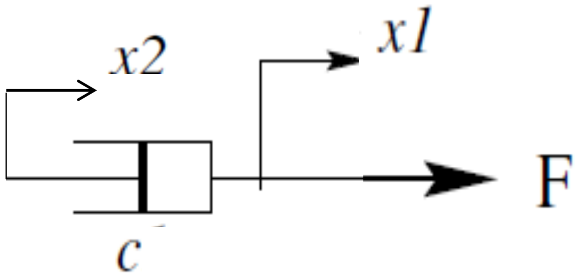
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Translational Damper

تصرفاته نفس تصرفات الزمبرك

ال Piston بكون محبى بغاز
ولازم الغاز يتحرك من فوق لتحت
عشان ال Piston يتحرك





$$F = C\dot{x}$$

اذا كان ال
Damper مثبت
من الجهتين

$$F = C(\dot{x}_1 - \dot{x}_2)$$

هي الـ Force التي بأثر
بها من جهة x_1 مقاومتاً
لحركة ال x_1

لأنها
انكتبت
أول

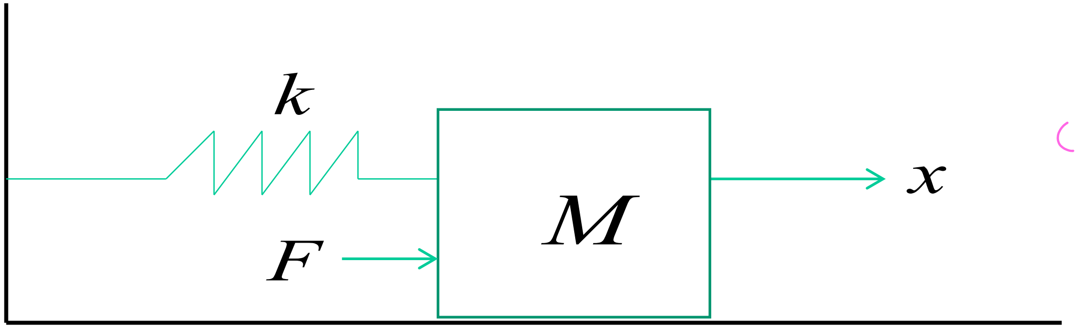
- Where *C* is damping coefficient (*N/ms⁻¹*).

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

مجموع القوة الخارجية = مجموع المقاومات الداخلية

Example-1

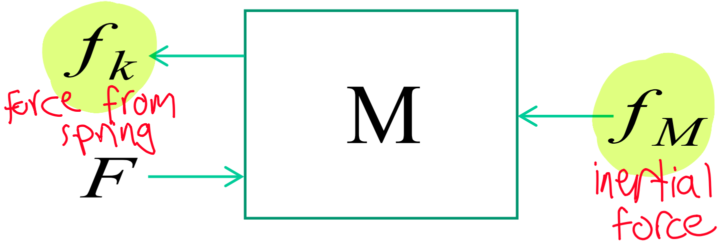
- Consider the following system (friction is negligible)



الزمبرك والـ mass try to impede us (prevent the motion)

$F = Kx + M\ddot{x}$

- Free Body Diagram



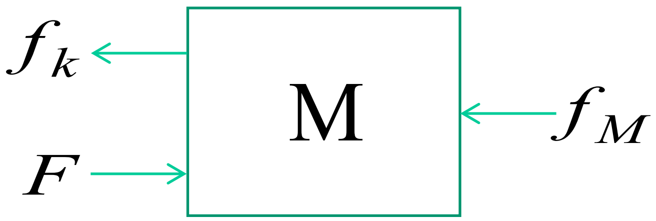
- Where f_k and f_M are force applied by the spring and inertial force respectively.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

in Laplace Transform :-

we assume all initial conditions are equall to zero

Example-1



$$F = f_k + f_M$$

- Then the differential equation of the system is:

$$F = kx + M\ddot{x}$$

$x(0)=0$
$\dot{x}(0)=0$
$\ddot{x}(0)=0$

Assume All initial conditions = zero

- Taking the Laplace Transform of both sides and ignoring initial conditions we get

$$F(s) = Ms^2X(s) + kX(s)$$

ثم نأخذ X عامل مشترك
و نسوي ضرب تبادلي

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

in Control → نعتمد على تعديل معادلة المقام بدون ما نغير أي component بال system

Example-1

$$F(s) = Ms^2X(s) + kX(s)$$

$$\left(\frac{\text{البسط}}{\text{المقام}}\right)$$

دائماً نركز على المقام لأنه اهم
(نعمل على تعديل معادلة المقام)

- The transfer function of the system is

معناها مشتقة ال x والتي هي velocity

$$\frac{S \times X(s)}{F(s)} = \frac{S \times 1}{\underbrace{Ms^2 + k}}$$

ما تأثر بالتغيرات → charactaristic function

نحن نركز على معادلة المقام

$$\frac{V}{F(s)} = \frac{S}{Ms^2+K}$$

- if

$$M = 1000kg$$

$$k = 2000Nm^{-1}$$

$$\frac{X(s)}{F(s)} = \frac{0.001}{s^2 + 2}$$

* اذا قسمنا على M البسط والمقام بصير يشبه sin أو cos

$$\frac{\frac{1}{M}}{S^2 + \frac{K}{M}}$$

البسط يختفي في ال Control (بصير Residual)

المقام يبقى كما هو (التركيز على المقام)

من هنا ال charactaristic Polynomial بقدر اوصف ال system كامل

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

في ال complex domain نقوم برسم ال Real part

Example-2

$$\frac{X(s)}{F(s)} = \frac{0.001}{(s^2+2)}$$

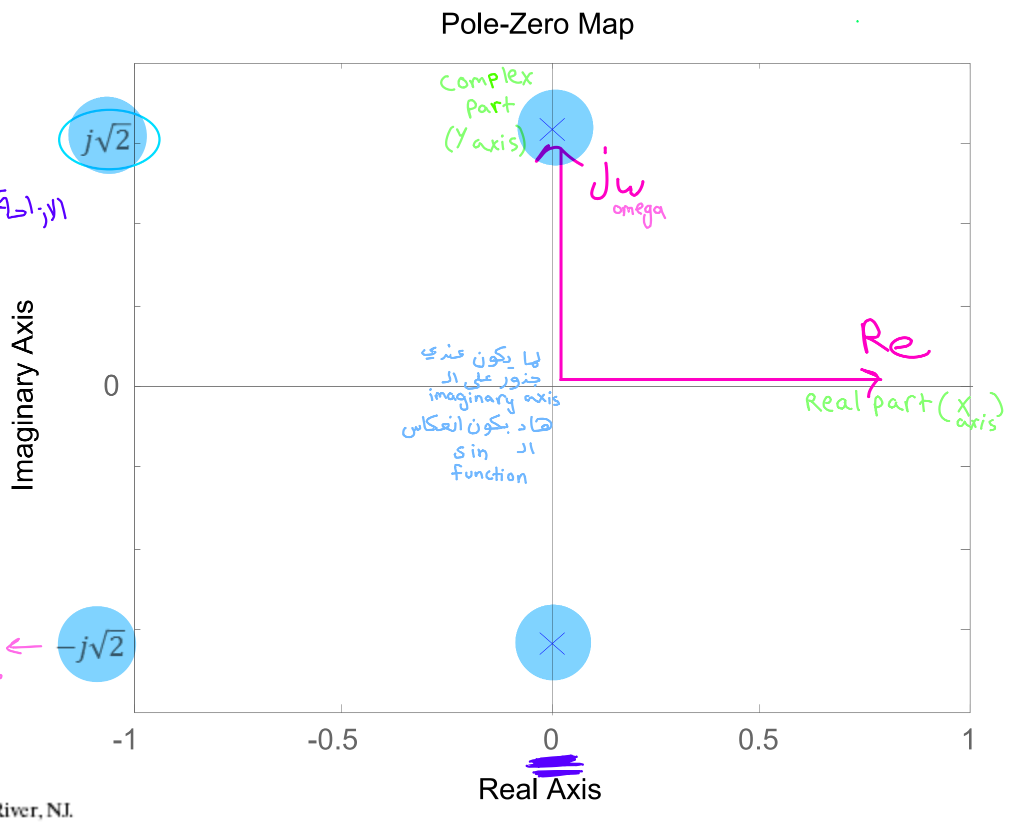
نرسم جذور المقام في ال complex Range

راح يعطيني sinusoidal

* كل جزر من جذور المقام راح يساهم ب Fraction

و هذه الجذور تساهم ب Something sinusoidal

- The pole-zero map of the system is



جذرين حقيقين

لما يكون عندي جذور على ال imaginary axis هاد بكون انعكاس ال sin function

الازاحة = $\frac{1}{٢}$ معامل s = $\frac{1}{٢}(0) = 0$

∴ لا يوجد ازاحة عن ال imaginary axis

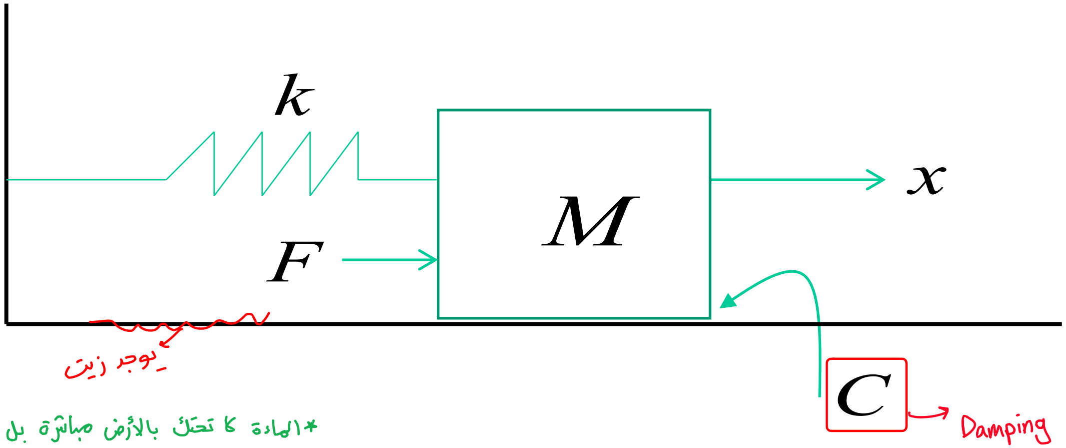
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-2

- Consider the following system

*الاحتكاك بين أي سائل وصلب
بمثل velocity
Resistance

لما تشحّم الأرض بتكون ال
friction مرتبطة بال velocity
وقوة المقاومة مرتبطة بال velocity



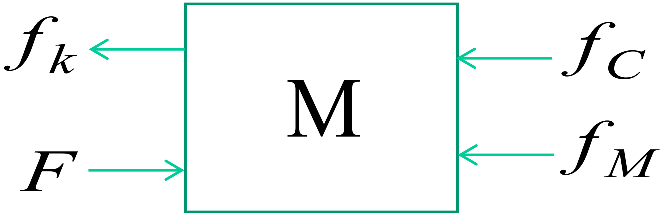
يوجد زيت

Damping coefficient

*المادة لا تحتك بالأرض مباشرة بل تحتك بالسائل يلي مغطي الأرض

- Free Body Diagram

$F_{friction} = \mu N$



Mass → يقاوم ال force
عن طريق
mass x acceleration

Damper → يحاول يقاوم عن طريق
$C \times \dot{X}$

Position K → تحاول تقاوم تغيير ال Position
عن طريق
$K \times X$

$$F = f_k + f_M + f_C$$

ال force يساوي مجموع المقاومات من جهة

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-3

* كل عنصر من العناصر الموجودة بالدائرة
سوف يِساهم بجزء من المقاومة

The (x) itself is the balance

Differential equation of the system is:

$$F = M\underset{\text{acceleratio}}{\underline{\ddot{x}}} + C\dot{x} + kx$$

Taking the Laplace Transform of both sides and ignoring Initial conditions we get

$$F(s) = Ms^2X(s) + CsX(s) + kX(s)$$

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + \underset{\text{The Damper}}{Cs} + k}$$

The Damper
بِين زي احتكاك.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-3

$$\frac{X(s)}{F(s)}=\frac{1}{Ms^2+Cs+k}$$

- if

$$M = 1000kg$$

$$k = 2000Nm^{-1}$$

$$C = 1000N/ms^{-1}$$

★ كل ما كانت رايحة لليسار اكتر
كلما كانت ال energy أكتر
وال damper أكبر
وال die vibration أسرع

* كل ما زاد ال damping
زادت ال exponential
e^{-at}

$F \longrightarrow \boxed{a} \longrightarrow X$

$x = aF$

a

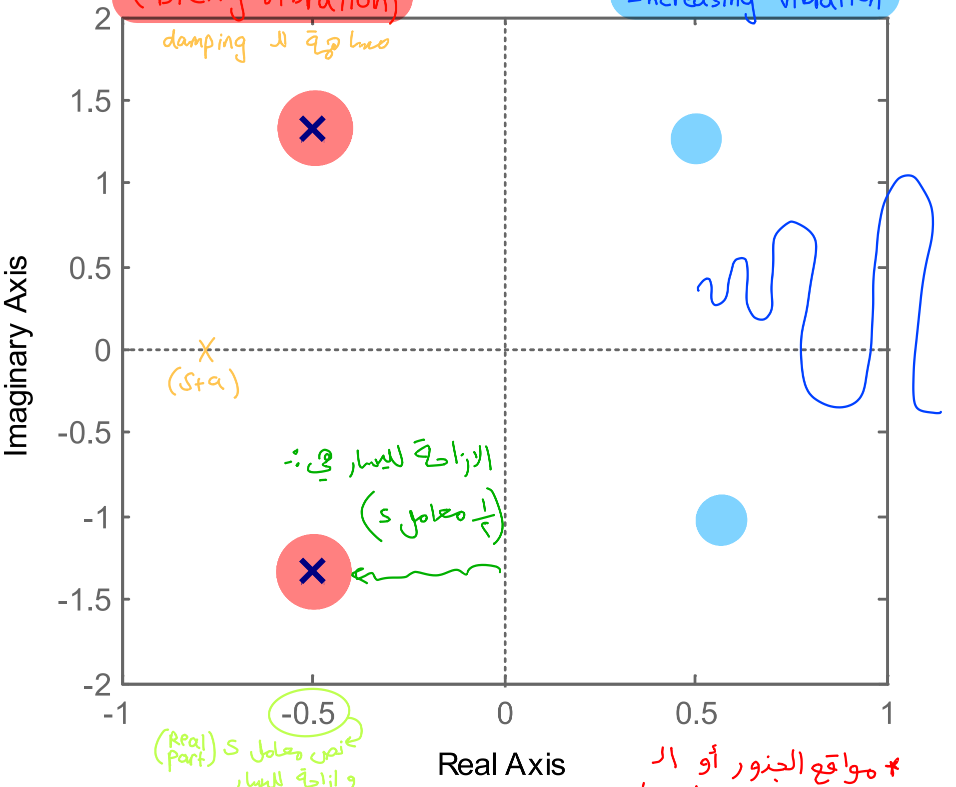
$$\frac{X(s)}{F(s)}=\frac{0.001}{s^2+s+\cancel{\text{[illegible]}}\;2}$$

اذا اخذنا نص معامل s وربعناه راح يعطيني complex

تظهر بالنهاية
$(s+a)^2+(\omega)^2$
راح يعطيني dieing vibration

whenever I see this it will be a
(Dieng Vibration)
معاملة ال damping

مثلاً لو كانت عالجهة هاي بكون اسمها
Increasing vibration



الازاحة لليسار هي:
(½ معامل s)

نص معامل s (Real part)
و ازاحة لليسار

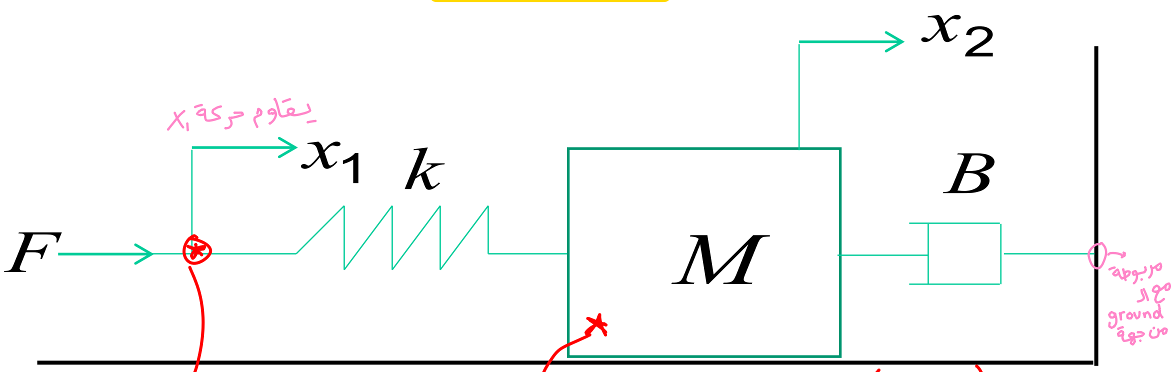
* مواقع الجذور أو ال characteristic Polynomial كافية لمعرفة وجود dieng vibration

بكون النقطة طايرة بالهوا
وفي ازاحة عن ال axis

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-4

- Consider the following system



$F = K(X_1 - X_2)$

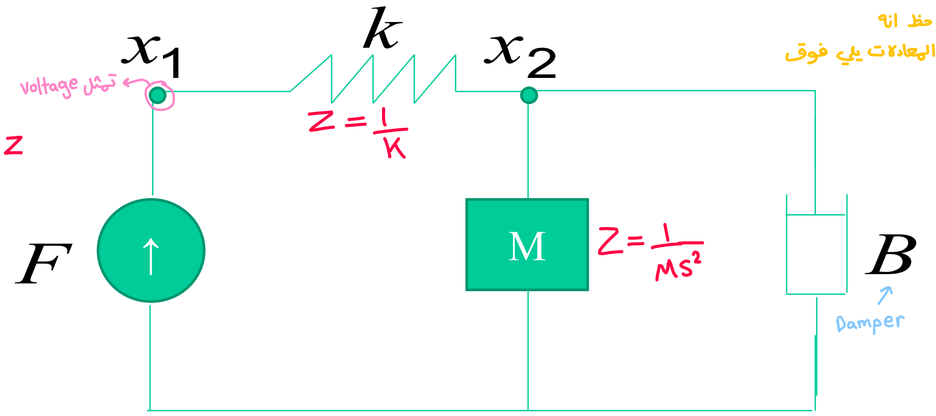
$F = M\ddot{X}_2 + B\dot{X}_2 + K(X_2 - X_1)$

$0 = M\ddot{X}_2 + B\dot{X}_2 + K(X_2 - X_1)$

- Mechanical Network

ليس من الضروري التحويل إلى هذا الشكل قبل البدء بالحل

impedences=z



$Z = \frac{1}{Bs}$

نلاحظ انه نفس المعادلات يلي فوق

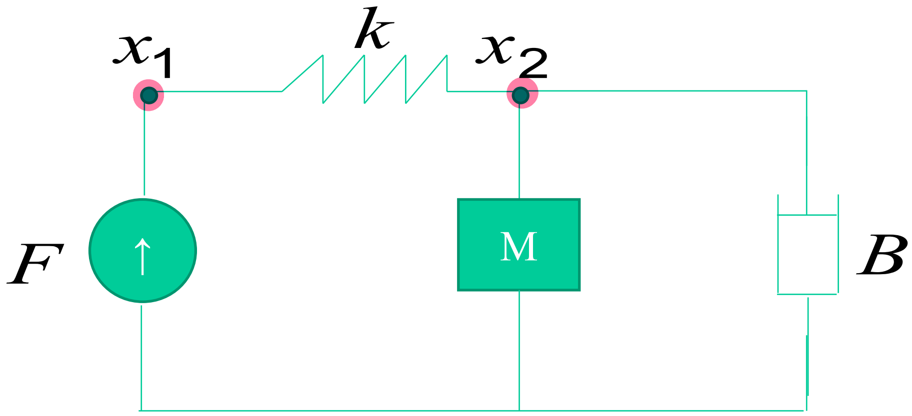
$F = \frac{X_1 - X_2}{\frac{1}{K}} \longrightarrow \left[F = K(X_1 - X_2)\right]$ نفس المعادلة يلي فوق

$0 = \frac{X_2 - X_1}{\frac{1}{K}} + \frac{X_2 - 0}{\frac{1}{Bs}} + \frac{X_2 - 0}{\frac{1}{Ms^2}}$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-4

- Mechanical Network



At node x_1

$$F = k(x_1 - x_2)$$

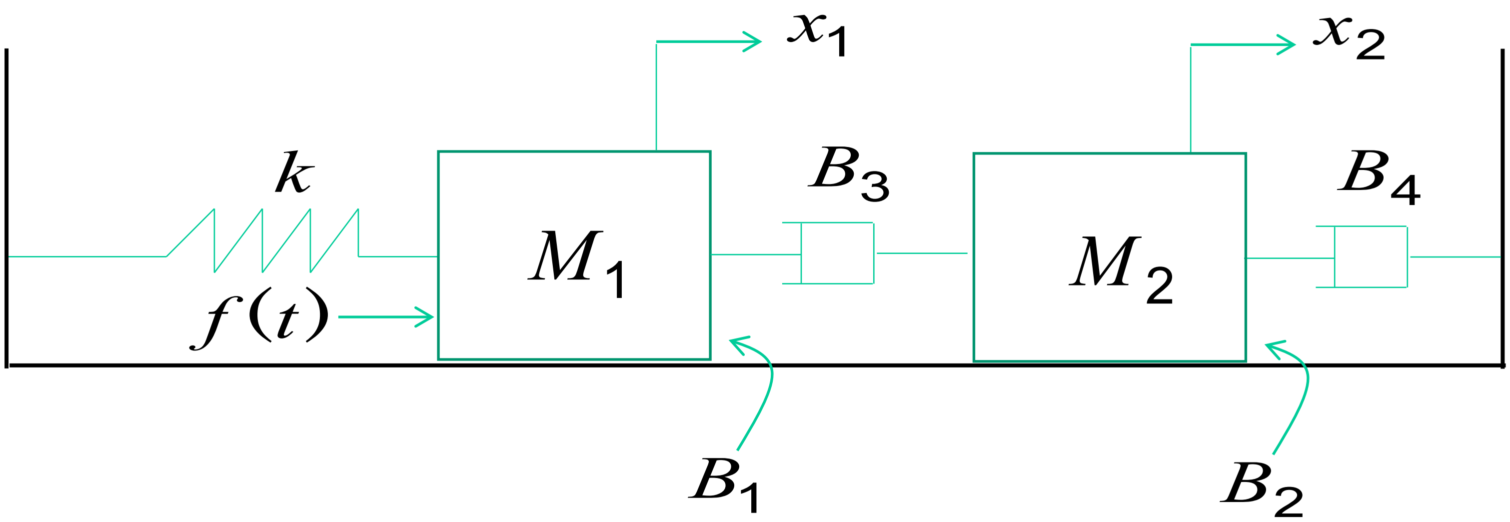
At node x_2

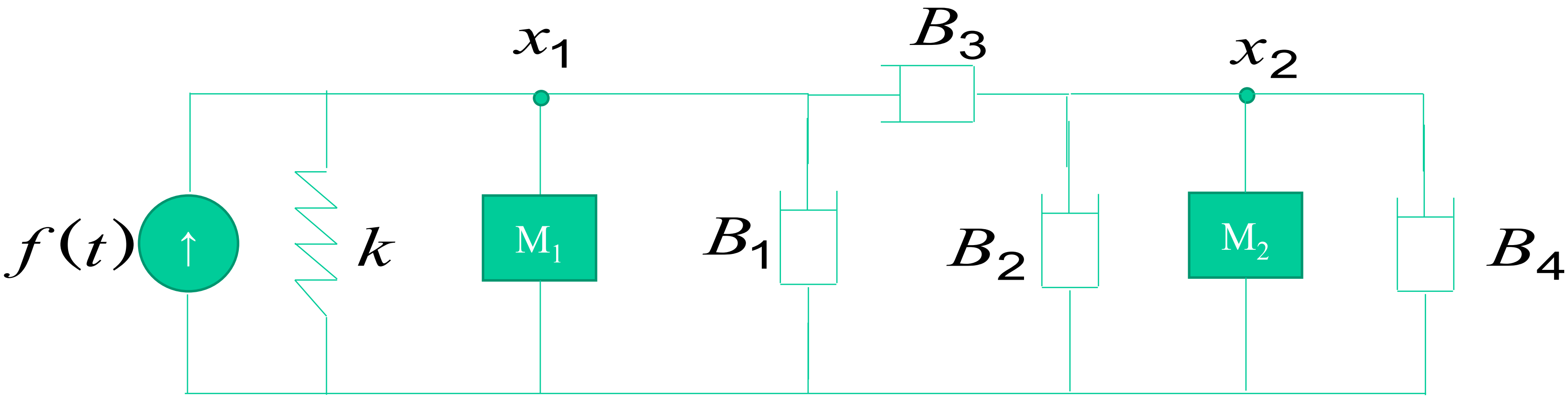
$$0 = k(x_2 - x_1) + M\ddot{x}_2 + B\dot{x}_2$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-6

اخدناه برضه على
ال MATLAB





Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

* تذكير :-

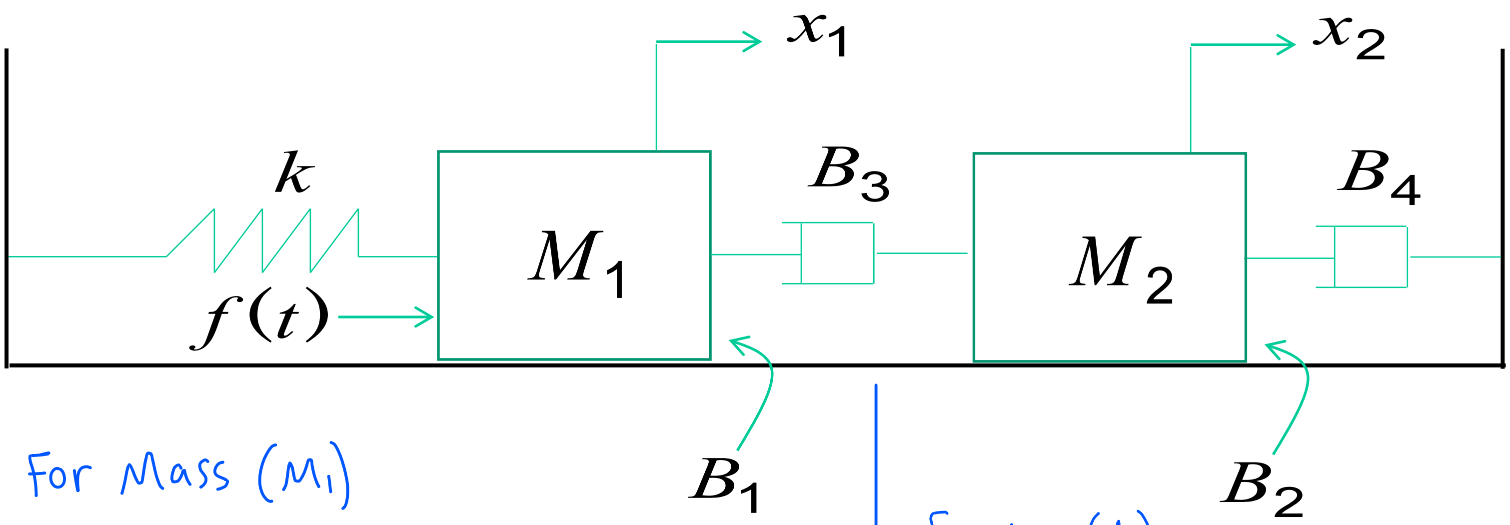
ال Mass يقاوم الacceleration اذاً $\ddot{x}$

ال damper يقاوم ال velocity اذاً $\dot{x}$

ال spring يقاوم ال distance اذاً x

Example-6

عندي 2 masses اذاً عندي معادلتين



For Mass (M_1)

$$F(t) = \underbrace{M_1\ddot{X}_1}_{\text{Mass}} + \underbrace{K X_1}_{\text{spring}} + \underbrace{B_1\dot{X}_1}_{\text{damper}} + \underbrace{B_3(\dot{X}_1-\dot{X}_2)}_{\text{damper}}$$

For Mass (M_2)

$$0 = M_2\ddot{X}_2 + B_2\dot{X}_2 + B_4\dot{X}_2 + B_3(\dot{X}_2-\dot{X}_1)$$

Example-7

- Find the transfer function of the mechanical translational system given in Figure-1.

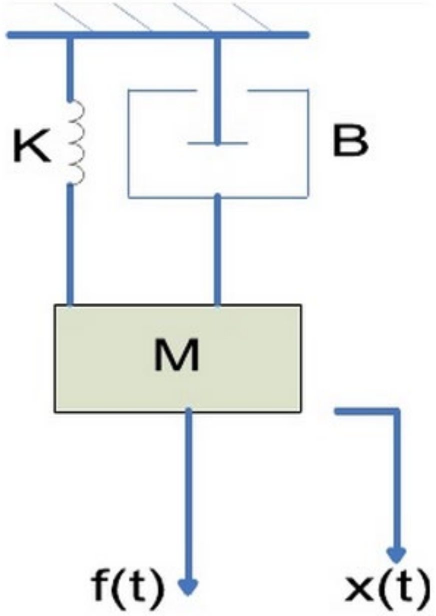
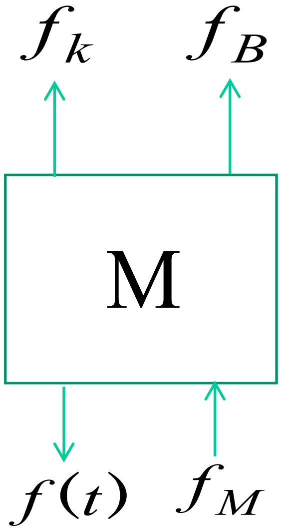


Figure-1

Free Body Diagram



$f(t) = Kx + M\ddot{x} + B\dot{x}$

$$f(t) = f_k + f_M + f_B$$

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + Bs + k}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Basic Elements of Rotational Mechanical Systems

اخذناه بمادة الفاينل وليس الميد

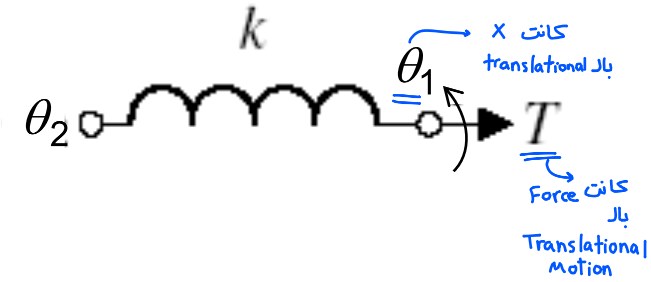
هنا Torque تقابل الـ Force

θ تقابل الـ Position

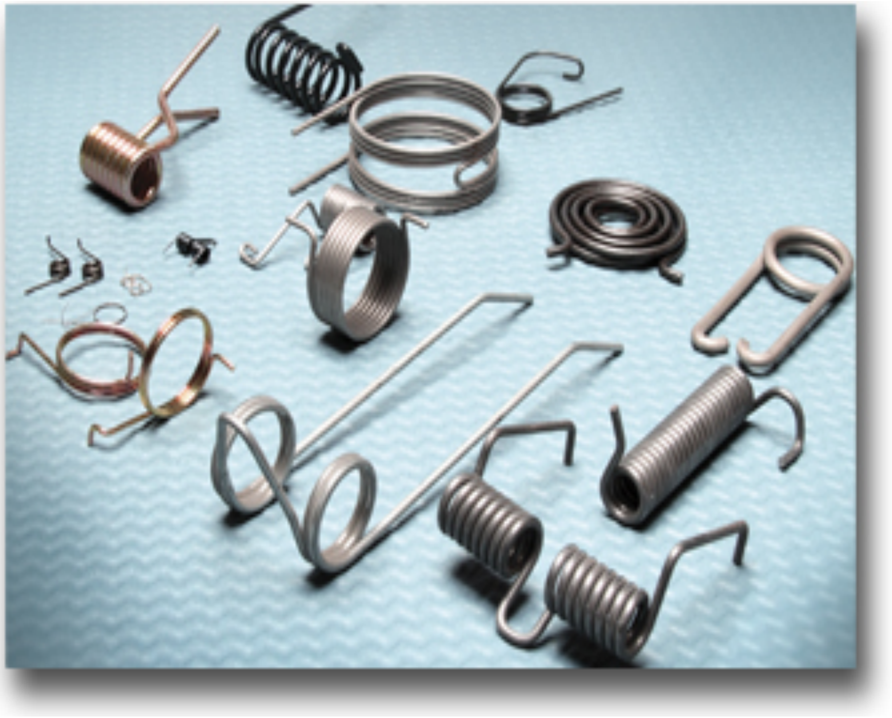
وباقي الاشياء نفسها

و الـ inhertia (J) تقابل الـ Mass

Rotational Spring



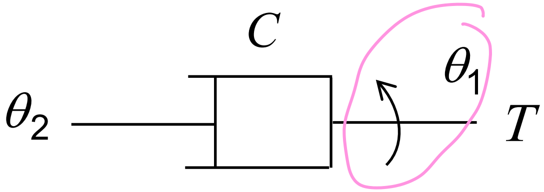
$$T = k(\theta_1 - \theta_2)$$



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Basic Elements of Rotational Mechanical Systems

Rotational Damper

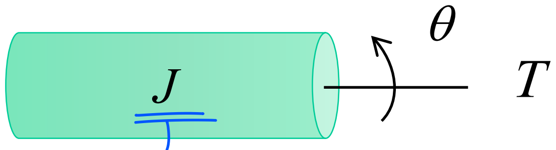


$$T = C(\dot{\theta}_1 - \dot{\theta}_2)$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Basic Elements of Rotational Mechanical Systems

Moment of Inertia



تقابل ال Mass في ال Translational Motion

$$T = J\ddot{\theta}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

اخذناه بمادة الفاينل
وليس الميد

Gears
Tanks
DC motors

Video
لتسهيل الفهم

https://youtu.be/RDUtnosb6yg?si=ZPUy15_7ESqDyvr_

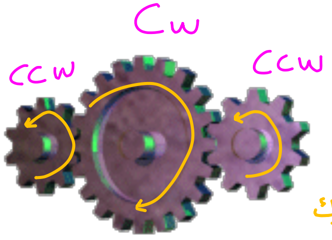
Power = Force x velocity $(\dot{x})$

Gear

Final Exam Material
(Not in Midterm)

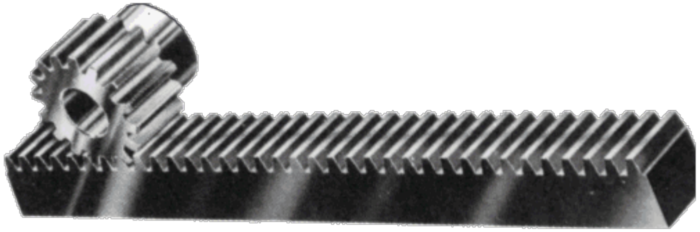
- **Gear** is a toothed machine part, such as a wheel or cylinder, that meshes with another toothed part to ① transmit motion or to ② change speed or ③ direction.

هدف ال gear



الصورة تتحرك

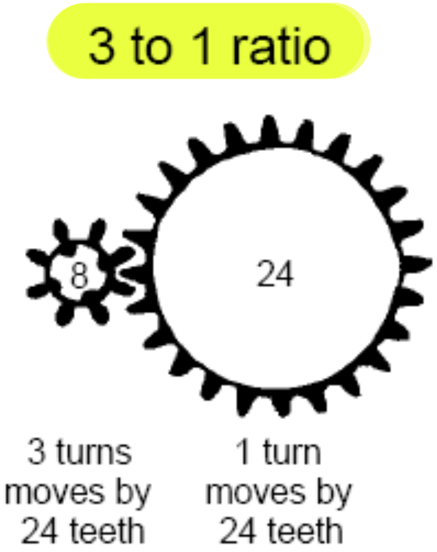
involute gear مبني على فكرة ال



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Gearing Up and Down

- Gearing up is able to convert torque to velocity.
- The more velocity gained, the more torque sacrifice.
- The ratio is exactly the same: if you get three times your original angular velocity, you reduce the resulting torque to one third.
- This conversion is symmetric: we can also convert velocity to torque at the same ratio.
- The price of the conversion is power loss due to friction.



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

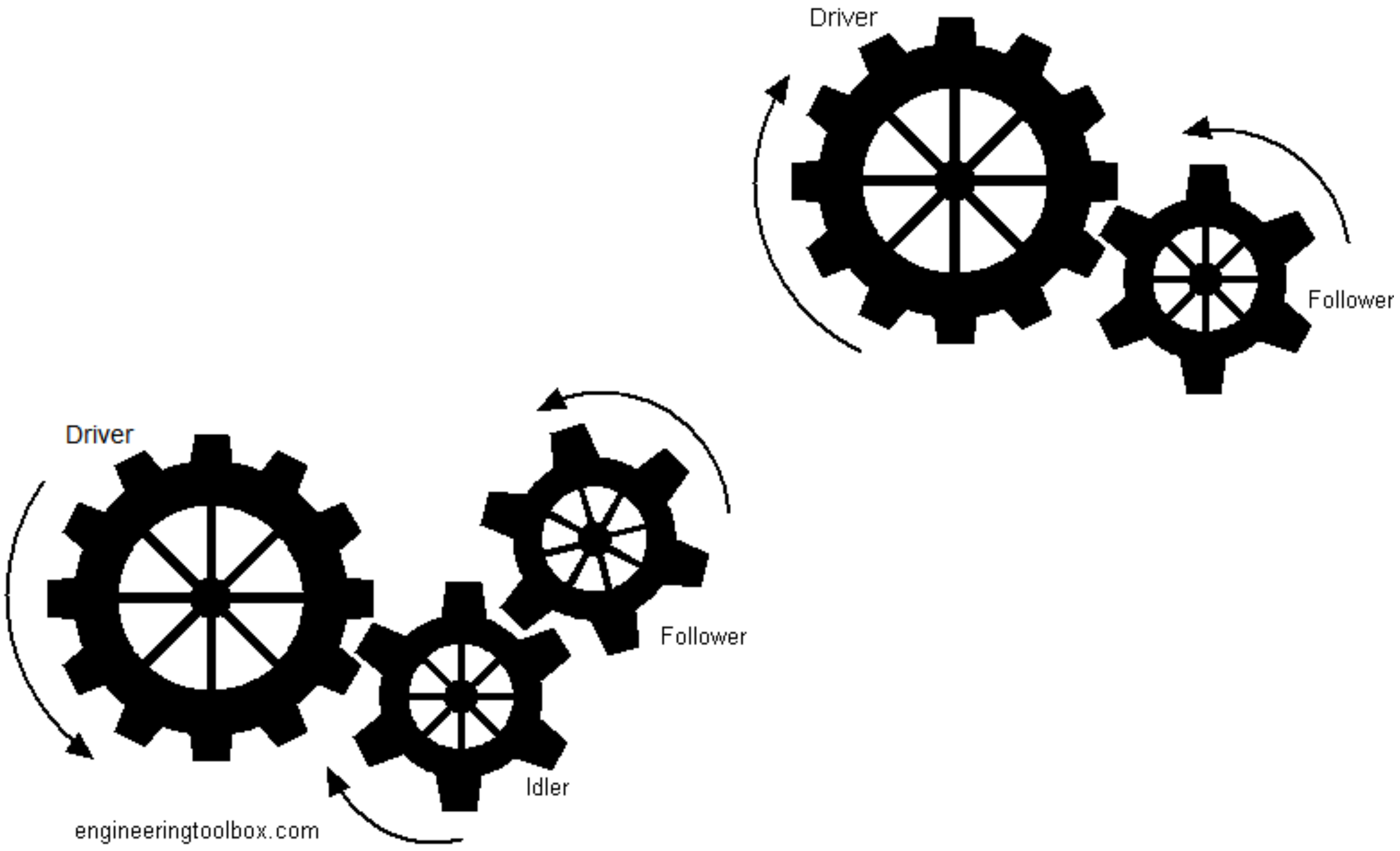
Why Gearing is necessary?

- A typical DC motor operates at speeds that are far too high to be useful, and at torques that are far too low.

- *Gear reduction* is the standard method by which a motor is made useful.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Gear Trains



engineeringtoolbox.com

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Gear Ratio

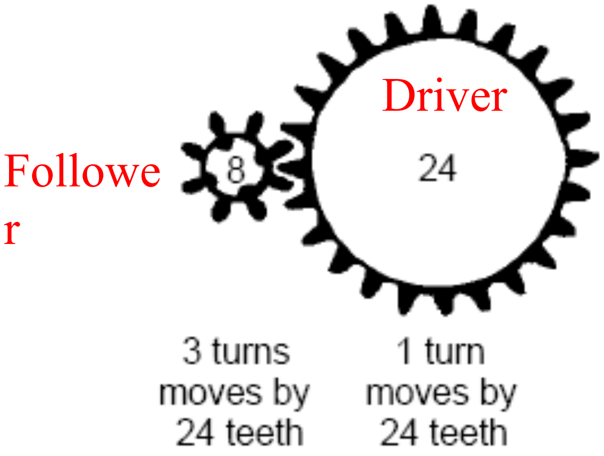
✓

- You can calculate the **gear ratio** by using the number of teeth of the *driver* divided by the number of teeth of the *follower*.

- We *gear up* when we increase velocity and decrease torque. (V↑) (T↓)
  Ratio: 3:1

- We *gear down* when we increase torque and reduce velocity. (V↓) (T↑)
  Ratio: 1:3

3 to 1 ratio



$$Gear\ ratio = \frac{number\ of\ teeth\ of\ input\ gear}{number\ of\ teeth\ of\ ouput\ gear} = \frac{Input\ Torque}{Ouput\ Torque} = \frac{Output\ Speed}{Input\ Speed}$$

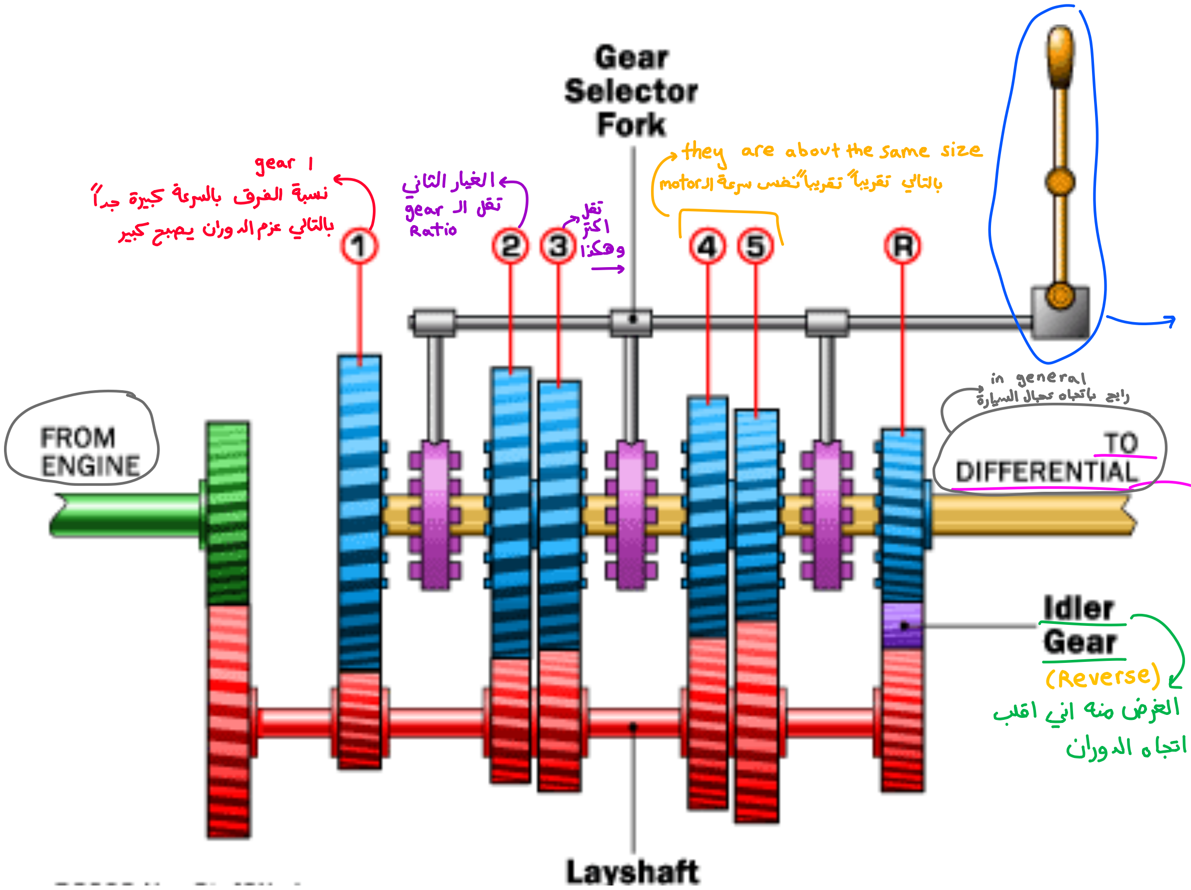
جاي من قانون
حفظ القدرة

اذا قلت السرعة يزداد الـ Torque

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example of Gear Trains

- A most commonly used example of gear trains is the gears of an automobile.



gear 1
نسبة الفرق بالسرعة كبيرة جداً
بالتالي عزم الدوران يصبح كبير

الغيار الثاني
نقل الـ gear
Ratio

نقل
اكثر
وهكذا

they are about the same size
بالتالي تقريباً تقريباً نفس سرعة الـmotor

من خلال هذه الذراع
نستطيع نقل الـ gear
من وضعية إلى وضعية أُخرى

in general
رابع باتجاه محال السيارة

شابك بين very small gear و very large gear
مما يؤدي انه السرعة تنزل كتير

(Reverse)
الغرض منه اني اقلب
اتجاه الدوران

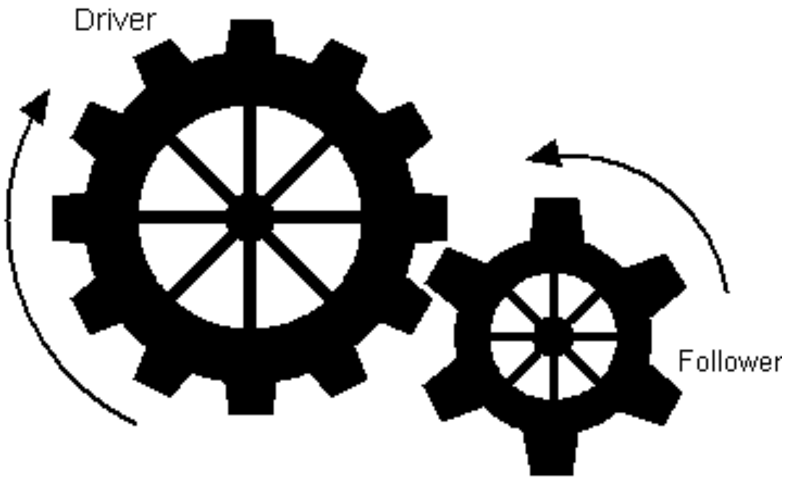
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Mathematical Modeling of Gear Trains

- Gears increase or descrease angular velocity (while simultaneously decreasing or increasing torque, such that energy is conserved).

Energy of Driving Gear = Energy of Following Gear

$$N_1\theta_1 \quad = \quad N_2\theta_2$$



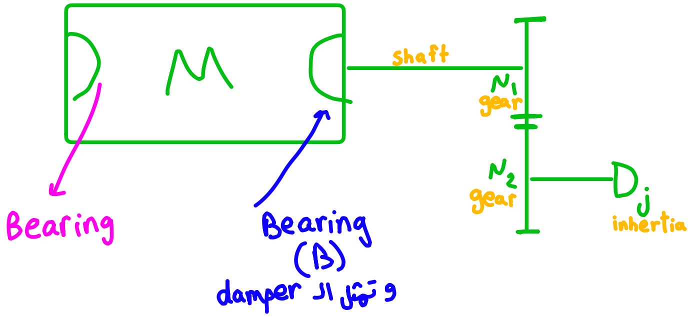
N_1 → Number of Teeth of Driving Gear

θ_1 → Angular Movement of Driving Gear

N_2 → Number of Teeth of Following Gear

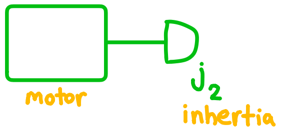
θ_2 → Angular Movement of Following Gear

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.



يكون في Bearing بالأول و بالآخر

the two systems يكافئو بعض



مبدأه أن الجزء الداخلي يلف و الجزء الخارجي ثابت

$$J_{eq} = \underbrace{J_1}_{\text{قبل الgear}} + \underbrace{\boxed{\left(\frac{N_1}{N_2}\right)^{2}} \underline{J_2}}_{\text{بعد ال gear}}$$

نقل Moment of inertia

بنسبة ال gear ratio تربيع

هنا عندي نسبة تؤثر على ال J_2

مثلاً لو $\frac{N_1}{N_2} = \frac{1}{3}$ ← بشوفها $\frac{1}{9}$ بسبب التربيع

$$B_{eq} = \underbrace{B_1}_{\text{قبل الgear}} + \underbrace{\left(\frac{N_1}{N_2}\right)^{2} B_2}_{\text{بعد ال gear}}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Mathematical Modelling of Gear Trains

- For three gears connected together

equivilant inhertia →

$$J_{eq} = \underbrace{J_1}_{\text{قبل ال gear الأول}} + \underbrace{\left(\frac{N_1}{N_2}\right)^2 J_2}_{\text{بين ال gear الأول وال gear الثاني}} + \underbrace{\left(\frac{N_1}{N_2}\right)^2 \left(\frac{N_3}{N_4}\right)^2 J_3}_{\text{بين ال gear الثاني وال gear الثالث}}$$

← مو متأكدة

$$B_{eq} = B_1 + \left(\frac{N_1}{N_2}\right)^2 B_2 + \left(\frac{N_1}{N_2}\right)^2 \left(\frac{N_3}{N_4}\right)^2 B_3$$

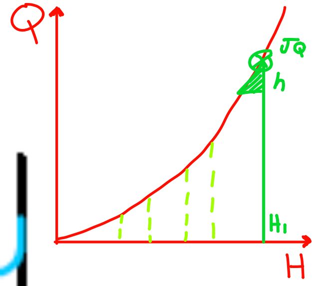
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Resistance of Liquid-Level Systems

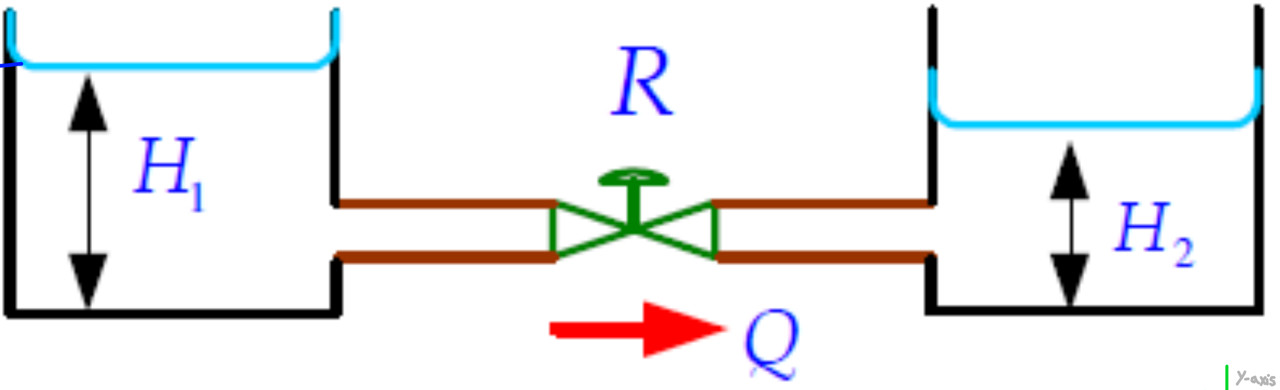
Mechanical Systems يشبه ال

it is extremely important
انه نحافظ على ارتفاع الخزان

- Consider the flow through a short pipe connecting two tanks as shown in Figure.



*العلاقة بين (H) و (Q)
is not Linear

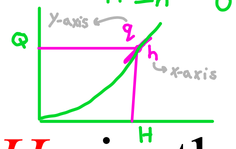


مثلاً انه لما يوصل level معين
تشتغل المضخة
و لما يوصل level معين تطفي

Q
التدفق

*to make Linearization

H ± h G ± q



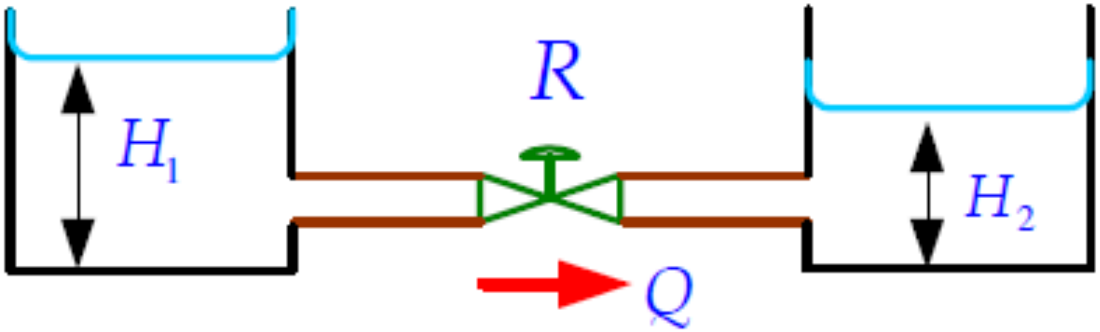
بفترض انه
علاقة ال curve
Linear

- Where H_1 is the height (or level) of first tank, H_2 is the height of second tank, R is the resistance in flow of liquid and Q is the flow rate.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Resistance of Liquid-Level Systems

- The resistance for liquid flow in such a pipe is defined as the change in the level difference necessary to cause a unit change inflow rate.



Voltage ال $\xleftarrow{\text{مقابل}}$ Position

current source $\xleftarrow{\text{مقابل ال}}$ Force

inductor $\xleftarrow{\text{مقابل ال}}$ damper

group of capacitors $\xleftarrow{\text{مقابل ال}}$ mass

$$Resistance = \frac{change\ in\ level\ difference}{change\ in\ flow\ rate} = \frac{m}{m^3 / s}$$

$$R = \frac{\Delta(H_1 - H_2)}{\Delta Q} = \frac{m}{m^3 / s}$$

$\Delta H_1 = h_1$

$\Delta H_2 = h_2$

$\Delta Q = q$

$$= \boxed{\frac{h_1 - h_2}{q}}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Resistance in Laminar Flow

- For laminar flow, the relationship between the steady-state flow rate and steady state height at the restriction is given by:

$$Q = k_l H$$

- Where Q = steady-state liquid flow rate in m/s^3
- K_l = *constant in m/s^2*
- *and H = steady-state* height in *m*.

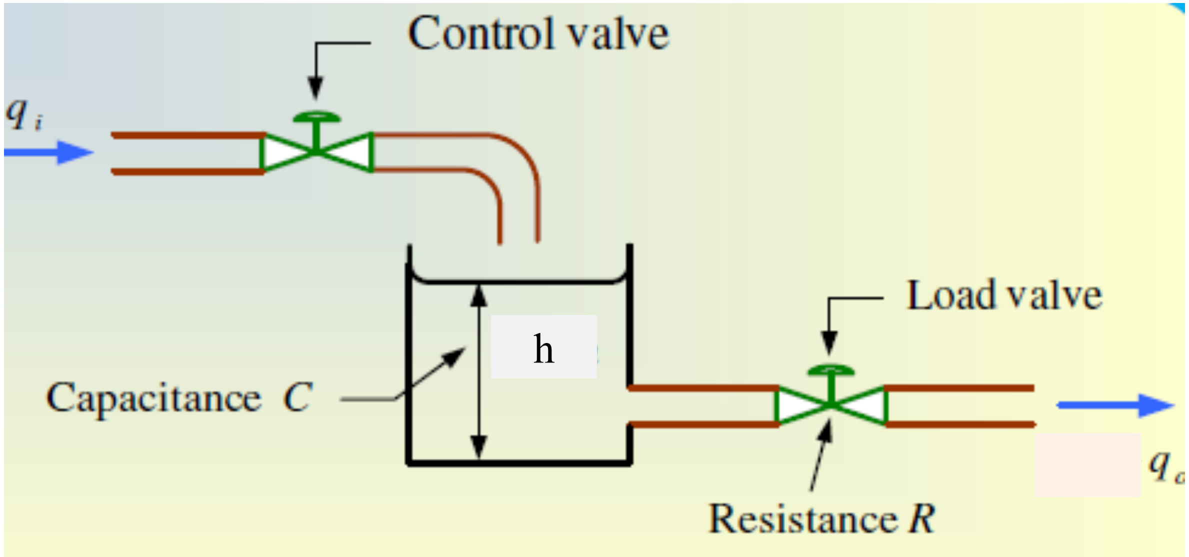
- The resistance R_ℓ is

$$R_l = \frac{dH}{dQ} = \frac{\Delta h}{q}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Capacitance of Liquid-Level Systems

- The capacitance of a tank is defined to be the change in quantity of stored liquid necessary to cause a unity change in the height.

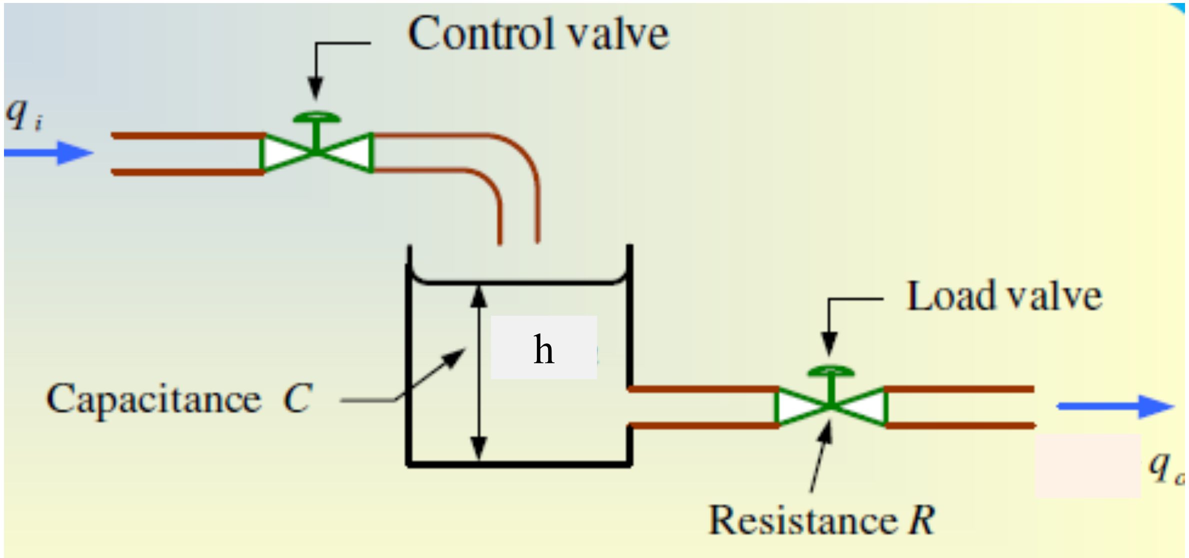


$$Capacitance = \frac{change\ in\ liquid\ \ stored}{change\ in\ height} = \frac{m^3}{m}\ or\ m^2$$

- Capacitance (C) is cross sectional area (A) of the tank.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Capacitance of Liquid-Level Systems



$$Rate\ of\ \ change\ of\ \ fluid\ \ volume\ in\ the\ tank\ =\ flow\ in-\ flow\ out$$

التغير في حجم الماء مثلاً في الخزان

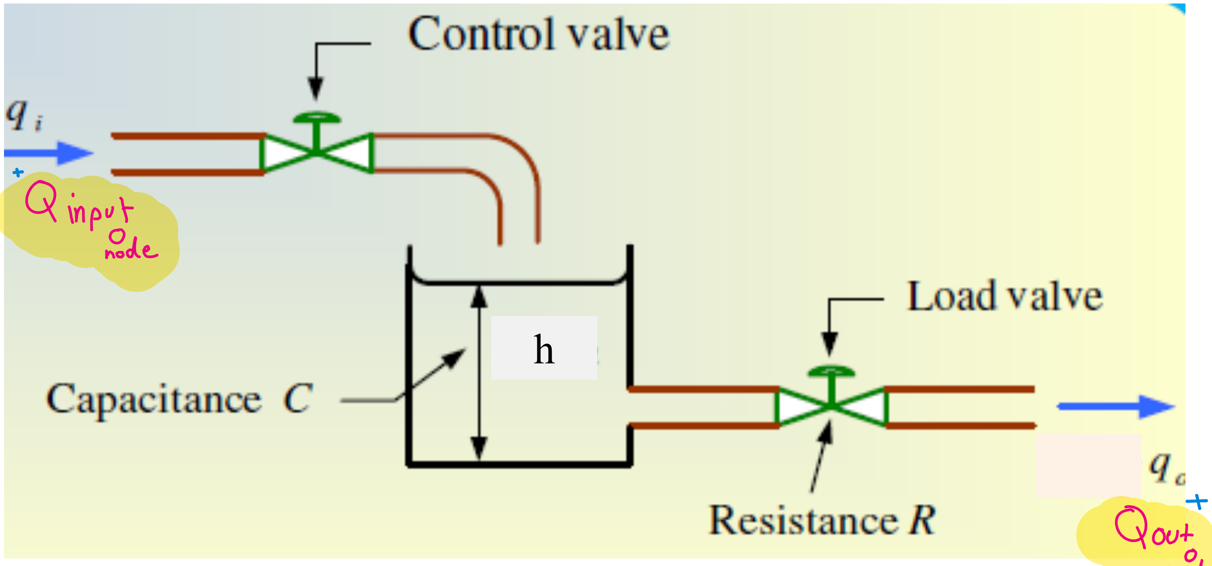
هو الفرق بين ال input و ال output

$$\frac{dV}{dt} = q_i - q_o$$

$$\frac{d(A \times h)}{dt} = q_i - q_o$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Capacitance of Liquid-Level Systems



مثلاً
0.25 m^3/min

Q input
o node

Qout
o node

مثلاً
0.25 m^3/min

$$A\frac{dh}{dt} = q_i - q_o$$

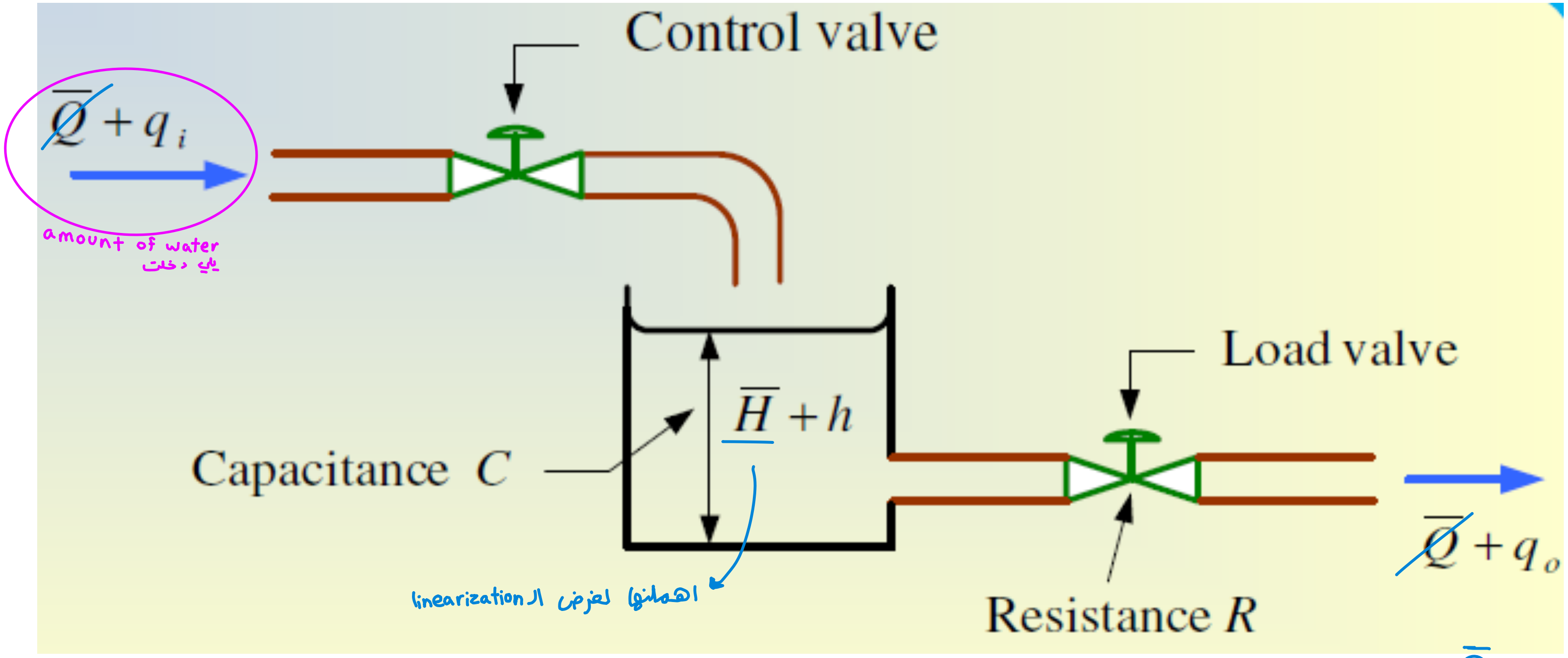
* لكي يبقى ال system stable يجب أن يكون

Oin_o و $Qout_o$ نفس القيمة

$$C\frac{dh}{dt} = q_i - q_o$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Modelling Example#1



هنا لغينا ال $\bar{Q}$

$\bar{H}$ = steady-state head (before any change has occurred), m.

h = small deviation of head from its steady-state value, m.

$\bar{Q}$ = steady-state flow rate (before any change has occurred), m^3/s.

q_i = small deviation of inflow rate from its steady-state value, m^3/s.

q_o = small deviation of outflow rate from its steady-state value, m^3/s.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Modelling Example#1

- The rate of change in liquid stored in the tank is equal to the flow in minus flow out.

تشبه معادلة ←
ال capacitor

$$C\frac{dh}{dt} = q_i - q_o \qquad \longrightarrow \qquad (1)$$

- The resistance *R* may be written as

تشبه معادلة ←
المقاومة

$$R = \frac{dH}{dQ} = \frac{h}{q_0} \qquad \longrightarrow \qquad (2)$$

- Rearranging equation (2)

$$q_0 = \frac{h}{R} \qquad \longrightarrow \qquad (3)$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Modelling Example#1

$$C\frac{dh}{dt} = q_i - q_o \longrightarrow (1) \qquad q_0 = \frac{h}{R} \longrightarrow (4)$$

- Substitute q_o in equation (3)

$$C\frac{dh}{dt} = q_i - \frac{h}{R}$$

- After simplifying above equation

$$\boxed{RC\frac{dh}{dt} + h = Rq_i}$$

- Taking Laplace transform considering initial conditions to zero

$$\boxed{RCsH(s) + H(s) = RQ_i(s)}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Modelling Example#1

$$RCsH(s) + H(s) = RQ_i(s)$$

- The transfer function can be obtained as

$$\boxed{\frac{H(s)}{Q_i(s)} = \frac{R}{(RCs + 1)}}$$

in the final value theorem → s = 0

$$\frac{H(s)}{Q_i(s)} = \frac{R}{(R\cancel{C}s + 1)} \longrightarrow \frac{H}{Q_i} = R$$

© 2001 by Prentice Hall, Upper Saddle River, NJ.

Example 3.7. A water heater.

The inflow of water to the water heater has the mass flow rate $\dot{m}_1$ and temperature T_1 whereas the outflow has the mass flow rate $\dot{m}_2$ and temperature T_2. The mass of water in the heater is M and it is heated to a temperature T with a heating power $\dot{Q}$. The mixing of water in the heater is assumed to be perfect.

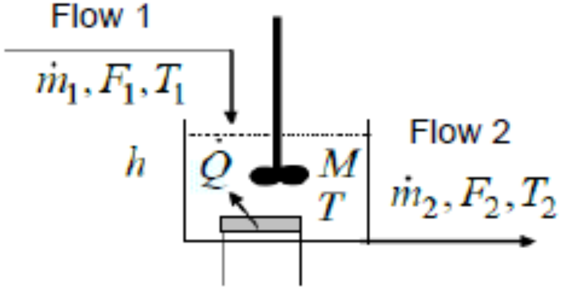


Fig. 3.8. A water heater.

How do the amount of water and the temperature in the heater depend on other variables?

Mass balance:

$$\frac{\mathrm{d}M}{\mathrm{d}t} = \dot{m}_1 - \dot{m}_2 \tag{1}$$

Energy balance:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \dot{E}_1 - \dot{E}_2 + \dot{Q} \tag{2}$$

Here, $\dot{E}_1$ and $\dot{E}_2$ are energy flows associated with the inflow and the outflow, respectively.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

The energy in a substance is proportional to its mass or mass flow rate. For liquids it applies with good accuracy that the energy is also proportional to its temperature. This results in the

constitutive relationships: $E = c_\mathrm{p} T M,\ \dot{E}_1 = c_\mathrm{p} T_1 \dot{m}_1,\ \dot{E}_2 = c_\mathrm{p} T_2 \dot{m}_2$ (3)

Here c_p is the *specific heat capacity* for water, which in this case is assumed to be constant independently of the water temperature. Combination of (2) and (3) and development of the derivative according to the product rule give

$$T \frac{\mathrm{d}M}{\mathrm{d}t} + M \frac{\mathrm{d}T}{\mathrm{d}t} = T_1 \dot{m}_1 - T_2 \dot{m}_2 + \frac{\dot{Q}}{c_\mathrm{p}} \tag{4}$$

Because of the assumption of perfect mixing, there is also a

constitutive relationship: $T_2 = T$ (5)

Elimination of $\mathrm{d}M/\mathrm{d}t$ from (4) by (1) and substitution of (5) give

$$M \frac{\mathrm{d}T}{\mathrm{d}t} = \dot{m}_1 (T_1 - T) + \frac{\dot{Q}}{c_\mathrm{p}} \tag{6}$$

Equation (1) and (6) show how the mass and the temperature in the heater depend on the inflow and the heating power $\dot{Q}$.

© 2001 by Prentice Hall, Upper Saddle River, NJ.

The energy in a substance is proportional to its mass or mass flow rate. For liquids it applies with good accuracy that the energy is also proportional to its temperature. This results in the

constitutive relationships: $E = c_\mathrm{p} T M,\ \dot{E}_1 = c_\mathrm{p} T_1 \dot{m}_1,\ \dot{E}_2 = c_\mathrm{p} T_2 \dot{m}_2$ (3)

Here c_p is the *specific heat capacity* for water, which in this case is assumed to be constant independently of the water temperature. Combination of (2) and (3) and development of the derivative according to the product rule give

$$T\frac{\mathrm{d}M}{\mathrm{d}t} + M\frac{\mathrm{d}T}{\mathrm{d}t} = T_1\dot{m}_1 - T_2\dot{m}_2 + \frac{\dot{Q}}{c_\mathrm{p}} \qquad (4)$$

Because of the assumption of perfect mixing, there is also a

constitutive relationship: $T_2 = T$ (5)

Elimination of $\mathrm{d}M/\mathrm{d}t$ from (4) by (1) and substitution of (5) give

$$M\frac{\mathrm{d}T}{\mathrm{d}t} = \dot{m}_1(T_1 - T) + \frac{\dot{Q}}{c_\mathrm{p}} \qquad (6)$$

Equation (1) and (6) show how the mass and the temperature in the heater depend on the inflow and the heating power $\dot{Q}$.

© 2001 by Prentice Hall, Upper Saddle River, NJ.

If we want to use volumetric units instead of mass units in the model, this can easily be accomplished by the substitutions

$$M = \rho A h, \quad \dot{m}_1 = \rho_1 F_1 \tag{7}$$

which applied to (6) yield

$$\rho A h \frac{\mathrm{d}T}{\mathrm{d}t} = \rho_1 F_1 (T_1 - T) + \frac{\dot{Q}}{c_\mathrm{p}} \tag{8}$$

Note that the *water density is not assumed to be constant* in equation (8).

Equation (1) expressed in volumetric units becomes more complicated when the water density is non-constant., i.e.,

$$A \frac{\mathrm{d}\rho h}{\mathrm{d}t} = \rho_1 F_1 - \rho_2 F_2 = \rho_1 F_1 - \rho F_2 \tag{9}$$

It is possible to show that even if $\rho \neq \rho_1$ due to the fact that $T \neq T_1$, the effects tend to cancel out in such a way that

$$A \frac{\mathrm{d}h}{\mathrm{d}t} \approx F_1 - F_2 \tag{10}$$

becomes a good approximation of (1) and (9).

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Electrical

mechanical

Electromechanical Systems

مادة فاينل
وليس ميد !!

- **Electromechanics** combines electrical and mechanical processes.
- Devices which carry out electrical operations by using moving parts are known as electromechanical.
  - Relays
  - Solenoids
  - Electric Motors
  - Switches and e.t.c

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

D.C Drives

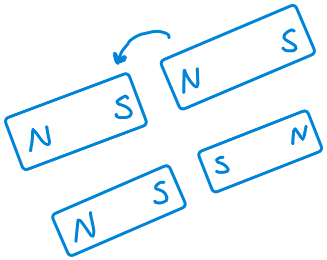
(T) Torque = constant x current (I)

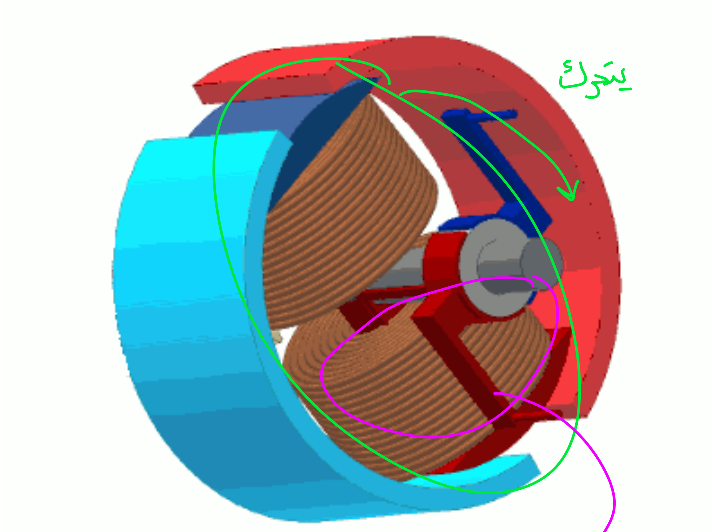
e_B = constant x omega

(T) و (I) علاقة طردية

كلما زادت (e_B)
تزداد (w)
(طردي)

- Speed control can be achieved using DC drives in a number of ways.
- Variable Voltage can be applied to the armature terminals of the DC motor .
- Another method is to vary the flux per pole of the motor.
- The first method involve adjusting the motor’s armature while the latter method involves adjusting the motor field. These methods are referred to as “armature control” and “field control.”





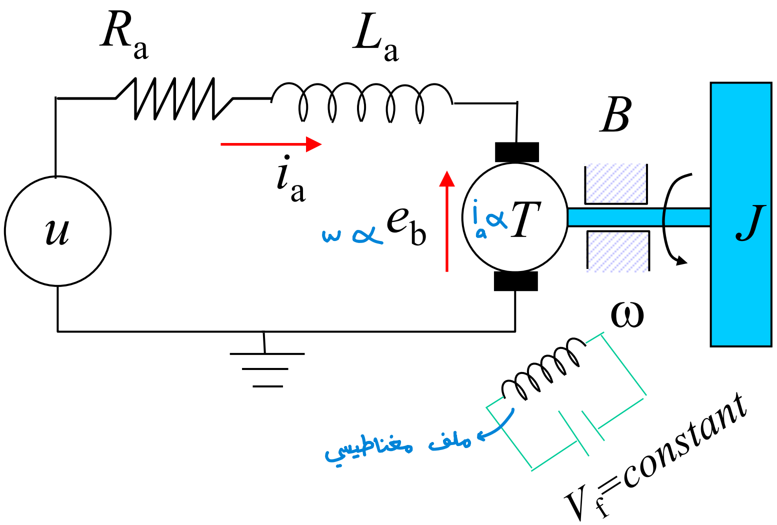
مرة بتكون موصولة مع ال N ومرة مع ال S وهكذا

* المبدأ لـ DC motor هو قلب القطبية

* قطر ال shaft هو الذي يتحكم .....

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-2: Armature Controlled D.C Motor



Input: voltage u

Output: Angular velocity ω

Electrical Subsystem (loop method):

$$u = R_a i_a + L_a \frac{di_a}{dt} + \underline{\underline{e_b}}, \qquad where \quad e_b = back\text{-}emf\ voltage$$

constant x ω omega

* كل مقاومة سوف تساهم

Mechanical Subsystem

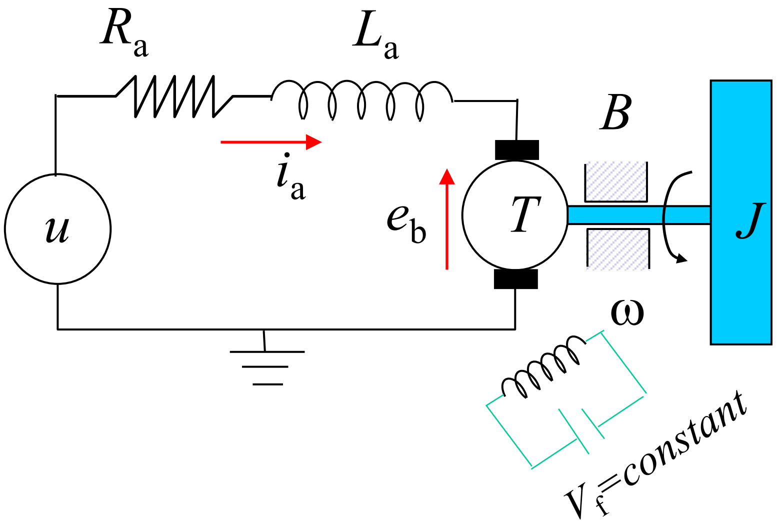
$$T_{motor} = J\dot{\omega} + B\omega$$

$$\dot{\Theta} = \omega$$

omega

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-2: Armature Controlled D.C Motor



Power Transformation:

Torque-Current: $T_{motor} = K_t i_a$

Voltage-Speed: $e_b = K_b \omega$

where K_t: torque constant, K_b: velocity constant For an ideal motor

$$K_t = K_b$$

Combing previous equations results in the following mathematical model:

$$\begin{cases} L_a \dfrac{di_a}{dt} + R_a i_a + K_b \omega = u \\ J\dot{\omega} + B\omega - K_t i_a = 0 \end{cases}$$

$\ddot{\theta} = \dot{\omega}$

acceleration

اشتقاق السرعة

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-2: Armature Controlled D.C Motor

Taking Laplace transform of the system's differential equations with zero initial conditions gives:

$$\begin{cases} \left(L_a s + R_a\right) I_a(s) + K_b \Omega(s) = U(s) \\ \left(Js + B\right) \Omega(s) - K_t I_a(s) = 0 \end{cases}$$

Eliminating I_a yields the input-output transfer function

$$\frac{\Omega(s)}{U(s)} = \frac{K_t}{L_a J s^2 + \left(J R_a + B L_a\right) s + B R_a + K_t K_b}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-2: Armature Controlled D.C Motor Reduced Order Model

S=0 ← عند ال Steady State

Assuming small inductance, $L_a \approx 0$

$$\frac{\Omega(s)}{U(s)} = \frac{\left(K_t / R_a\right)}{\underbrace{Js} + \left(B + K_t K_b / R_a\right)}$$

لها علاقة بال timing

(كلما زادت ال inertia J زاد الوقت الازم للوصول لل steady state)

* final value theorem

$S = 0$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Block Diagram
+
Mason’s Rule

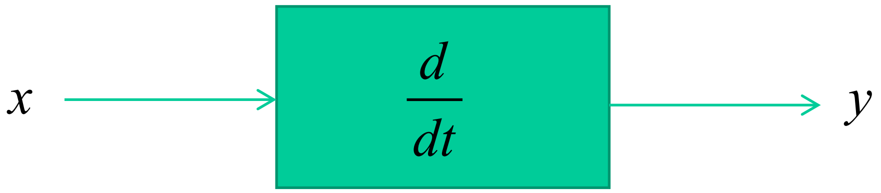
ال Modelling ينتهي ب transfer function

Introduction

- A Block Diagram is a shorthand pictorial representation of the cause-and-effect relationship of a system.

- The interior of the rectangle representing the block usually contains a description of or the name of the element, gain, or the symbol for the mathematical operation to be performed on the input to yield the output.

- The arrows represent the direction of information or signal flow.

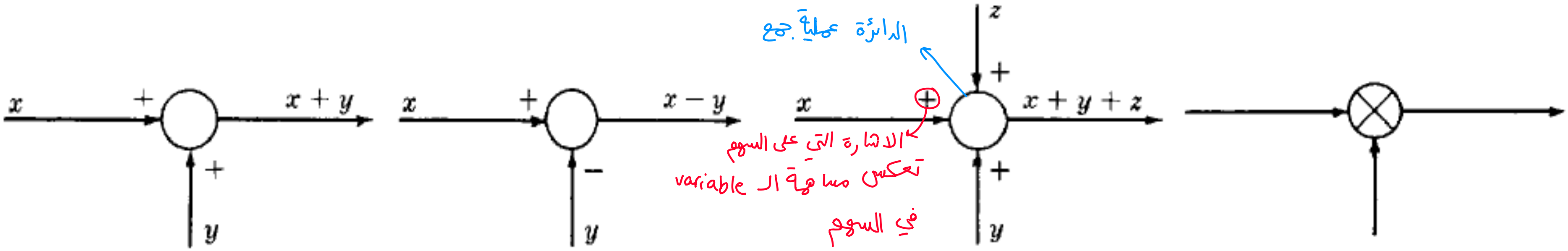


Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

تمثل الدائرة عملية جمع

Introduction

- The operations of addition and subtraction have a special representation.
- The block becomes a small circle, called a summing point, with the appropriate plus or minus sign associated with the arrows entering the circle.
- The output is the algebraic sum of the inputs.
- Any number of inputs may enter a summing point.
- Some books put a cross in the circle.

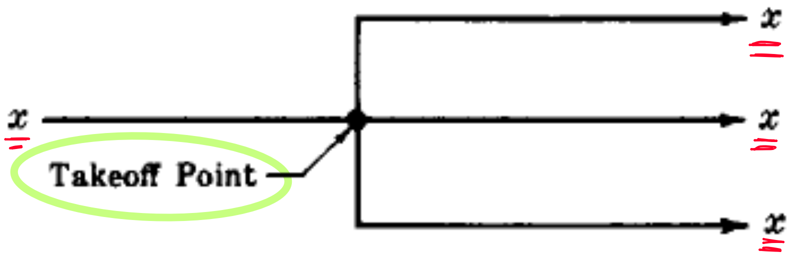


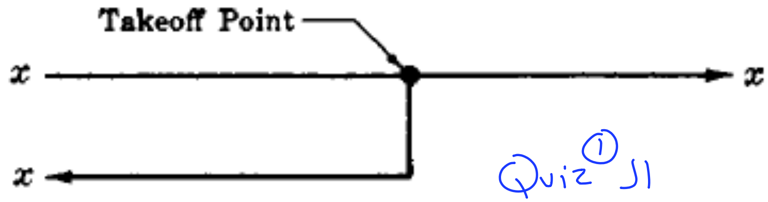
الاشارة التي على السهم تعكس مساهمة الـ variable في السهم

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Introduction

- In order to have the same signal or variable be an input to more than one block or summing point, a takeoff (or pickoff) point is used.
- This permits the signal to proceed unaltered along several different paths to several destinations.





ال Quiz ①

Find L^{-1}

$$\boxed{\frac{1}{s(s+1)}}$$

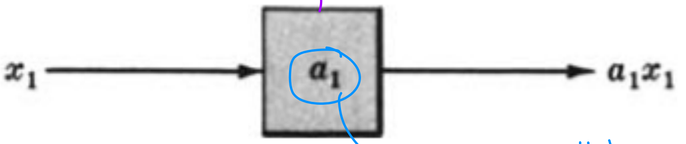
تتأثر قيمة السهم اذا ضربته باشي أو جمعته بأشي

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-1

- Consider the following equations in which x_1, x_2, x_3, are variables, and a_1, a_2 *are* general coefficients or mathematical operators.

$$x_3 = a_1x_1 + a_2x_2 - 5$$



يمثل المربع عملية ضرب

يلي جوا المربع هو Transfer function

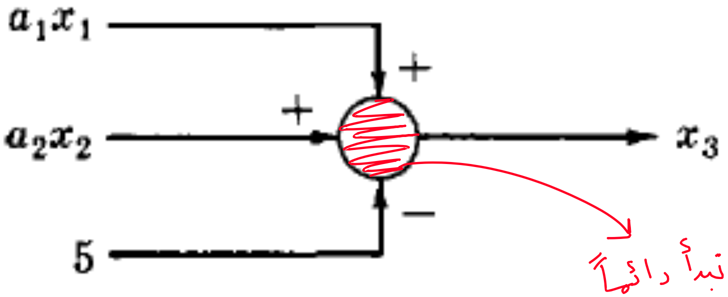
(For simplification Purpose)

مثلاً بتكون $a_1 =$

$$\frac{X(s)}{F(s)} = \frac{1}{Ms^2 + Cs + k}$$

$$X = \frac{1}{Ms^2 + Cs + K} \times F$$

$$X = a_1 \times F$$

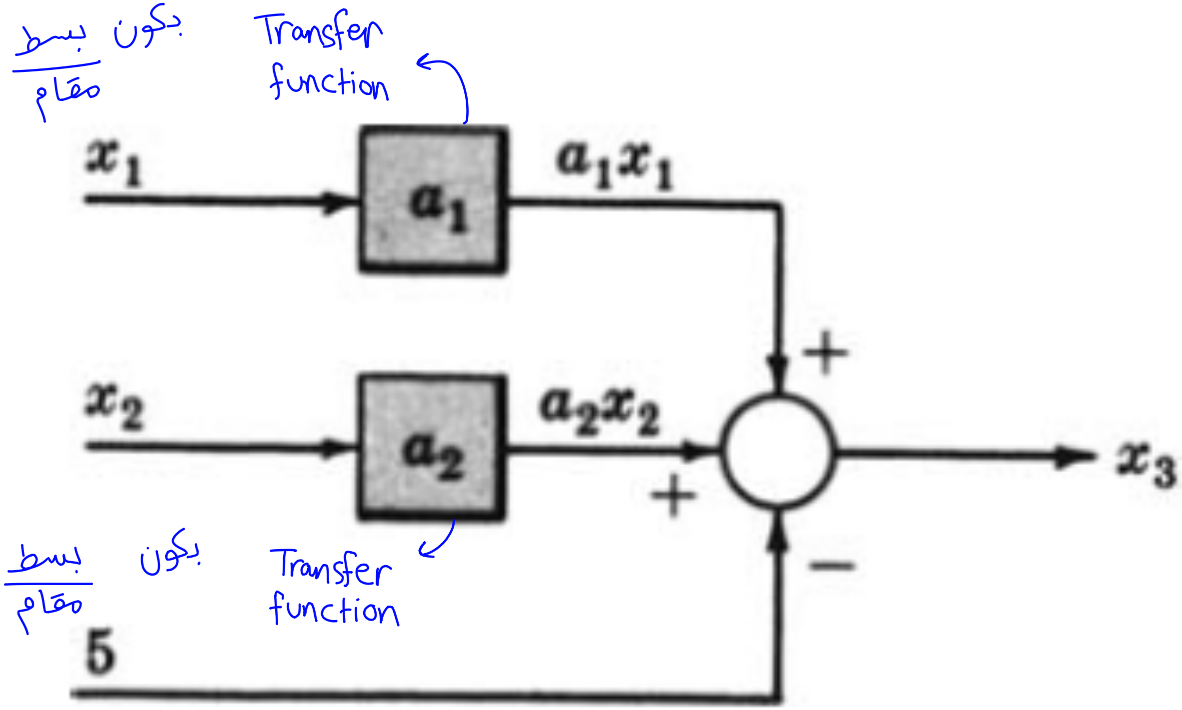


لتمثيل المعادلة نبدأ دائماً بعملية الجمع

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-1

$$x_3 = a_1 x_1 + a_2 x_2 - 5$$



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

$\int \rightarrow \frac{1}{s}$

تكامل

$\frac{dx}{dt} \rightarrow s$

الاشتقاق

ال Control متعلق بالتحكم ب one item only

اما ال Automation متعلق بالنظام كله

Example-2

مبني على ال transfer function

- Draw the Block Diagrams of the following equations.

التحويل إلى صورة ال Laplace →

اشتقاق تكامل

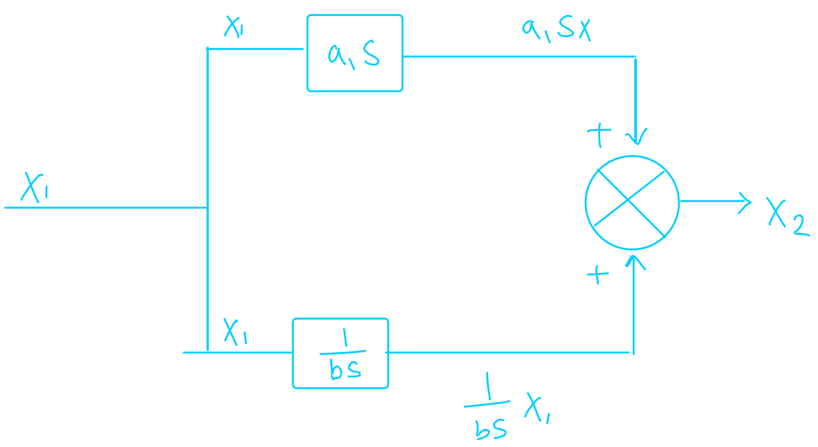
$a_1 \, S \, X_1 + \frac{1}{bS} X_1$

(1) $$x_2 = a_1 \frac{dx_1}{dt} + \frac{1}{b} \int x_1 dt$$

$a_1 s X_1 + \frac{1}{bs} X_1$

(2) $$x_3 = a_1 \frac{d^2 x_2}{dt^2} + 3 \frac{dx_1}{dt} - bx_1$$

حل فرع ①



Canonical Form of A Feedback Control System

Tuesday 24/10

Regulation تشمل الـ

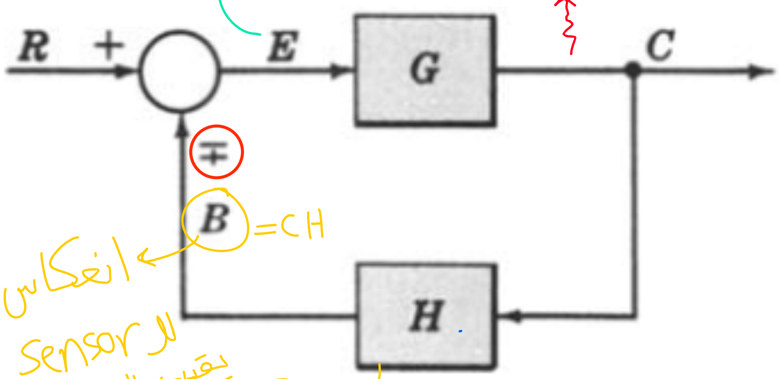
$E = R \mp CH$

tuning $E\,K$ = power

Error = $R - B$

Required − Actual

المجاهيل الوحيدة هم R و C



B = CH

انعكاس للـ sensor

يقيس الحرارة أو أي اشي

مربع ← عملية الضرب

دائرة ← عملية الجمع

$B = CH$

$(R \mp CH) \times G = C$

الاشتقاق غير مطلوب

الان أي اشي مضروب بال C حاطه على جهة C

وأي اشي مضروب بال R حاطه على جهة R

$GR \mp CHG = C$

انقلب الاشارة

$GR = C \pm CHG$

$GR = C(1 \pm GH)$

$$\frac{C}{R} = \frac{G}{(1 \pm GH)}$$

$G \equiv$ direct transfer function $\equiv$ forward transfer function

$H \equiv$ feedback transfer function

هذه الدائرة مثلاً تعمل عمل العوامة تبعت الخزان

$GH \equiv$ loop transfer function $\equiv$ open-loop transfer function

$C/R \equiv$ closed-loop transfer function $\equiv$ control ratio

$$\frac{C}{R} = \frac{G}{1 \pm GH}$$

EG

عكس الاشارة التي في الدائرة

$E/R \equiv$ actuating signal ratio $\equiv$ error ratio

$$\frac{E}{R} = \frac{1}{1 \pm GH}$$

$B/R \equiv$ primary feedback ratio

$$\frac{B}{R} = \frac{GH}{1 \pm GH}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Characteristic Equation

- The control ratio is the closed loop transfer function of the system.

ال feed back loop غيرت بال characteristic equation

اذا غيرت معادلة المقام

وبالتالي غيرت مواقع جذور ال equation

و غيرت بتصرفات النظام

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 \pm G(s)H(s)}$$

عكس اشارة السهم

المقام

- The denominator of closed loop transfer function determines the characteristic equation of the system.

- Which is usually determined as:

هذه المعادلة هي أساس عملية التصميم في ال control

وبلاقي منها جذور الاقتران و كل جذر من الجذور الو معنى

$$1 \pm G(s)H(s) = 0$$

اذا اجى على ال imaginary axis ← Sinusoidal

اذا الجذور على ال real axis ← exponential

اذا الجذر مكرر على ال real axis ← t x exponential

بالعادة معادلة المقام Loop بكون فيها K

tuning parameter

ولما اغيره هذا يؤثر على مواقع جذور الاقتران

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Block Diagram Reduction

Example-3

مرةً السؤال بكون

find the equavilant transfer function

$= \frac{C}{R} =$

1. Open loop transfer function

$$\frac{B(s)}{E(s)} = G(s)H(s)$$

2. Feed Forward Transfer function

$$\frac{C(s)}{E(s)} = G(s)$$

3. control ratio

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)}$$

4. feedback ratio

$$\frac{B(s)}{R(s)} = \frac{G(s)H(s)}{1+G(s)H(s)}$$

5. error ratio

$$\frac{E(s)}{R(s)} = \frac{1}{1+G(s)H(s)}$$

6. closed loop transfer function

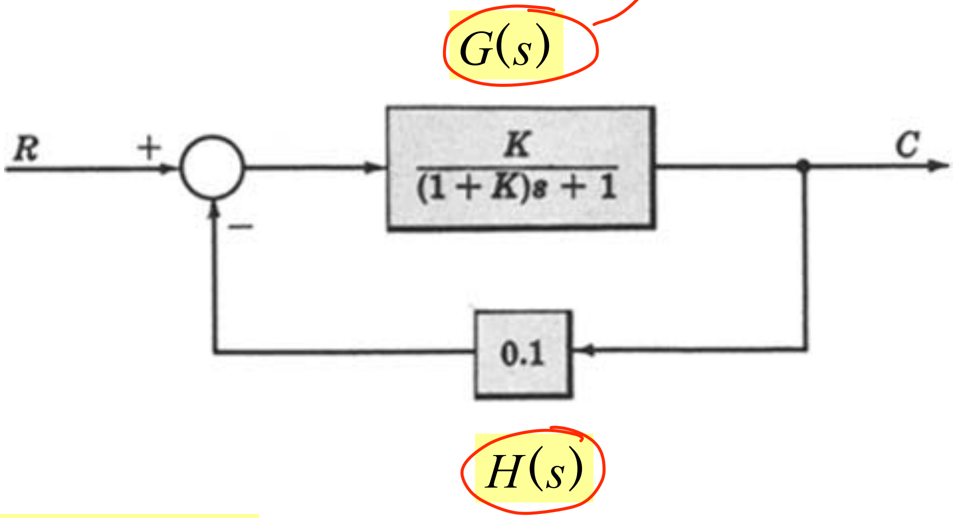
$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)}$$

7. characteristic equation

$$1+G(s)H(s) = 0$$

8. Open loop poles and zeros if 9. closed loop poles and zeros if K=10.

بحط الرموز تبعي وبعوض عالقانون ثم نبسط



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

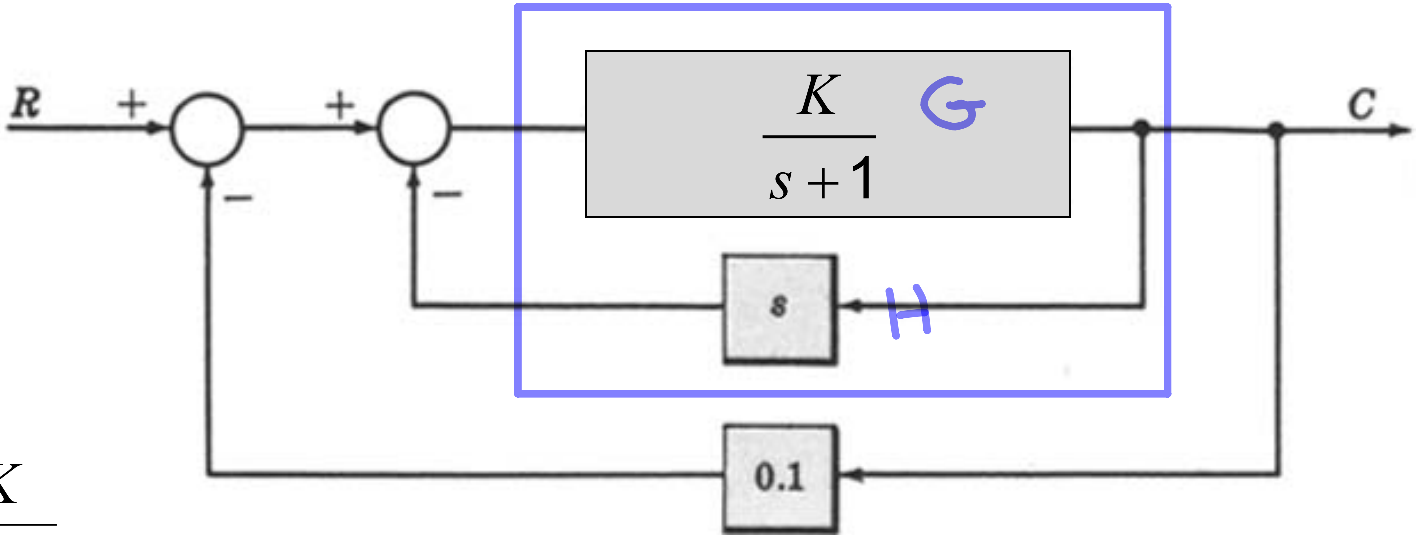
Example-5

$$\frac{C}{R}=\frac{G}{1+GH}=\frac{\frac{K}{s+1}}{1+\frac{Ks}{s+1}}$$

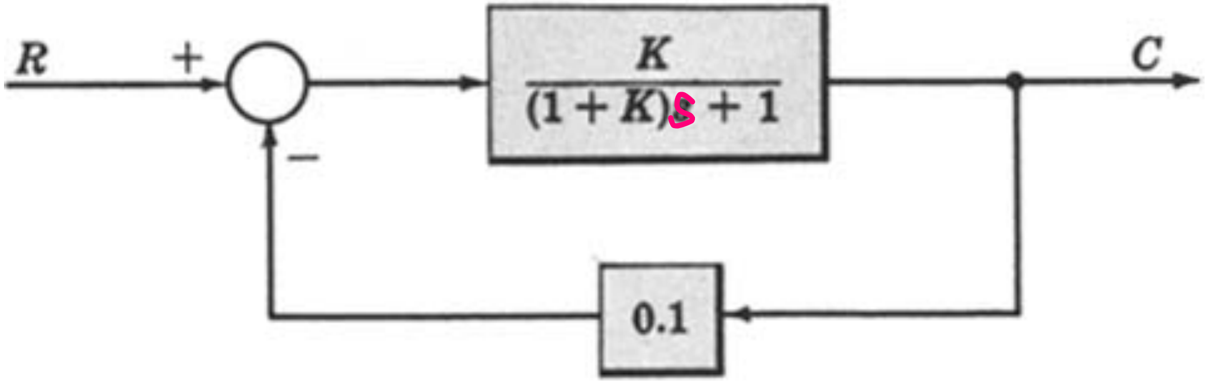
X ضرب تبادلي

$$=\frac{\frac{K}{\cancel{s+1}}}{\frac{s+1+Ks}{\cancel{s+1}}}$$

$$=\frac{K}{(1+K)s+1}$$



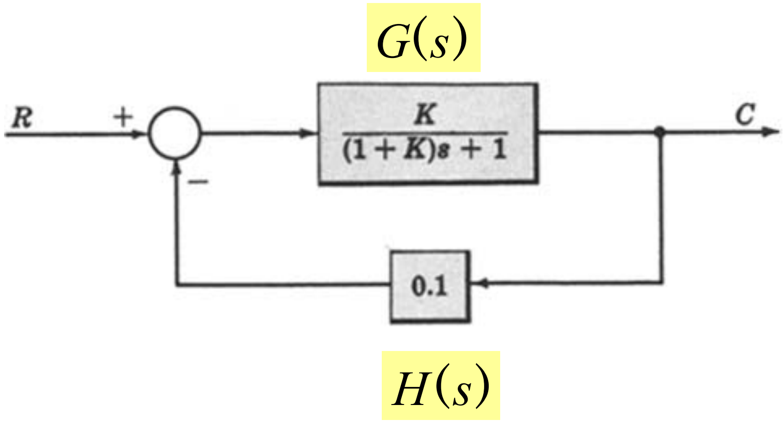
$$\frac{G}{1+GH}=\frac{\frac{K}{s+1}}{1+\frac{K}{s+1}s}$$



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-5 (see example-3)

1. Open loop transfer function $\frac{B(s)}{E(s)} = G(s)H(s)$
2. Feed Forward Transfer function $\frac{C(s)}{E(s)} = G(s)$
3. control ratio $\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)}$
4. feedback ratio $\frac{B(s)}{R(s)} = \frac{G(s)H(s)}{1+G(s)H(s)}$
5. error ratio $\frac{E(s)}{R(s)} = \frac{1}{1+G(s)H(s)}$
6. closed loop transfer function $\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)}$
7. characteristic equation $1+G(s)H(s)=0$
8. closed loop poles and zeros if K=10.



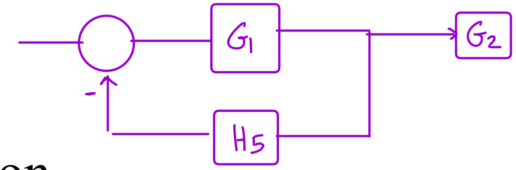
$$\frac{C}{R} = \frac{\frac{K}{(1+K)s+1}}{1+\frac{0.1K}{(1+K)s+1}}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

I can solve it on Masons Rule

Example-6

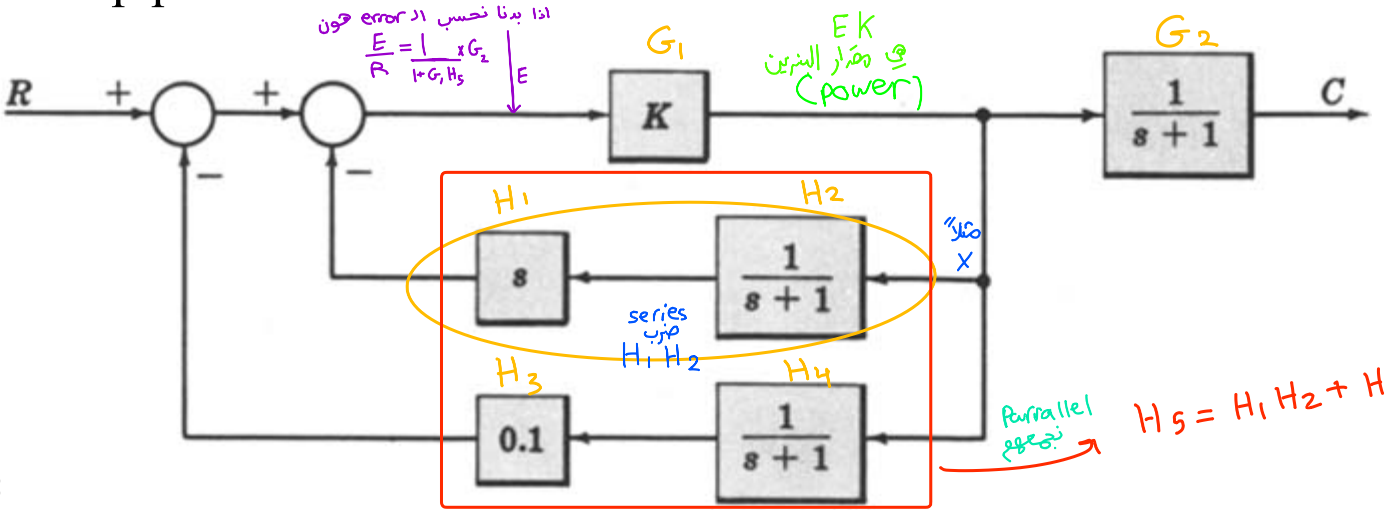
- For the system represented by the following block diagram determine:
  1. Open loop transfer function
  2. Feed Forward Transfer function
  3. control ratio
  4. feedback ratio
  5. error ratio
  6. closed loop transfer function
  7. characteristic equation
  8. closed loop poles and zeros if K=100.



series → ضرب

parrallel → جمع.

$$\frac{C}{R} = \frac{G_1}{1+G_1H_5} = \frac{K}{1+K(H_1H_2+H_3H_4)}$$



$$H_5 = H_1H_2 + H_3H_4$$

Illustrations © 2001 by Prentice

Example-6

- For the system represented by the following block diagram determine:
  1. Open loop transfer function
  2. Feed Forward Transfer function
  3. control ratio
  4. feedback ratio
  5. error ratio
  6. closed loop transfer function
  7. characteristic equation
  8. closed loop poles and zeros if K=100.

Solve by Masons Rule :-

$$\frac{C}{R} = \frac{\sum \text{Paths} \times \Delta_i}{\underset{\text{الكلية}}{\Delta}}$$

*ممنوع نمر بنفس الممر مرتين

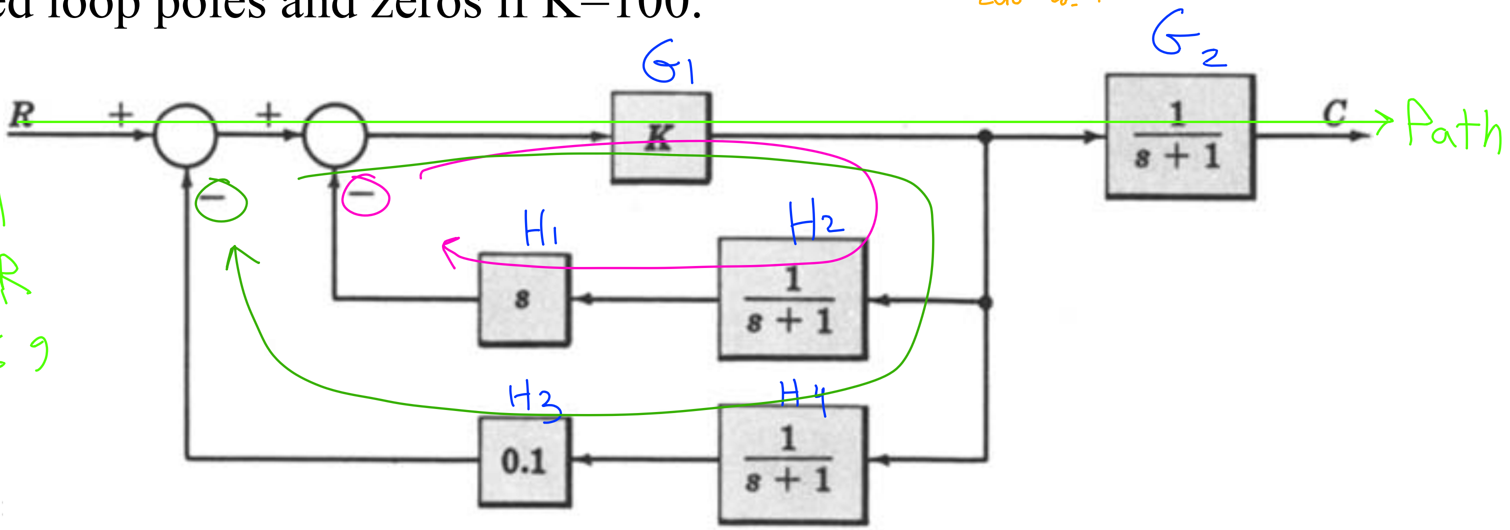
$$\frac{C}{R} = \frac{G_1 \; G_2 \; (1+0+0)}{1-(\overset{\ominus}{G_1} H_1 H_2 + \overset{\ominus}{G_1} H_3 H_4) + (0)}$$

لأن G1 مشترك

مجموع ال loops

مجموع مضروب ال 2 loops لكن بشرط ما يكون في اشي مشترك (اذا كان في مشترك بفرض القيمة صفر)

عندي G1 مشترك و ما بقدر اضربهم ببعض اذاً القيمة zero



ال Path من R الى C و كلشي بطريقي باخذه

Illustrations © 2001 by Prentice

(by Masin Rule) → أسهل

$$\frac{C}{R} = \frac{G_1G_2G_3\,(1-0-0)}{1-(\oplus G_1G_2H_1 - G_1G_2G_3 - G_2G_3H_2 + 0}$$

Loop (1) — loop (2)

ضربنا بناقص واحد

+ مجموع مضروب الـ 2loops بشرط إذا مافي ولا item مشترك

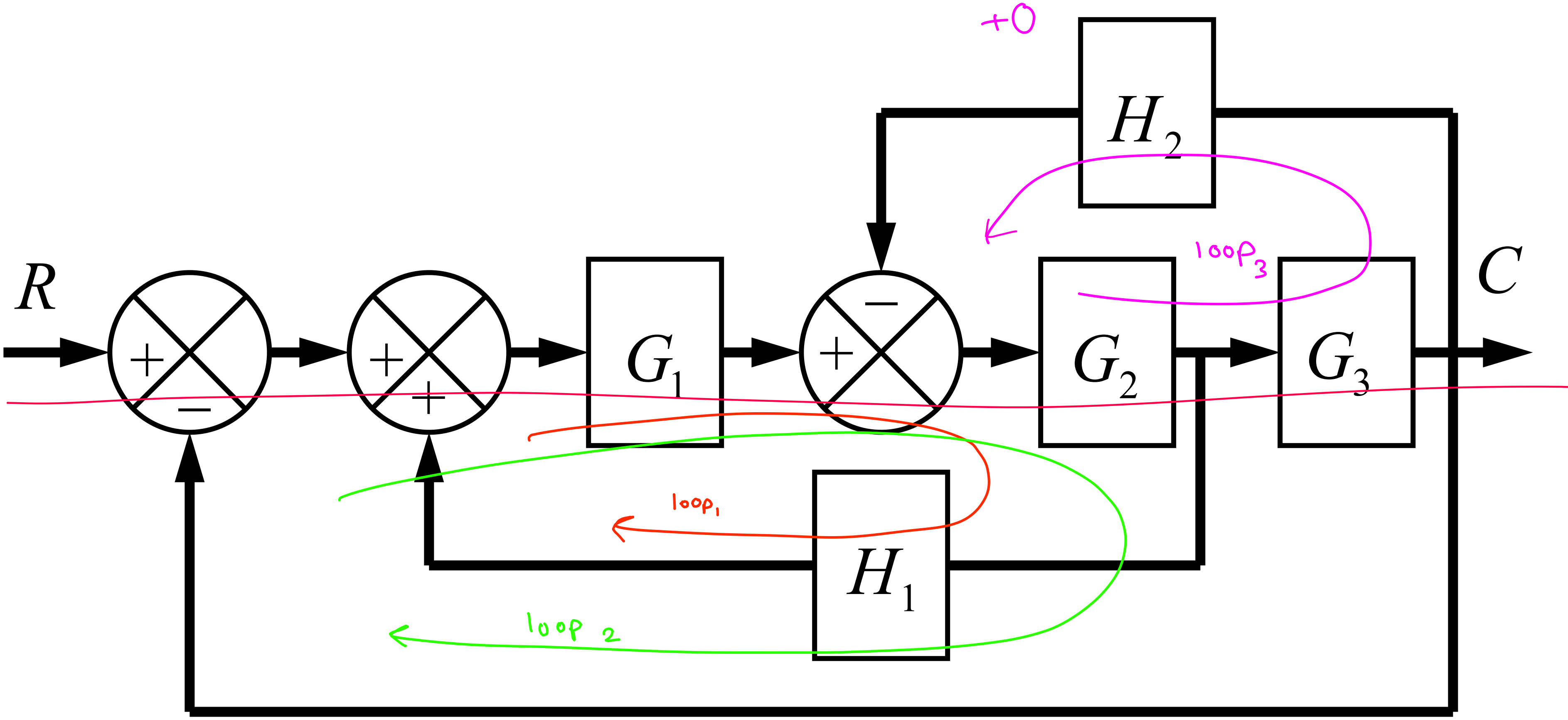
Example-7

$$\frac{C}{R} = \frac{\sum \text{Paths} \times \Delta_i}{\Delta}$$

الكلية

- Reduce the following block diagram to canonical form.

*ممنوع نمر بنفس الممر مرتين



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-7

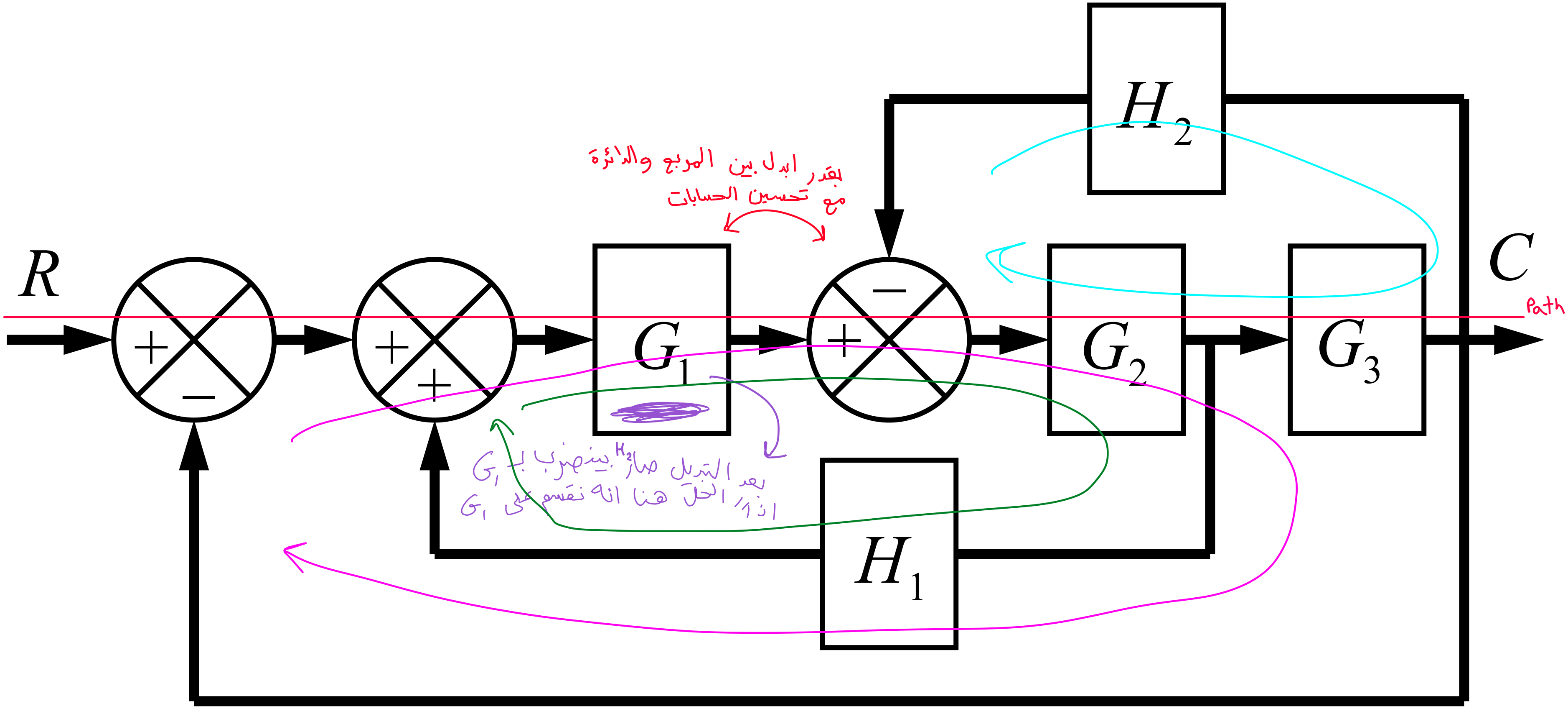
*by Mason's Rule

$$\frac{C}{R} = \frac{G_1 G_2 G_3 \; (1-0-0-0)}{1-(G_1 G_2 H_1 \overset{\ominus}{+} G_1 G_2 G_3 \overset{\ominus}{+} G_2 G_3 H_2)+(0)}$$

كلهم مشتركين بـ G_2

you can exchage the
circle and square
by modifying the calculations

- Reduce the following block diagram to canonical form.



بقدر ابدل بين المربع والدائرة
مع تحسين الحسابات

بعد التبديل صار H_2 بينضرب بـ G_1
اذاً الحل هنا انه نقسم على G_1

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-7

*by block diagram

بدور على

Series

Parrallel

loop

ممنوع ابدل ال takeoff Point مع الدائرة

لأنه يخالف المعادلات

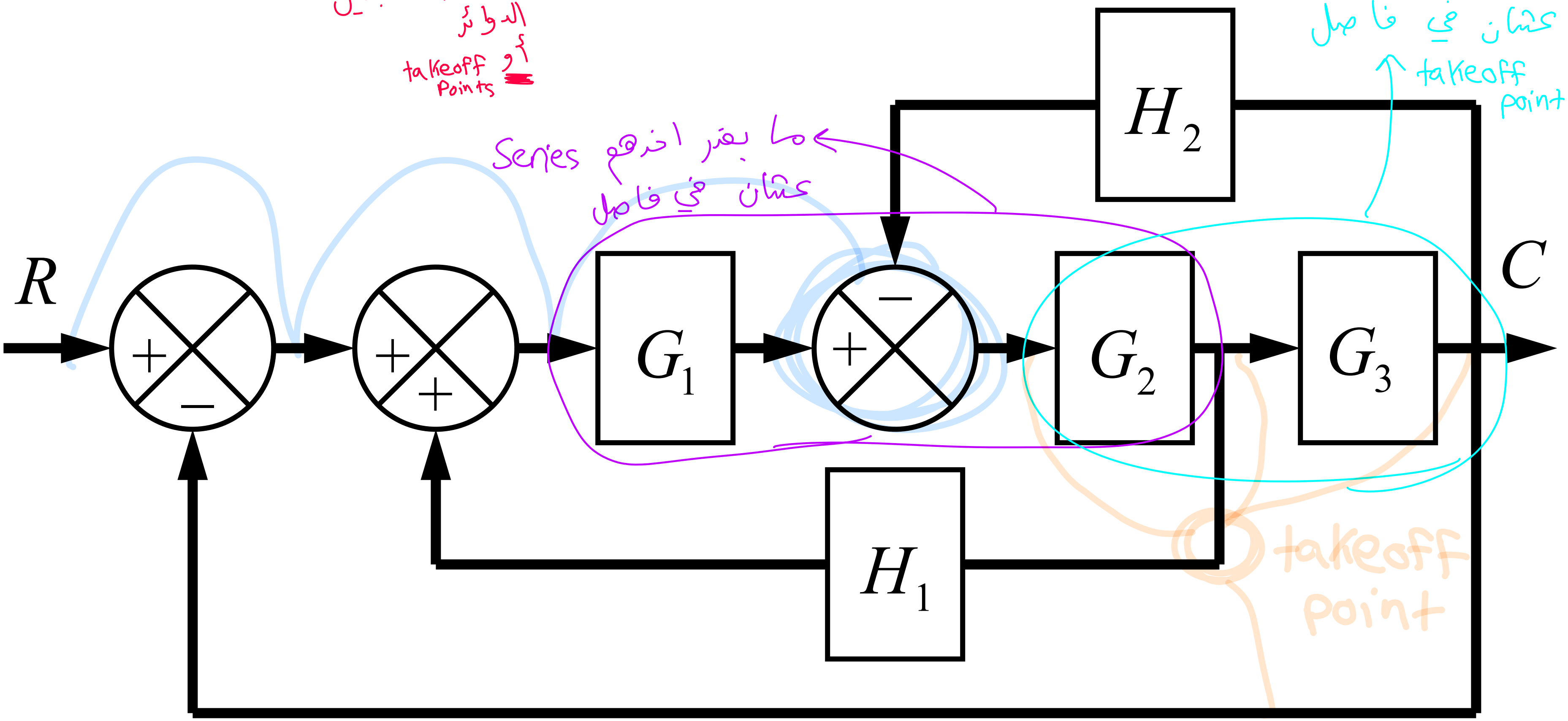
- Reduce the following block diagram to canonical form.

أو بحاول اوصل الهم

عن طريق تبديل

الدوائر

أو takeoff Points



ما بقدر اخذهم Series

عشان في فاصل

ما بقدر اخذهم Series

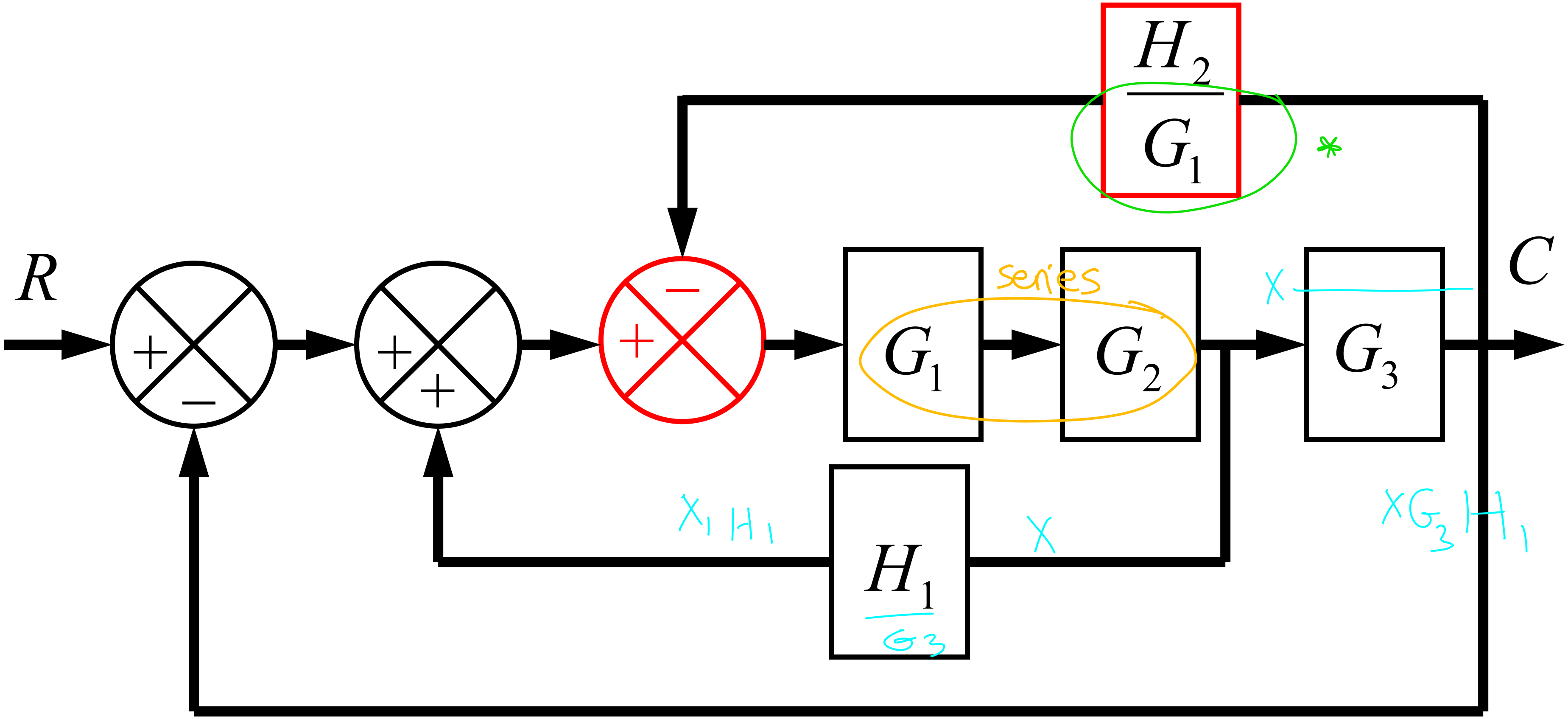
عشان في فاصل

takeoff point

takeoff point

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

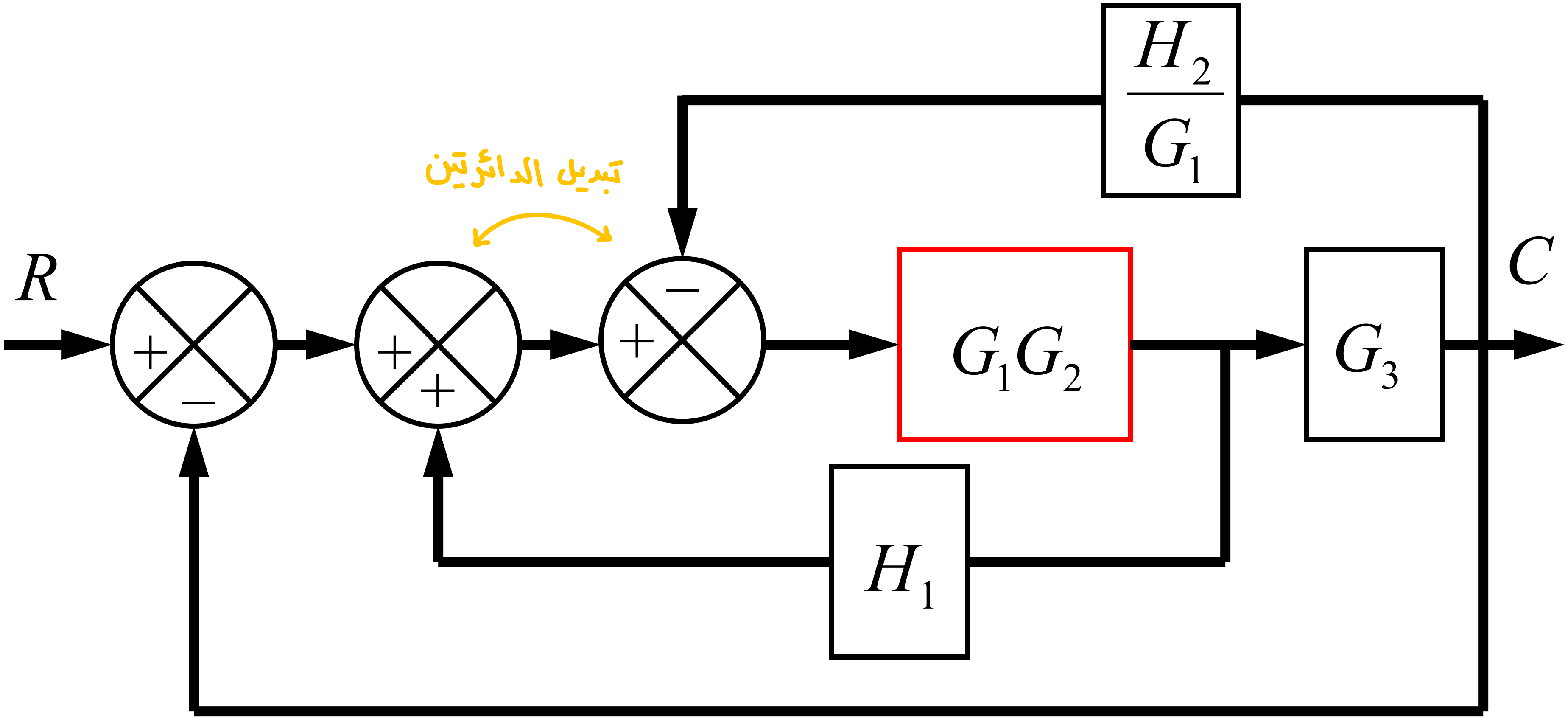
Example-7



لازم بعد التحريك ال output تبقى كما هي
المخارج

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

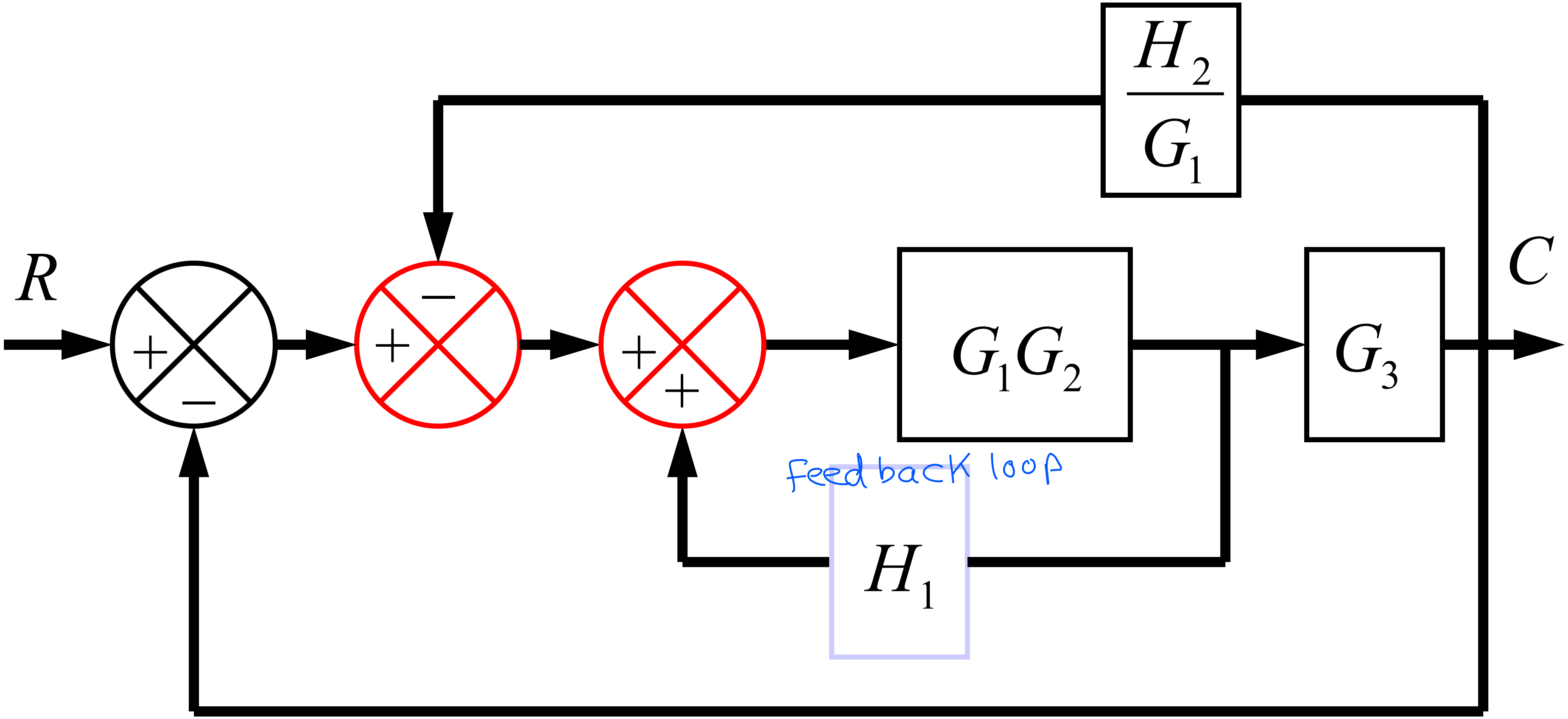
Example-7



تبديل الدائرتين

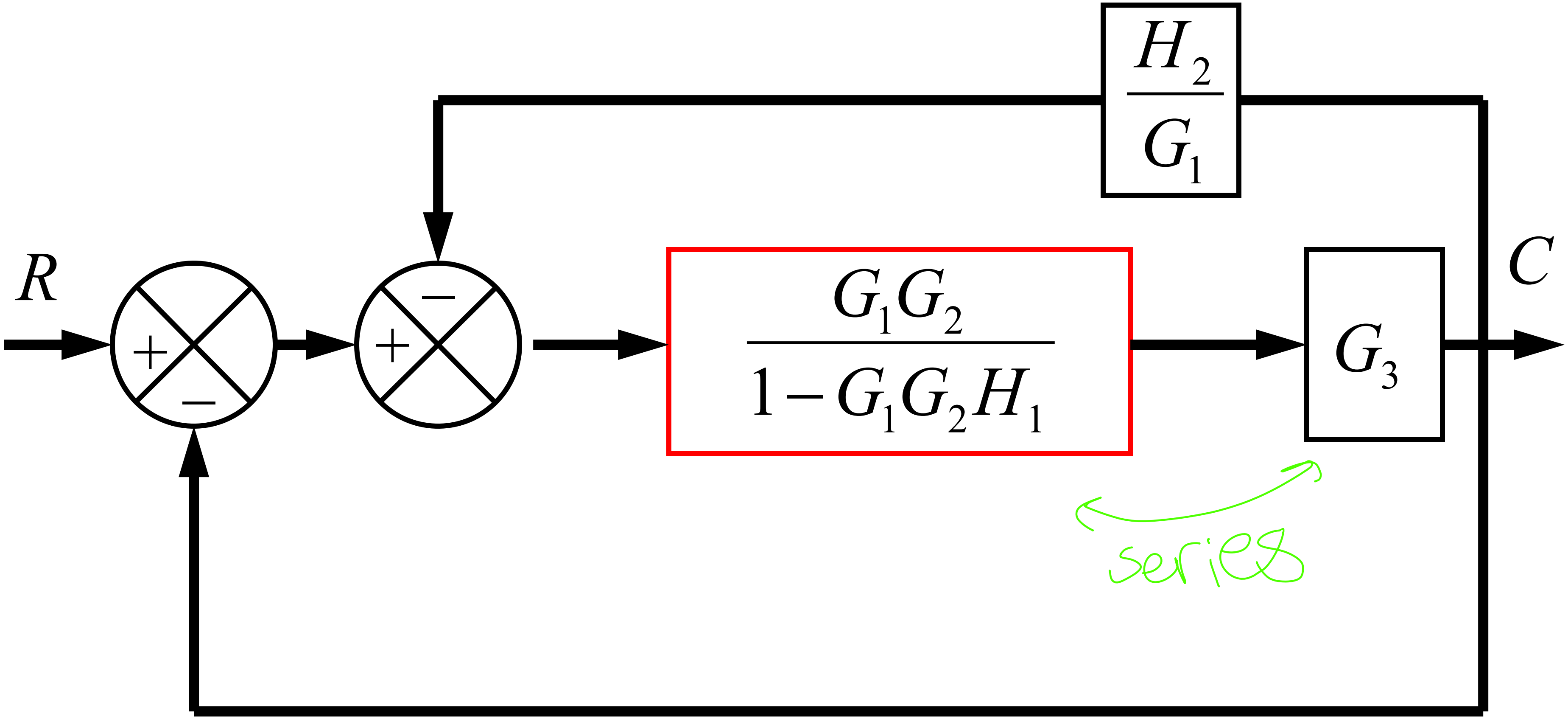
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-7



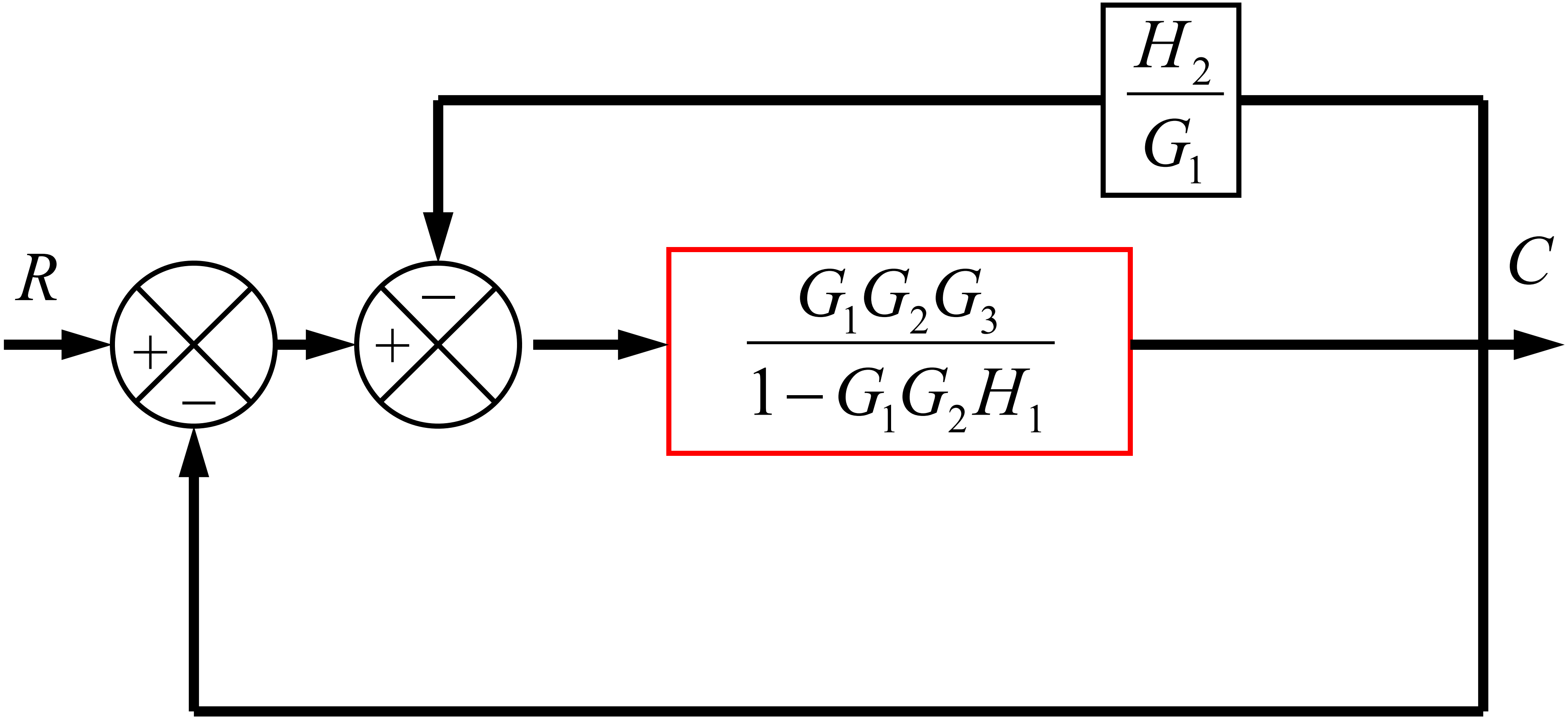
© 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-7



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-7



$$\frac{G}{1+GH} = \frac{\dfrac{G_1G_2G_3}{1-G_1G_2H_1}}{1+\left(\dfrac{G_1G_2G_3}{1-G_1G_2H_1}\right)\times\left(\dfrac{H_2}{G_1}\right)}$$

عكس الاشارة

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

$$\frac{G}{1+GH} = \frac{\dfrac{G_1G_2G_3}{1-G_1G_2H_1}}{1+\left(\dfrac{\cancel{G_1}G_2G_3}{1-G_1G_2H_1}\right)\times\left(\dfrac{H_2}{\cancel{G_1}}\right)}$$

$$= \frac{\dfrac{G_1G_2G_3}{\cancel{1-G_1G_2H_1}}}{\dfrac{1-G_1G_2H_1+G_2G_3H_2}{\cancel{1-G_1G_2H_1}}}$$

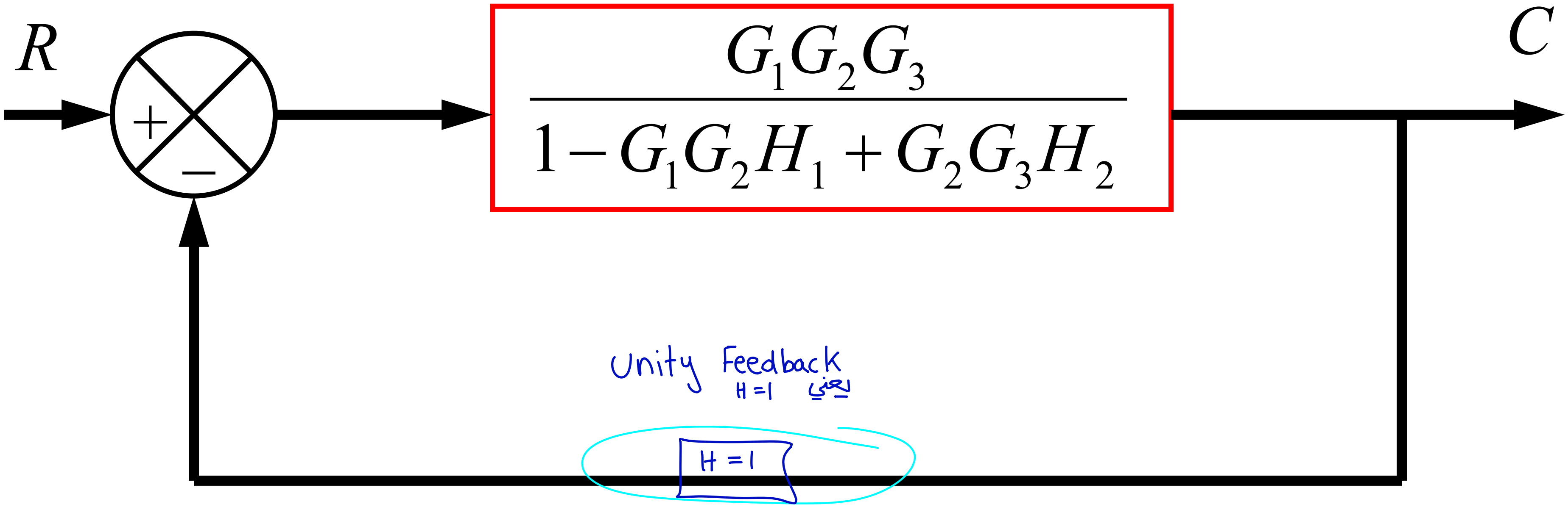
$$= \frac{G_1G_2G_3}{1-G_1G_2H_1+G_1G_3H_2}$$

Example-7

* احنا بنختار اذا نحل على ال Block diagram

Mason's Rule

أسهل لكن لها محاذير



Unity Feedback يعني H=1

Final Solution:

$$\frac{G_1G_2G_3}{1-G_1G_2H_1+G_2G_3H_2+\boxed{G_1G_2G_3}}$$

اجت بعد توحيد المقامات

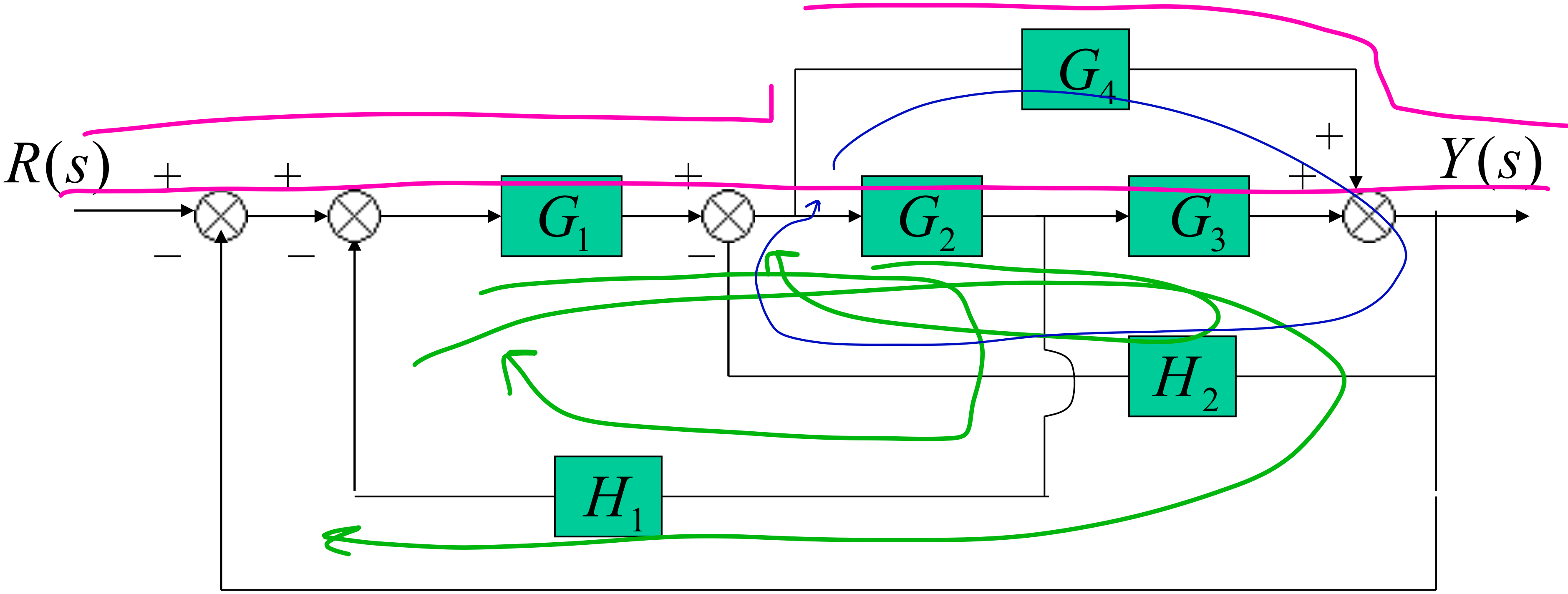
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example 8

by Masons Rule →

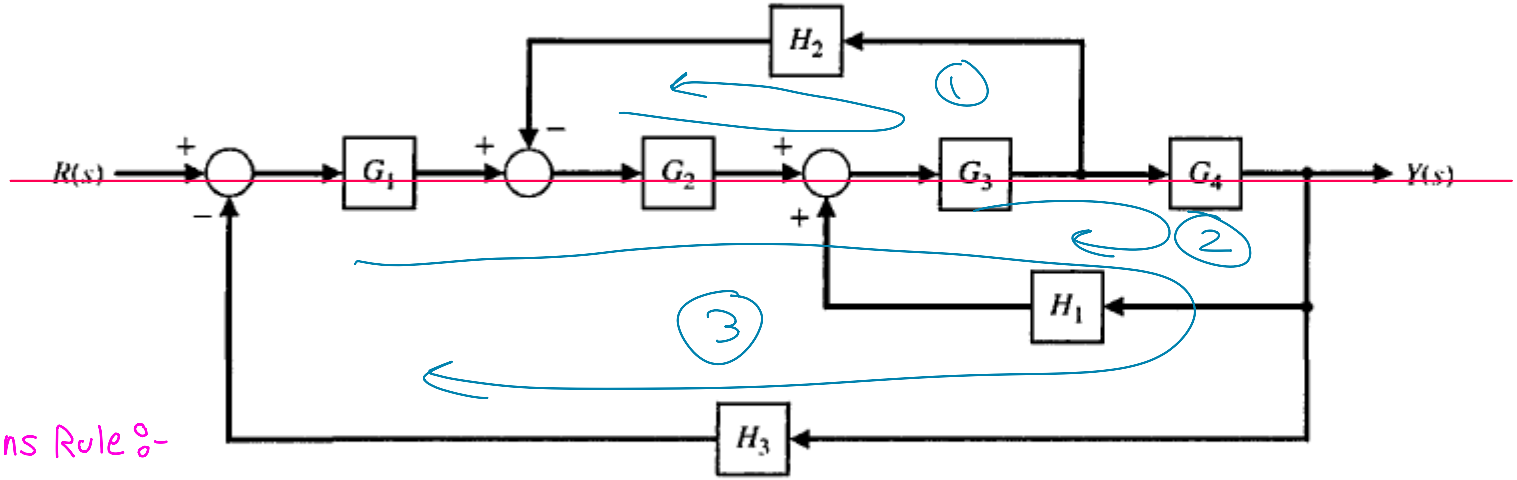
$$\frac{G_1G_2G_3 + G_1G_4}{1 - (\ominus G_2G_3H_2 \ominus G_1G_2H_1 \ominus G_1G_2G_3 \ominus \underbrace{G_4H_2}_{\text{موجودة}})}$$

Find the transfer function of the following block diagram



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

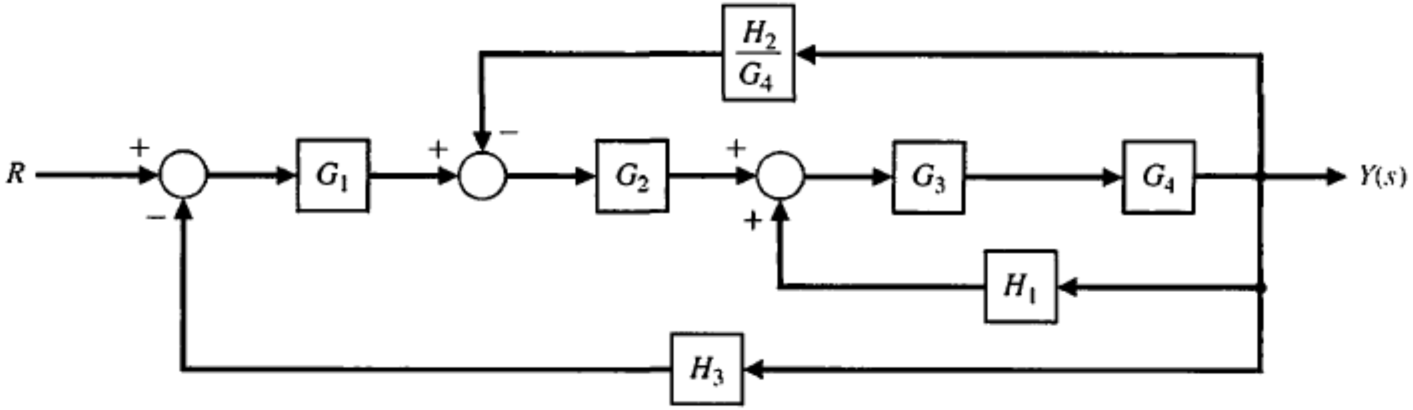
Example-10: Reduce the Block Diagram.



by Masons Rule :-

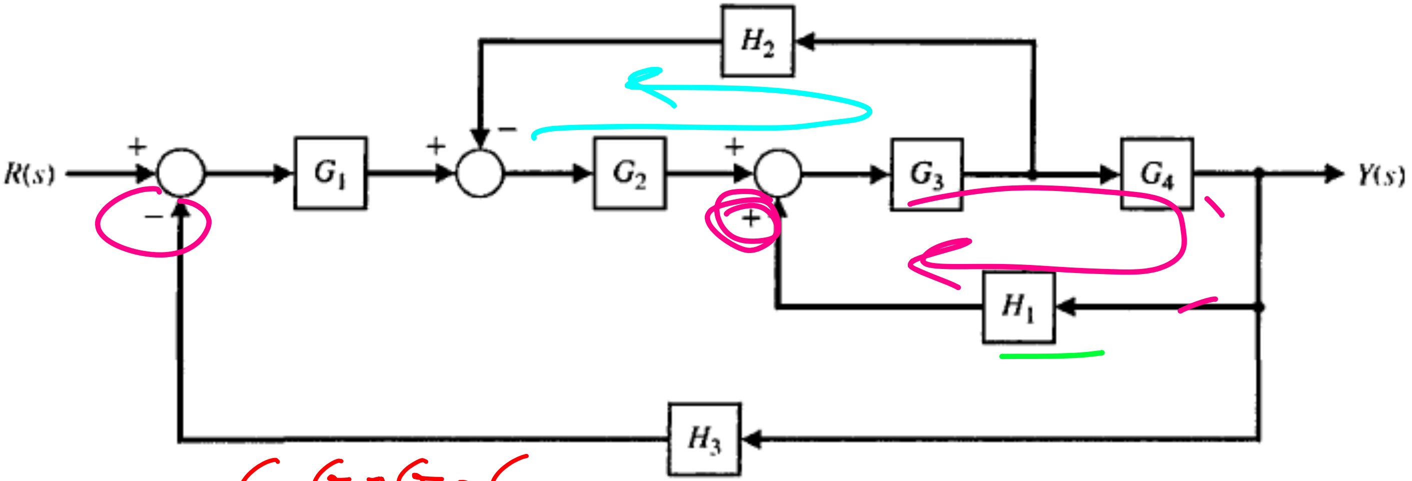
$$\frac{G_1G_2G_3G_4}{1-(\ominus G_2G_3H_2 \oplus G_3G_4H_1 \ominus G_1G_2G_3G_4H_3)+0}$$

First, to eliminate the loop $G_3G_4H_1$, we move H_2 behind block G_4



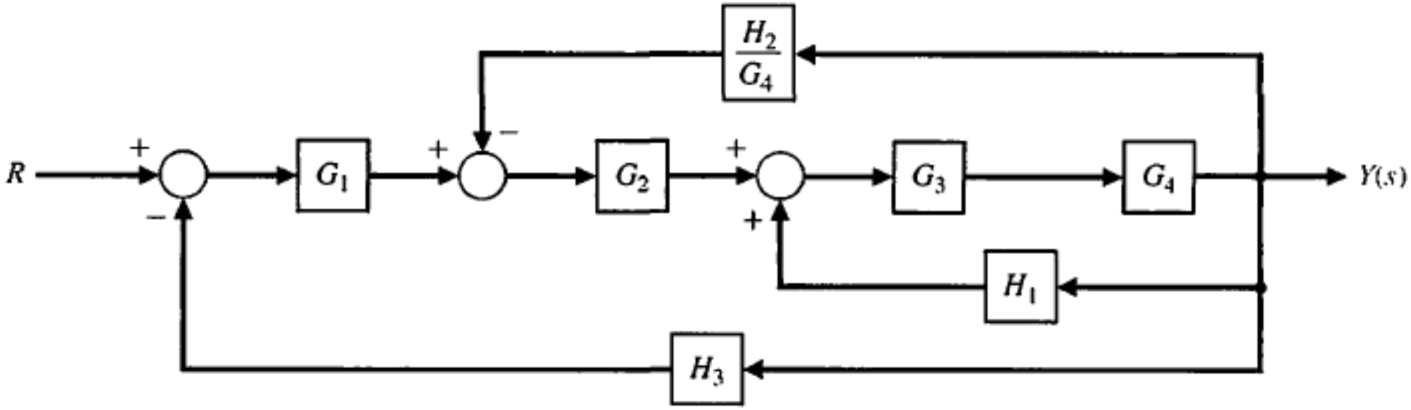
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-10: Reduce the Block Diagram.



$$\frac{C}{R} = \frac{G_1G_2G_3G_4}{1-(\ominus G_1G_2G_3G_4H_3 + G_3G_4H_1 \ominus G_2G_3H_2)+0}$$

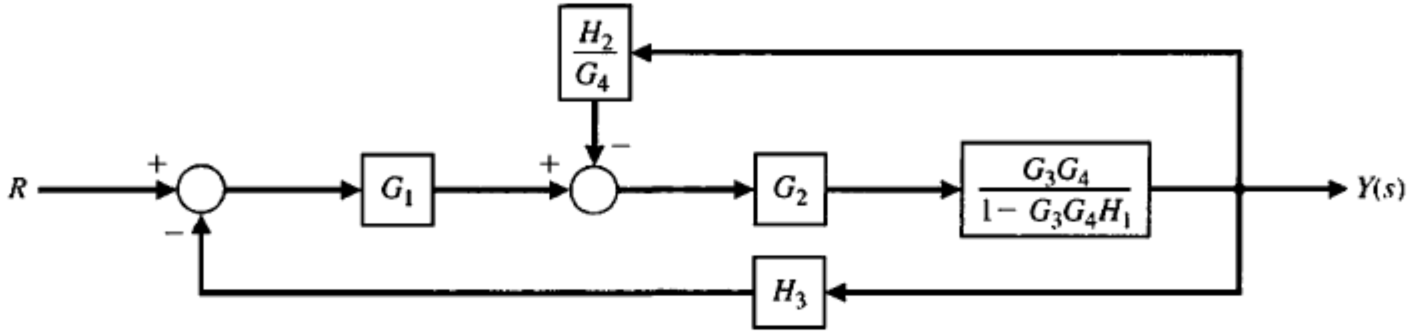
First, to eliminate the loop $G_3G_4H_1$, we move H_2 behind block G_4



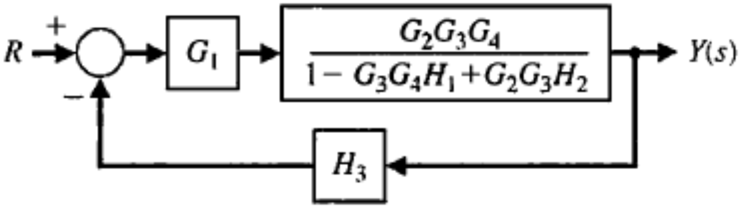
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-10: Continue.

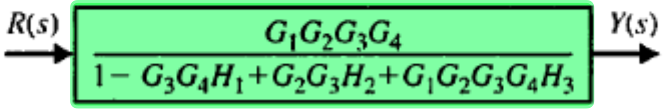
Eliminating the loop $G_3G_4H_1$ we obtain



Then, eliminating the inner loop containing H_2/G_4, we obtain

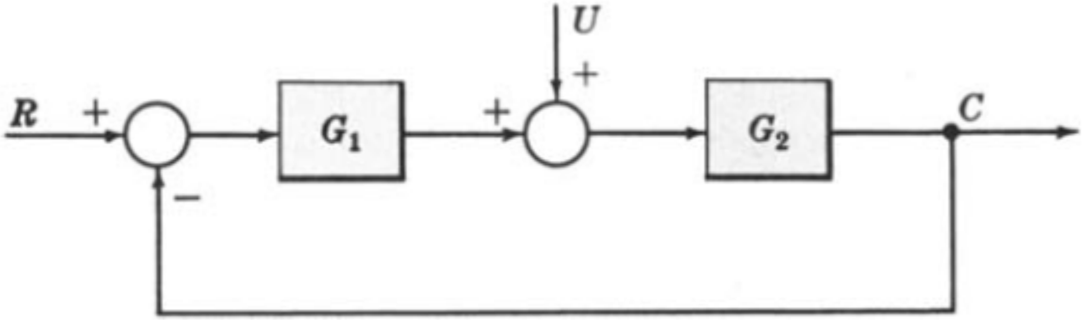


Finally, by reducing the loop containing H_3, we obtain



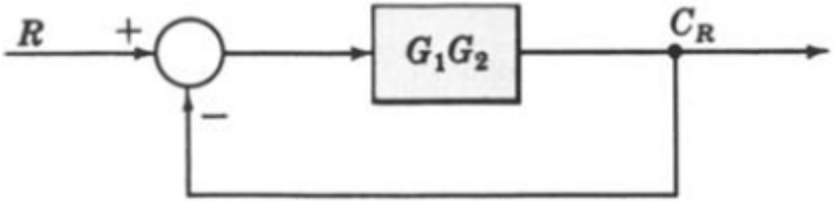
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-12: Multiple Input System. Determine the output C due to inputs R and U using the Superposition Method.



Step 1: Put $U \equiv 0$.

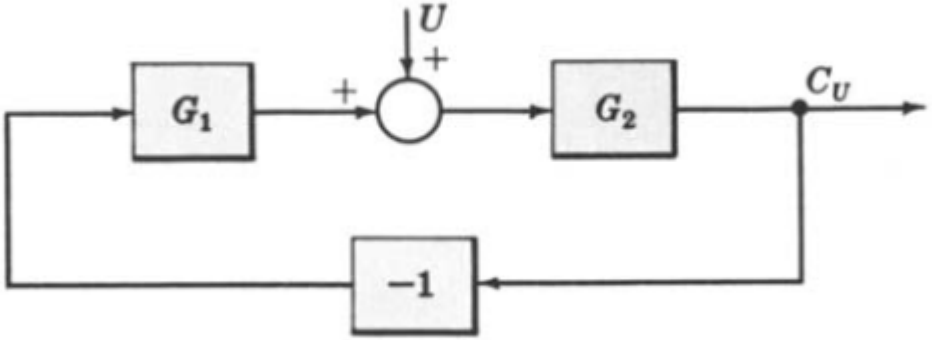
Step 2: The system reduces to



Step 3: the output C_R due to input R is $C_R = [G_1G_2/(1 + G_1G_2)]R$.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

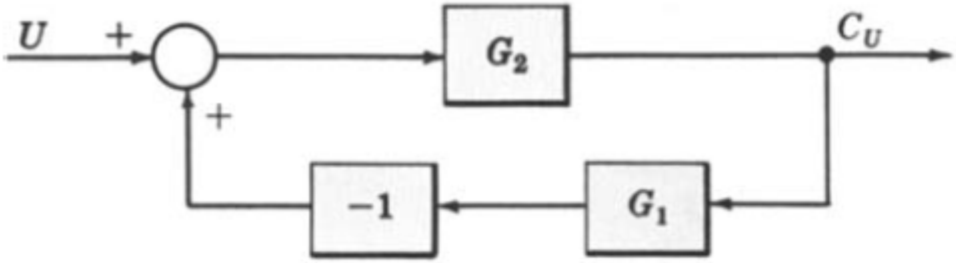
Example-12: Continue.



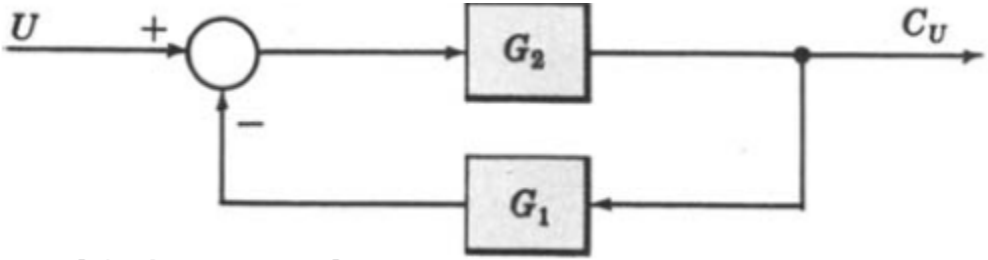
Step 4a: Put $R = 0$.

Step 4b: Put -1 into a block, representing the negative feedback effect:

Rearrange the block diagram:



Let the -1 block be absorbed into the summing point:



Step 4c: the output C_U due to input U is $C_U = [G_2/(1 + G_1G_2)]U$.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-12: Continue.

Step 5: The total output is $C = C_R + C_U$

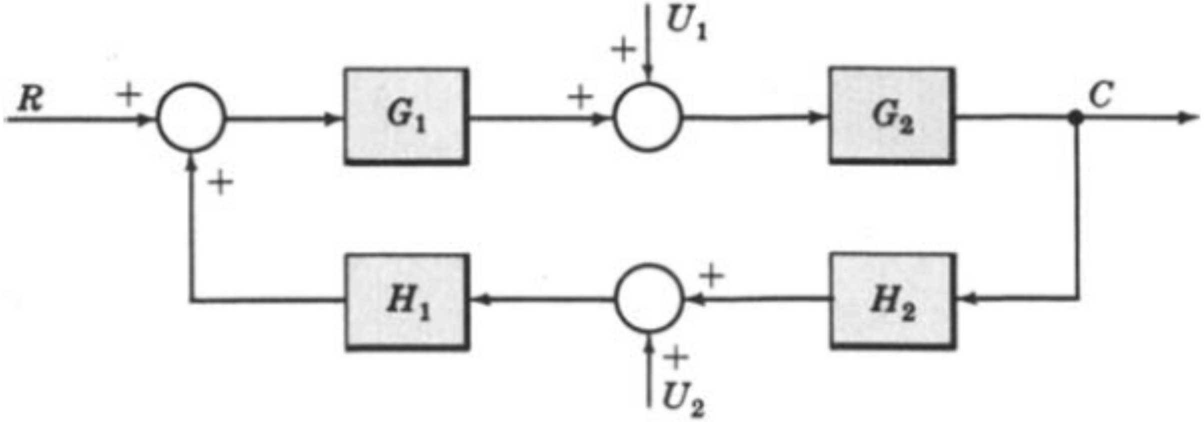
$$= \left[\frac{G_1G_2}{1 + G_1G_2}\right]R + \left[\frac{G_2}{1 + G_1G_2}\right]U$$

$$= \left[\frac{G_2}{1 + G_1G_2}\right][G_1R + U]$$

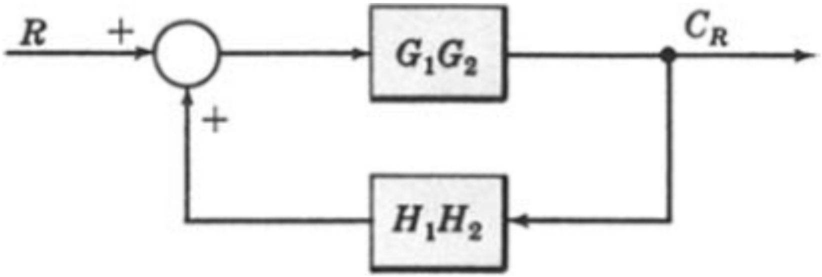
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-13: Multiple-Input System. Determine the output C due to inputs R, U1 and U2 using the Superposition Method.

ما بتذكر اخذناه



Let $U_1 = U_2 = 0$.



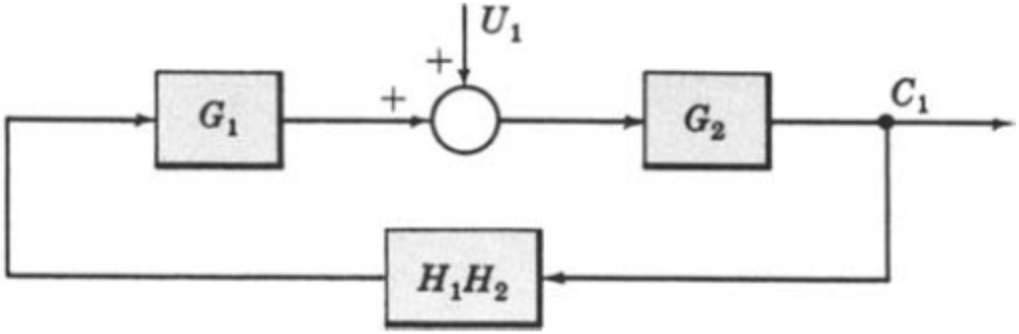
$$C_R = [G_1G_2/(1 - G_1G_2H_1H_2)]R$$

where C_R is the output due to R acting alone.

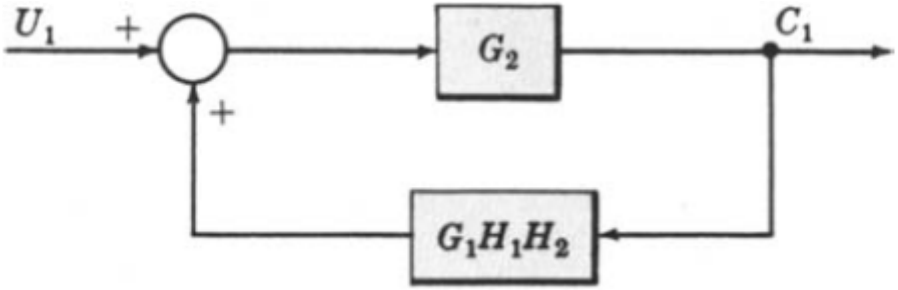
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-13: Continue.

Now let $R = U_2 = 0$.



Rearranging the blocks, we get



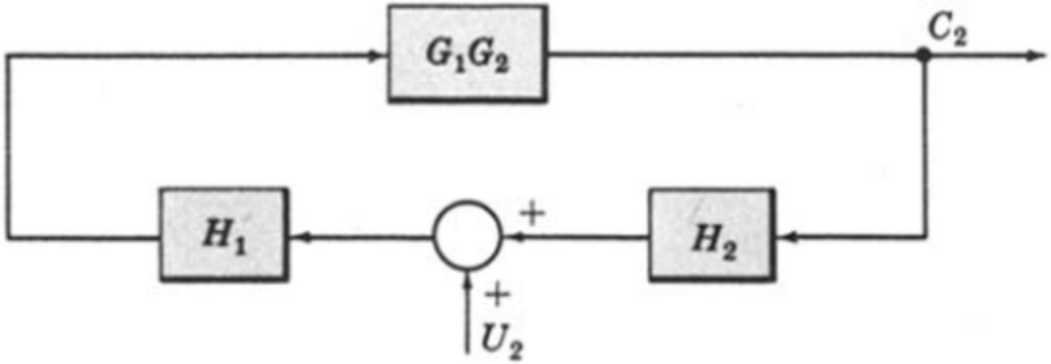
$$C_1 = [G_2/(1 - G_1G_2H_1H_2)]U_1$$

where C_1 is the response due to U_1 acting alone.

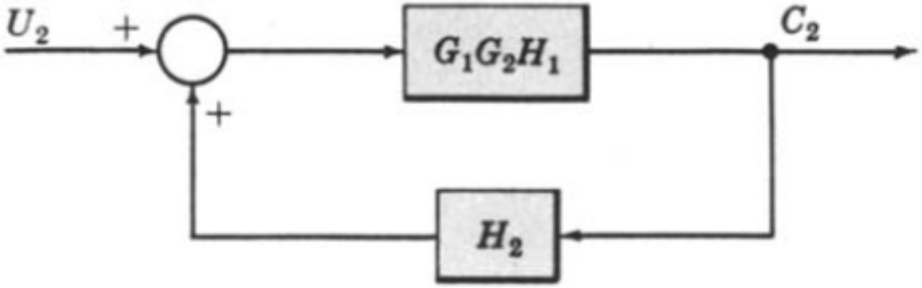
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-13: Continue.

Finally, let $R = U_1 = 0$.



Rearranging the blocks, we get



$$C_2 = [G_1G_2H_1/(1 - G_1G_2H_1H_2)]U_2$$

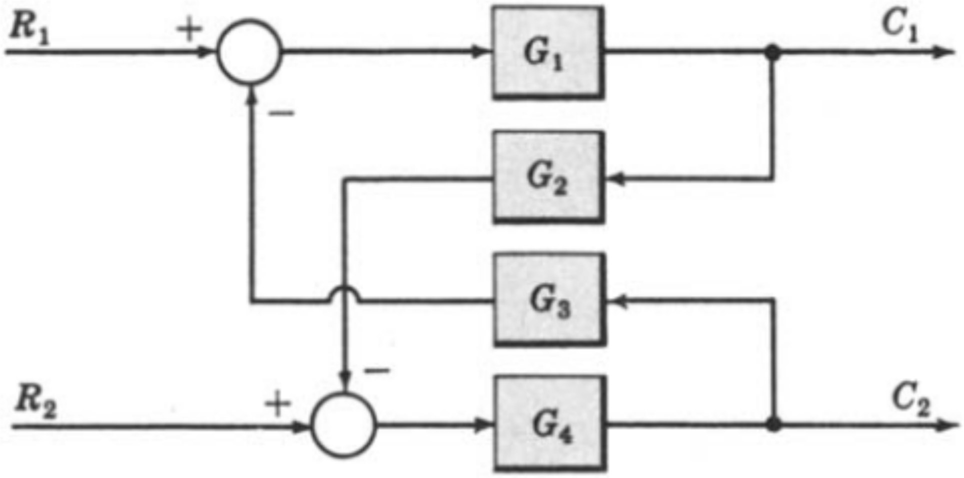
where C_2 is the response due to U_2 acting alone.

By superposition, the total output is

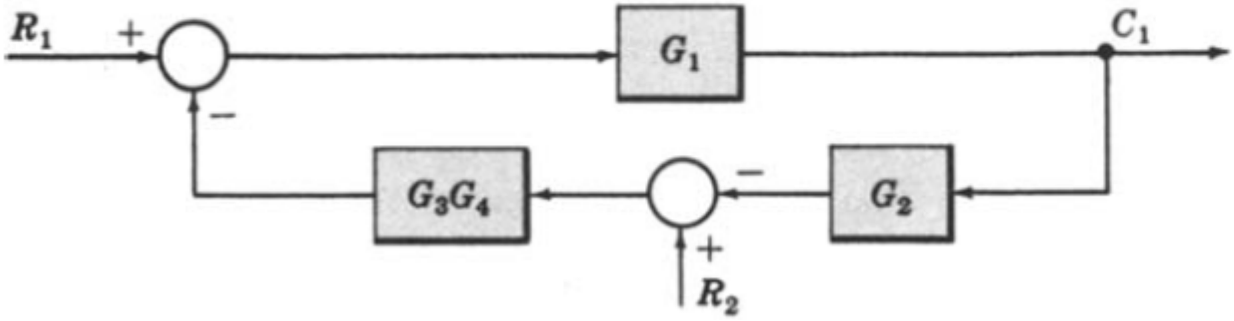
$$C = C_R + C_1 + C_2 = \frac{G_1G_2R + G_2U_1 + G_1G_2H_1U_2}{1 - G_1G_2H_1H_2}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-14: Multi-Input Multi-Output System. Determine C1 and C2 due to R1 and R2.



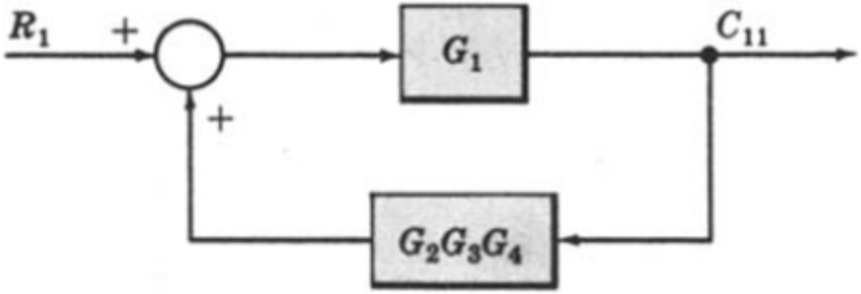
First ignoring the output C_2.



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

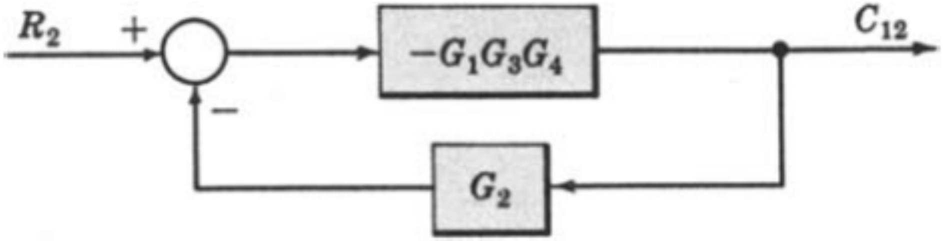
Example-14: Continue.

Letting $R_2 = 0$ and combining the summing points,



Hence C_{11}, the output at C_1 due to R_1 alone, is $C_{11} = G_1R_1/(1 - G_1G_2G_3G_4)$.

For $R_1 = 0$,



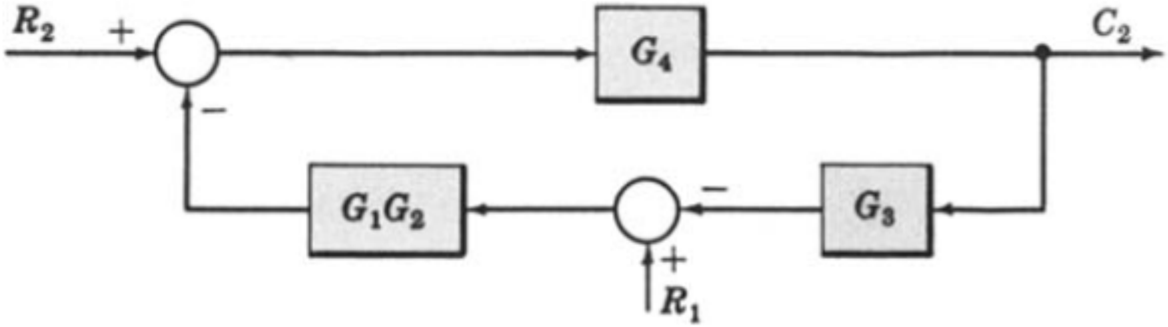
Hence $C_{12} = -G_1G_3G_4R_2/(1 - G_1G_2G_3G_4)$ is the output at C_1 due to R_2 alone.

Thus $C_1 = C_{11} + C_{12} = (G_1R_1 - G_1G_3G_4R_2)/(1 - G_1G_2G_3G_4)$

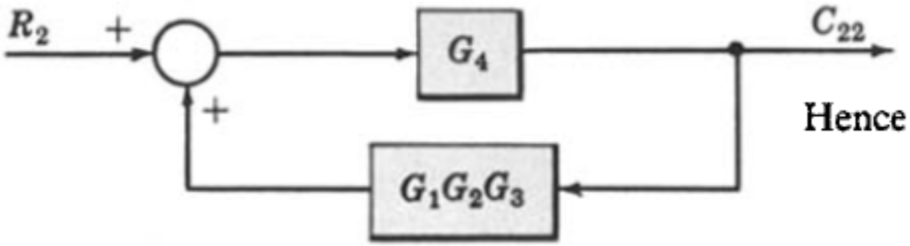
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example-14: Continue.

Now we reduce the original block diagram, ignoring output C_1.

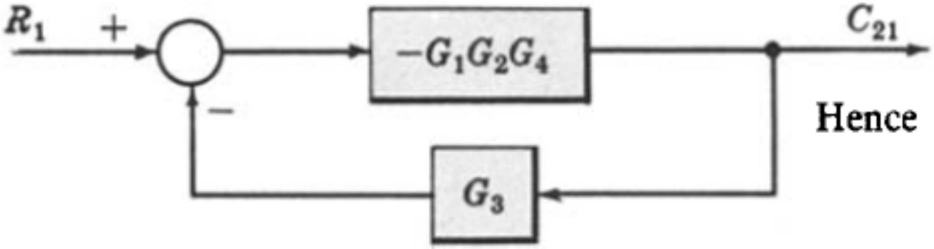


When R1 = 0,



Hence $C_{22} = G_4 R_2/(1 - G_1G_2G_3G_4)$

When R2 = 0,



Hence $C_{21} = -G_1G_2G_4 R_1/(1 - G_1G_2G_3G_4)$

Finally, $C_2 = C_{22} + C_{21} = (G_4 R_2 - G_1G_2G_4 R_1)/(1 - G_1G_2G_3G_4)$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Introduction

- Alternative method to block diagram representation, developed by Samuel Jefferson Mason.
- Advantage: the availability of a flow graph gain formula, also called Mason's gain formula.
- A signal-flow graph consists of a network in which nodes are connected by directed branches.
- It depicts the flow of signals from one point of a system to another and gives the relationships among the signals.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

نحن غير مجبرين على استخدامه (عادي بنحل بالطرق الثانية)

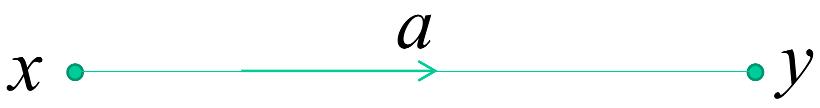
Fundamentals of Signal Flow Graphs

* هنا لا يوجد مربعات (فقط اسهم و دوائر)

- Consider a simple equation below and draw its signal flow graph:

$$y = ax$$

- The signal flow graph of the equation is shown below;



السهم عملية ضرب

- Every variable in a signal flow graph is designed by a **Node**.
- Every transmission function in a signal flow graph is designed by a **Branch**.
- Branches are always **unidirectional**.
- The arrow in the branch denotes the **direction** of the signal flow.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

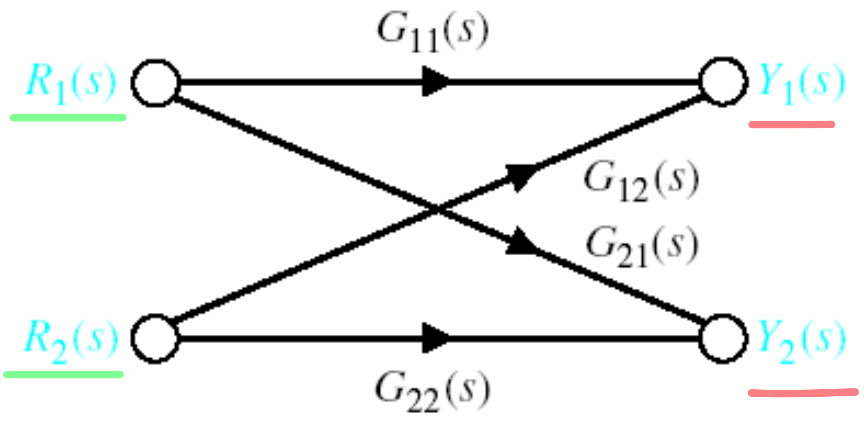
Signal-Flow Graph Models

output

variables

$$Y_1(s) = G_{11}(s) \cdot R_1(s) + G_{12}(s) \cdot R_2(s)$$

$$Y_2(s) = G_{21}(s) \cdot R_1(s) + G_{22}(s) \cdot R_2(s)$$



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Signal-Flow Graph Models

r_1 *and* r_2 are inputs and x_1 and x_2 are outputs

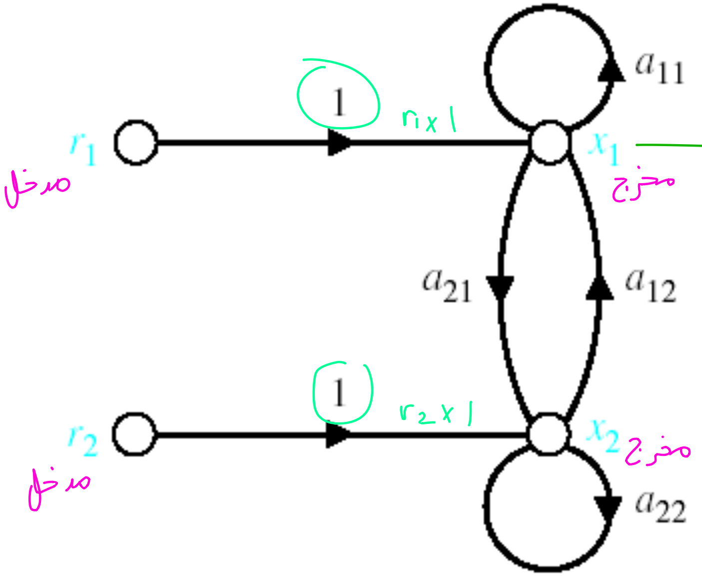
$$a_{\underline{11}} \cdot x_1 + a_{\underline{12}} \cdot x_2 + \underline{1}\, r_1 = x_1$$

مكتوبة على x_1

مكتوبة على السهم لـ x_2

مكتوب على السهم لـ r_1 واحد

$$a_{21} \cdot x_1 + a_{22} \cdot x_2 + r_2 = x_2$$



مجموع ثلاث اسهم يدخلوها

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Signal-Flow Graph Models

هنا نكون مجبرين على ال
Mason's Rule

a, b, c, d, e, f, g, h, e → كلهم بسط على مقام
(Laplace Transform)

x_o is input and x_4 is output

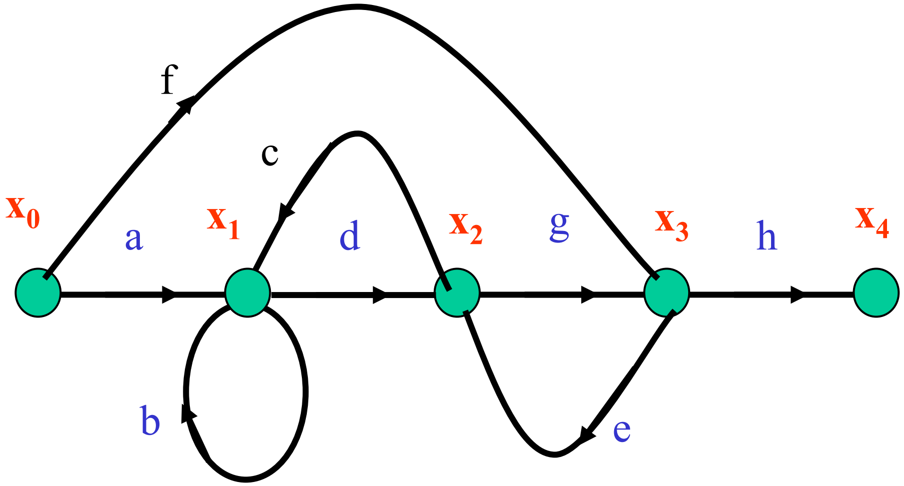
$$x_1 = ax_0 + bx_1 + cx_2$$

$$x_2 = dx_1 + ex_3$$

$$x_3 = fx_0 + gx_2$$

$$x_4 = \underline{hx_3}$$

يعني بدخله
سهم واحد



ما بيجي سؤال نحول لـ Block diagram أو هيك

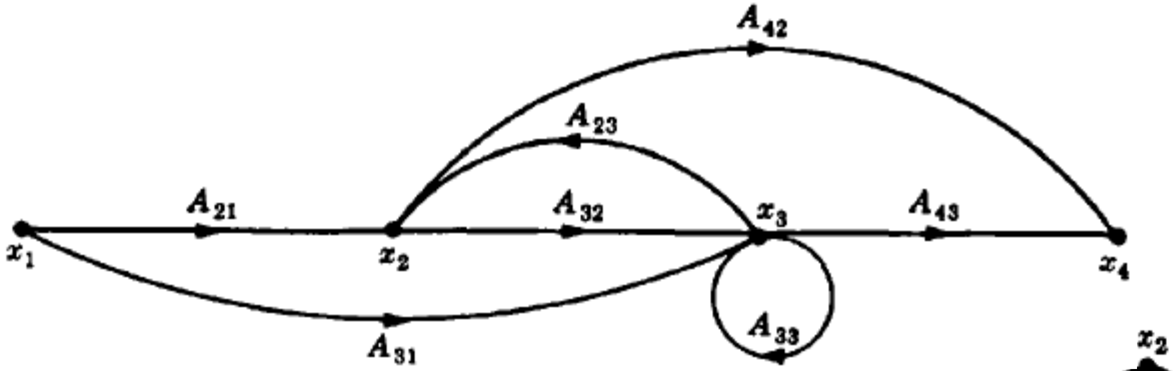
← بيجي على ال signal flow graph ← solve for the equivelant

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

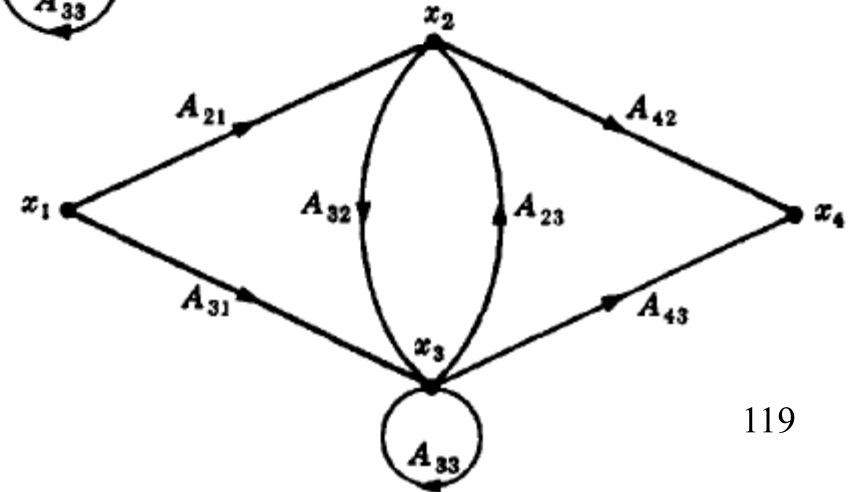
Construct the signal flow graph for the following set of simultaneous equations.

$$x_2 = A_{21}x_1 + A_{23}x_3 \qquad x_3 = A_{31}x_1 + A_{32}x_2 + A_{33}x_3 \qquad x_4 = A_{42}x_2 + A_{43}x_3$$

- There are four variables in the equations (i.e., x_1,x_2,x_3,and x_4) therefore four nodes are required to construct the signal flow graph.
- Arrange these four nodes from left to right and connect them with the associated branches.



- Another way to arrange this graph is shown in the figure.



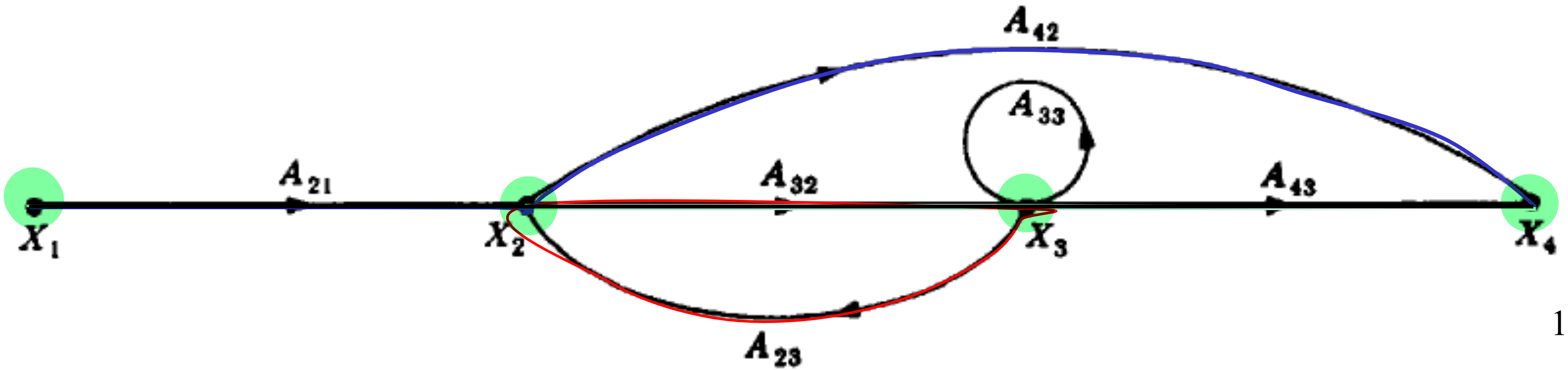
Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Terminologies

- An **input node** or source contain only the outgoing branches. i.e., X_1
- An **output node** or sink contain only the incoming branches. i.e., X_4
- A **path** is a continuous, unidirectional succession of branches along which no node is passed more than ones. i.e.,

  X_1 *to* X_2 *to* X_3 *to* X_4 X_1 *to* X_2 *to* X_4 X_2 *to* X_3 *to* X_4 ?

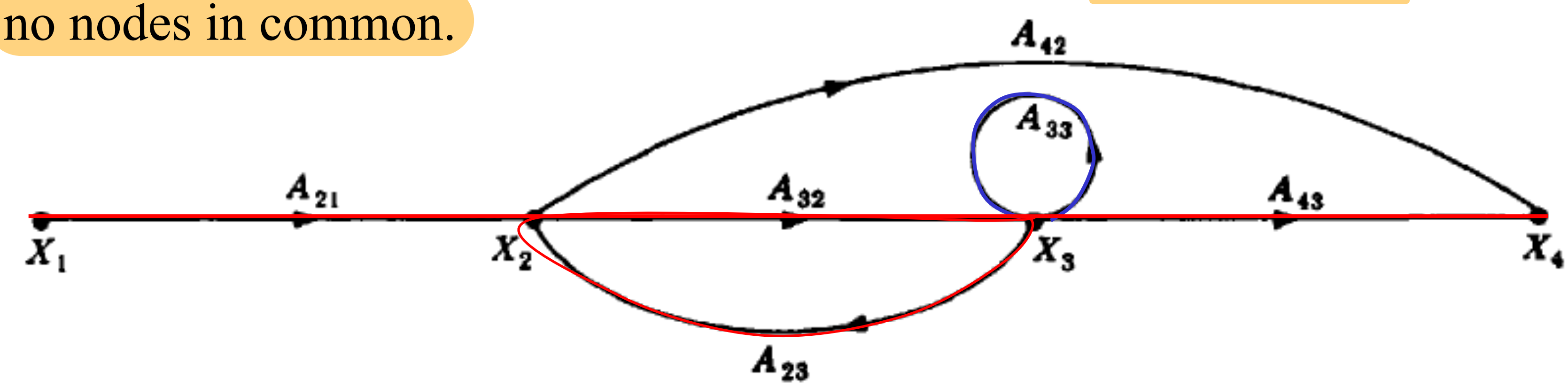
- A **forward path** is a path from the input node to the output node. i.e., X_1 *to* X_2 *to* X_3 *to* X_4, and X_1 *to* X_2 *to* X_4, are forward paths.
- A **feedback path** or feedback loop is a path which originates and terminates on the same node. i.e.; X_2 *to* X_3 **and back to** X_2 is a feedback path.



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

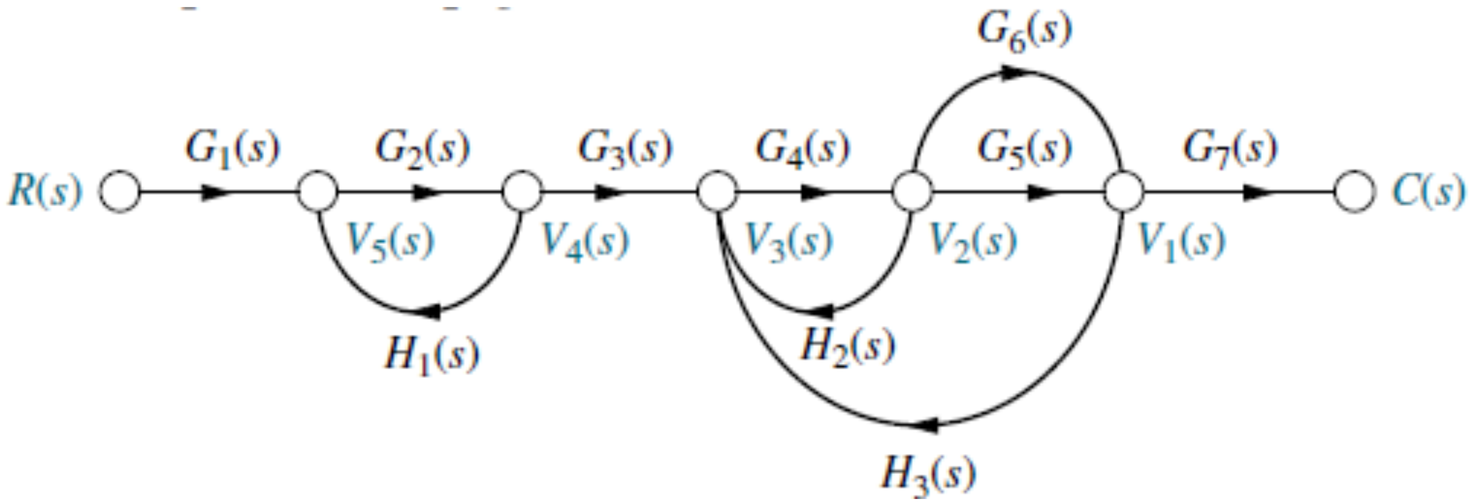
Terminologies

- A **self-loop** is a feedback loop consisting of a single branch. i.e.; A_{33} is a self loop.
- The **gain** of a branch is the transmission function of that branch.
- The **path gain** is the product of branch gains encountered in traversing a path. i.e. the gain of forwards path X_1 *to* X_2 *to* X_3 *to* X_4 is $A_{21}A_{32}A_{43}$
- The **loop gain** is the product of the branch gains of the loop. i.e., the loop gain of the feedback loop from X_2 to X_3 and back to X_2 is $A_{32}A_{23}$.
- Two loops, paths, or loop and a path are said to be **non-touching** if they have no nodes in common.



Illustrat

Consider the signal flow graph below and identify the following



a) Input node. R

b) Output node. C

c) Forward paths. $G_1G_2G_3G_4G_5G_7$ or $G_1G_2G_3G_4G_6G_7$

d) Feedback paths (loops). G_2H_1

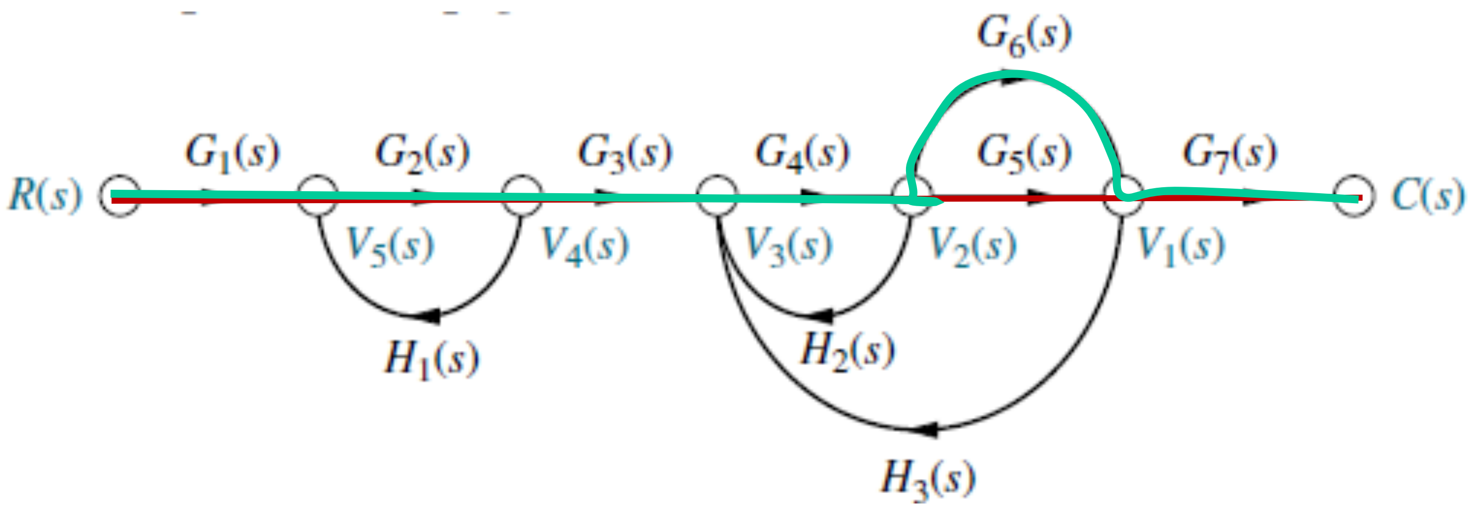
e) Determine the loop gains of the feedback loops.

f) Determine the path gains of the forward paths.

g) Non-touching loops

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Consider the signal flow graph below and identify the following



- There are two forward path gains;

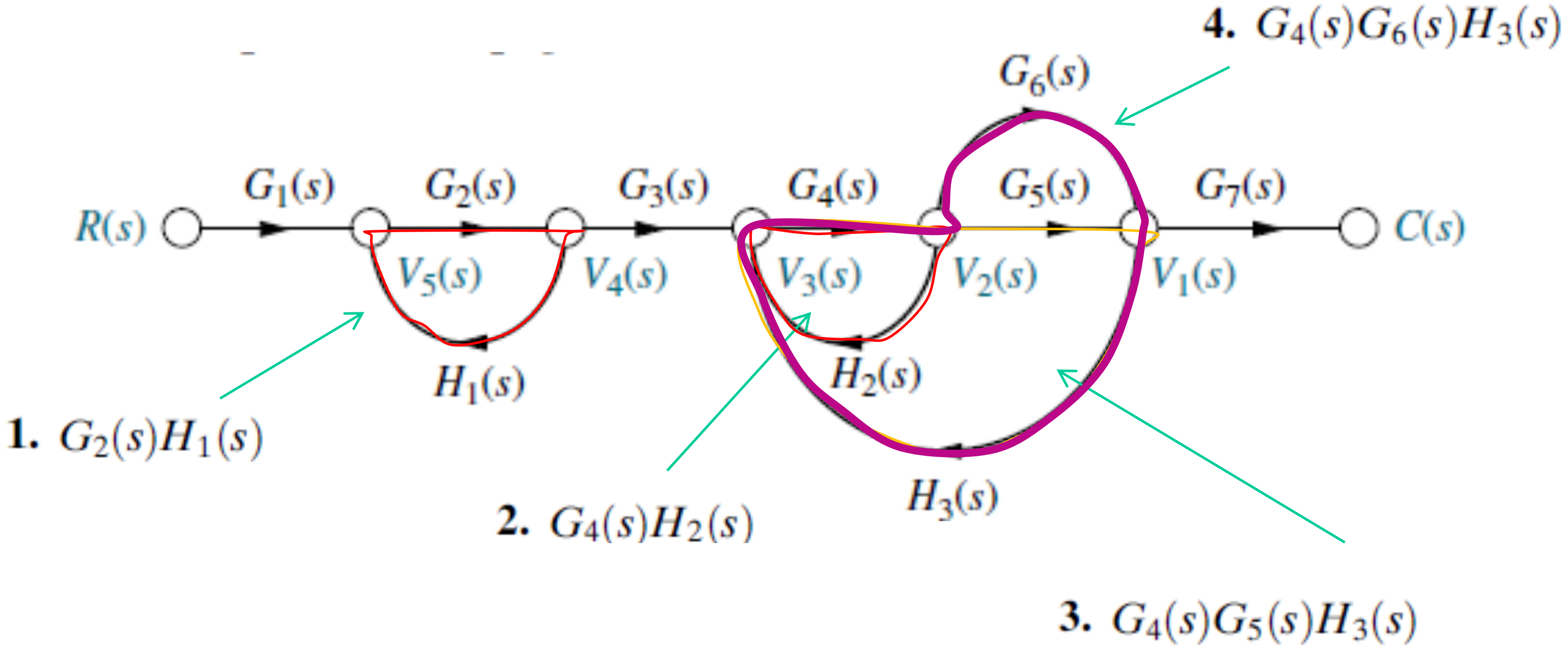
1. $G_1(s)G_2(s)G_3(s)G_4(s)G_5(s)G_7(s)$

2. $G_1(s)G_2(s)G_3(s)G_4(s)G_6(s)G_7(s)$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

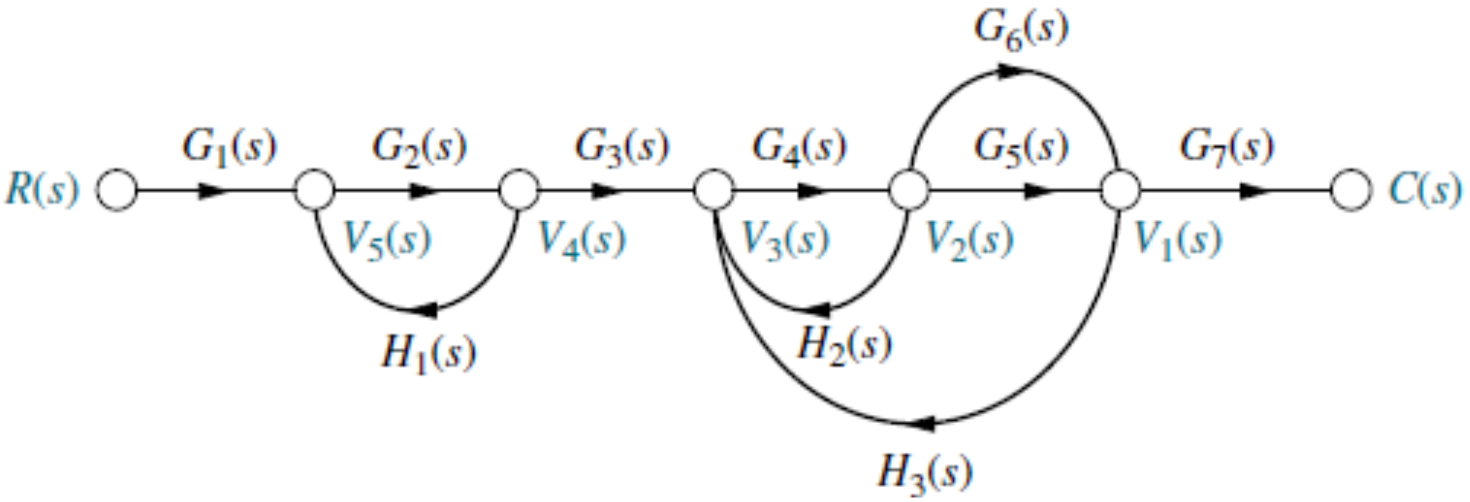
Consider the signal flow graph below and identify the following

- There are four loops



Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Consider the signal flow graph below and identify the following



- Nontouching loop gains;

1. $[G_2(s)H_1(s)][G_4(s)H_2(s)]$
2. $[G_2(s)H_1(s)][G_4(s)G_5(s)H_3(s)]$
3. $[G_2(s)H_1(s)][G_4(s)G_6(s)H_3(s)]$

✓

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Mason's Rule (Mason, 1953)

- The block diagram reduction technique requires successive application of fundamental relationships in order to arrive at the system transfer function.
- On the other hand, Mason's rule for reducing a signal-flow graph to a single transfer function requires the application of one formula.
- The formula was derived by S. J. Mason when he related the signal-flow graph to the simultaneous equations that can be written from the graph.

© 2001 by Prentice Hall, Upper Saddle River, NJ.

Mason's Rule:

- The transfer function, ***C(s)/R(s),*** of a system represented by a signal-flow graph is;

$$\boxed{\frac{C(s)}{R(s)} = \frac{\sum_{i=1}^{n} P_i \Delta_i}{\Delta}}$$

Where

n = number of forward paths.
P_i = the i^{th} forward-path gain.
Δ = Determinant of the system
Δ_i = Determinant of the i^{th} forward path

- Δ is called the signal flow graph determinant or characteristic function. Since $\Delta=0$ is the system characteristic equation.

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Mason's Rule:

$$\boxed{\frac{C(s)}{R(s)} = \frac{\sum_{i=1}^{n} P_i \Delta_i}{\Delta}}$$

Δ = 1 - (sum of all individual loop gains) + (sum of the products of the gains of all possible two loops that do not touch each other) – (sum of the products of the gains of all possible three loops that do not touch each other) + … and so forth with sums of higher number of non-touching loop gains

Δ_i = value of Δ for the part of the block diagram that does not touch the i-th forward path (Δ_i = 1 if there are no non-touching loops to the i-th path.)

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

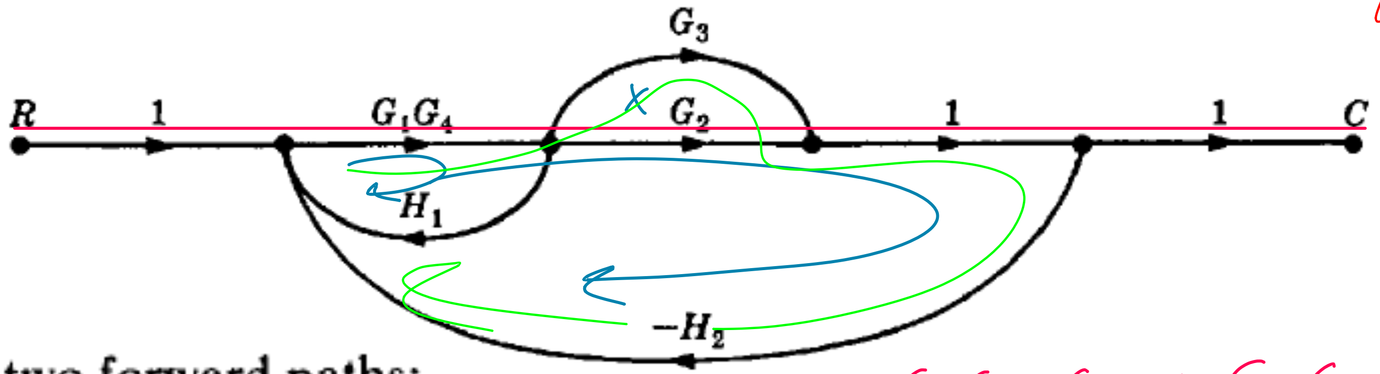
Systematic approach

1. Calculate forward path gain P_i for each forward path i.
2. Calculate all loop transfer functions
3. Consider non-touching loops 2 at a time
4. Consider non-touching loops 3 at a time
5. etc
6. Calculate Δ from steps 2,3,4 and 5
7. Calculate Δ_i as portion of Δ not touching forward path i

© 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#1: Apply Mason's Rule to calculate the transfer function of the system represented by following Signal Flow Graph

Find the equivelant transfer function



There are two forward paths:

$$P_1 = G_1G_2G_4 \qquad P_2 = G_1G_3G_4$$

$$\frac{G_1G_4G_2 + G_1G_4G_3}{1-(\ominus G_1G_4G_2H_2 + G_1G_4H_1 \ominus G_1G_4G_3H_2)}$$

Therefore,

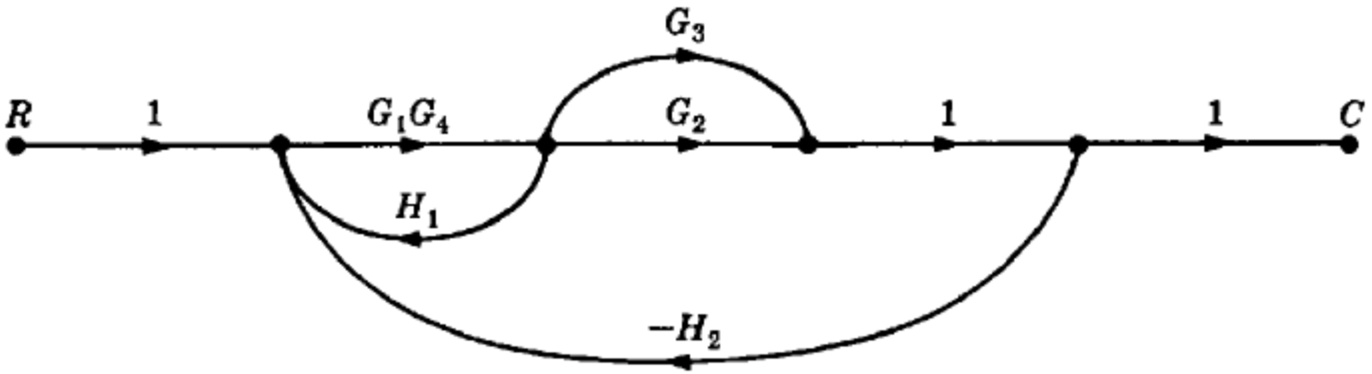
$$\frac{C}{R} = \frac{P_1\Delta_1 + P_2\Delta_2}{\Delta}$$

There are three feedback loops

$$L_1 = G_1G_4H_1, \quad L_2 = -G_1G_2G_4H_2, \quad L_3 = -G_1G_3G_4H_2$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#1: Apply Mason's Rule to calculate the transfer function of the system represented by following Signal Flow Graph



There are no non-touching loops, therefore

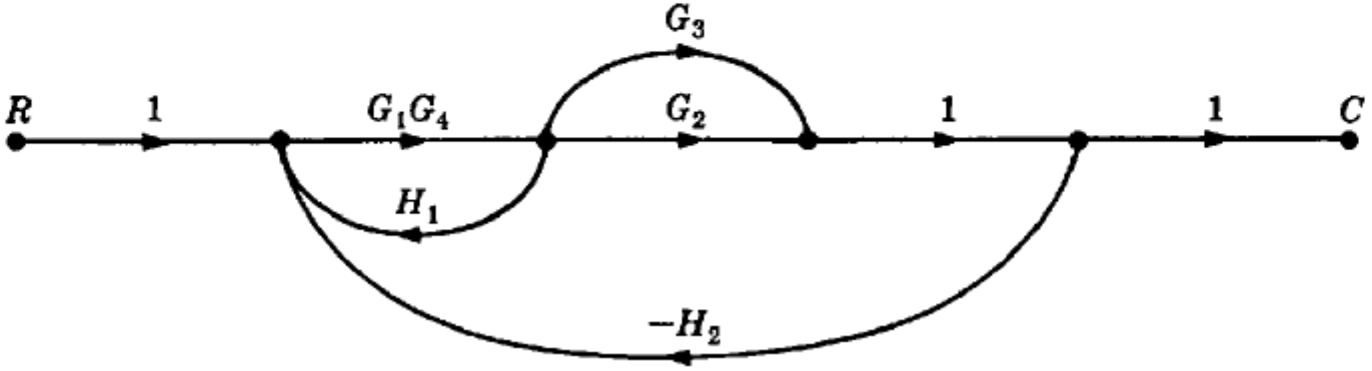
Δ = 1- (sum of all individual loop gains)

$$\Delta = 1 - \left(L_1 + L_2 + L_3\right)$$

$$\Delta = 1 - \left(G_1G_4H_1 - G_1G_2G_4H_2 - G_1G_3G_4H_2\right)$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#1: Apply Mason's Rule to calculate the transfer function of the system represented by following Signal Flow Graph



Eliminate forward path-1

$$\Delta_1 = 1\text{- (sum of all individual loop gains)+...}$$
$$\Delta_1 = 1$$

Eliminate forward path-2

$$\Delta_2 = 1\text{- (sum of all individual loop gains)+...}$$
$$\Delta_2 = 1$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

Example#1: Continue

$$\frac{C}{R} = \frac{P_1\Delta_1 + P_2\Delta_2}{\Delta} = \frac{G_1G_2G_4 + G_1G_3G_4}{1 - G_1G_4H_1 + G_1G_2G_4H_2 + G_1G_3G_4H_2}$$

$$= \frac{G_1G_4(G_2 + G_3)}{1 - G_1G_4H_1 + G_1G_2G_4H_2 + G_1G_3G_4H_2}$$

Illustrations © 2001 by Prentice Hall, Upper Saddle River, NJ.

CHAPTER 4

تحليل استجابة الحالة العابرة والثابتة

Transient & Steady State Response Analysis

repeated → قليل بتطلع
(حالة صدفة)

1st
2nd order
بالمعتاد بكونو

complex أو جذر حقيقي → بالعادة بتطلع

Introduction

The time response of a control system consists of two parts:

e^{-4t} → بيهمل في التصنيع
لأنه بروح بسرعة

1. Transcient response

   - from initial state to the final state – purpose of control systems is to provide a desired response.

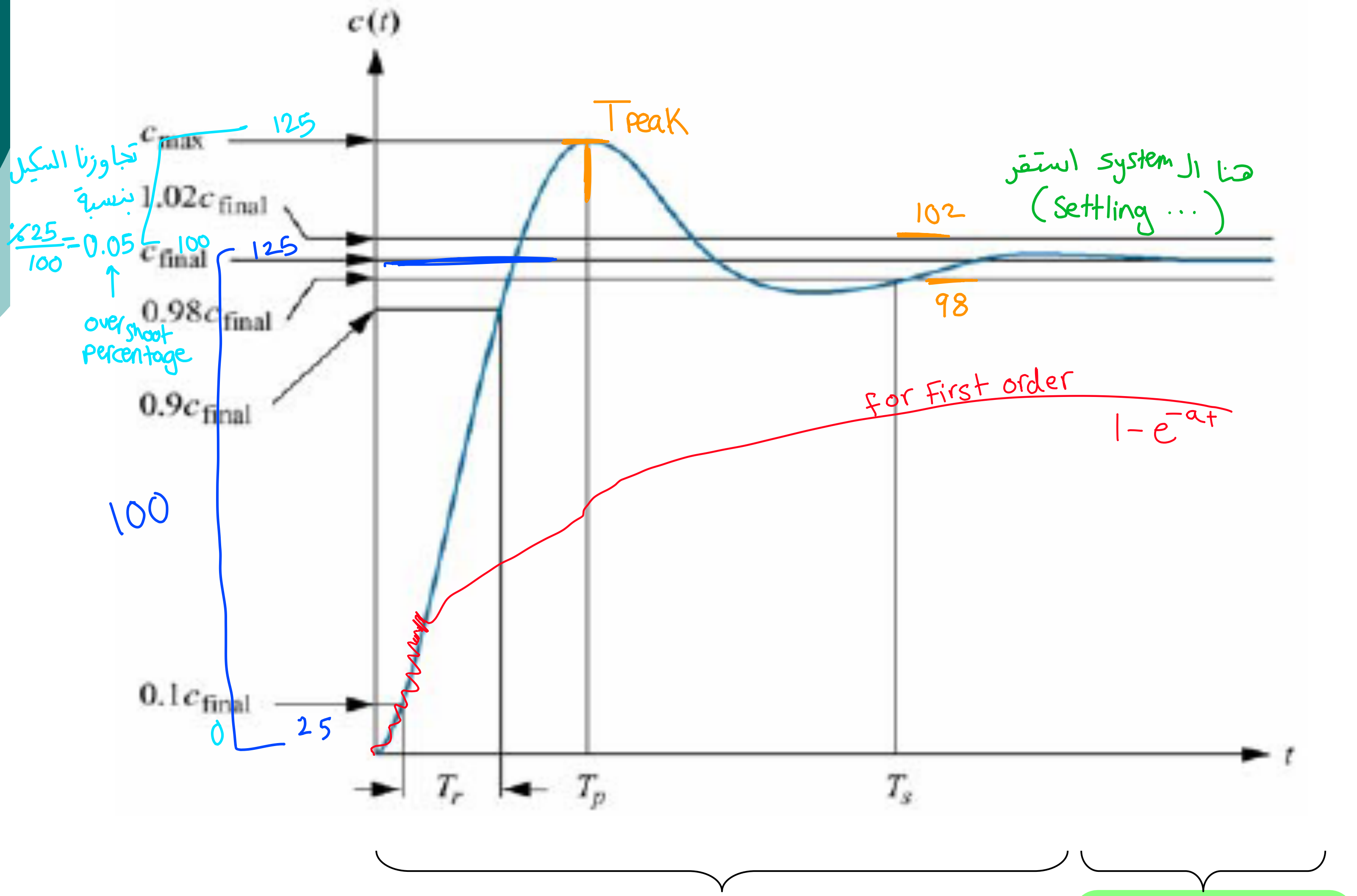
2. Steady-state response

   - the manner in which the system output behaves as *t* approaches infinity – the error after the transient response has decayed, leaving only the continuous response.

$t \to \infty$

Example of a General Response

Introduction



Transient

Steady-state

الـ exponent متناقص

First – order system

A first-order system without zeros can be represented by the following transfer function

الـ steady state يساوي واحد لهذه المعادلة →

$$\cancel{R(s)} \times \frac{C(s)}{\cancel{R(s)}} = \frac{1}{\tau s + 1} \times R(s)$$

تاو

ثم نحل عن الـ Partial fraction

$$\frac{1}{s+a} = \frac{1}{\tau}$$

- Given a step input, i.e., $R(s) = 1/s$, then the system output (called **step response** in this case) is

اذا كان repeated e^{-at}

زي حكاية first order

* كل جذر من الجزور بصير fraction و يساهم كل جذر بـ exponential (e^{-at}) أو $e^{-at}\sin(wt)$

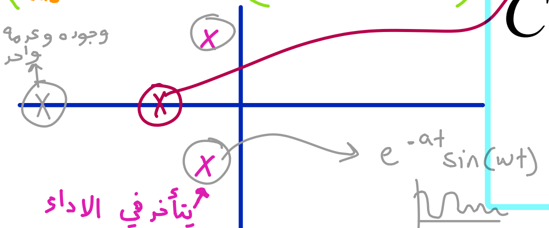
sin يكون بين الـ 0 والـ 1 فما بأثر كتير

* يلي بقرر تصرفات الـ system هو الـ (Dominant Poles)

(الجزور القريبة للـ imaginary axis)(الجذور التي تتأخر)

$$C(s) = \frac{1}{\tau s + 1} R(s) = \frac{1}{s(\tau s + 1)} = \frac{1}{s} - \frac{1}{s + \frac{1}{\tau}}$$

Taw = RC

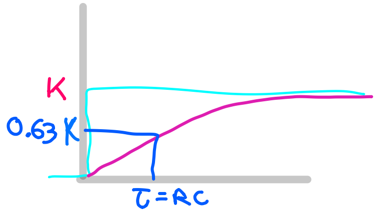


First – order system

Taking inverse Laplace transform, we have the step response

$$\boxed{c(t) = 1 - e^{-\frac{t}{\tau}}}$$

$\tau = RC$



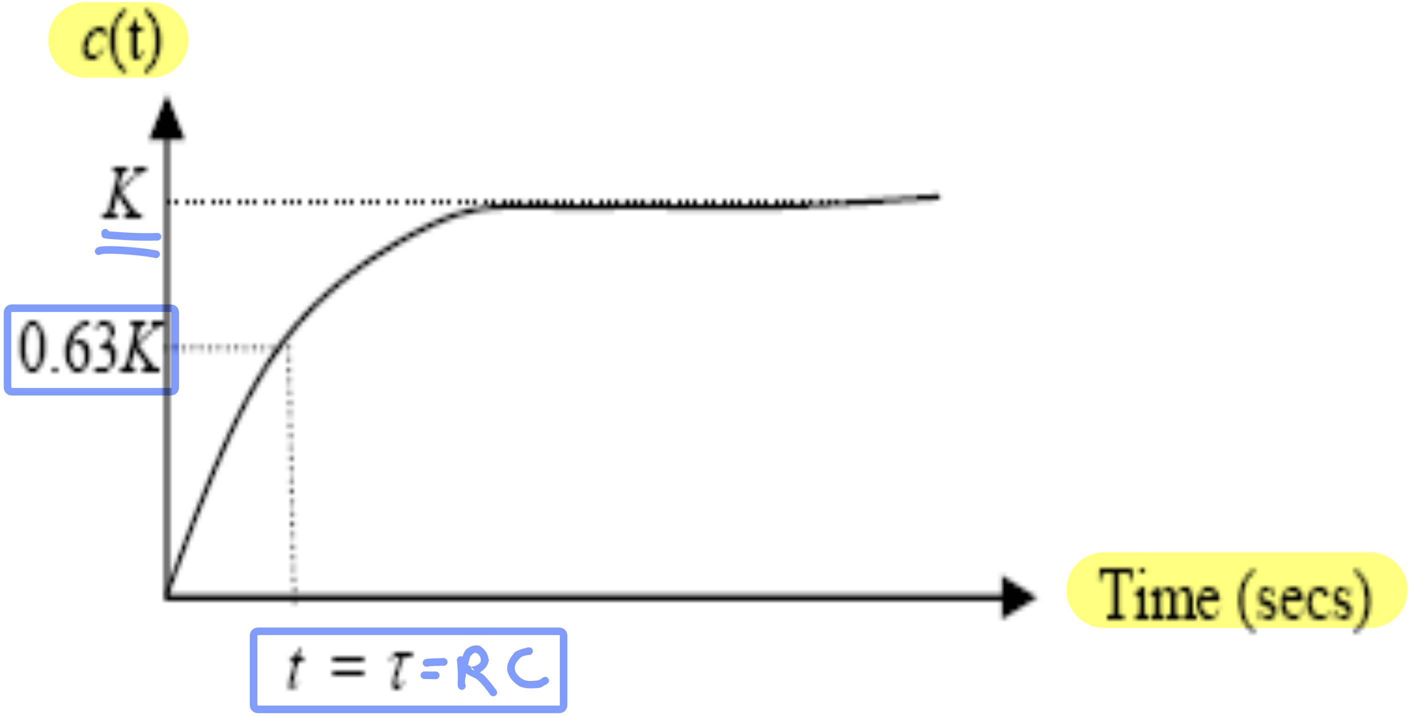
Time Constant: If t= τ , So the step response is

$$\boxed{C(\tau) = (1 - 0.37) = 0.63}$$

τ is referred to as the **time constant** of the response. In other words, the time constant is the time it takes for the step response to rise to 63% of its final value. Because of this, the time constant is used to measure how fast a system can respond. The time constant has a unit of seconds.

First – order system

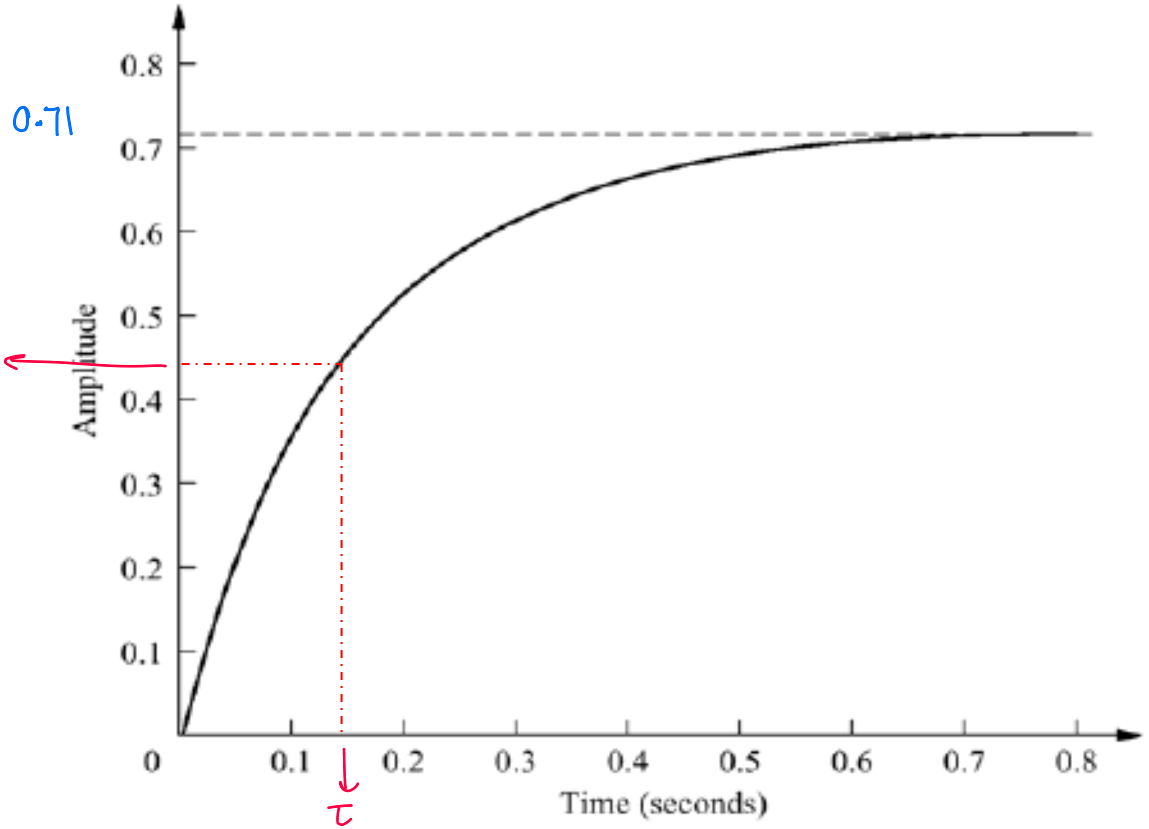
Plot *c*(*t*) versus time:



First – order system

Example 1

The following figure gives the measurements of the step response of a first-order system, find the transfer function of the system.



$K = 0.71$

$0.63 \times K$
$= 0.63 \times 0.71$
$= 0.4473$

τ
اظن

First – order system
Transient Response Analysis

Rise Time *Tr*:

The rise-time (symbol *Tr* units s) is defined as the time taken for the step response to go from **10% to 90%** of the final value.

اذا كان الـ system تبعي Dominat Pole
و كان single nonrepeated Pole بكون الـ
Tr=2.2τ ← Rise time
Ts= 4τ ← settling time

$$\boxed{T_r = 2.31\tau - 0.11\tau = 2.2\tau}$$

Settling Time *Ts*:

Defined the settling-time (symbol *Ts* units s) to be the time taken for the step response to come to within **2% of the final value** of the step response.

* كل ما كان مقدار الجزر اقل
كل ما كان الجزر ابطأ
يعني كل ما كان الـexponential ا صغر
يأخذ مدة اطول to settle ف همة الـDominant
و يلي بحاول اغير فيهم بحيث يعطيني performance أفضل

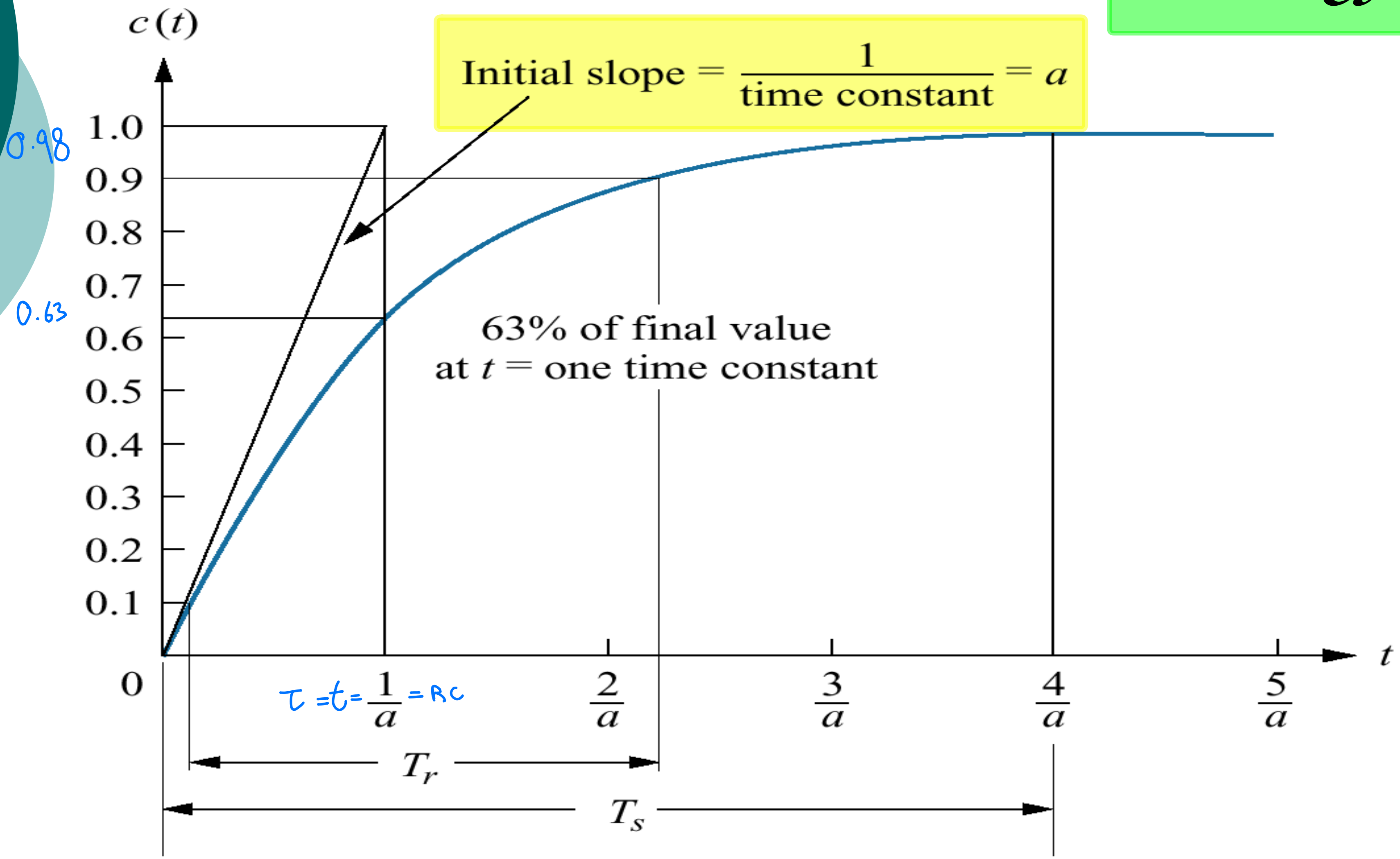
$\tau = \frac{1}{a}$

$$\boxed{T_s = \underline{\underline{4\tau}}} = \frac{4}{a}$$

معناه بعد 4τ بقدر اعتبر الـ system settled

First – order system

$$\tau = \frac{1}{a}$$



Initial slope $= \dfrac{1}{\text{time constant}} = a$

63% of final value at $t =$ one time constant

0.98

0.63

$\tau = t = \frac{1}{a} = RC$

Second – Order System

فرضية ال Autotuning Rule

مبنية على فرضية انه ال system أو first order / second order

- *Second-order systems* exhibit a wide range of responses which must be analyzed and described.
  - Whereas for a *first-order system*, varying a single parameter changes the speed of response, changes in the parameters of a *second order system* can change the form of the response.

- *For example*: a second-order system can display characteristics much like a first-order system or, depending on component values, display damped or *pure oscillations* for its *transient response*.

لازم تكون المعادلة complex حتى تكون second order

Second – Order System

التصميم مبني على فكرة ال complex because they make the system faster

★ (Real Poles) and (Repeated Poles) are weak

- A general second-order system is characterized by the following transfer function:

* كتير مهم نفهم ال complex *

$$G(s) = \frac{b}{s^2 + as + b}$$

لازم اخلص من معامل (s^2) اذا كان في مبقسم عليه

→ بهمني فقط المقام

$\omega_n = \sqrt{b}$

- We can re-write the above transfer function in the following form (closed loop transfer function):

$\frac{\cancel{M}s^2}{\cancel{M}} + \frac{\boxed{C}s}{M} + \frac{K}{M}$ → عشان اخلص من معامل s^2 قبل ما ابلش

واذا ما عندي Damper ما بيظهر عندي damping ratio

و دورها انها بتاكل ال Energy و بالتالي بتبطأ ال system

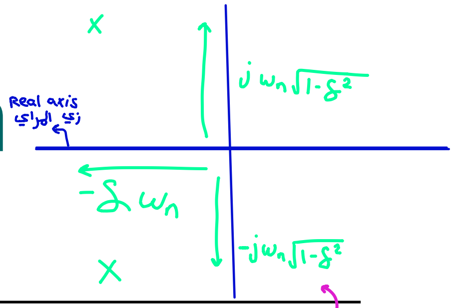
اذا ثبتت ال $\frac{K}{M}$ وغيرت ال c بكون عم ارسم دائرة وعم بغير باد θ و بالتالي Performance of vibration will change

The Damping Ratio zeta ζ

$\zeta = \frac{a}{2\sqrt{b}}$

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Second – Order System



memorize it

$\omega_n \quad (\omega_n = \sqrt{b}\,)$ - referred to as *the* *un-damped natural frequency* of the second order system, which is the frequency of oscillation of the system without damping.

تتأثر فقط بتغير b

لازم نحفظ عملية التحويل ←

$\zeta \quad (\zeta = \frac{a}{2\sqrt{b}})$ - referred to as *the* *damping ratio* of the second order system, which is a measure of the degree of resistance to change in the system output.

$\sqrt{-1} \times \sqrt{-1}$

$\sqrt{-1}$ j

$\Rightarrow (-\omega_n\zeta + j\omega_n\sqrt{1-\zeta^2}) \rightarrow$ عشان اقدر ارسمه على الcomplex plane

Poles;

$$-\omega_n\zeta + \omega_n\sqrt{\zeta^2 - 1}$$
$$-\omega_n\zeta - \omega_n\sqrt{\zeta^2 - 1}$$

Poles are complex if ζ< 1!

لازم نحفظ المعادلة ←

Damping Ratio

Second – Order System

- According the value of ζ, a second-order system can be set into one of the four categories:

damping ←

بزيادة ذا'' هي مش مقبولة

1. *Overdamped* - when the system has two real distinct poles ($\zeta > 1$).
2. *Underdamped* - when the system has two complex conjugate poles ($0 < \zeta < 1$)

→ أفضل اشي

3. *Undamped* - when the system has two imaginary poles ($\zeta = 0$).

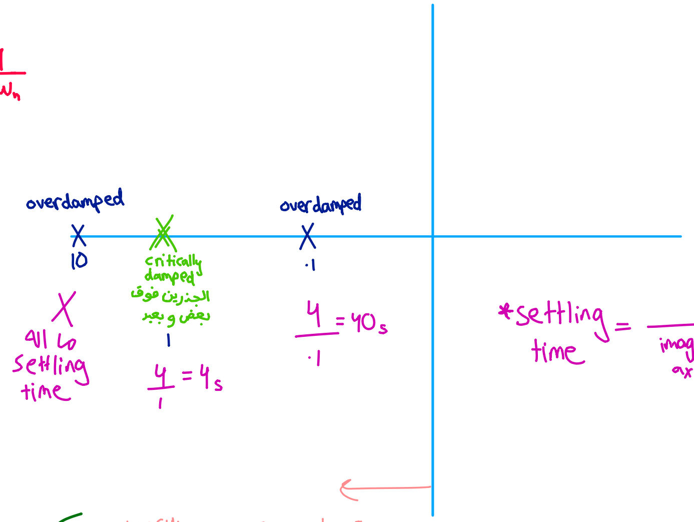
→ UnDamped
هنا الجذور جاي على ال
imaginary axis بالزبط

* Remember *
البعد عن ال imaginary axis
هو ($\zeta\omega_n$) فهناهو صفر

4. *Critically damped* - when the system has two real but equal poles ($\zeta = 1$).

حدوثهم في
الواقع هو صدفة

$T_s = \frac{4}{\zeta \omega_n}$



ما ال settling time

$\frac{4}{1} = 4s$

$\frac{4}{\cdot 1} = 40s$

*settling time = $\frac{4}{\text{اقرب جزر للimaginary axis}}$

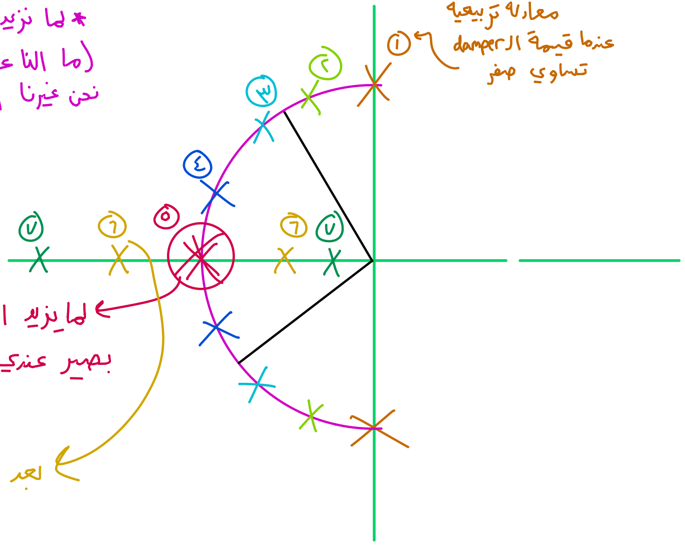
كلما بعدت عن الimaginary axis

كلما كان Settling time أفضل ؟

* مو متأكدة

* لما نزيد ال damper

(ما النا علاقة بال K نحن غيرنا ال c)



① عندما قيمة الdamper تساوي صفر

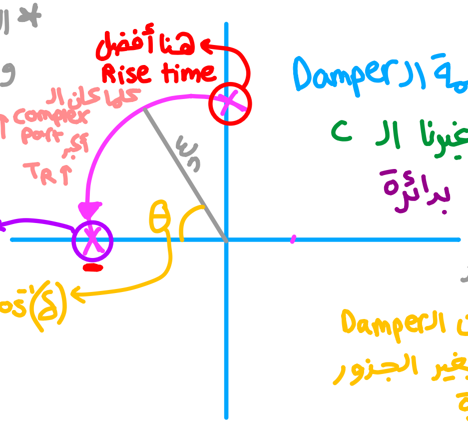
معادلة تربيعية

لما يزيد ال damping لمرحلة معينة بصير عندي جذرين فوق بعض في المنطقة هاي

لعد ما زدنا ال damping أكثر

* البعد عن المركز هو W_n

و بنغير الdamper و الزاوية



هنا أفضل Rise time

كلما كان ال complex part أكبر ↑

TR ↑

عندما نصل هذه النقطة يكون Critically Damped

$\theta = \cos^{-1}(\zeta)$

مع تغيير قيمة الDamper

ما غيرنا ال K غيرنا ال c

اذاً نحن نتحرك بدائرة

فالزاوية سوف تتغير

وال ζ محصورة من الDamper

فلما اغيره بكون بغير الجذور بصورة نص دائرة

و هنا يكون أفضل Settling time

مدري

كلما بعدنا عن ال imaginary axis أكثر كان ال Settling time أكثر

Time-Domain Specification

Given that the closed loop TF

$$T(s)=\frac{C(s)}{R(s)}=\frac{\omega_n^{\ 2}}{s^2+2\varsigma\omega_n s+\omega_n^{\ 2}}$$

× R(s)

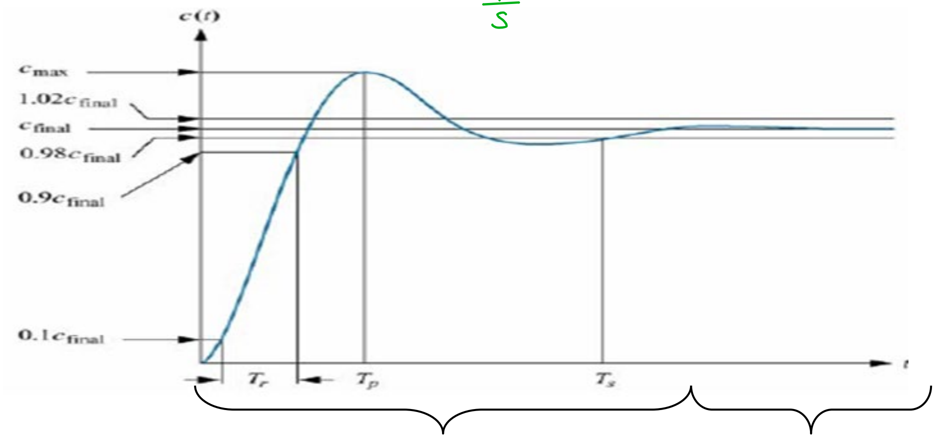
$\frac{1}{s}$ عوض

The system (2nd order system) is parameterized by ς and ω_n

For $\boxed{0< \varsigma <1}$ and $\omega_n > 0$, we like to investigate its response due to a $\underbrace{\text{unit step input}}_{\frac{1}{s}}$

$$C(t) = 1 - \frac{1}{\sqrt{1-\varsigma^2}} \times e^{-\varsigma\omega_n t} \sin(\omega_n \sqrt{1-\varsigma^2}\, t + \theta)$$

complex part
يؤثر في الاهتزازات



Transient Steady State

Two types of responses that are of interest:
(A)Transient response
(B)Steady state response

(A) For transient response, we have 4 specifications:

(a) T_r – rise time = $\dfrac{\pi-\theta}{\omega_n\sqrt{1-\varsigma^2}}$ θ in radians

يتأثر بالجزء ال complex part وكلما زاد قل ال Rise time

(b) T_p – peak time = $\dfrac{\pi}{\left(\omega_n\sqrt{1-\varsigma^2}\right)}$ complex part

(c) %MP – percentage maximum overshoot = $e^{-\frac{\pi\varsigma}{\sqrt{1-\varsigma^2}}} x100\%$

لها علاقة بالزاوية وال damping ratio

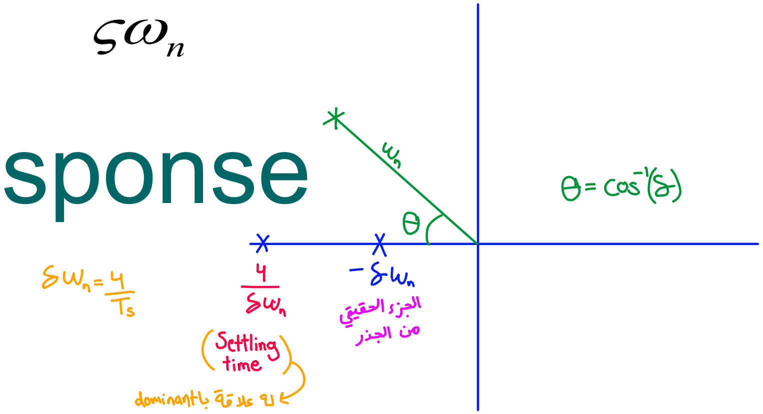
(d) T_s – settling time (2% error) = $\dfrac{4}{\varsigma\omega_n}$

متعلق بالجزء الحقيقي للجذر وما دخله بال complex

Real part

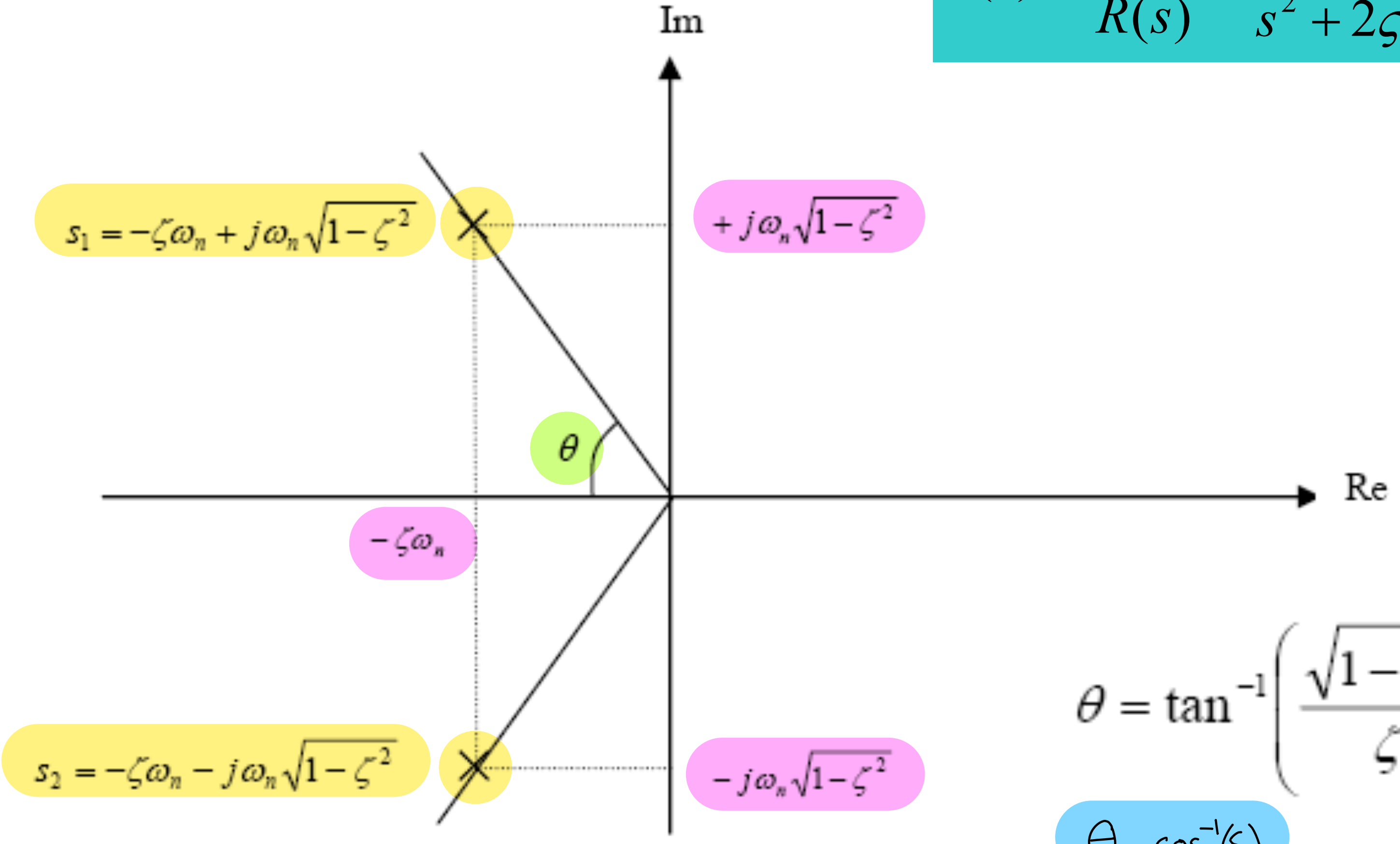
(B) Steady State Response

(a) Steady State error



Second – Order System

$$T(s)=\frac{C(s)}{R(s)}=\frac{\omega_n{}^2}{s^2+2\varsigma\omega_n s+\omega_n{}^2}$$



$$\theta=\tan^{-1}\left(\frac{\sqrt{1-\varsigma^2}}{\varsigma}\right)$$

$\theta=\cos^{-1}(\zeta)$

Mapping the poles into s-plane

Therefore,

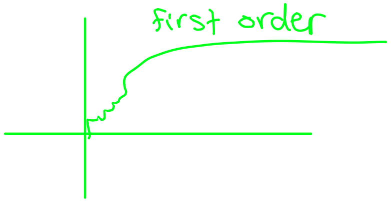
$$\%MP = e^{-\frac{\pi\varsigma}{\sqrt{1-\varsigma^2}}} x100\%$$

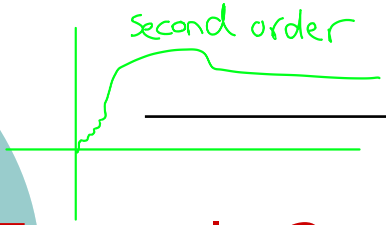
ς

30 % = .3

- For given %*OS*, the damping ratio can be solved from the above equation;

$$\varsigma = \frac{-\ln(\%MP/100)}{\sqrt{\pi^2+\ln^2(\%MP/100)}}$$





UNDERDAMPED

ω_n

Example 2: *Find the natural frequency and damping ratio for the system with transfer function*

ζ

$$G(s)=\frac{36}{s^2+4.2s+36}$$

→ البسط ننساه

↑ $b=6$

Solution:

Compare with general TF

$\frac{4.2}{12}$

$$G(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

- ωn= 6 $=\sqrt{b}=\sqrt{36}$
- ξ =0.35 $=\frac{a}{2\sqrt{b}}=\frac{4.2}{2\sqrt{36}}$

UNDERDAMPED

Example 3: Given the transfer function

$$G(s) = \frac{100}{s^2 + \underline{15}s + 100}$$

$\frac{4}{7.5}$ ← نص ال 15

find $\quad T_s, \%OS, T_p$

$W_n = \sqrt{} = 10$

$\zeta = \frac{15}{20}$

Solution:

$w_n = \sqrt{100}$ $\quad$ $\zeta = \frac{15}{2 \times 10}$

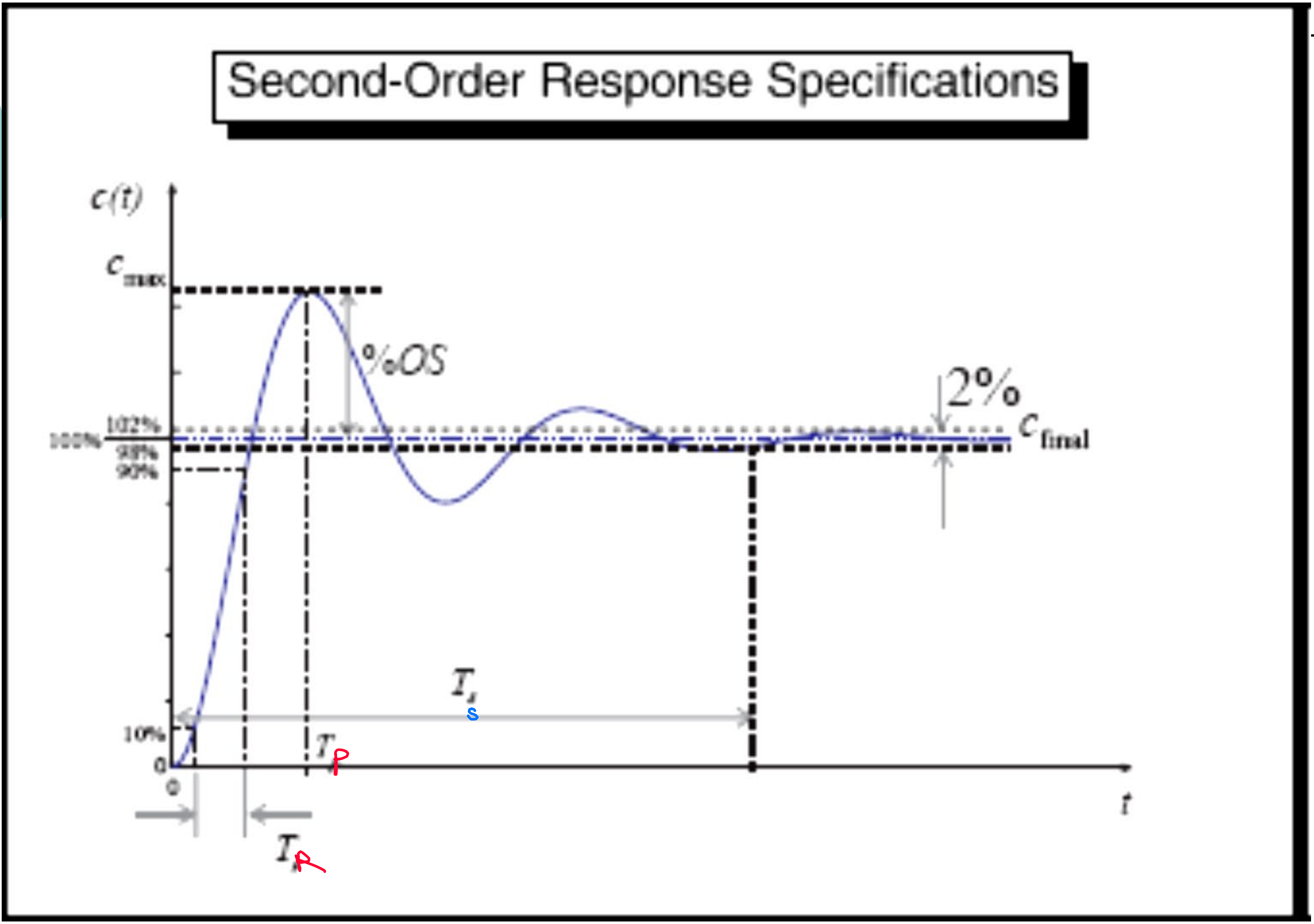
$$\omega_n = 10 \qquad \xi = 0.75$$

second

$$T_s = 0.533\,s,\ \%OS = 2.838\%,\ T_p = 0.475s$$

$T_s = \frac{4}{\zeta w_n}$ $\quad$ $\%OS = e^{\frac{-\pi \zeta}{\sqrt{1-\zeta^2}}} \times 100\,\%$ $\quad$ $T_p = \frac{\pi}{w_n \sqrt{1-\zeta^2}}$

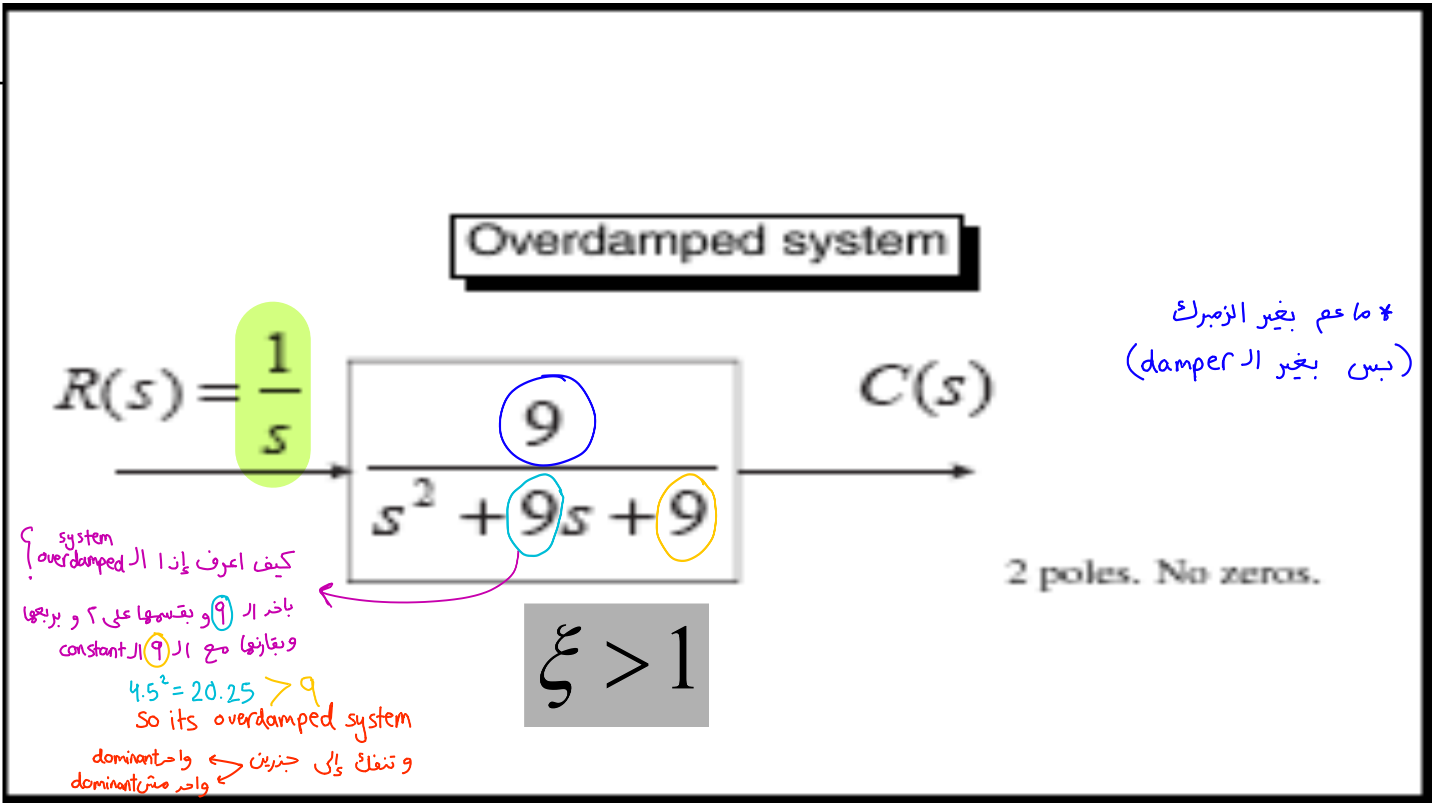
UNDERDAMPED



Overdamped Response

$$G(s) = \frac{b}{s^2 + as + b}$$

a = 9



* ماعم بغير الزمبرك
(بس بغير ال damper)

كيف اعرف إذا ال overdamped system ؟

باخذ ال ⑨ و بقسمها على ٢ و بربعها
وبقارنها مع ال ⑨ الconstant

$4.5^2 = 20.25 > 9$

So its overdamped system

و تنفك إلى جذرين → واحد dominant
→ واحد مش dominant

$$C(s) = \frac{9}{s(s^2 + 9s + 9)} = \frac{9}{s(s + 7.854)\underbrace{(s + 1.146)}}$$

$\frac{4}{7.854} = 0.50$ $\frac{4}{1.146} = 3.49$

→ This is the dominant
وسوف يتأخر لحد مايختفي

s= 0; s = -7.854; s = -1.146 (*two real poles)*

بعد نص second هذا الجذر يهمل

اذا نفرضها مهملة

$$c(t)=K_1+\underline{\underline{K_2e^{-7.854t}}}+K_3e^{-1.146t}$$

ممكن يجيب السؤال هاي الرسمة لحالها ويسألنا عن

Natural frequency $= \sqrt{1.5^2+2.5^2}$

Damping Ratio $= \frac{1.5}{3}$

complex part $= w_n\sqrt{\ldots}$

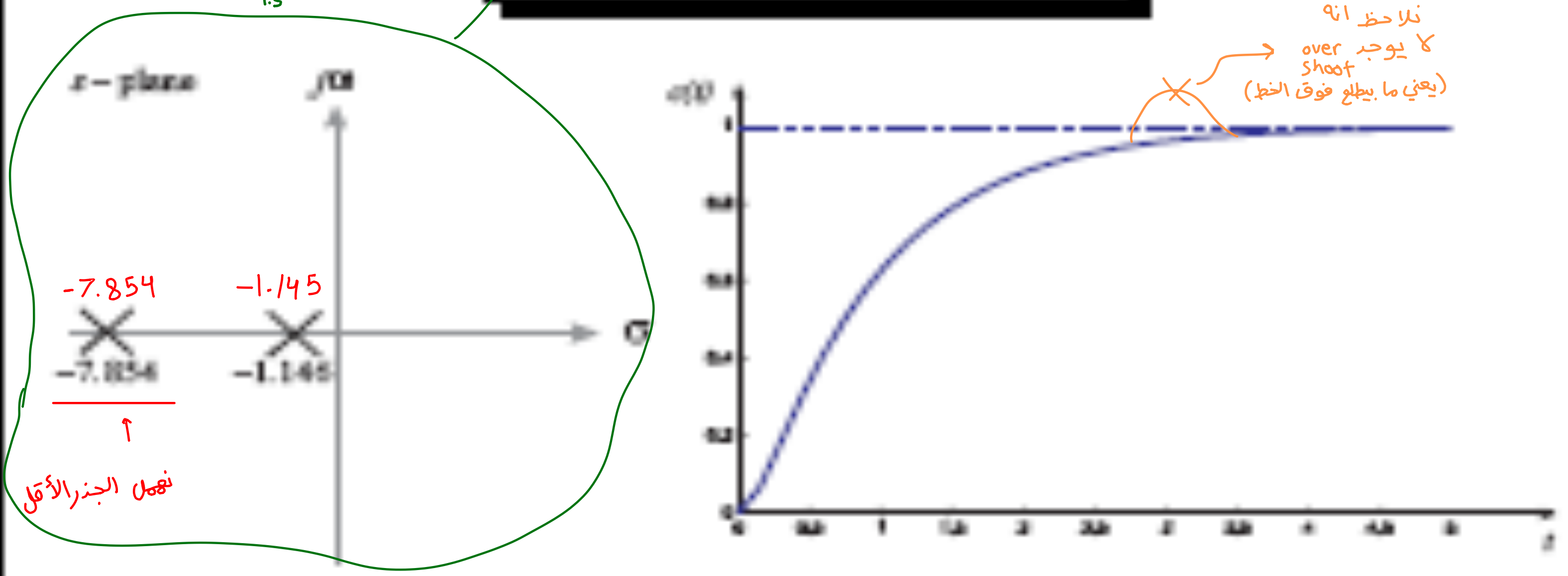
Peak time $= \frac{4}{2.598}$

settling time $= \frac{4}{1.5}$

يعني زي كأنه ال system

$c(t) = K_1 + K_3 e^{-1.146t}$

Overdamped response



نلاحظ انه

لا يوجد over shoot

(يعني ما بيطلع فوق الخط)

نهمل الجذر الأقل

OVERDAMPED RESPONSE !!!

الاكبر

Settling time

Underdamped Response

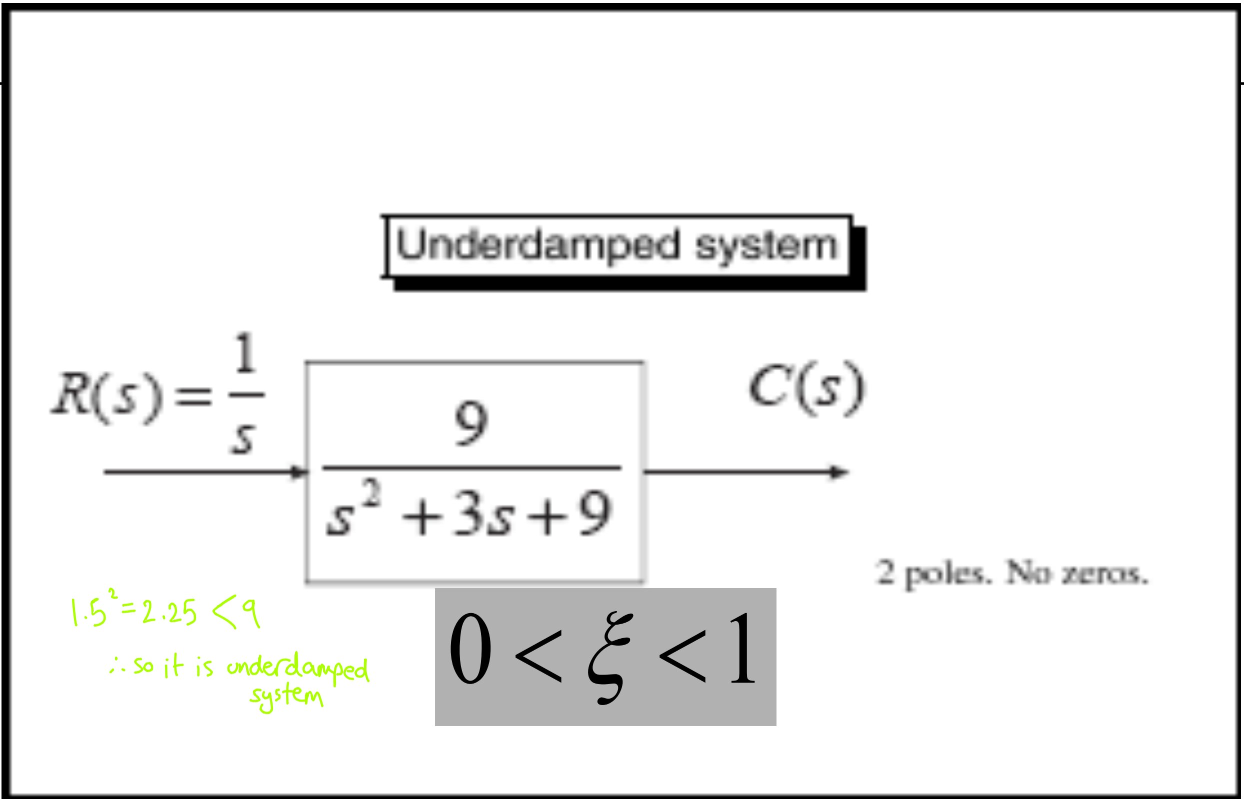
$\rightarrow (0 < \delta < 1)$

and we will have two complex conjegute poles

$$G(s) = \frac{b}{s^2 + as + b}$$

a = 3

$$\frac{9}{s(-1.5 \pm 2.59i)}$$



$1.5^2 = 2.25 < 9$

∴ so it is underdamped system

$$c(t) = K_1 + \underbrace{e^{-1.5t}}_{\text{settling time}} \left(K_2 \cos \underbrace{2.598t}_{\frac{\pi}{2.598}} + K_3 \sin 2.598t \right)$$

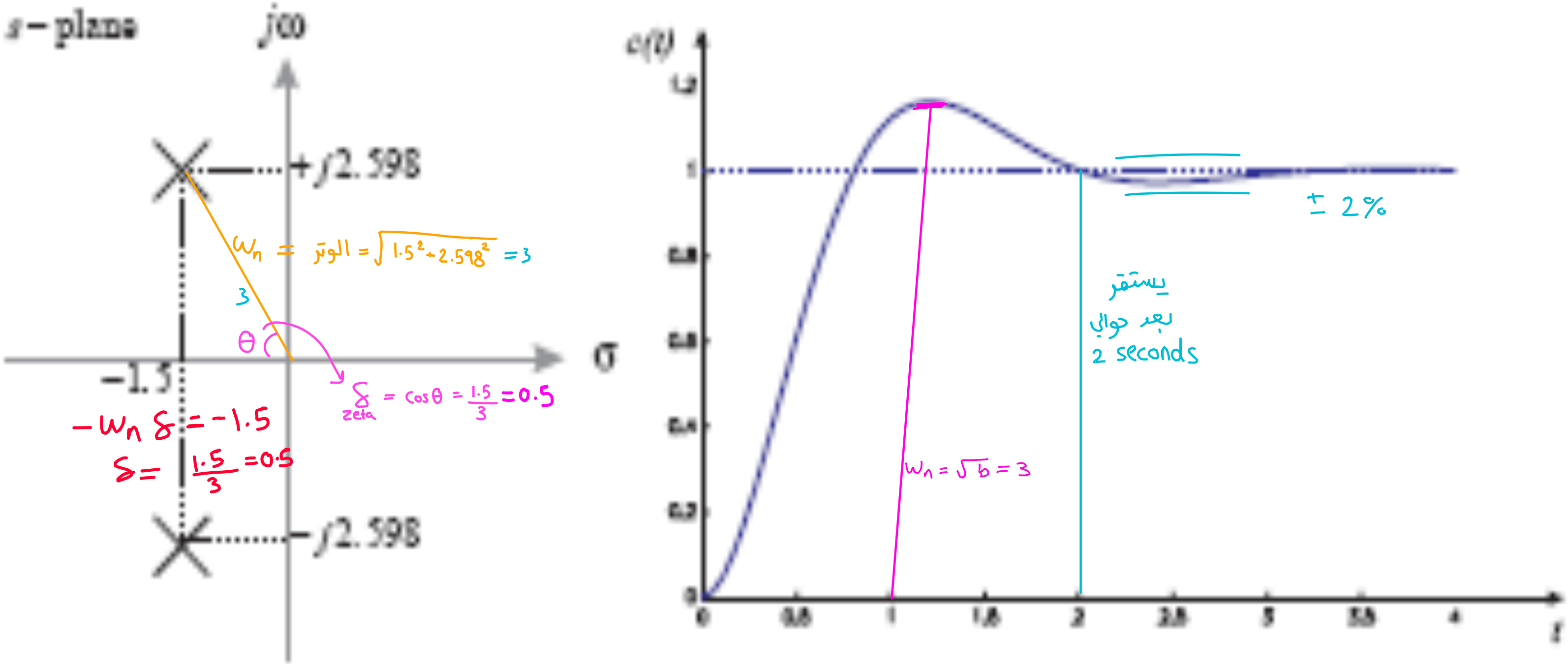
settling time $= \frac{4}{1.5} = \boxed{2.66}$

إذا ال settling time هنا أخفض من ال overdamped

s = 0; s = -1.5 ± j2.598 (*two complex poles)*

Underdamped response

لأن ال 0 < Damping Ratio < 1



UNDERDAMPED RESPONSE !!!

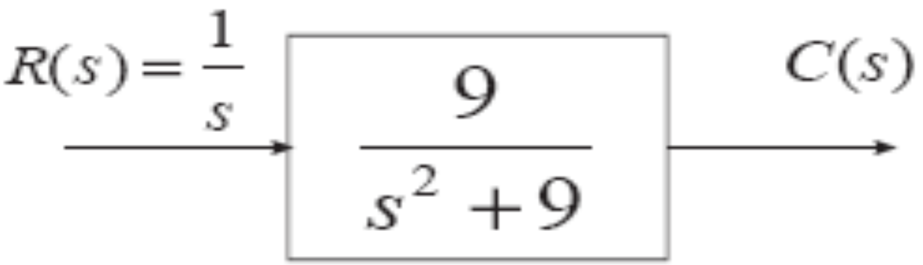
Undamped Response

ما فيو damping بالمرة
وبكون ال Rise time سريع جداً
بس مشكلته انه ما بستقر

$$G(s) = \frac{b}{s^2 + as + b}$$

a = 0

Undamped system



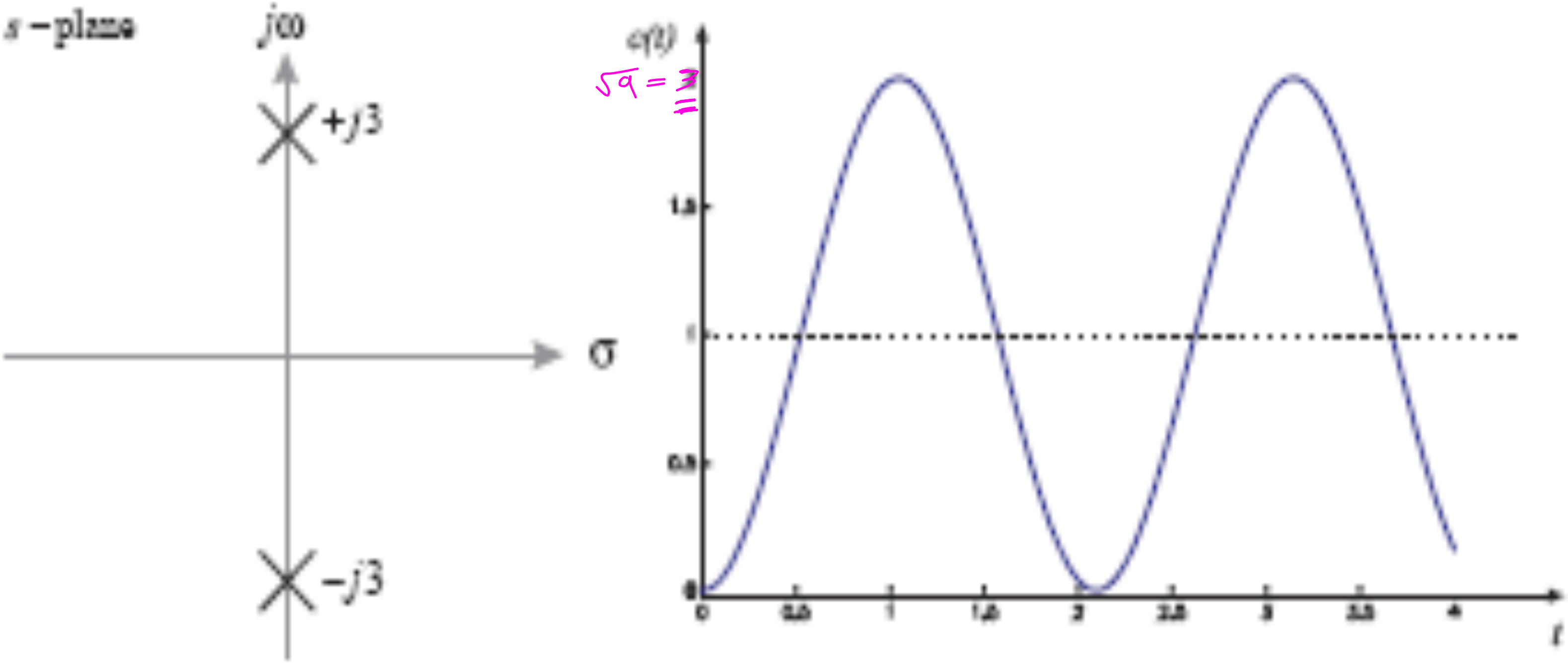
2 poles. No zeros.

$$\xi = 0$$

$$c(t) = K_1 + K_2 \cos 3t$$

s = 0; s = ± j3 *(two imaginary poles)*

Undamped response



$\sqrt{9} = \underline{\underline{3}}$

UNDAMPED RESPONSE !!!

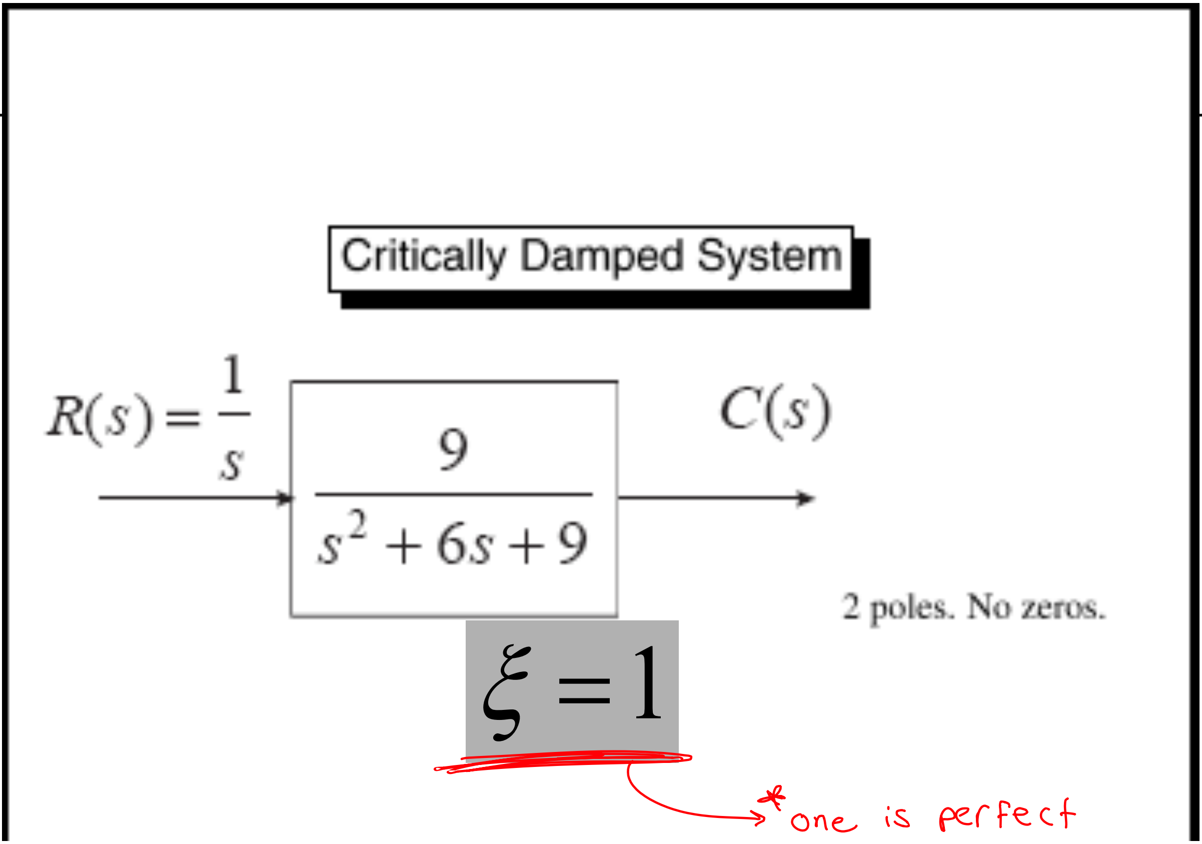
Critically Damped System

→ منيح جداً
لكن صعب نجيبه

$$G(s) = \frac{b}{s^2 + as + b}$$

Response سريع
Rise time سريع
Settling time $\frac{4}{3}$

a = 6

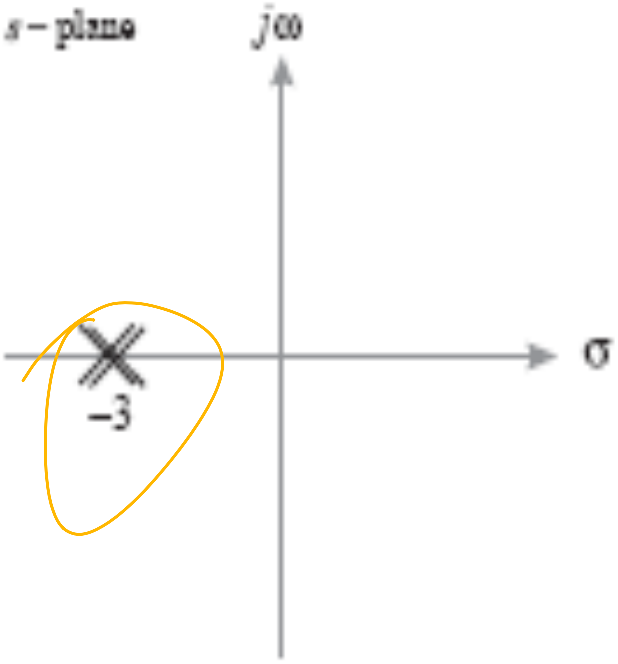


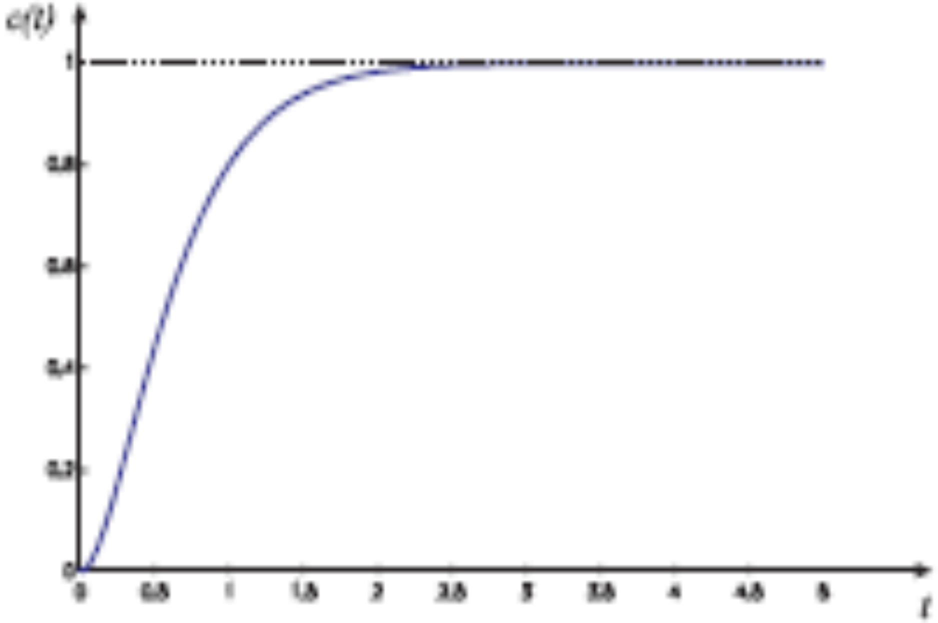
*one is perfect
*more than one is not good

$$c(t) = K_1 + K_2 e^{-3t} + K_3 t e^{-3t}$$

S = 0; s = -3,-3 (*two real and equal poles)*

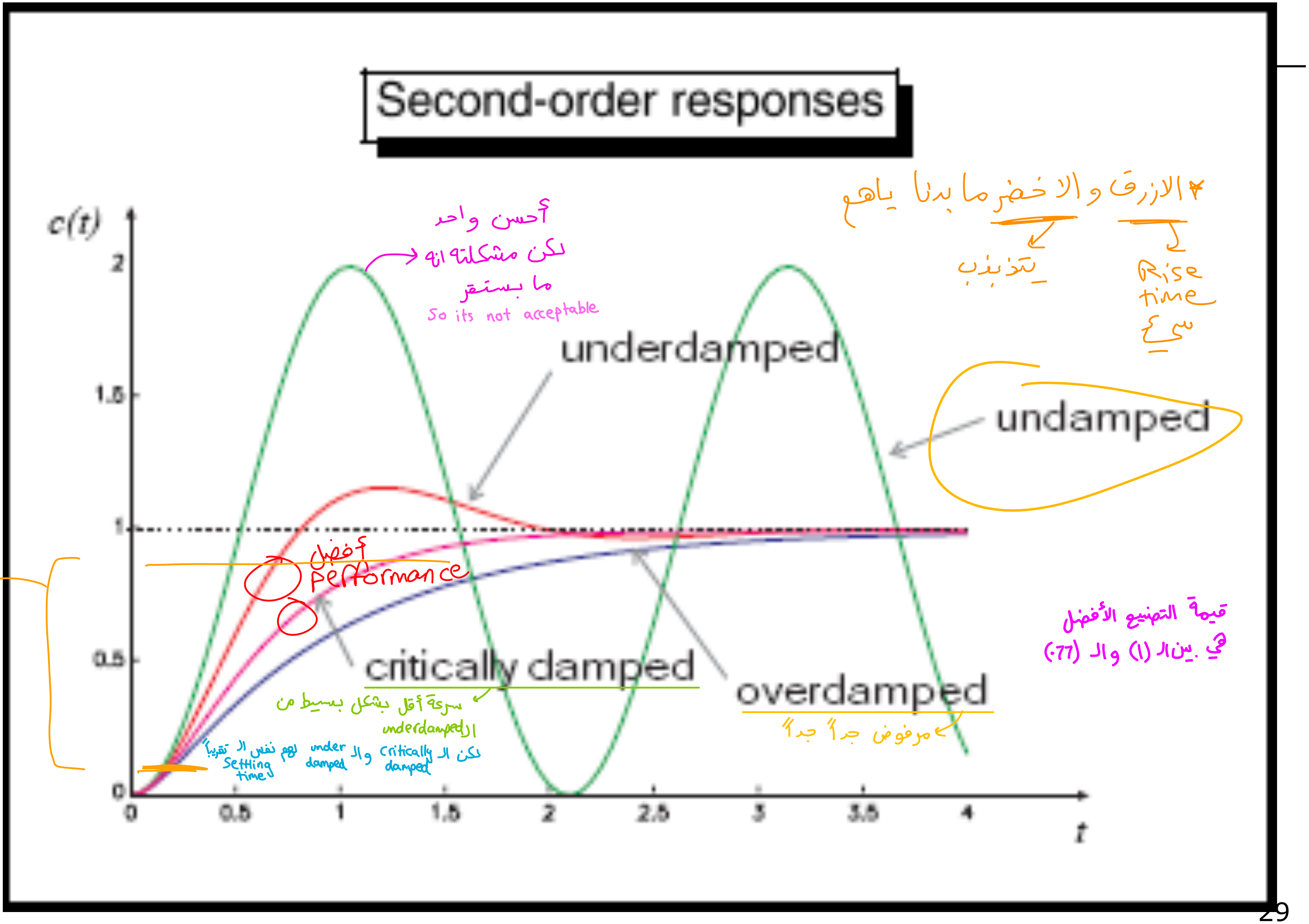
Critically Damped Response





CRITICALLY DAMPED RESPONSE !!!

Second – Order System



أحسن واحد
لكن مشكلته انه
ما بستقر
So its not acceptable

* الازرق والاخضر ما بدنا ياهم

تذبذب

Rise time
سريع

أفضل performance

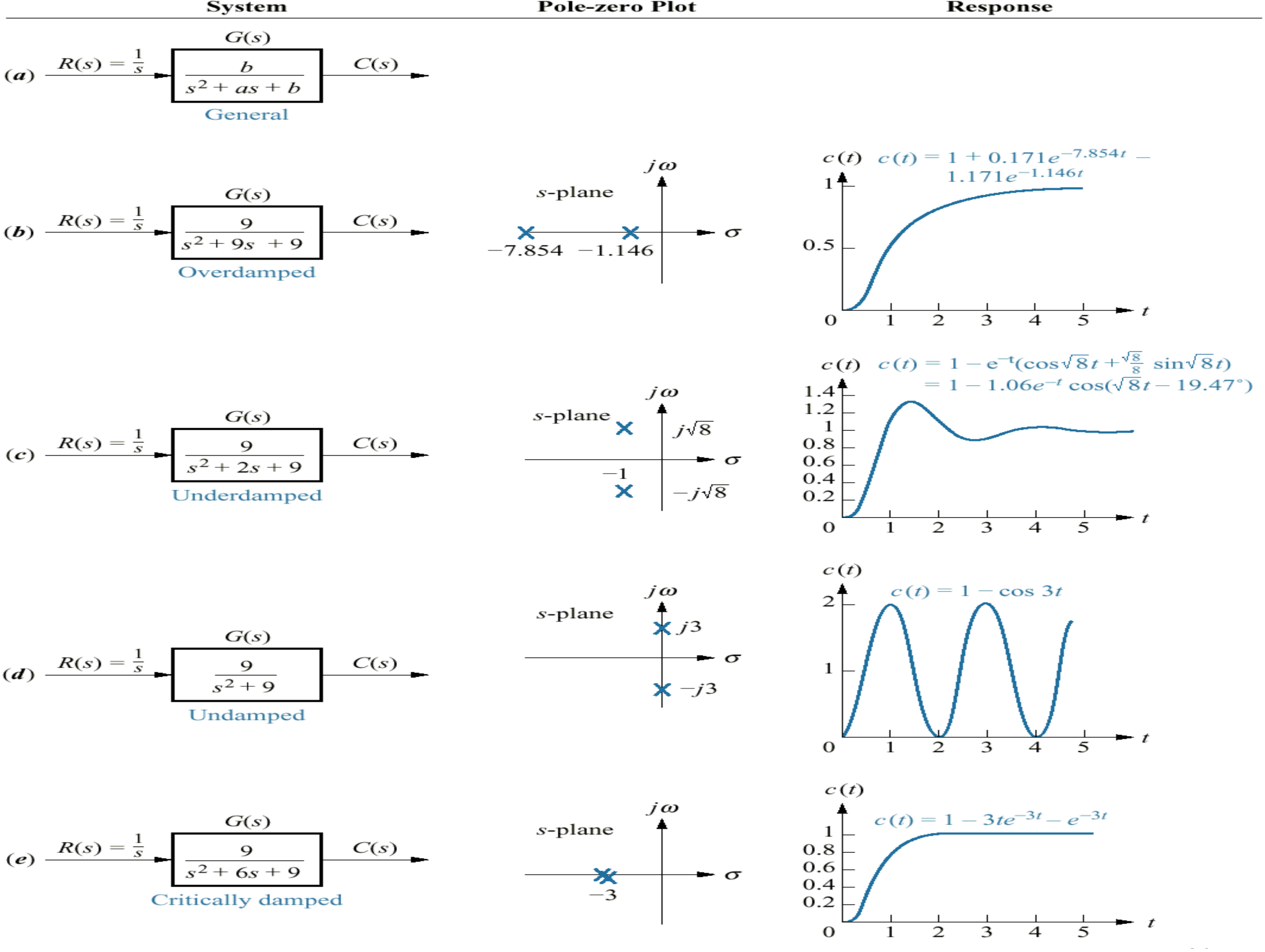
Rise time
هو الانطلاق
من ١٠% إلى ٩٠%

سرعة أقل بشكل بسيط من
الunderdamped

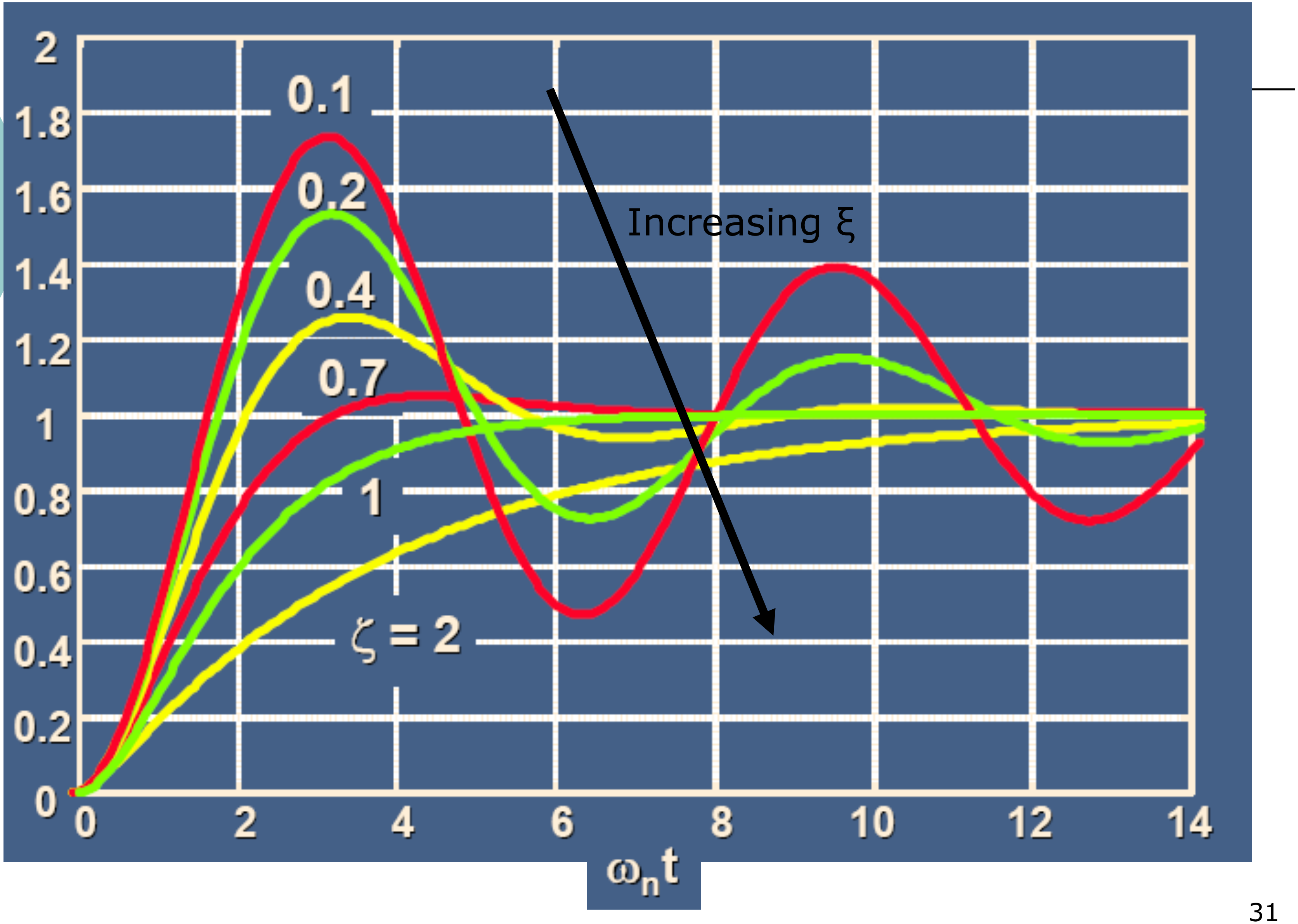
لكن الـ critically damped والـ under damped لهم نفس الـ Settling time تقريباً

مرفوض جداً

قيمة التضميع الأفضل
هي بين الـ (1) والـ (.77)


System
Pole-zero Plot
Response
(a) $R(s) = \frac{1}{s}$, $G(s) = \frac{b}{s^2 + as + b}$, $C(s)$
General
(b) $R(s) = \frac{1}{s}$, $G(s) = \frac{9}{s^2 + 9s + 9}$, $C(s)$
Overdamped
s-plane, jω, σ, −7.854, −1.146
$c(t) = 1 + 0.171e^{-7.854t} - 1.171e^{-1.146t}$
(c) $R(s) = \frac{1}{s}$, $G(s) = \frac{9}{s^2 + 2s + 9}$, $C(s)$
Underdamped
s-plane, jω, σ, −1, $j\sqrt{8}$, $-j\sqrt{8}$
$c(t) = 1 - e^{-t}(\cos\sqrt{8}t + \frac{\sqrt{8}}{8}\sin\sqrt{8}t)$
$= 1 - 1.06e^{-t}\cos(\sqrt{8}t - 19.47°)$
(d) $R(s) = \frac{1}{s}$, $G(s) = \frac{9}{s^2 + 9}$, $C(s)$
Undamped
s-plane, jω, σ, j3, −j3
$c(t) = 1 - \cos 3t$
(e) $R(s) = \frac{1}{s}$, $G(s) = \frac{9}{s^2 + 6s + 9}$, $C(s)$
Critically damped
s-plane, jω, σ, −3
$c(t) = 1 - 3te^{-3t} - e^{-3t}$
c(t), t, 0, 1, 2, 3, 4, 5

Effect of different damping ratio, ξ



Second – Order System

Example 4: Describe the nature of the second-order system response via the value of the damping ratio for the systems with transfer function

$$1.\quad G(s) = \frac{12}{s^2 + \underset{a}{8}s + \underset{b}{12}}$$

$w_n = \sqrt{b} = \sqrt{12} = 3.46$

$\zeta = \frac{a}{2\sqrt{b}} = \frac{8}{2 \times 3.46} = 1.15 \longrightarrow$ So its overdamped

$$2.\quad G(s) = \frac{16}{s^2 + 8s + 16}$$

Do them as your own revision

$$3.\quad G(s) = \frac{20}{s^2 + 8s + 20}$$

Chapter 4

Transient & Steady State Response Analysis

Previous Class

- Chapter 4:
  - First Order System
  - Second Order System

Today's class

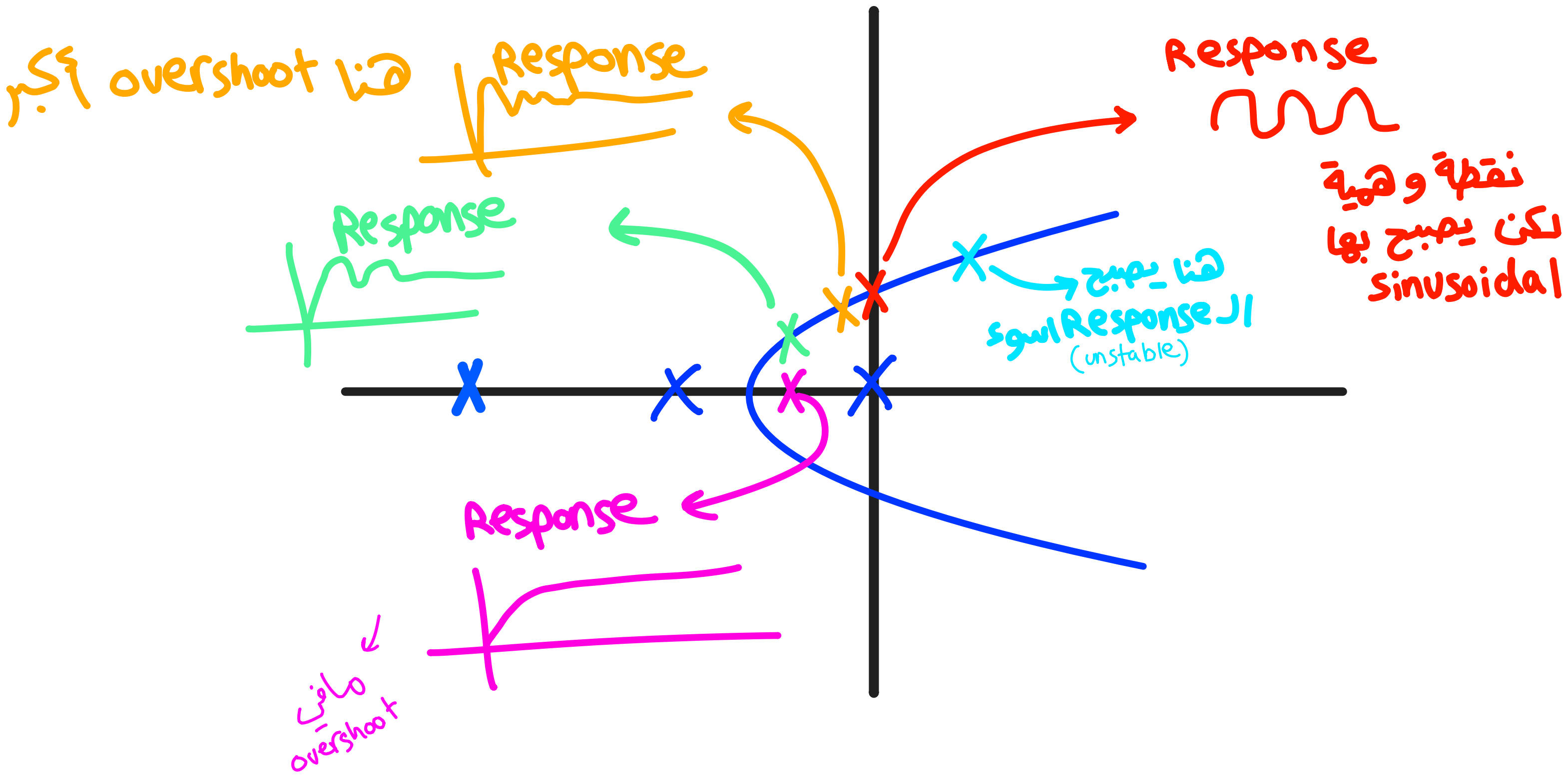
- Routh-Hurtwitz Criterion
- Steady-state error

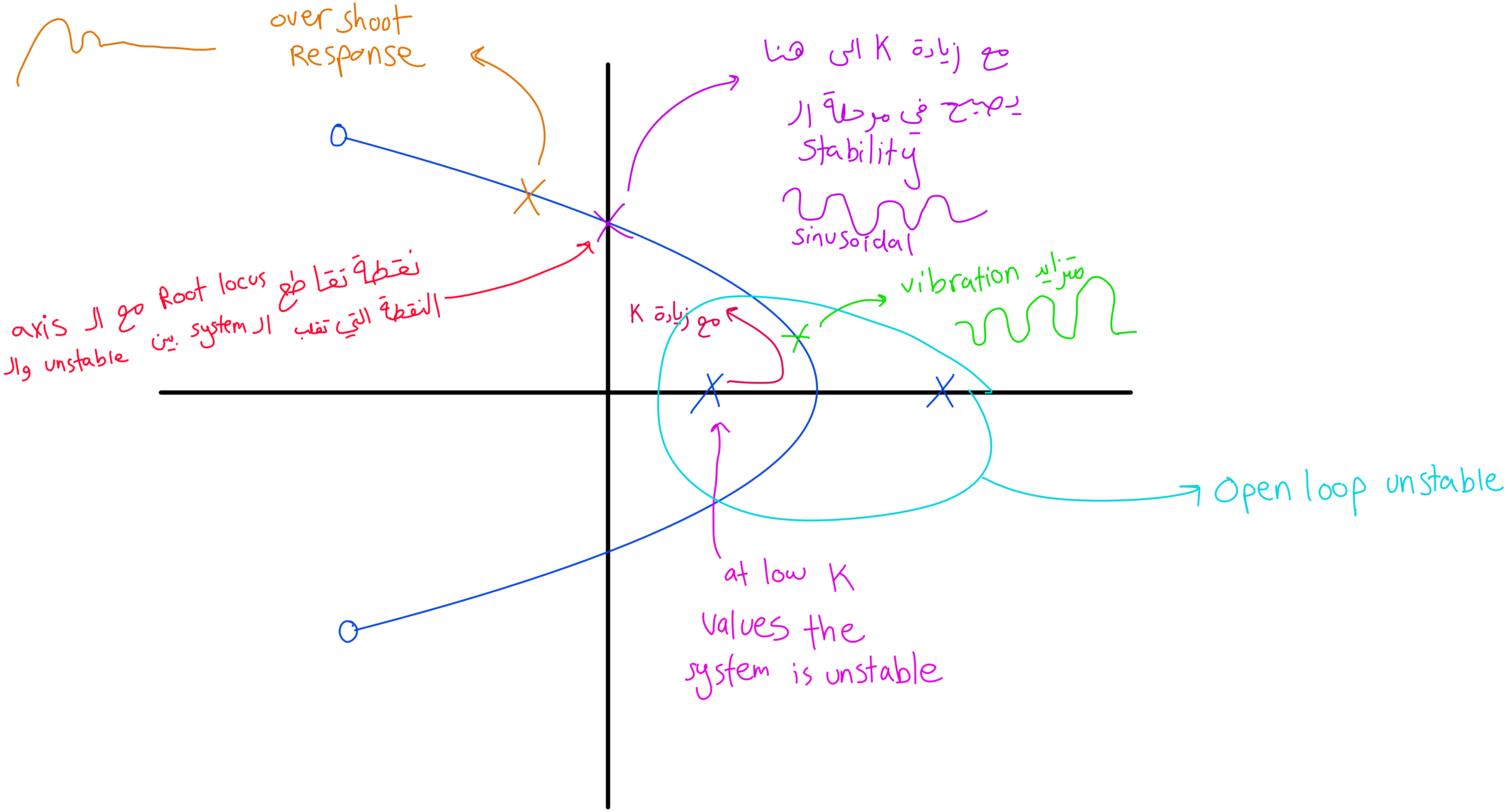
يكون بعد ال Settling time واستقر

و ال Error هو فرق بين ال Actual و ال Required

Routh-Hurwitz Criterion

To check for stability of a system


هنا overshoot أكبر
Response
Response
Response
نقطة وهمية
لكن يصبح بها
sinusoidal
هنا يصبح
الResponse اسوء
(unstable)
Response
مافي
overshoot



overshoot Response

نقطة تقاطع Root locus مع ال axis

النقطة التي تقلب ال system بين unstable وال stable

مع زيادة K الى هنا

يصبح في مرحلة ال Stability

sinusoidal

vibration متزايد

مع زيادة K

at low K values the system is unstable

Open loop unstable

Stability

12 min

- in order to know the location of the poles, we need to find the roots of the closed-loop characteristic equation.

- It turned out, however, that in order to judge a system's stability we don't need to know the actual location of the poles, just their sign. that is whether the poles are in the right-half or left-half plane.

- The Hurwitz criterion can be used to indicate that a characteristic polynomial with negative or missing coefficients is unstable.

لازم نفك ال Loop عن طريق Block diagram reduction

- The *Routh-Hurwitz Criterion* is called a *necessary* and *sufficient test* of stability because a polynomial that satisfies the criterion is guaranteed to stable. The criterion can also tell us how many poles are in the right-half plane or on the imaginary axis.

Stability

في ال Root locus و ال Steady State error استخدمنا ال (G) open loop response

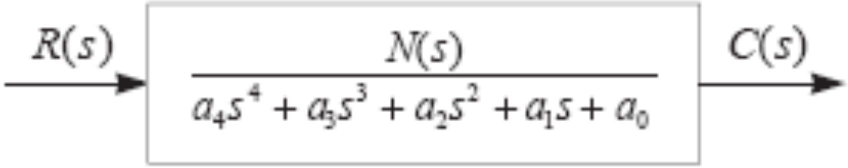
- need to construct a Routh array.

*اما هنا يجب أن نفك ال loop
نفك الجذر ونأخذ معادلة المقام كما هي

Consider the system shown in the Figure. The closed-loop characteristic equation is:

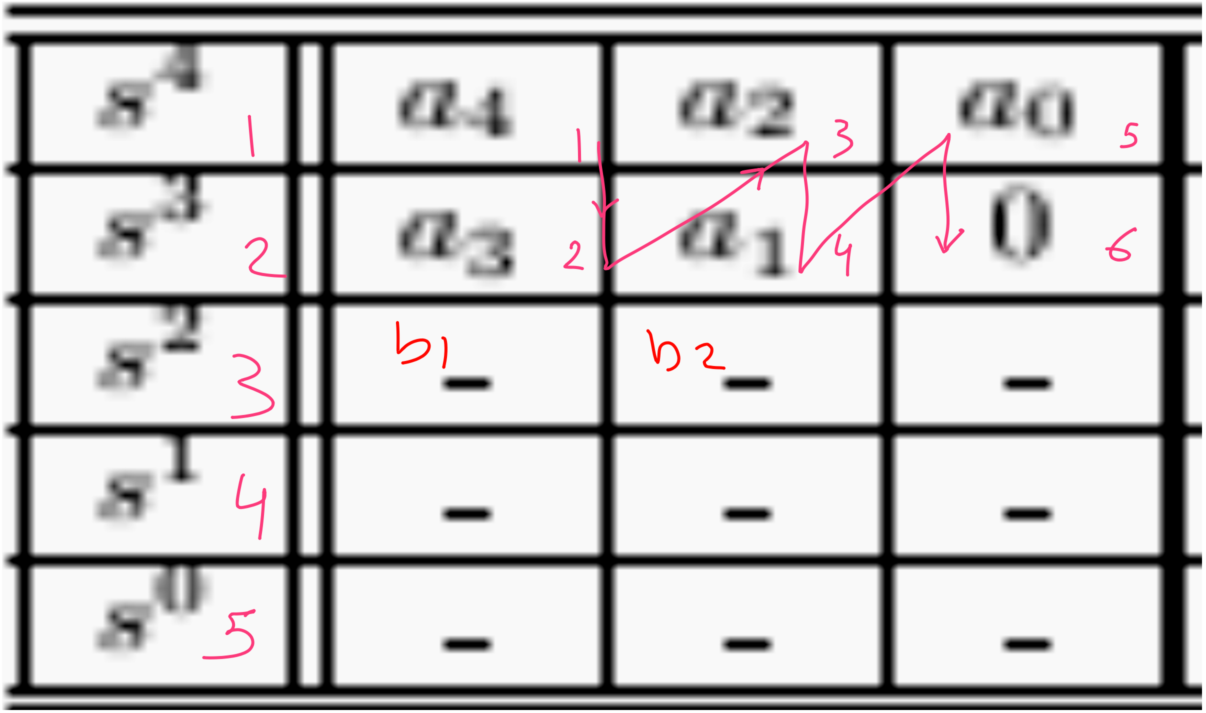
معادلة المقام

$$a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0 = 0.$$



- The Routh array is simply a rectangular matrix with one row for each power of *s* in the closed-loop characteristic polynomial

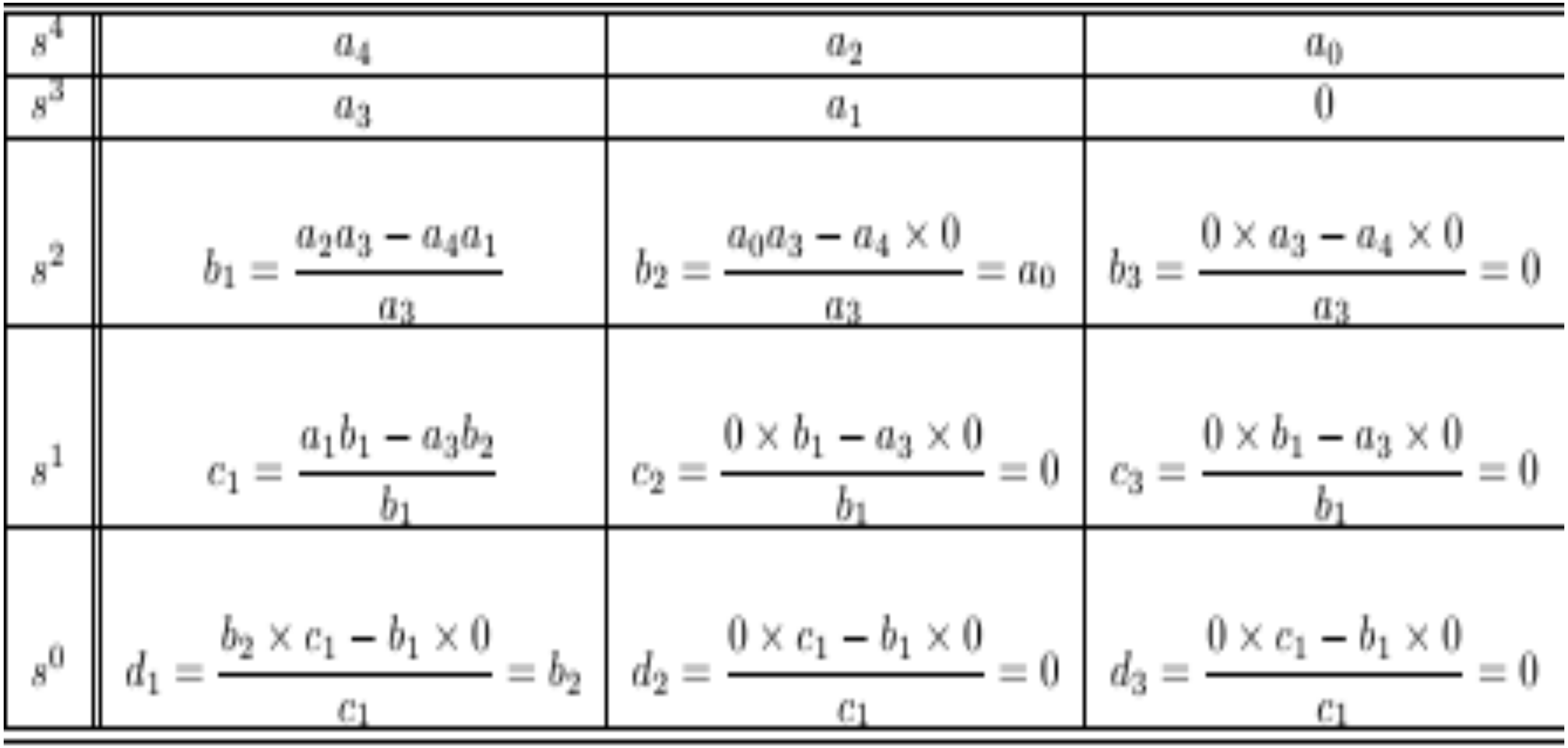
Stability



s^4 1	a_4	a_2	a_0
s^3 2	a_3	a_1	0
s^2 3	b1 –	b2 –	–
s^1 4	–	–	–
s^0 5	–	–	–

Table 1: Starting layout for Routh array

Stability



s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	$b_1 = \frac{a_2 a_3 - a_4 a_1}{a_3}$	$b_2 = \frac{a_0 a_3 - a_4 \times 0}{a_3} = a_0$	$b_3 = \frac{0 \times a_3 - a_4 \times 0}{a_3} = 0$
s^1	$c_1 = \frac{a_1 b_1 - a_3 b_2}{b_1}$	$c_2 = \frac{0 \times b_1 - a_3 \times 0}{b_1} = 0$	$c_3 = \frac{0 \times b_1 - a_3 \times 0}{b_1} = 0$
s^0	$d_1 = \frac{b_2 \times c_1 - b_1 \times 0}{c_1} = b_2$	$d_2 = \frac{0 \times c_1 - b_1 \times 0}{c_1} = 0$	$d_3 = \frac{0 \times c_1 - b_1 \times 0}{c_1} = 0$

Table 2: Completed Routh array

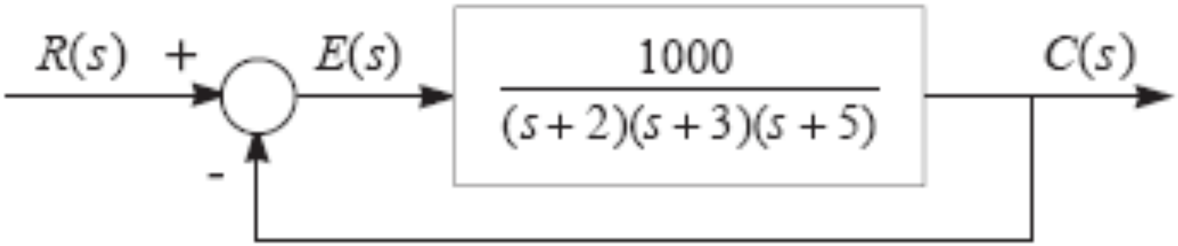
Stability

The Routh-Hurwitz Criterion: The number of roots of the characteristic polynomial that are in the right-half plane is equal to the number of sign changes in the first column of the *Routh Array*. If there are no sign changes, the system is stable.

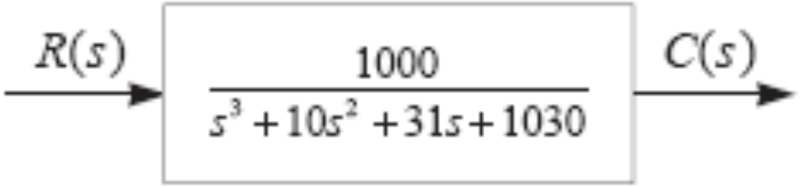
Example:

Test the stability of the closed-loop system

* أول اشي نفك ال Loop
by Block Reduction
$$\frac{G}{1+GH}$$



أو طريقة أسهل لفك ال loop
لأنها unity feedback
(H = 1)



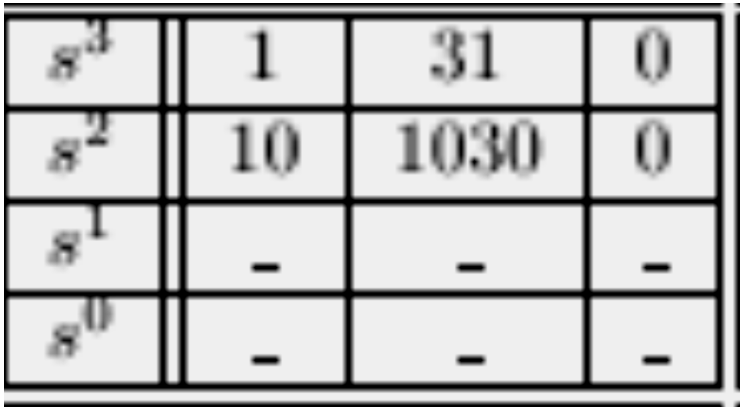
نجمع البسط إلى المقام
و هيك بنكون فكينا ال loop

و ضروري نبسط
و نضرب الأقواس ببعض
عشان نعرف
نفرغ الجدول للبدء

Stability

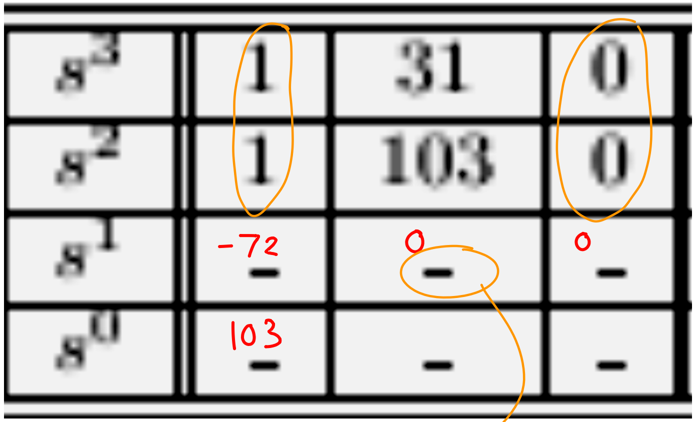
Solution: Since all the coefficients of the closed-loop characteristic equation $s^3 + 10s^2 + 31s + 1030$ are present, the system passes the Hurwitz test. So we must construct the Routh array in order to test the stability further.

*استطيع في أي مرحلة القسمة على عود موجب



s^3	1	31	0
s^2	10	1030	0
s^1	–	–	–
s^0	–	–	–

this Row ÷10



s^3	1	31	0
s^2	1	103	0
s^1	-72 –	0 –	0 –
s^0	103 –	–	–

لحساب هذا العنصر
باخذ مصفوفة بأول عمود
والعمود الذي بعدي

Stability

s^3	1	31
s^2	1	103
s^1	$\frac{31 \times 1 - 1 \times 103}{1} = -72$	$\frac{0 \times 1 - 1 \times 0}{1} = 0$
s^0	$\frac{-\cancel{72} \times 103 - 1 \times 0}{-\cancel{72}} = 103$	$\frac{-72 \times 0 - 1 \times 0}{-72} = 0$

For clarity, we can rewrite the array:

$$\begin{array}{c} \\ + \\ - \\ + \end{array} \begin{pmatrix} 1 & 31 & 0 \\ 1 & 103 & 0 \\ -72 & 0 & 0 \\ 103 & 0 & 0 \end{pmatrix}$$

* في ال Routh Array نحدد عدد جذور المقام الموجبة عشان نحدد ال system stable or unstable

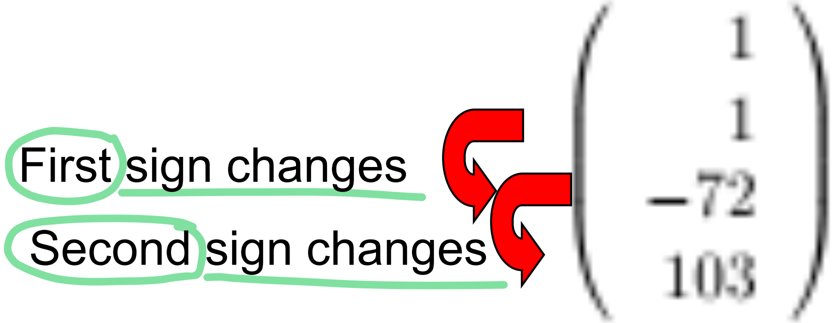
* عدد جذور المقام الموجبة = عدد تغيرات الاشارة في العمود الأول

إذا هذا ال system is unstable

(جذر واحد يكفي انه يخلي ال system unstable)

Stability

and now it is clear that column 1 of the Routh array is:



First sign changes

Second sign changes

$$\begin{pmatrix} 1 \\ 1 \\ -72 \\ 103 \end{pmatrix}$$

- and it has two sign changes (from 1 to -72 and from -72 to 103). Hence the system is *unstable* with *two poles in the right-half plane*.

Any Q's so far ?

Stability

← اهم اشي نفكك ال loop قبل البدء بالحل هنا

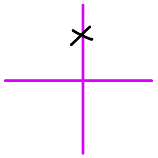
Special Case:

1. a zero may appear in the first column of the array
   - Zero Only in the First Column

في هذه الحالة بستبدل الصفر بـ $\in$ (قريبة جداً من الصفر) وبكمل حل

2. a complete row can become zero
   - Entire Row Is Zero

هذه الحالة هي يلي بتعطيني (K) Marginaly stable (النقطة يلي على ال imaginary axis)



Stability (Special Case 1)

Consider the control system with closed-loop transfer function:

$$G_c(s) = \frac{10}{s^5 + 2s^4 + 3s^3 + 6s^2 + 5s + 3}.$$

Routh array will be:

s^5	1	3	5
s^4	2	6	3
s^3	$0 \to \epsilon$	$7/2$	0
s^2	− $\frac{6\epsilon - 7}{\epsilon}$	3	0
s^1	+ $\frac{42\epsilon - 49 - 6\epsilon^2}{12\epsilon - 14}$	0	0
s^0	+ 3	0	0

Considering just the sign changes in column 1:

Label	First column	$\epsilon \to 0^+$	$\epsilon \to 0^-$
s^5	1	+	+
s^4	2	+	+
s^3	ϵ	+	−
s^2	$\frac{6\epsilon - 7}{\epsilon}$	−	+
s^1	$\frac{42\epsilon - 49 - 6\epsilon^2}{12\epsilon - 14}$	+	+
s^0	3	+	+

- If is chosen *positive* there are *two sign changes*. If is chosen *negative* there are also *two sign changes*. Hence the system has two poles in the right-half plane and it doesn't matter whether we chose to approach zero from the positive or the negative side.

Stability (Special Case 1)

Consider the control system with closed-loop transfer function:

$$G_c(s) = \frac{10}{s^5 + 2s^4 + 3s^3 + 6s^2 + 5s + 3}.$$

Routh array will be:

s^5	1	3	5
s^4	2	6	3
s^3	$0 \rightarrow \epsilon$	$7/2$	0
s^2	$\frac{6\epsilon - 7}{\epsilon}$	3	0
s^1	$\frac{42\epsilon - 49 - 6\epsilon^2}{12\epsilon - 14}$	0	0
s^0	3	0	0

Considering just the sign changes in column 1:

Label	*First column*	$\epsilon \rightarrow 0^+$	$\epsilon \rightarrow 0^-$
s^5	1	+	+
s^4	2	+	+
s^3	ϵ	+	−
s^2	$\frac{6\epsilon - 7}{\epsilon}$	−	+
s^1	$\frac{42\epsilon - 49 - 6\epsilon^2}{12\epsilon - 14}$	+	+
s^0	3	+	+

- If is chosen *positive* there are *two sign changes*. If is chosen *negative* there are also *two sign changes*. Hence the system has two poles in the right-half plane and it doesn't matter whether we chose to approach zero from the positive or the negative side.

Stability (Special Case 2)

Consider the control system with closed-loop transfer function:

$$G_c(s) = \frac{10}{s^5 + 7s^4 + 6s^3 + 42s^2 + 8s + 56}.$$

* إذا تماثل أي سطرين في أي مصفوفة
تصبح الـ determinant تساوي zero
كما حصل هنا

Routh array will be:

s^5	1	6	8
s^4	$7 \rightarrow 1$	$42 \rightarrow 6$	$56 \rightarrow 8$
s^3	0	0	0
s^2	–	–	–
s^1	–	–	–
s^0	–	–	–

÷7

divide by '7' for convenience

Auxiliary equation

مكونة من السطر الذي قبل الاصفار

replace the zero row with a row formed from the coefficients of the derivative:

s^5	1	6	8
s^4	1	6	8
s^3	$0 \rightarrow 4 \rightarrow 1$	$0 \rightarrow 12 \rightarrow 3$	0
s^2	–	–	–
s^1	–	–	–
s^0	–	–	–

*

divide by '4' for convenience

Auxilliary equation

$$Q(s) = s^4 + 6s^{2} + 8$$

Differentiate

$$\frac{dQ(s)}{ds} = 4s^3 + 12s + 0$$

s^5	1	6	8
s^4	1	6	8
s^3	1	3	0
s^2	3	8	0
s^1	1/3	0	0
s^0	8	0	0

There are no sign changes in the completed Routh array, hence the system is *stable*.

Example 1:

Construct a Routh table and determine the number of roots with *positive real parts* for the equation;

$$2s^3 + 4s^2 + 4s + 12 = 0$$

Solution:

s^3	2	4	0	⊕
s^2	4	12	0	⊕
s^1	-2	0	0	⊖
s^0	12	0	0	⊕

✓ Since there are two changes of sign in the first columm of Routh table, the equation above have two roots at right side (positive real parts).

$$2s^3 + 4s^2 + 4s + 12 = 0$$

S^3	2	4	0
S^2	⊕ 4	12	0
S^1	⊖2	0	0
S^0	⊕ ⑫ ؟؟	0	0

$$\frac{1}{4} x (16-24) = \frac{-8}{4} = -2$$

تغيرت الاشارة مرتين

عدد الجذور (٢) اذاً

the system is unstable

Example 2:

The characteristic equation of a given system is:

اضافة للسؤال "مثلا

$$s^{4}+6s^{3}+11s^{2}+6s+K\overset{+\,1}{\uparrow}=0$$

What restrictions must be placed upon the parameter *K* in order to ensure that the system is stable?

ما بدي اياه يغير اشارته

Solution:

For the system to be stable, $60 - 6K$ ~~$<$~~ > 0, or $k < 10$, and $K > 0$. Thus $0 < K < 10$

$$s^4 + 6s^3 + 11s^2 + 6s + K = 0$$

S^4	1	11	K	0
S^3	6	6	0	0
S^2	10	K	0	
S^1	$\frac{60-6K}{10}$	0	0	
S^0	K			

لازم كله يكون Positive

$60 - 6K > 0$

$$\left[\ 10 > K > 0\ \right] \rightarrow$$ in this range of (K) the system is stable

$\frac{1}{6}(66-6)$

$\frac{1}{10}(10\times6-6K)$

اضافة للسؤال للفهم

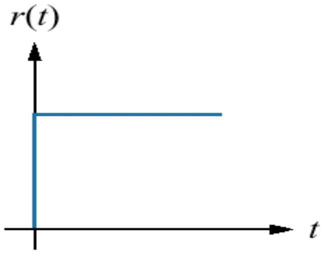
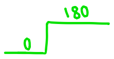
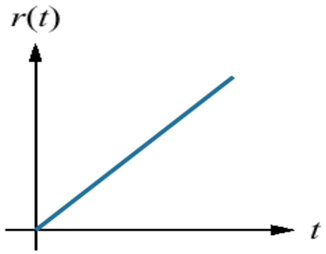
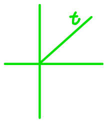
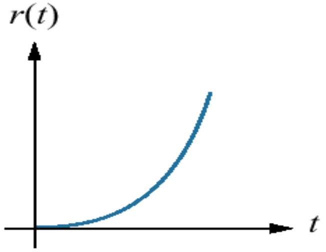
$$s^4 + 6s^3 + 11s^2 + 6s + K + \underline{1} = 0$$

S^4	1	11	$\underline{K+1}$
S^3	6	6	0
S^2			
S^1			
S^0			

Steady State Error Analysis

Test Waveform for evaluating steady-state error

٣ انواع ال input

Waveform	Name	Physical interpretation	Time function	Laplace transform
	Step	Constant position	1	$\frac{1}{s}$ 
	Ramp	Constant velocity	t	$\frac{1}{s^2}$ 
	Parabola	Constant acceleration	$\frac{1}{2}t^2$	$\frac{1}{s^3}$ Prabolic motion

google من

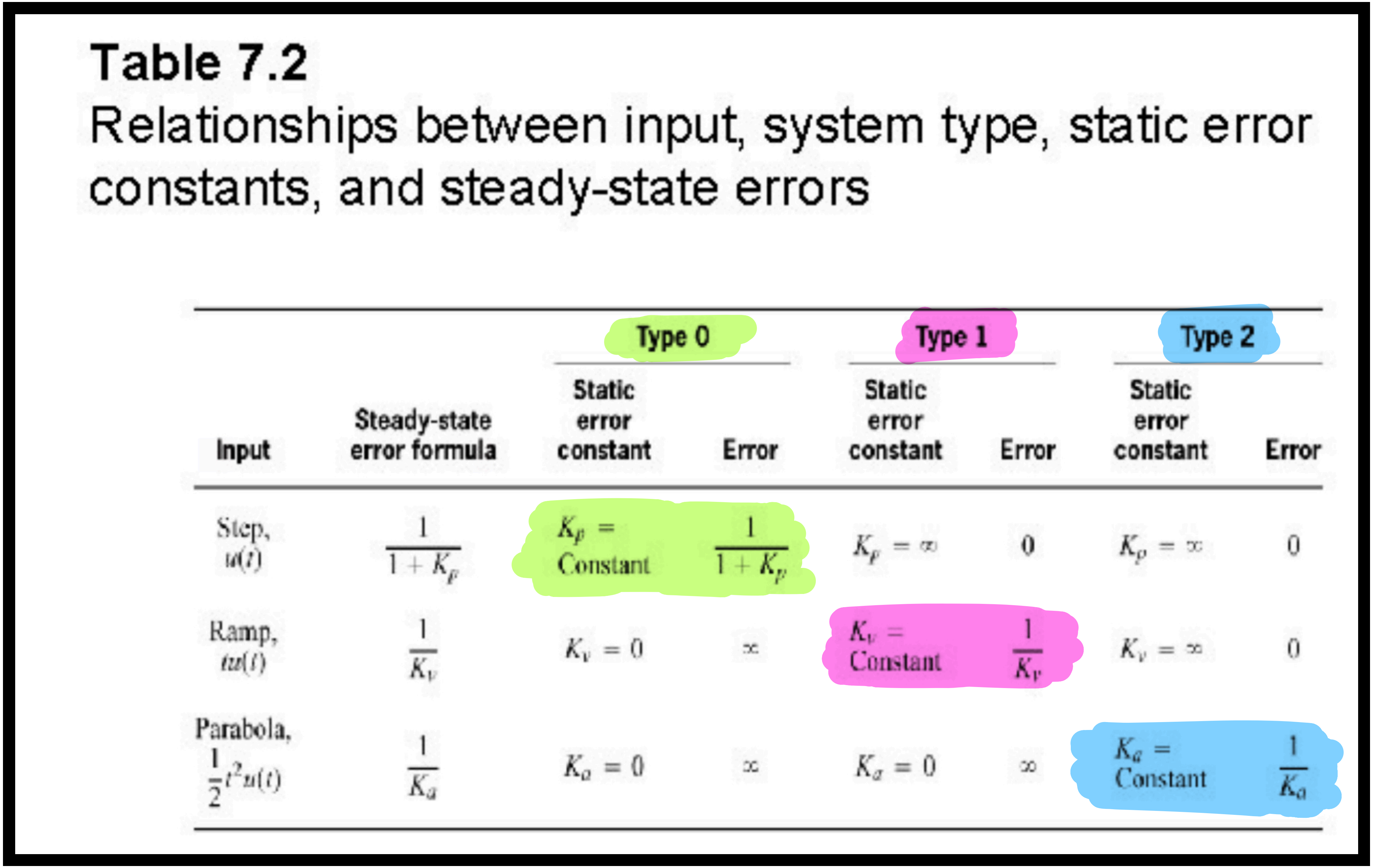


Table 7.2

Relationships between input, system type, static error constants, and steady-state errors

		Type 0		Type 1		Type 2	
Input	Steady-state error formula	Static error constant	Error	Static error constant	Error	Static error constant	Error
Step, $u(t)$	$\frac{1}{1+K_p}$	$K_p =$ Constant	$\frac{1}{1+K_p}$	$K_p = \infty$	0	$K_p = \infty$	0
Ramp, $tu(t)$	$\frac{1}{K_v}$	$K_v = 0$	∞	$K_v =$ Constant	$\frac{1}{K_v}$	$K_v = \infty$	0
Parabola, $\frac{1}{2}t^2u(t)$	$\frac{1}{K_a}$	$K_a = 0$	∞	$K_a = 0$	∞	$K_a =$ Constant	$\frac{1}{K_a}$

*تقريباً تقريباً اختصار لكل حاجة

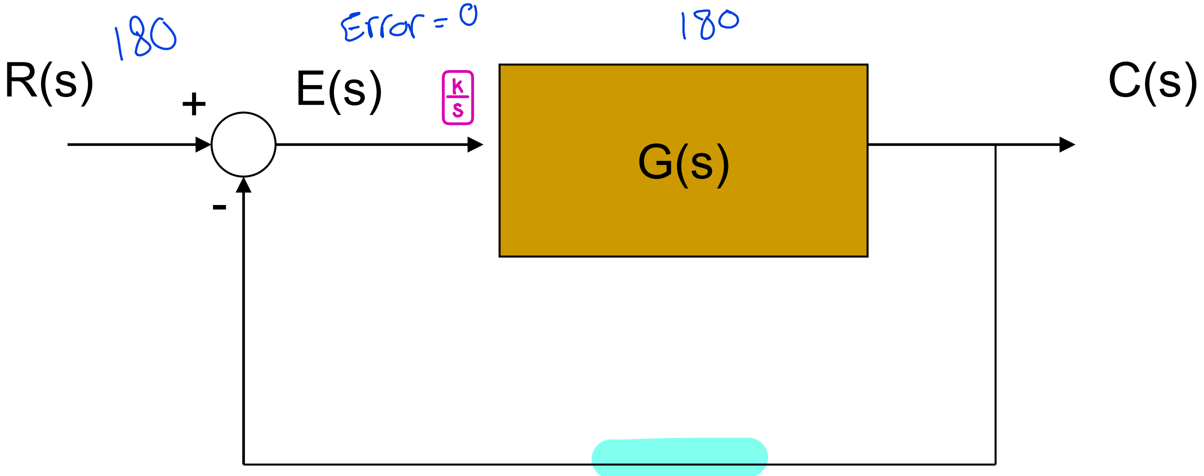
SSE:

	SSE for		
System Type	Step	Ramp	Parabolic
0	$\frac{1}{1+K_p}$	∞	∞
1	0	$\frac{1}{K_v}$	∞
2	0	0	$\frac{1}{K_a}$

Steady-state error analysis

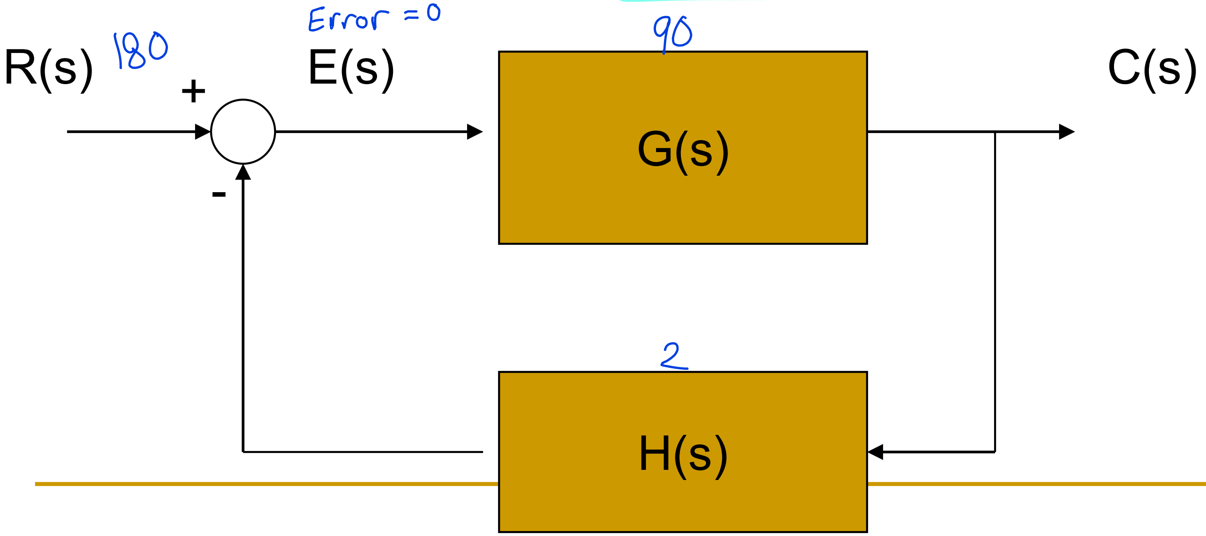
* في كل مرة نضيف integrater بزيد كمان $\left(\frac{1}{s}\right)$

مثلا//



Unity feedback
H(s)=1

← more common



Non-unity feedback
H(s)≠1

Steady-state error analysis

→ x s

→ s = 0

For unity feedback system:

more common

$$E(s) = R(s) - C(s)$$

→ System error

Error = input − output

For a non-unity feedback system:

$$E(s) = R(s) - \underline{H(s)}C(s)$$

→ Actuating error

Steady-state error analysis

ال (s) تسبب اشكاليات و مشاكل

يأتي من

- input
- final value theorem
- integrater
- derivative

Consider a unity feedback system, if the inputs are step response, ramp & parabolic (no sinusoidal input). We want to find the steady-state error

*the system have to be (Error=zero)

input →	output
20	20
100	100
12	12
77	77

I want the error to be zero

$\text{Error} = 0 = \frac{1}{\infty}$ → و هذا ينطبق عند الجذر S

$$e_{ss} = \lim_{t\to\infty} e(t)$$

Where, $$e(t) = r(t) - c(t)$$

By Final Value Theorem:

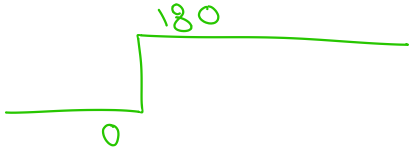
$$\boxed{e_{ss} = \lim_{t\to\infty} e(t) \cong \lim_{s\to 0} sE(s)}$$

يشبه

Final value theorem

Steady-state error analysis

Step Response



Consider Unity Feedback System

$$E(s)=R(s)-C(s) \longrightarrow (1)$$

$$\frac{C(s)}{R(s)}=\frac{G(s)}{1+G(s)} \longrightarrow (2)$$

Substitute (2) into (1)

Error =

$$\therefore E(s)=R(s)-\frac{G(s)}{1+G(s)}R(s)=\frac{1}{1+G(s)}R(s) \longrightarrow (3)$$

Steady-state error analysis

Based on equation (3), it can be seen that E(s) depends on:

(a) Input signal, R(s)

(b) G(s), open loop transfer function

Error

from block diagram reduction

$$\lim_{s\to 0} s\times\frac{1}{1+G}\times R$$

Final value theorem

ثم نعوض قيم مختلفة لـ R

راح يغلبني

$\frac{1}{s}$ ←

أكبر load $\frac{1}{s^2}$ ←

أكبر أكبر load $\frac{1}{s^3}$ ←

$\frac{1}{s=0}=\infty$

Cases to be considered:

$$(A)\,R(s)=\frac{1}{s}$$

← Step

$$(B)\,R(s)=\frac{1}{s^2}$$

← Ramp

$$(C)\,R(s)=\frac{1}{s^3}$$

← Parabolic

Now, assume:

$(s+1)\times(s+2)\times(s+3)\ldots.$ يعني عملية ضرب اقواس

$\sum_{i=1}^{M}(s+z_i)$ أو

zeros

$$G(s)=\frac{K\overset{M}{\underset{i=1}{\pi}}(s+\underline{z_i})}{S^N\overset{\theta}{\underset{j=1}{\pi}}(s+\underline{p_j})}$$

poles

الجذور يلي قيمتهم صفر

type N

$*\ N=1\rightarrow$

$*\ N\geq 1\rightarrow$ Error = zero

Case (A): Input is a unit step R(s)=1/s

$$E(s)=\frac{1}{1+G(s)}R(s)=\frac{1/s}{1+G(s)}$$

$$e_{ss}=Steady-State\_Error\cong\lim_{s\to 0}sE(s)$$

$$e_{ss}=\lim_{s\to 0}s\left[\frac{1/s}{1+G(s)}\right]=\lim_{s\to 0}\left[\frac{1}{1+G(s)}\right]$$

$$=\left[\frac{1}{1+\lim_{s\to 0}G(s)}\right]=\left[\frac{1}{1+K_p}\right]$$

where $\boxed{K_p=\lim_{s\to 0}G(s)}$ → "Static Position Error Constant"

If N = 0, K_p = constant

$$e_{ss} = \frac{1}{1+K_p} = finite$$

If N ≥ 1, K_p = infinite

$$e_{ss} = \frac{1}{1+K_p} = \frac{1}{1+\infty} = 0$$

For unit step response, as the type of system increases (N ≥ 1), the steady state error goes to zero

Case (B): Input is a unit ramp R(s)=1/s^2

$$E(s) = \frac{1}{1+G(s)}R(s) = \frac{1/s^2}{1+G(s)}$$

$$e_{ss} = Steady - State\_Error \cong \lim_{s\to 0} sE(s)$$

$$e_{ss} = \lim_{s\to 0} s\left[\frac{1/s^2}{1+G(s)}\right] = \lim_{s\to 0}\left[\frac{1}{s+sG(s)}\right]$$

$$= \left[\frac{1}{0+\lim_{s\to 0} sG(s)}\right] = \left[\frac{1}{\lim_{s\to 0} sG(s)}\right] \cong \frac{1}{K_V}$$

where $\boxed{K_v = \lim_{s\to 0} sG(s)}$ → "Static Velocity Error Constant"

If N = 0, $K_v = s\dfrac{\pi(s+z_i)}{\pi(s+p_j)} = 0,$ $\qquad e_{ss} = \dfrac{1}{K_v} = \infty$

If N =1, K_v = finite $\qquad e_{ss} = \dfrac{1}{K_v} = finite$

If N ≥2, K_v = infinite $\qquad e_{ss} = \dfrac{1}{K_v} = \dfrac{1}{\infty} = 0$

For unit ramp response, the steady state error in infinite for system of type zero, finite steady state error for system of type 1, and zero steady state error for systems with type greater or equal to 2.

Case (C): Input is a parabolic, $R(s)=1/s^3$

$$E(s)=\frac{1}{1+G(s)}R(s)=\frac{1/s^3}{1+G(s)}$$

$$e_{ss}=Steady-State\_Error\cong\lim_{s\to0}sE(s)$$

$$e_{ss}=\lim_{s\to0}s\left[\frac{1/s^3}{1+G(s)}\right]=\lim_{s\to0}\left[\frac{1}{s^2+s^2G(s)}\right]$$

$$=\left[\frac{1}{0+\lim_{s\to0}s^2G(s)}\right]=\left[\frac{1}{\lim_{s\to0}s^2G(s)}\right]\cong\frac{1}{K_a}$$

where $\boxed{K_a=\lim_{s\to0}s^2G(s)}$ → "Static Acceleration Error Constant"

If N = 0,$K_a = s^2 \dfrac{\pi(s+z_i)}{\pi(s+p_j)} = 0,$ $e_{ss} = \dfrac{1}{K_a} = \infty$

If N =1, K_a = 0 $e_{ss} = \dfrac{1}{K_a} = \infty$

If N = 2, K_a = constant $e_{ss} = \dfrac{1}{K_a} = finite$

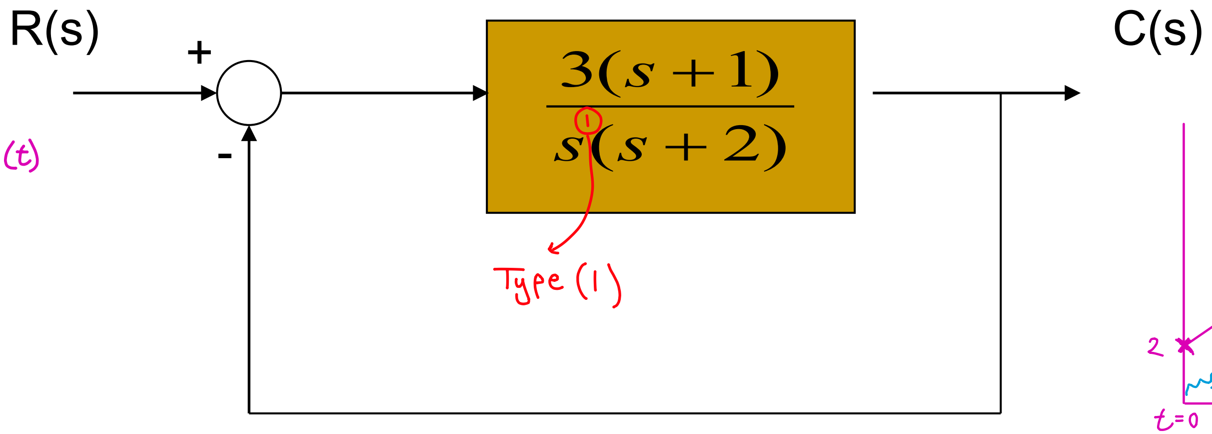
If N ≥3 , K_a = infinite $e_{ss} = \dfrac{1}{K_a} = \dfrac{1}{\infty} = 0$

→ Increasing system type (N) will accommodate more different inputs.

SSE :

		SSE for	
System Type	Step	Ramp	Parabolic
0	$\frac{1}{1+K_p}$	∞	∞
1	0	$\frac{1}{K_v}$	∞
2	0	0	$\frac{1}{K_a}$

Example 3

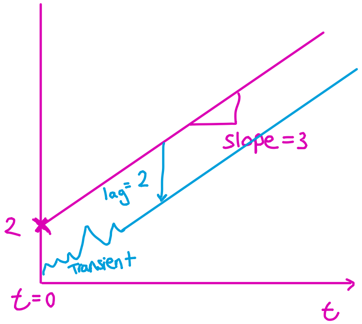


$r(t) = (2+3t) \times u(t)$

$2 \downarrow \frac{2}{s}$ step $\downarrow$ $\frac{1}{1+K_p}$

the error in laplace domain

$3t \downarrow \frac{3}{s^2}$ Ramp $\downarrow$ $\frac{3}{K_v}$



If r(t) = (2+3t)u(t), find the steady state error (e_{ss}) for the given system.

معناها انه النظام بلش
عند ال zero

Solution:

Infinite Error → $K_p = \lim_{s\to 0} G(s) = \underline{\infty}$

من الجدول
يلي حفظ

Finite Error → $K_v = \lim_{s\to 0} sG(s) = 3/2$

$\lim_{s\to 0} \cancel{s} \times \frac{3(s+1)}{\cancel{s}(s+2)}$

ثم نعوض ال s=0

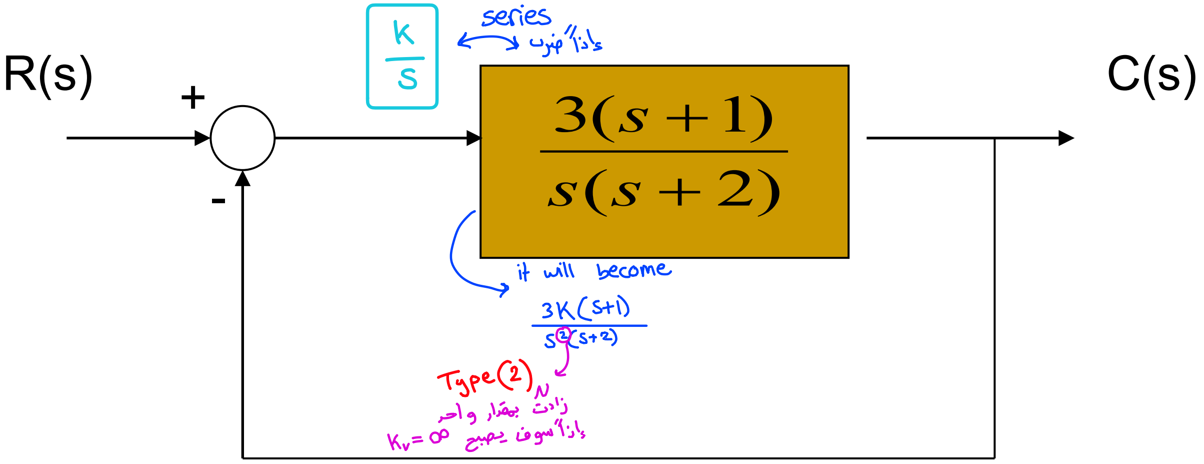
$\frac{3(0+1)}{0+2} = \frac{3}{2}$

$$e_{ss} = \frac{2}{1+K_p} + \frac{3}{K_v} = \frac{2}{1+\infty} + \frac{3}{3/2} = 2$$

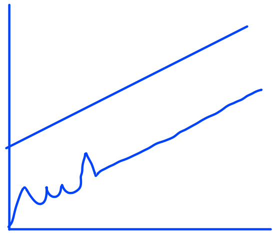
Any Q's?

Example 3

*if the Question was design the controller to achieve Zero steady state error



If r(t) = (2+3t)u(t), find the steady state error (e_{ss}) for the given system.



Solution:

$$K_p = \lim_{s\to 0} G(s) = \infty$$

$$K_v = \lim_{s\to 0} sG(s) = \cancel{\frac{3}{2}} \quad \infty$$

*مو متأكدة اذا لازم نحسبها أو لا

$$K_a = \lim_{S\to 0} S^2 G(S) = \lim_{S\to 0} \cancel{S^2} \times \frac{3K(S+1)}{\cancel{S^2}(S+2)}$$

بعد تعويض S=0

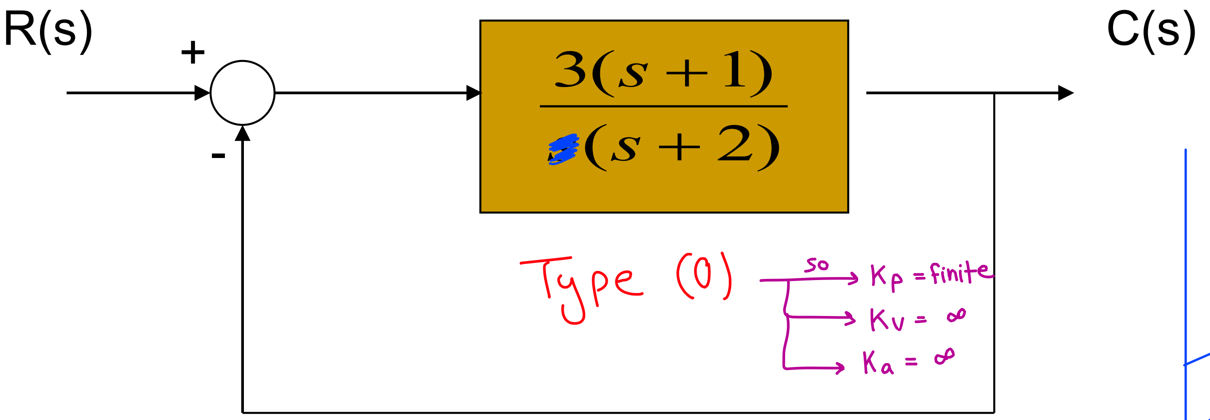
$$\frac{K}{2}$$

$$e_{ss} = \frac{2}{1+K_p} + \frac{3}{K_v} = \frac{2}{1+\infty} + \frac{3}{\cancel{\frac{3}{2}}_{\infty}} = \cancel{2} \ \text{zero}$$

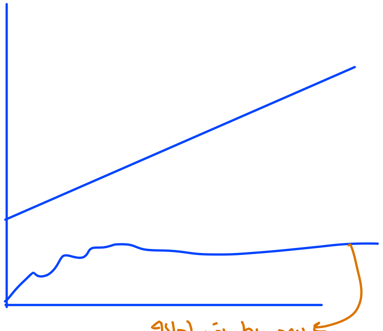
Any Q's?

Example 3

اذا كان السؤال هكذا:



Type (0) $\xrightarrow{so}$ Kp = finite
Kv = ∞
Ka = ∞



بروح بطريق لحاله
يعني لو شلنا ال integrater
بصير مشكلة كبيرة

If r(t) = (2+3t)u(t), find the steady state error (e_{ss}) for the given system.

Solution:

$$K_p = \lim_{s\to 0} G(s) = \cancel{\infty}\ \frac{3}{2}$$

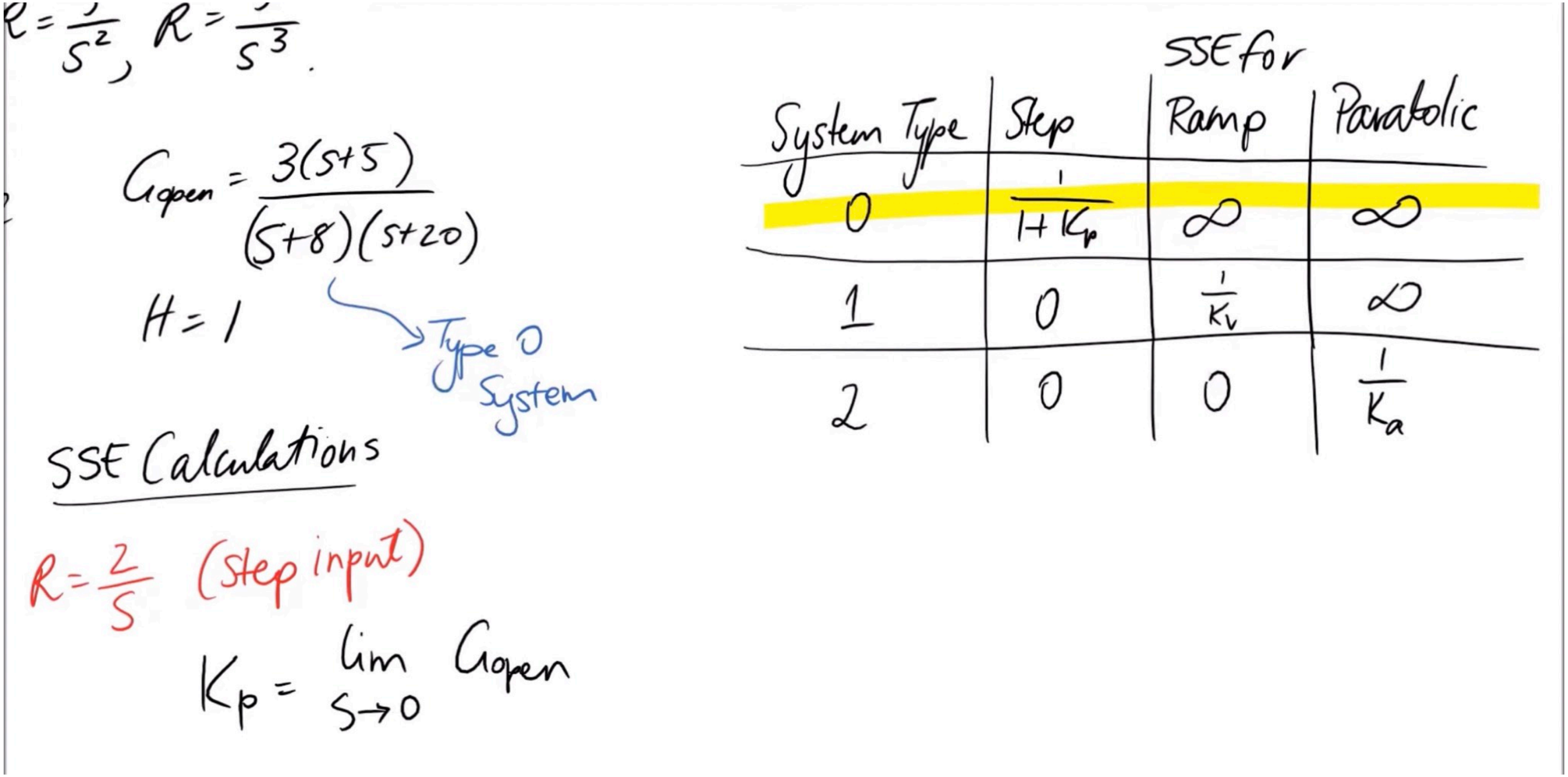
$$K_v = \lim_{s\to 0} sG(s) = \cancel{3/2}\ \text{zero}$$

$$e_{ss} = \frac{2}{1+\cancel{K_p}\frac{3}{2}} + \frac{3}{\cancel{K_v}\,\text{zero}} = \frac{2}{1+\overset{\frac{3}{2}}{\cancel{\infty}}} + \frac{3}{\cancel{3/2}\,\text{zero}} = \overset{\infty}{\cancel{2}}$$

Any Q's?

https://youtu.be/Idk9OkB2fuY?si=bughEpP3EsMcXf0z

https://youtu.be/AQNk2bydOY4?si=fhjxa3k_Rf7NgFxQ



$R = \frac{\ }{s^2},\ R = \frac{\ }{s^3}.$

$$G_{open} = \frac{3(s+5)}{(s+8)(s+20)}$$

$H = 1$

SSE Calculations

$R = \frac{2}{s}$ (Step input)

$$K_p = \lim_{s \to 0} G_{open}$$

	SSE for		
System Type	Step	Ramp	Parabolic
0	$\frac{1}{1+K_p}$	∞	∞
1	0	$\frac{1}{K_v}$	∞
2	0	0	$\frac{1}{K_a}$

$e(\infty) = 0$ | $K_V = \lim_{s\to 0} s\,G_{open}$ | $e(\infty) = \infty$

$$= \lim_{s\to 0} \cancel{s} \frac{6(s+3)}{\cancel{s}(s+9)(s+18)}$$

$$= \frac{6(3)}{(9)(18)}$$

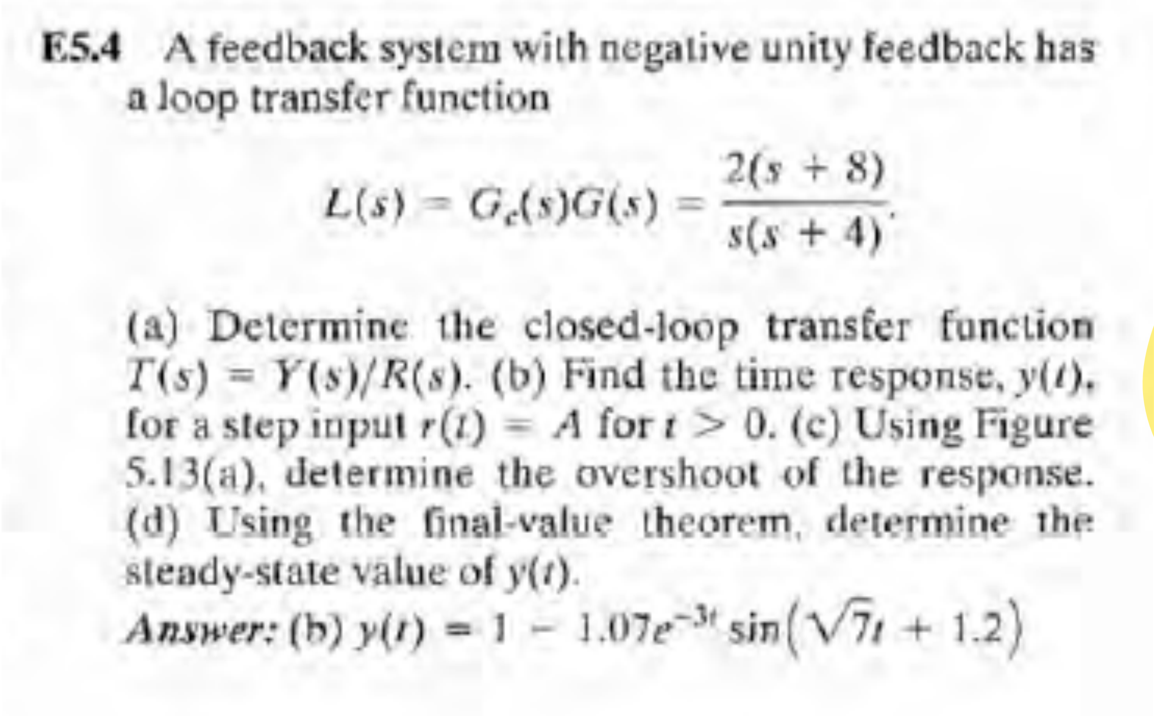
$$= \frac{1}{9}$$

$$e(\infty) = \frac{1}{K_V} \times 5$$

$$= \frac{1}{\left(\frac{1}{9}\right)} \times 5$$

$$= 45$$

RECORDED WITH
SCREENCAST-O-MATIC



E5.4 A feedback system with negative unity feedback has a loop transfer function

$$L(s) = G_c(s)G(s) = \frac{2(s+8)}{s(s+4)}.$$

(a) Determine the closed-loop transfer function $T(s) = Y(s)/R(s)$. (b) Find the time response, $y(t)$, for a step input $r(t) = A$ for $t > 0$. (c) Using Figure 5.13(a), determine the overshoot of the response. (d) Using the final-value theorem, determine the steady-state value of $y(t)$.

Answer: (b) $y(t) = 1 - 1.07e^{-3t}\sin\left(\sqrt{7}t + 1.2\right)$

SOLVE HERE

في التصميم نحدد عدد ال integraters وبضل اضيف $\frac{K}{S}$

في ال steady state error بضل اضيف $\frac{K}{S}$ لحد ما يعطيني → finite Error / zero Error

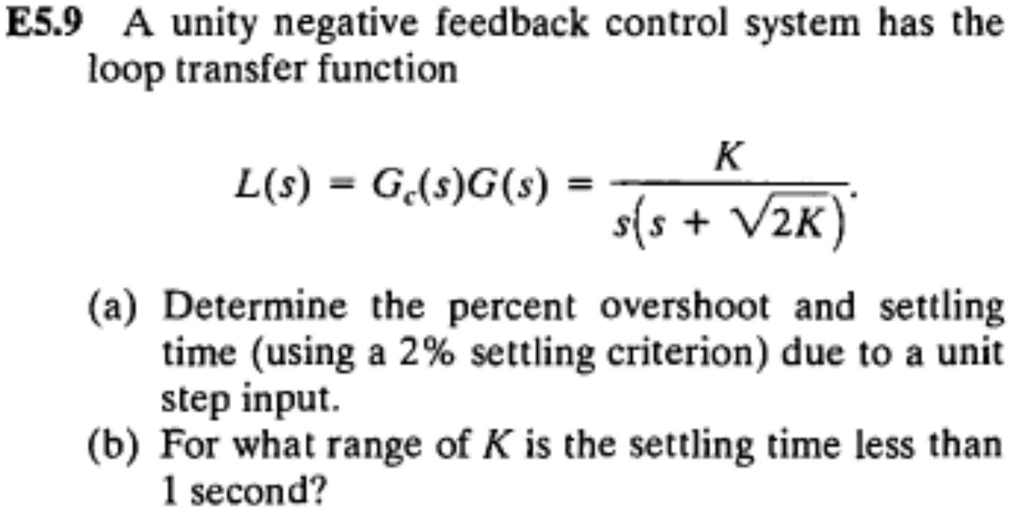
finite Error ← هنا احدد قيمة ال K

E5.8 A control system for positioning the head of a floppy disk drive has the closed-loop transfer function

$$T(s) = \frac{11.1(s + 18)}{(s + 20)(s^2 + 4s + 10)}.$$

Plot the poles and zeros of this system and discuss the dominance of the complex poles. What overshoot for a step input do you expect?

SOLVE HERE



E5.9 A unity negative feedback control system has the loop transfer function

$$L(s) = G_c(s)G(s) = \frac{K}{s\left(s + \sqrt{2K}\right)}.$$

(a) Determine the percent overshoot and settling time (using a 2% settling criterion) due to a unit step input.

(b) For what range of K is the settling time less than 1 second?

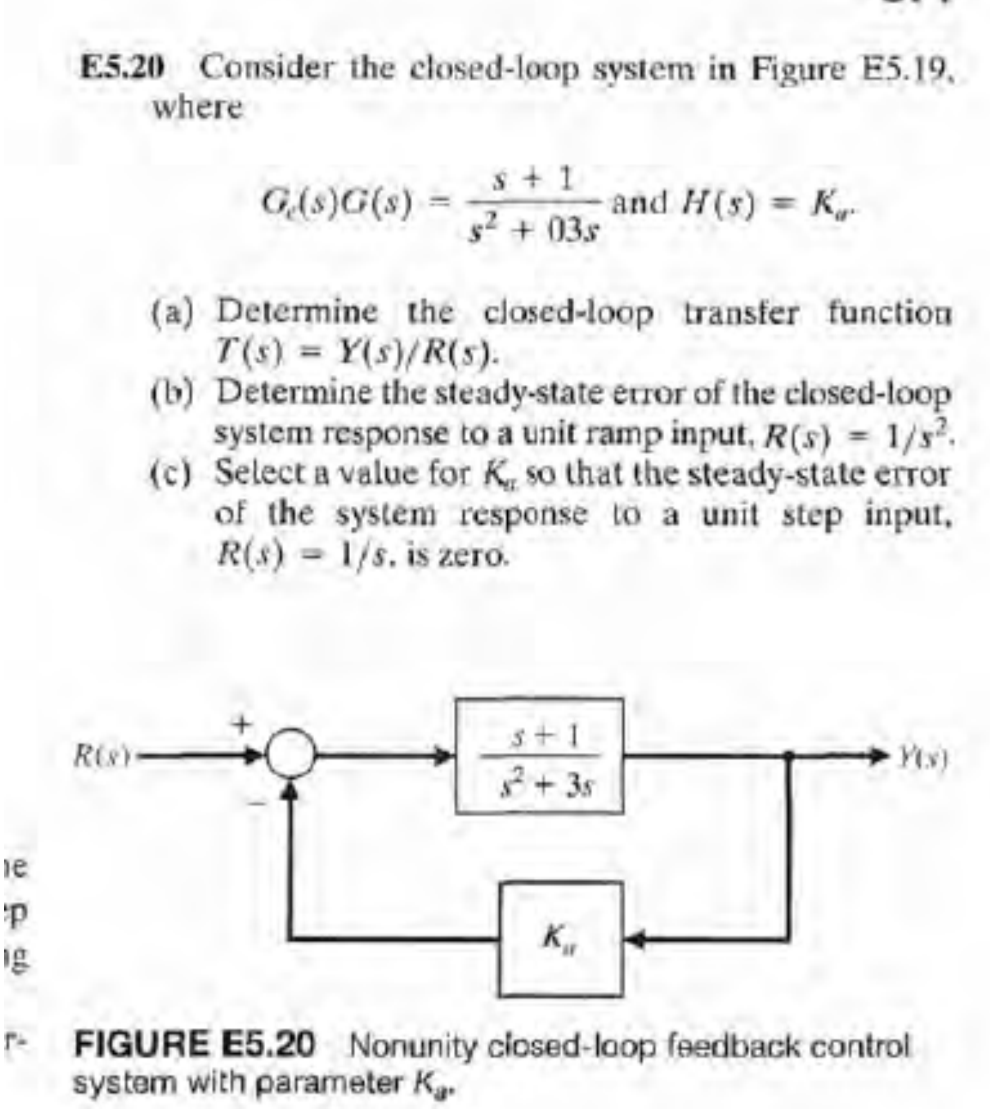
SOLVE HERE

E5.13 For the system with unity feedback shown in Figure E5.11, determine the steady-state error for a step and a ramp input when

$$G(s) = \frac{20}{s^2 + 14s + 50}.$$

Answer: $e_{ss} = 0.71$ for a step and $e_{ss} = \infty$ for a ramp.

SOLVE HERE



E5.20 Consider the closed-loop system in Figure E5.19, where

$$G_c(s)G(s) = \frac{s+1}{s^2 + 03s} \text{ and } H(s) = K_a.$$

(a) Determine the closed-loop transfer function $T(s) = Y(s)/R(s)$.

(b) Determine the steady-state error of the closed-loop system response to a unit ramp input, $R(s) = 1/s^2$.

(c) Select a value for K_a so that the steady-state error of the system response to a unit step input, $R(s) = 1/s$, is zero.

FIGURE E5.20 Nonunity closed-loop feedback control system with parameter K_a.

SOLVE HERE

P5.20 A system is shown in Figure P5.20.

(a) Determine the steady-state error for a unit step input in terms of K and K_1, where $E(s) = R(s) - Y(s)$.

(b) Select K_1 so that the steady-state error is zero.

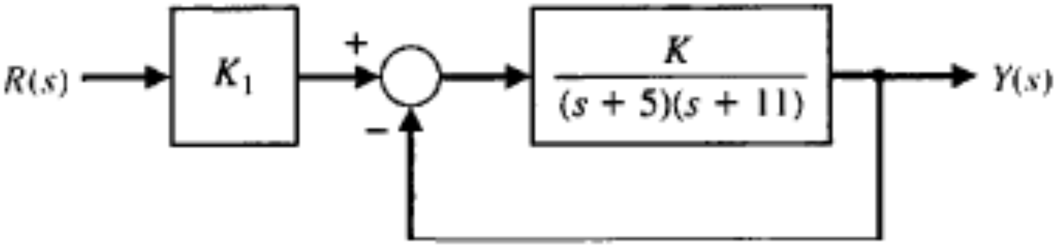


FIGURE P5.20 System with pregain, K_1.

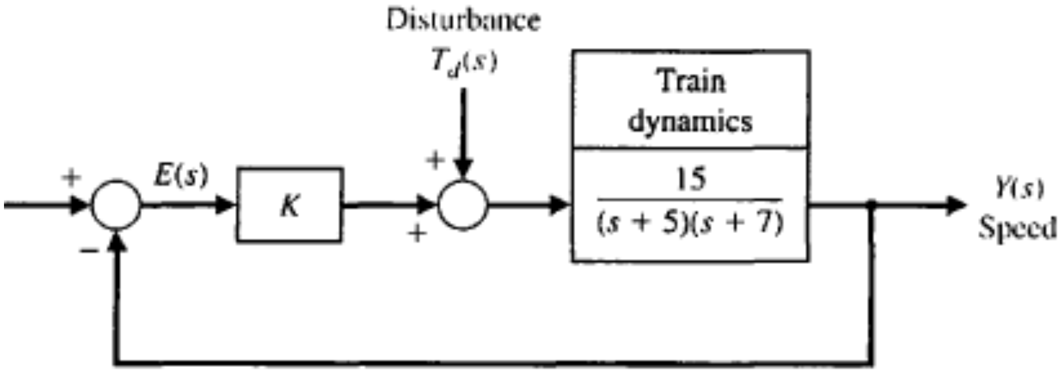
SOLVE HERE

AP5.4 The speed control of a high-speed train is represented by the system shown in Figure AP5.4 [17]. Determine the equation for steady-state error for K for a unit step input $r(t)$. Consider the three values for K equal to 1, 10, and 100.

(a) Determine the steady-state error.

(b) Determine and plot the response $y(t)$ for (i) a unit step input $R(s) = 1/s$ and (ii) a unit step disturbance input $T_d(s) = 1/s$.

(c) Create a table showing overshoot, settling time (with a 2% criterion), e_{ss} for $r(t)$, and $|y/t_d|_{\max}$ for the three values of K. Select the best compromise value.



SOLVE HERE

Modern Control Systems (MCS)

Root Locus

9/11/2023

Lecture Outline

- Construction of root loci
    - Angle and Magnitude Conditions
    - Illustrative Examples
- Closed loop stability via root locus
- Example of Root Locus
    - Root Locus of 1st order systems
    - Root Locus of 2nd order systems
    - Root Locus of Higher order systems

in this topic we will discuss
من الصفر لحد ال settling time

Construction of Root Loci

- Finding the roots of the characteristic equation of degree higher than 3 is laborious and will need computer solution.

*جذور المقام هي التي تحدد → settling time / Transient Response

*المؤثر الحقيقي → اما يكون جذر واحد (one zero) / أو جذرين complex (Dominant poles)

- A simple method for finding the roots of the characteristic equation has been developed by W. R. Evans and used extensively in control engineering.

- This method, called the *root-locus method,* is one in which the roots of the characteristic equation are plotted for all values of a system parameter.

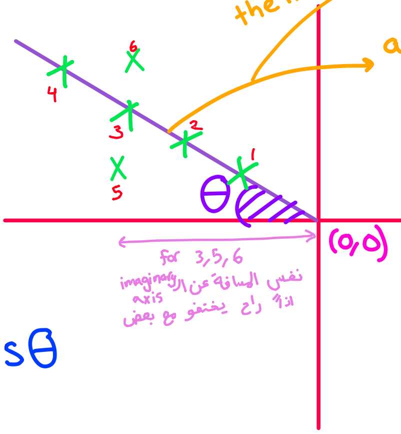
Construction of Root Loci

- The roots corresponding to a particular value of this parameter can then be located on the resulting graph.

- Note that the parameter is usually the gain, but any other variable of the open-loop transfer function may be used.

- By using the root-locus method the designer can predict the effects on the location of the closed-loop poles of varying the gain value or adding open-loop poles and/or open-loop zeros.

Any point in the line have the same damping Ratio



for 3,5,6
نفس المسافة عن ال imaginary axis
اذاً راح يختفو مع بعض

the line have → constant damping Ratio

and constant percent overshoot

$e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100\%$

Damping Ratio = $\zeta = \cos\theta$

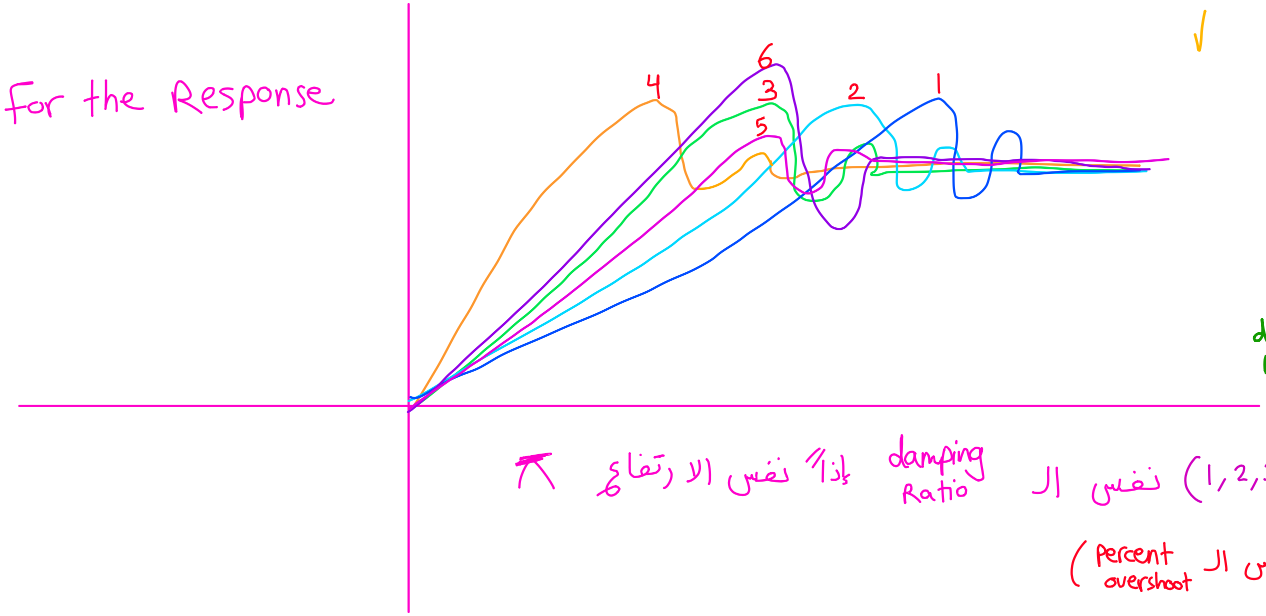
*العلاقة بين θ و ال ζ علاقة عكسية

كلما زادت θ قل ال ζ

3,5,6
they will settle at the same time
(لهم نفس البعد عن ال imaginary axis)

(6) ← أعلى زاوية وأقل damping
$\zeta = \cos\theta$

* For the Response



√

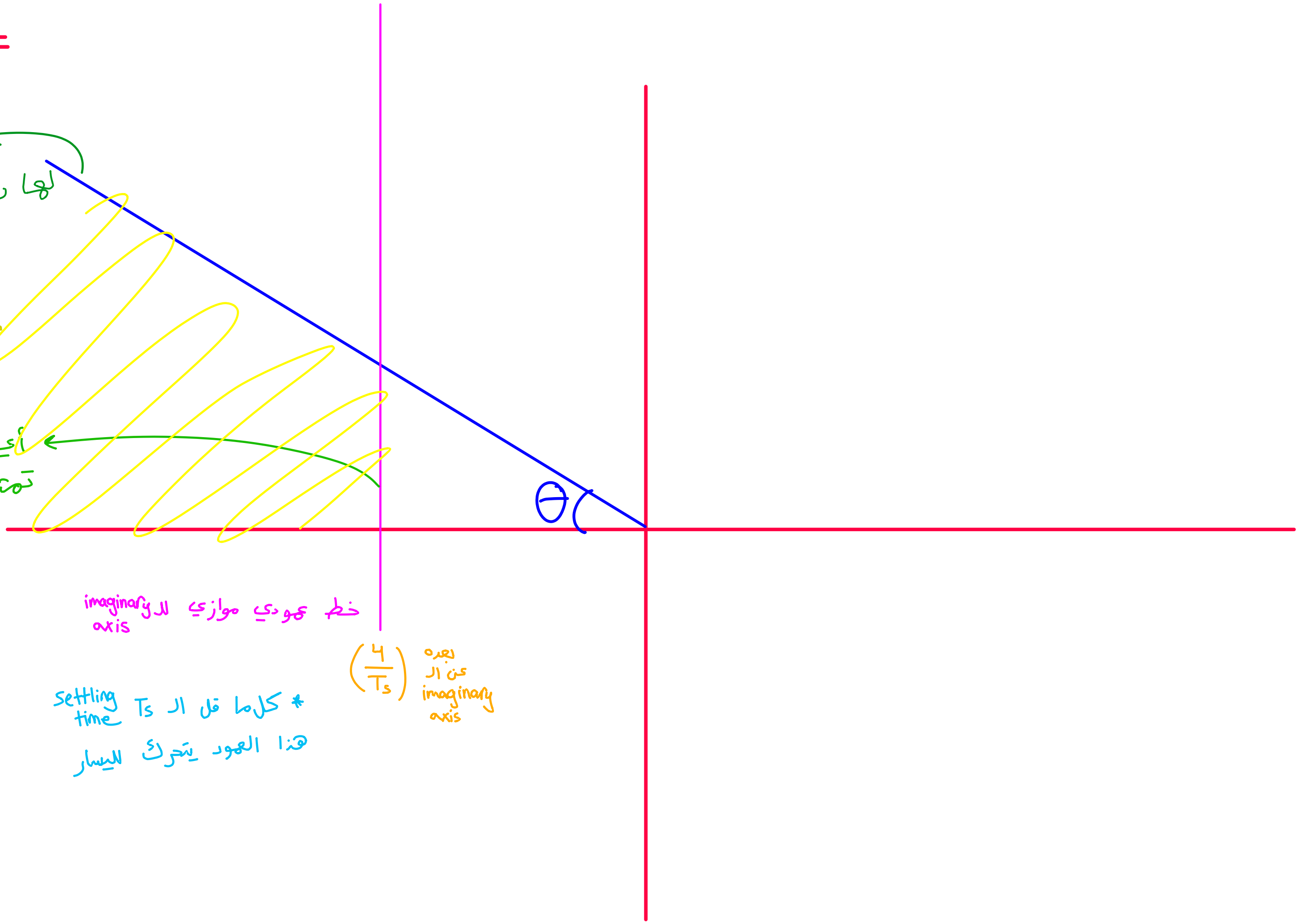
5 ← أعلى damping Ratio إذاً أقل Percent overshoot

6 ← أسوءهم ، أعلى Percent overshoot و أقل damping Ratio

*(1,2,3,4) نفس ال damping Ratio إذاً نفس الارتفاع ↑

(نفس ال Percent overshoot)

Summary :-



جميع النقاط على الخط لها نفس ال damping Ratio ونفس ال Percent overshoot

أي نقطة ضمن هذا ال Region تحقق الشرط

أي نقطة على هذا الخط تمثل constant Settling time

خط عمودي موازي لل imaginary axis

$\left(\frac{4}{T_s}\right)$ بعده عن ال imaginary axis

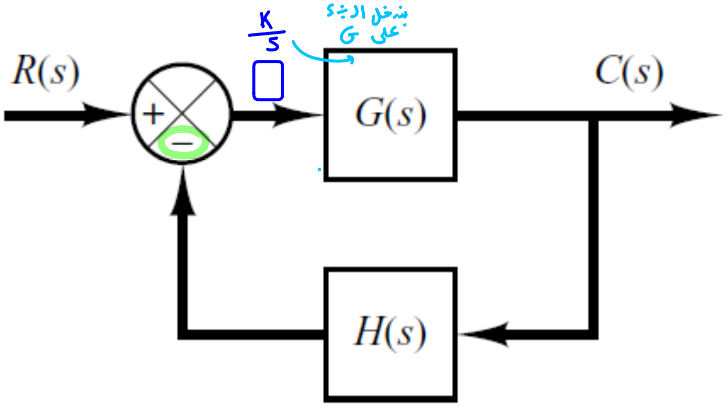
* كلما قل ال Ts settling time هذا العمود يتحرك لليسار

Angle & Magnitude Conditions

- In constructing the root loci angle and magnitude conditions are important.

- Consider the system shown in following figure.

دورنا الان هو تحديد قيمة ال K



نحن نزيد الجزر الأقرب على ال imaginary axis
(the dominant pole)

- The closed loop transfer function is

*اذاً ممكن يكون برا ال Region ومع تغيير قيمة K يصير جوا ال Region

$$\frac{C(s)}{R(s)} = \frac{G(s)}{(1+G(s)H(s))}$$

هذه ال loop تغير جذور المقام

Construction of Root Loci

- The characteristic equation is obtained by setting the denominator polynomial equal to zero.

$$\boxed{1+KG(s)H(s)=0}$$

- Or

$$G(s)H(s)=-1$$

$0 \longrightarrow \infty$

- Where *G(s)H(s)* is a ratio of polynomial in s.

- Since *G(s)H(s)* is a complex quantity it can be split into angle and magnitude part.

Angle & Magnitude Conditions

نحول المعادلة إلى معادلتين → معادلة (angle)
معادلة (magnitude)

- The angle of *G(s)H(s)=-1* is

$$\angle G(s)H(s) = \angle -1$$

$$\angle G(s)H(s) = \pm 180^{\circ}(2k+1)$$

- Where *k*=1,2,3…

- The magnitude of *G(s)H(s)=-1* is

← ايجاد K عن طريق معادلة ال magnitude

*لو طلعت الزاوية 180 أو مكرراتها
إذا هي نقطة بقدر اوصلها
عن طريق تغيير قيمة K

$$|G(s)H(s)| = |-1|$$

$$|G(s)H(s)| = 1$$

Angle & Magnitude Conditions

- Angle Condition

$$\boxed{\angle G(s)H(s) = \pm 180^\circ(2k+1)} \qquad (k = 1,2,3\ldots)$$

- Magnitude Condition

$$\boxed{|G(s)H(s)| = 1}$$

- The values of s that fulfill both the angle and magnitude conditions are the roots of the characteristic equation, or the closed-loop poles.

- A locus of the points in the complex plane satisfying the angle condition alone is the root locus.

Angle and Magnitude Conditions (Graphically)

- To apply Angle and magnitude conditions graphically we must first draw the poles and zeros of G(s)H(s) in s-plane.

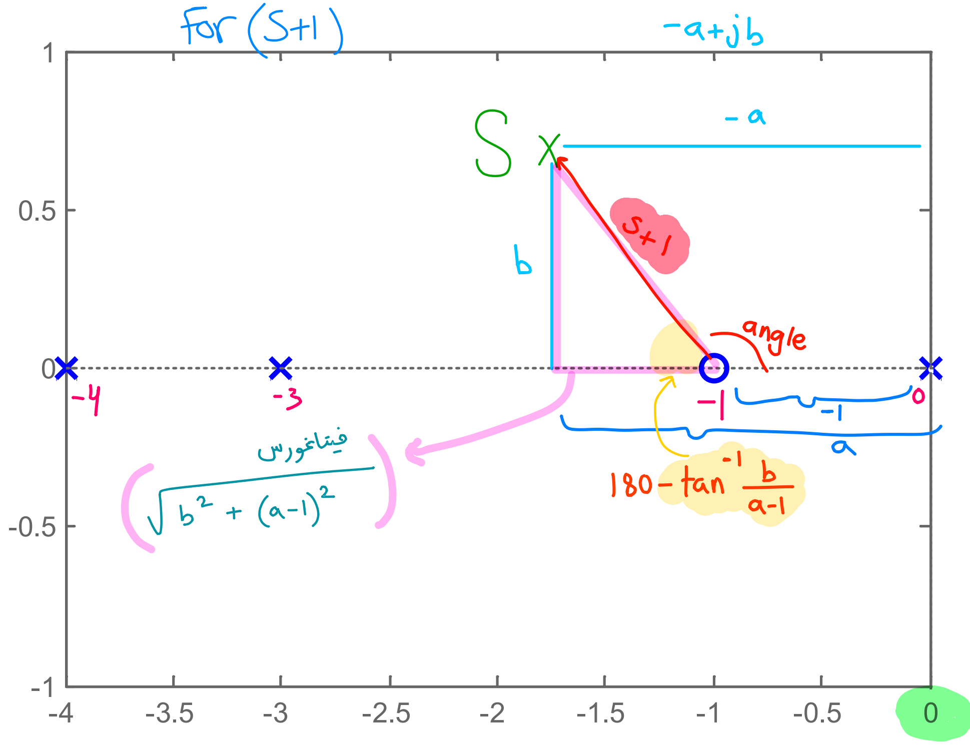
رسمنا جذور البسط ← دوائر (o)

جذو المقام ← crosses (X)

- For example if G(s)H(s) is given by

بدي احول ال complex ل magnitude وزاوية

$$G(s)H(s)=\frac{s+1}{s(s+3)(s+4)}$$



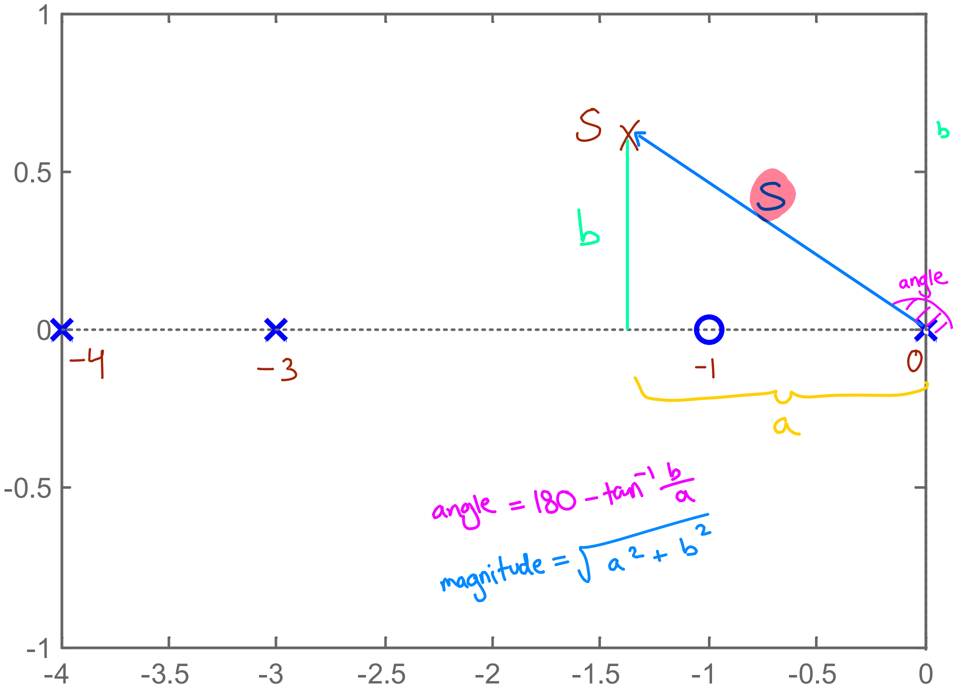
Angle and Magnitude Conditions (Graphically)

- To apply Angle and magnitude conditions graphically we must first draw the poles and zeros of G(s)H(s) in s-plane.

for (S)

- For example if G(s)H(s) is given by

$$G(s)H(s) = \frac{s+1}{s(s+3)(s+4)}$$



angle = 180 − $\tan^{-1}\frac{b}{a}$

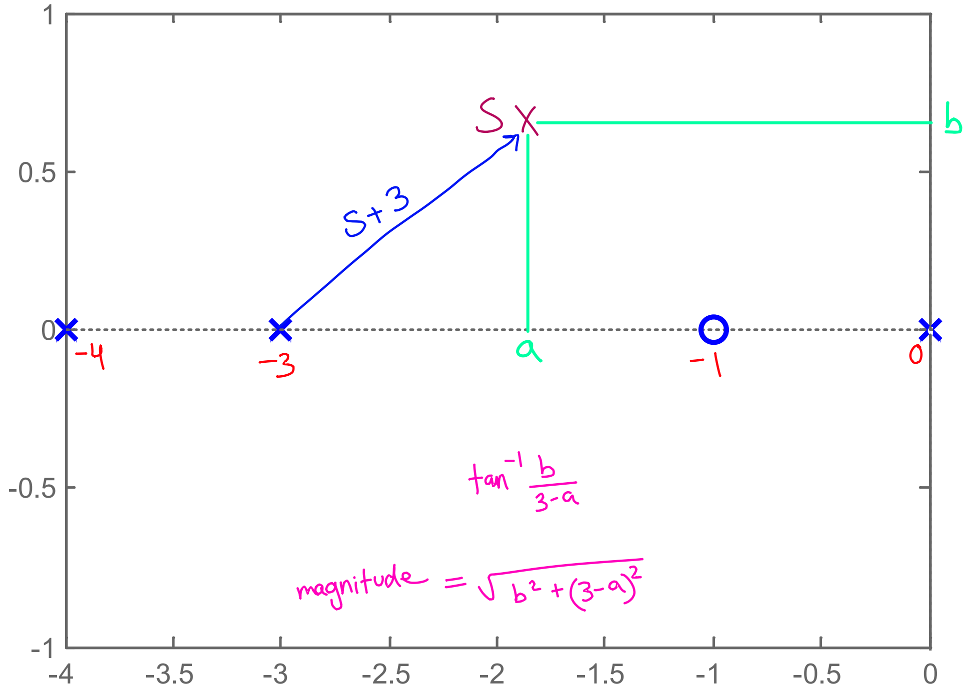
magnitude = $\sqrt{a^2+b^2}$

Angle and Magnitude Conditions (Graphically)

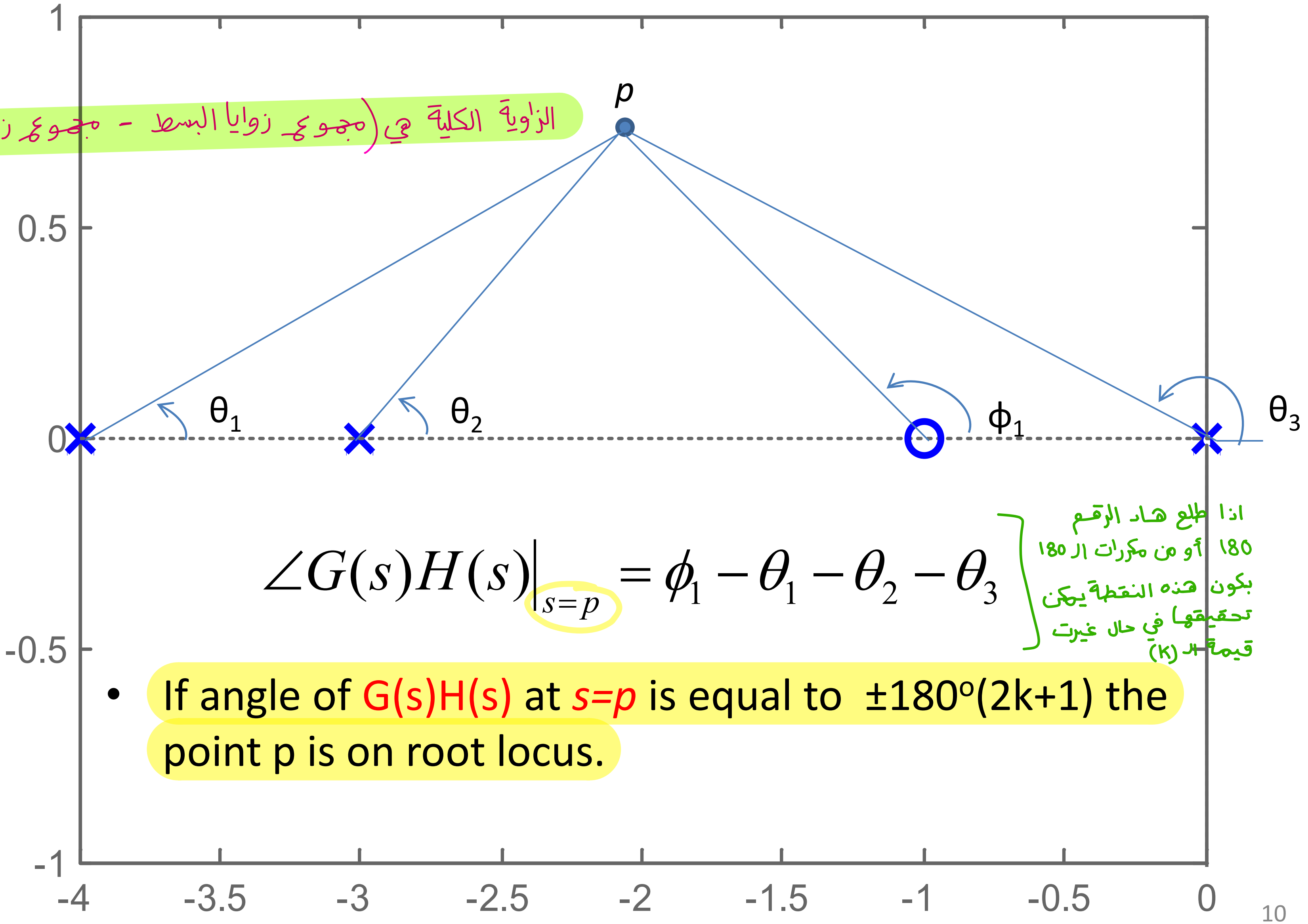
- To apply Angle and magnitude conditions graphically we must first draw the poles and zeros of G(s)H(s) in s-plane.

- For example if G(s)H(s) is given by

$$G(s)H(s) = \frac{s+1}{s(s+3)(s+4)}$$



Angle and Magnitude Conditions (Graphically)



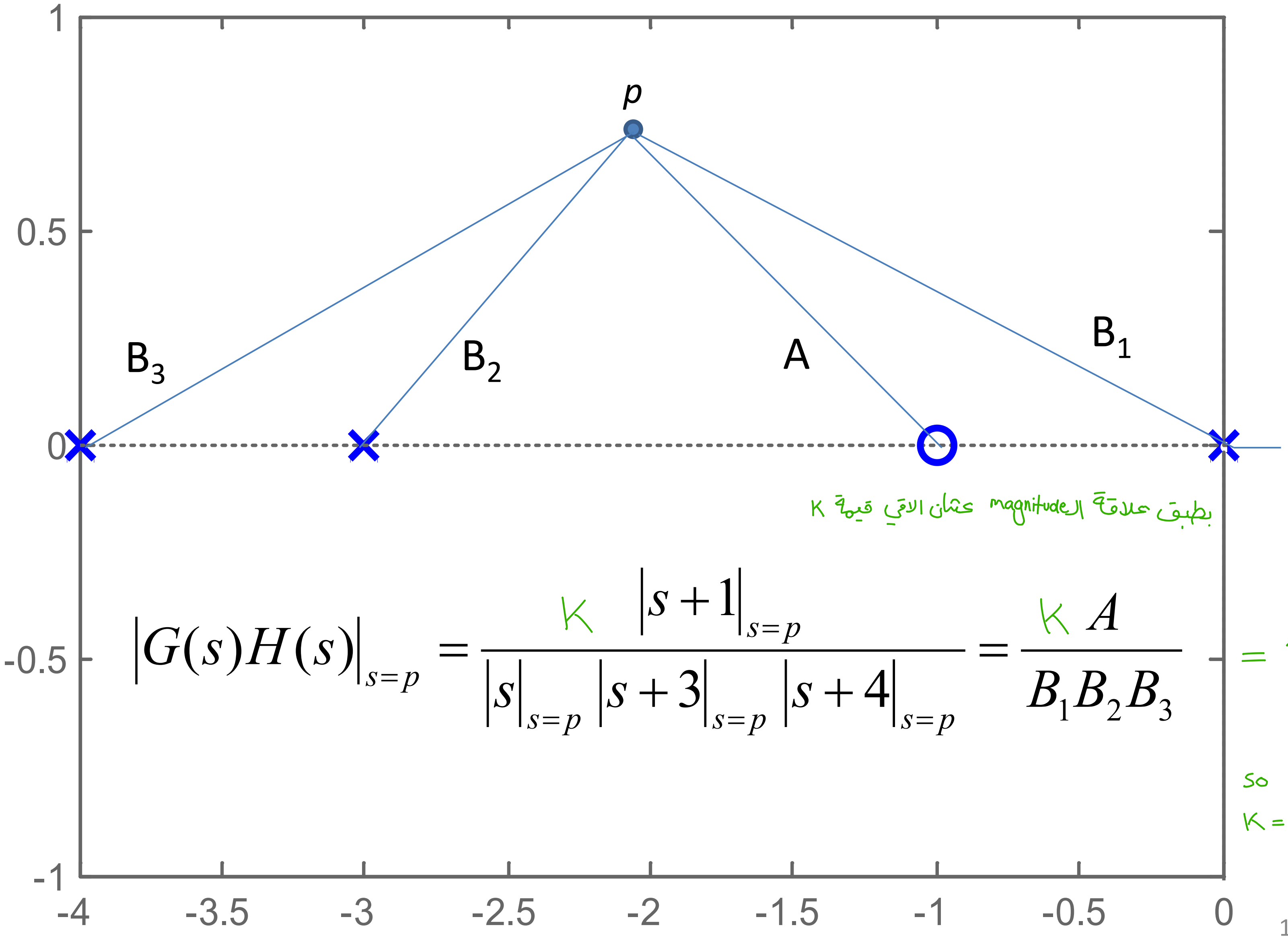
الزاوية الكلية هي (مجموع زوايا البسط - مجموع زوايا المقام)

$$\angle G(s)H(s)\Big|_{s=p} = \phi_1 - \theta_1 - \theta_2 - \theta_3$$

اذا طلع هاد الرقم ١٨٠ أو من مكررات ال ١٨٠ بكون هذه النقطة يمكن تحقيقها في حال غيرت قيمة ال (K)

- If angle of G(s)H(s) at *s=p* is equal to ±180°(2k+1) the point p is on root locus.

Angle and Magnitude Conditions graphically



بطبق علاقة الmagnitude عشان الاقي قيمة K

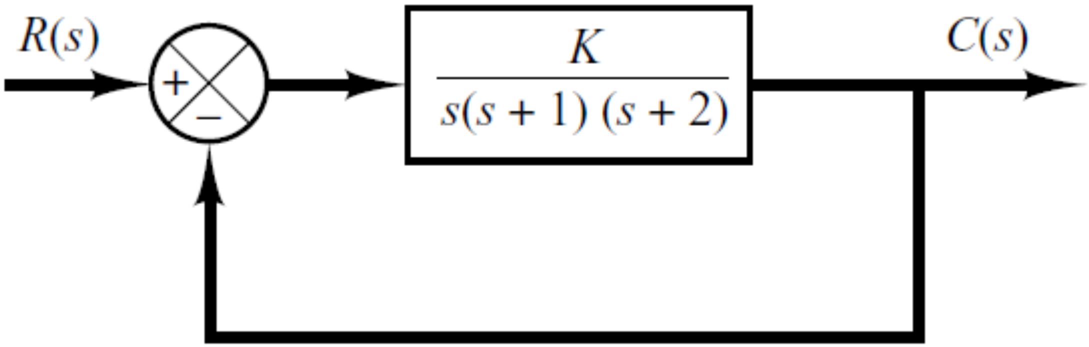
$$\left|G(s)H(s)\right|_{s=p} = \frac{K \;\left|s+1\right|_{s=p}}{\left|s\right|_{s=p}\left|s+3\right|_{s=p}\left|s+4\right|_{s=p}} = \frac{K\;A}{B_1B_2B_3} = 1$$

So K is

$$K = \frac{B_1B_2B_3}{A}$$

Illustrative Example#1

- Apply angle and magnitude conditions (Analytically as well as graphically) on following unity feedback system.



Illustrative Example#1

- Here $$G(s)H(s) = \frac{K}{s(s+1)(s+2)}$$

* لإيجاد الزوايا :-

الزوايا التي في البسط ← تجمع

الزوايا التي في المقام ← تطرح

- For the given system the angle condition becomes

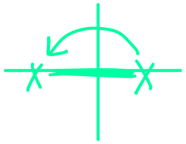
$$\angle G(s)H(s) = \angle \frac{K}{s(s+1)(s+2)}$$

$$\angle G(s)H(s) = \angle K - \angle s - \angle(s+1) - \angle(s+2)$$

$$\angle K - \angle s - \angle(s+1) - \angle(s+2) = \pm 180^\circ(2k+1)$$

Illustrative Example#1

- For example to check whether *s=-0.25* is on the root locus or not we can apply angle condition as follows.



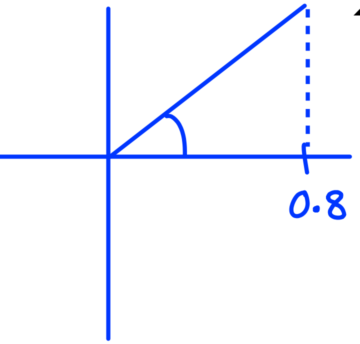
$$\angle G(s)H(s)\Big|_{s=-0.25} = \left(\angle K\Big|_{s=-0.25} - \angle s\Big|_{s=-0.25} - \angle(s+1)\Big|_{s=-0.25} - \angle(s+2)\Big|_{s=-0.25}\right)$$

zero دائماً ⊝ 180 دائماً

بعد ما اجمعهم راح يعطيني 180°

المقابل ماراح يحتلف

(يلي بختلف دائماً هو المجاور)



$$\angle G(s)H(s)\Big|_{s=-0.25} = -\angle(-0.25) - \angle(0.75) - \angle(1.75)$$

$$\angle G(s)H(s)\Big|_{s=-0.25} = \underline{-180^\circ} - 0^\circ - 0^\circ$$

اذاً يمكن من خلال تغيير قيمة ال K انو اوصل للجذر ال Dominant بالتالي نلجأ لمعادلة ال magnitude

-.25

$$\angle G(s)H(s)\Big|_{s=-0.25} = \pm 180^\circ(2k+1)$$

we have to practice the complex calculator

Illustrative Example#1

- Here $G(s)H(s) = \frac{K}{s(s+1)(s+2)}$
- And the Magnitude condition becomes

$$\left|G(s)H(s)\right| = \left|\frac{K}{s(s+1)(s+2)}\right| = 1$$

Illustrative Example#1

- Now we know from angle condition that the point s=-0.25 is on the rot locus. But we do not know the value of gain K at that specific point.

- We can use magnitude condition to determine the value of gain at any point on the root locus.

$$\boxed{\left|\frac{K}{s(s+1)(s+2)}\right|_{s=-0.25}=1}$$

$$\left|\frac{K}{(-0.25)(-0.25+1)(-0.25+2)}\right|_{s=-0.25}=1$$

Illustrative Example#1

$$\left|\frac{K}{(-0.25)(-0.25+1)(-0.25+2)}\right|_{s=-0.25} = 1$$

$$\left|\frac{K}{(-0.25)(0.75)(1.75)}\right| = 1$$

$$\left|\frac{K}{-0.3285}\right| = 1$$

★ قيمة مطلقة

$$\frac{K}{0.328} = 1$$

$$\boxed{K = 0.328}$$

Illustrative Example#1

- **Home work:**
  - check whether $s=-0.2+j0.937$ is on the root locus or not (Graphically as well as analytically) ?
  - check whether $s=-1+j2$ is on the root locus or not (Graphically as well as analytically) ?

Illustrative Example#1

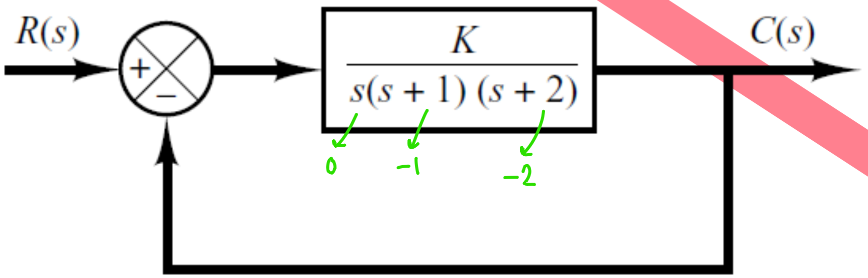
- **Home work:**
  - If $s=-0.2+j0.937$ is on the root locus determine the value of gain K at that point.
  - If $s=-1+j2$ is on the root locus determine the value of gain K at that point.

Construction of root loci

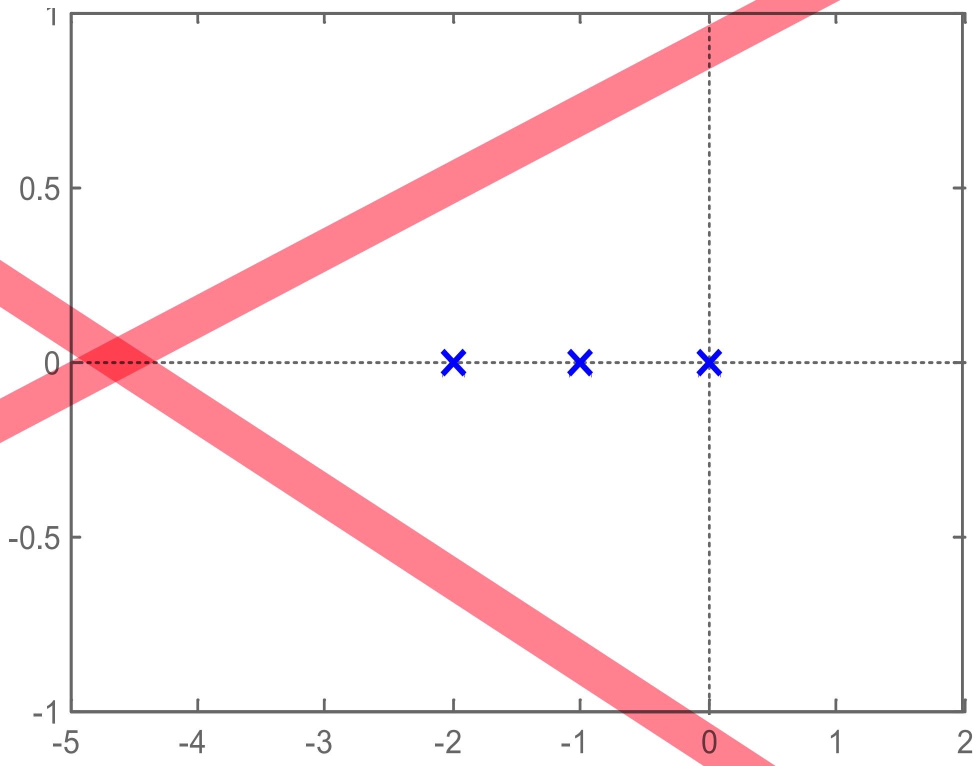
→ Not Required

كأنه مو مطلوب احنا نرسمه

- Step-1: The first step in constructing a root-locus plot is to locate the open-loop poles and zeros in s-plane.

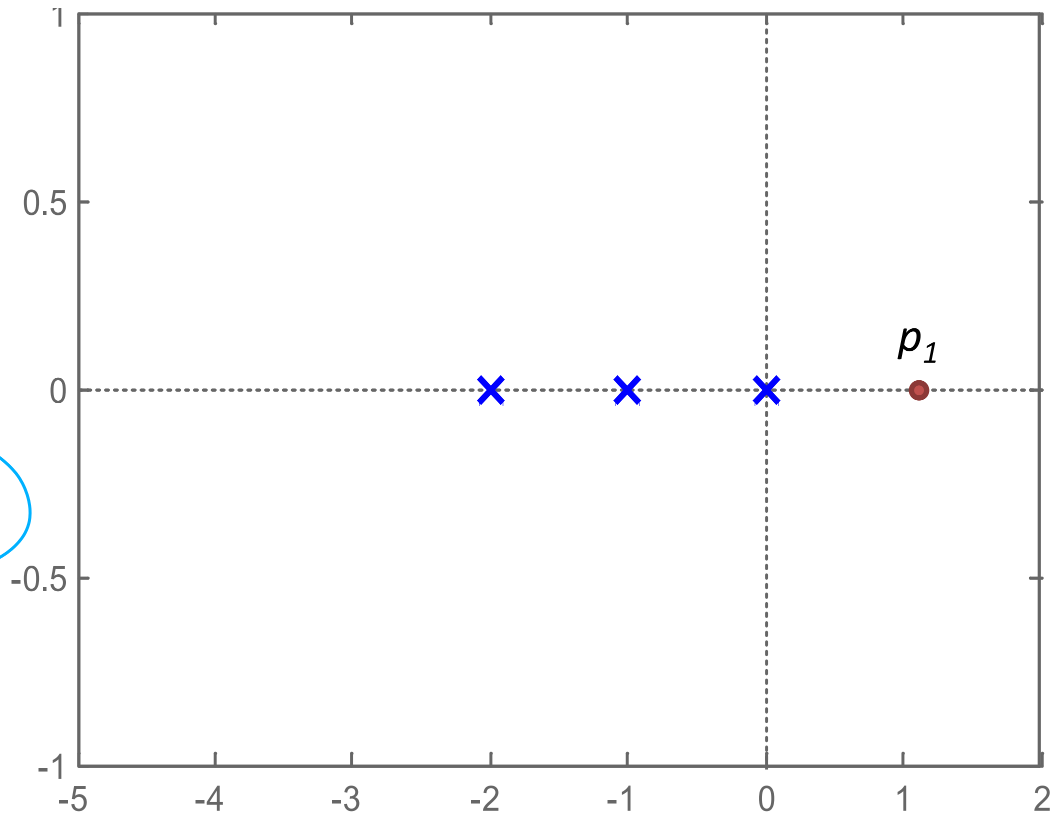


$$G(s)H(s)=\frac{K}{s(s+1)(s+2)}$$



Construction of root loci

- Step-2: Determine the root loci on the real axis.



- To determine the root loci on real axis we select some test points.
- e.g: p_1 (on positive real axis).

$$\angle s = \angle s+1 = \angle s+2 = 0°$$

- The angle condition is not satisfied.

- Hence, there is no root locus on the positive real axis.

Construction of root loci

- Step-2: Determine the root loci on the real axis.

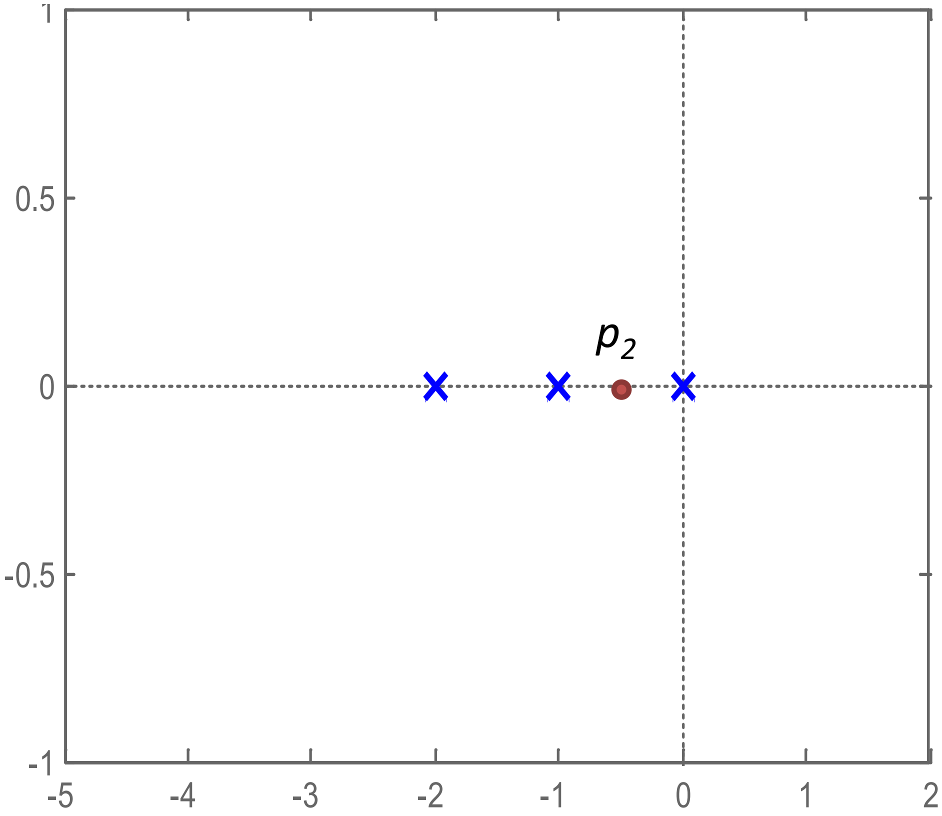
- Next, select a test point on the negative real axis between 0 and –1.
- Then

$$\underline{/s} = 180°, \qquad \underline{/s + 1} = \underline{/s + 2} = 0°$$

- Thus

$$-\underline{/s} - \underline{/s + 1} - \underline{/s + 2} = -180°$$

- The angle condition is satisfied. Therefore, the portion of the negative real axis between 0 and –1 forms a portion of the root locus.



Construction of root loci

- Step-2: Determine the root loci on the real axis.

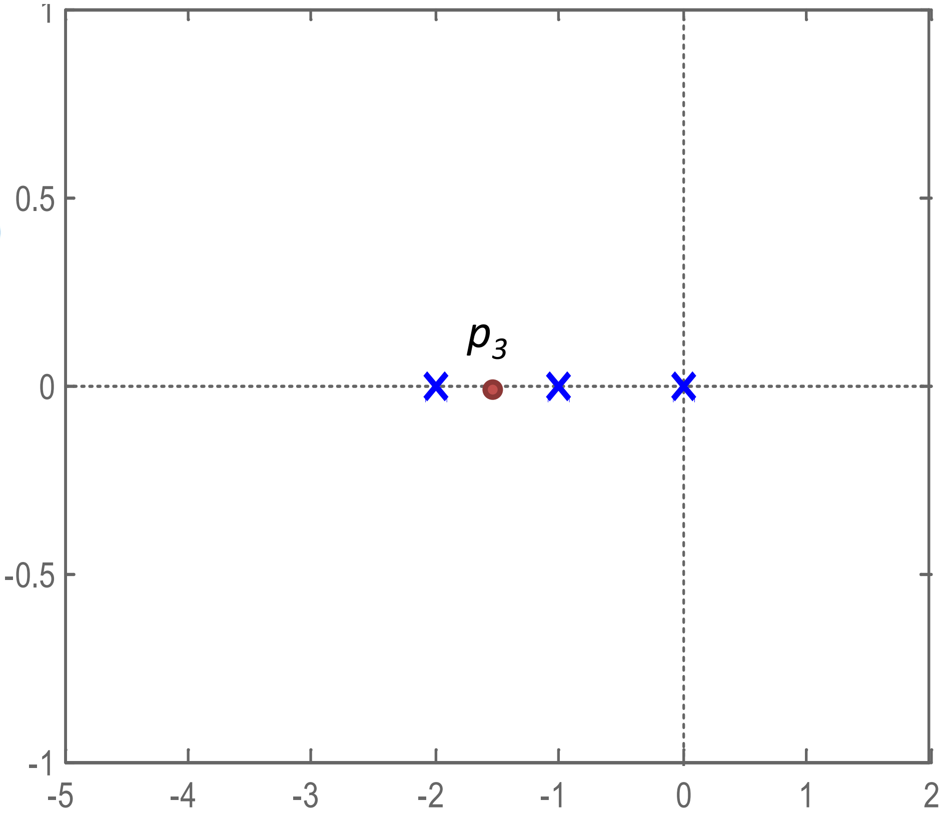
- Now, select a test point on the negative real axis between -1 and –2.
- Then

$$\underline{/s} = \underline{/s+1} = 180°, \qquad \underline{/s+2} = 0°$$

- Thus

$$-\underline{/s} - \underline{/s+1} - \underline{/s+2} = -360°$$

- The angle condition is not satisfied. Therefore, the negative real axis between -1 and –2 is not a part of the root locus.

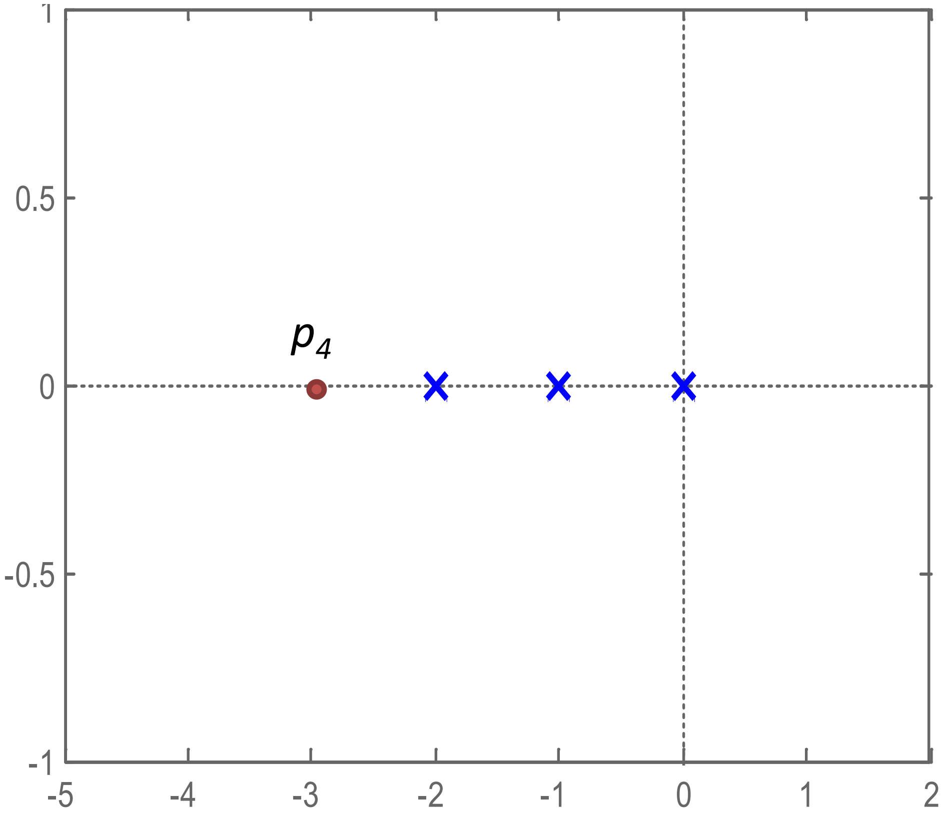


Construction of root loci

- Step-2: Determine the root loci on the real axis.

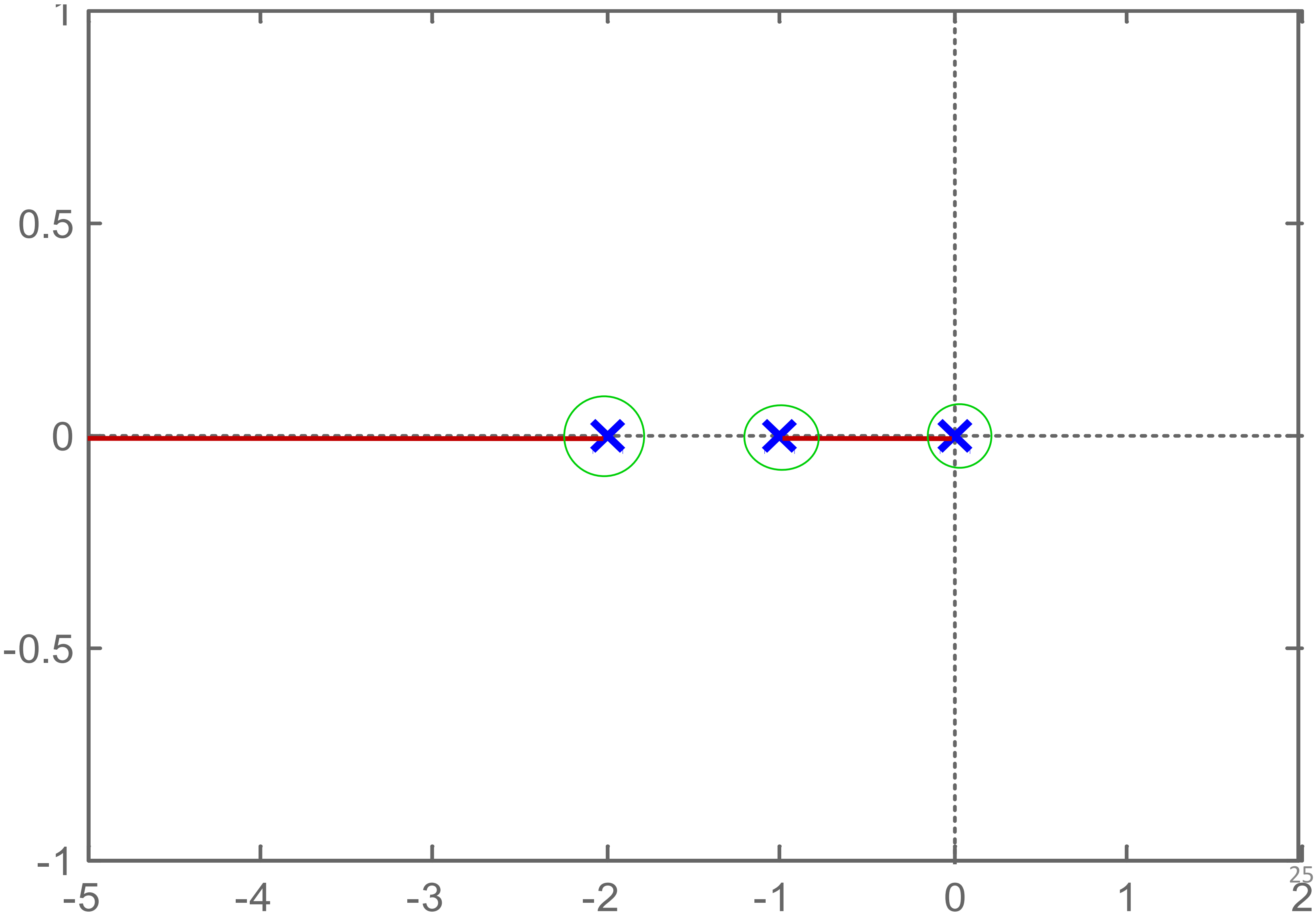
- Similarly, test point on the negative real axis between -3 and $-\infty$ satisfies the angle condition.

- Therefore, the negative real axis between -3 and $-\infty$ is part of the root locus.



Construction of root loci

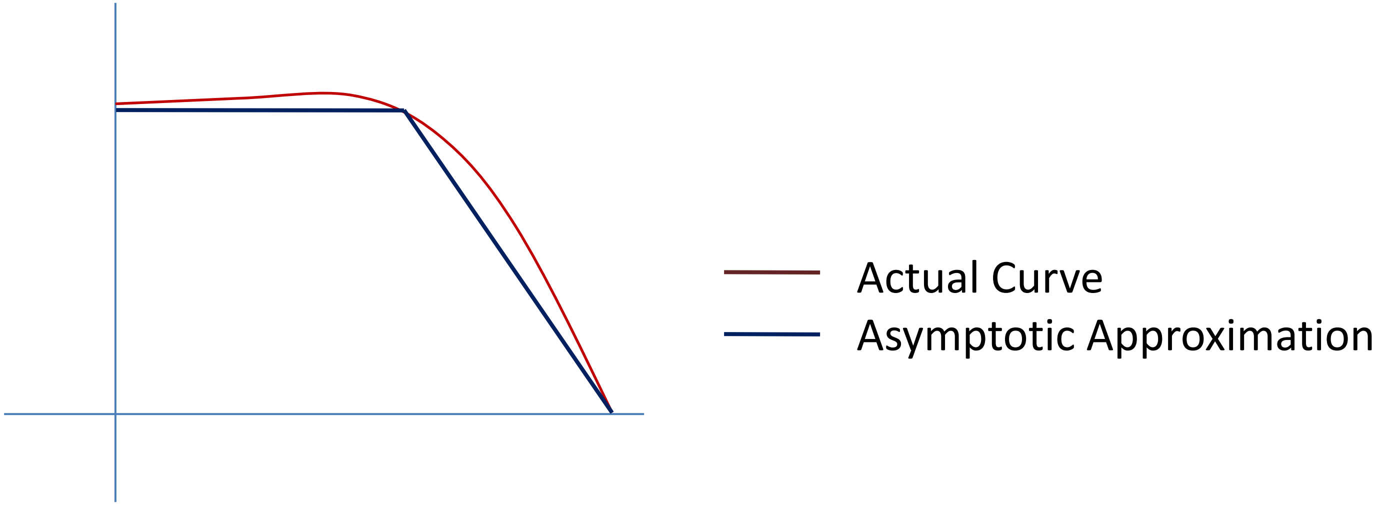
- Step-2: Determine the root loci on the real axis.



Construction of root loci

- Step-3: Determine the *asymptotes* of the root loci.

Asymptote is the straight line approximation of a curve



Construction of root loci

- Step-3: Determine the *asymptotes* of the root loci.

$$Angle\ of\ asymptotes = \psi = \frac{\pm 180^\circ(2k+1)}{n-m}$$

- where
- n-----> number of poles
- m-----> number of zeros
- For this Transfer Function $G(s)H(s) = \frac{K}{s(s+1)(s+2)}$

$$\psi = \frac{\pm 180^\circ(2k+1)}{3-0}$$

Construction of root loci

- Step-3: Determine the *asymptotes* of the root loci.

$$
\begin{aligned}
\psi &= \pm 60^\circ && \text{when } k = 0 \\
&= \pm 180^\circ && \text{when } k = 1 \\
&= \pm 300^\circ && \text{when } k = 2 \\
&= \pm 420^\circ && \text{when } k = 3
\end{aligned}
$$

- Since the angle repeats itself as k is varied, the distinct angles for the asymptotes are determined as 60°, –60°, -180°and 180°.
- Thus, there are three asymptotes having angles 60°, –60°, 180°.

Construction of root loci

- Step-3: Determine the *asymptotes* of the root loci.

- Before we can draw these asymptotes in the complex plane, we must find the point where they intersect the real axis.

- Point of intersection of asymptotes on real axis (or centroid of asymptotes) can be find as out

$$\sigma = \frac{\sum poles - \sum zeros}{n - m}$$

Construction of root loci

- Step-3: Determine the *asymptotes* of the root loci.

- For $G(s)H(s) = \dfrac{K}{s(s+1)(s+2)}$

$$\sigma = \frac{(0-1-2)-0}{3-0}$$

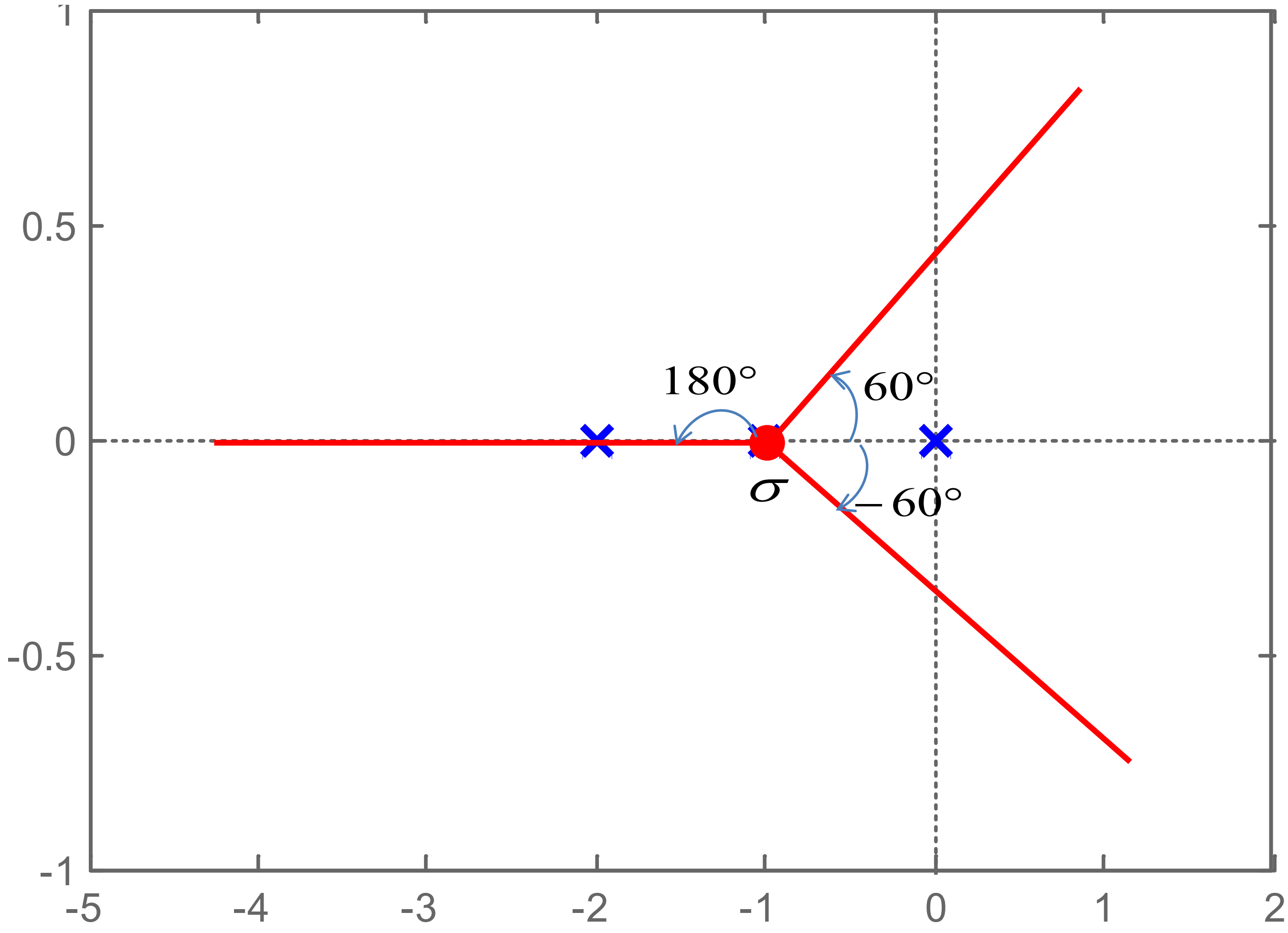
$$\sigma = \frac{-3}{3} = -1$$

Construction of root loci

- Step-3: Determine the *asymptotes* of the root loci.

$$\psi = 60^\circ, -60^\circ, 180^\circ$$

$$\sigma = -1$$

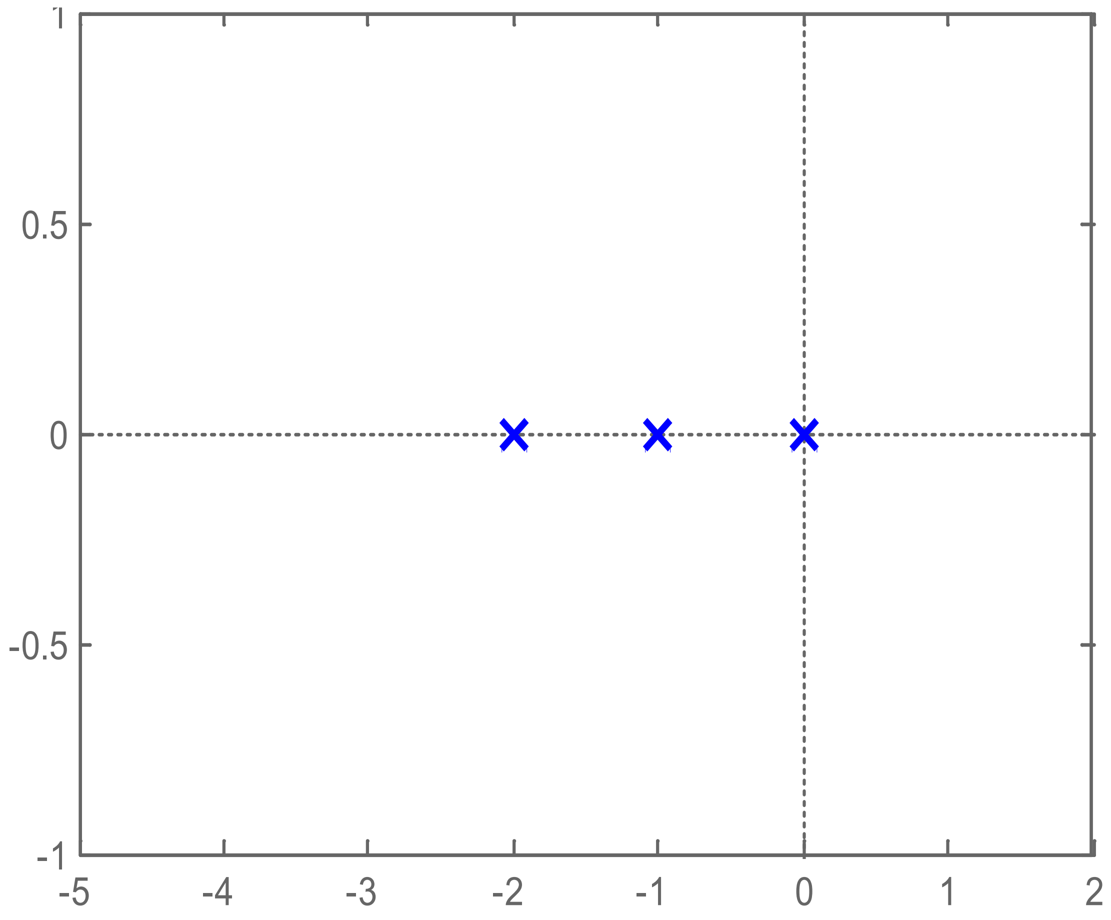


Construction of root loci

- Step-4: Determine the *breakaway point*.

  - The breakaway point corresponds to a point in the s plane where multiple roots of the characteristic equation occur.

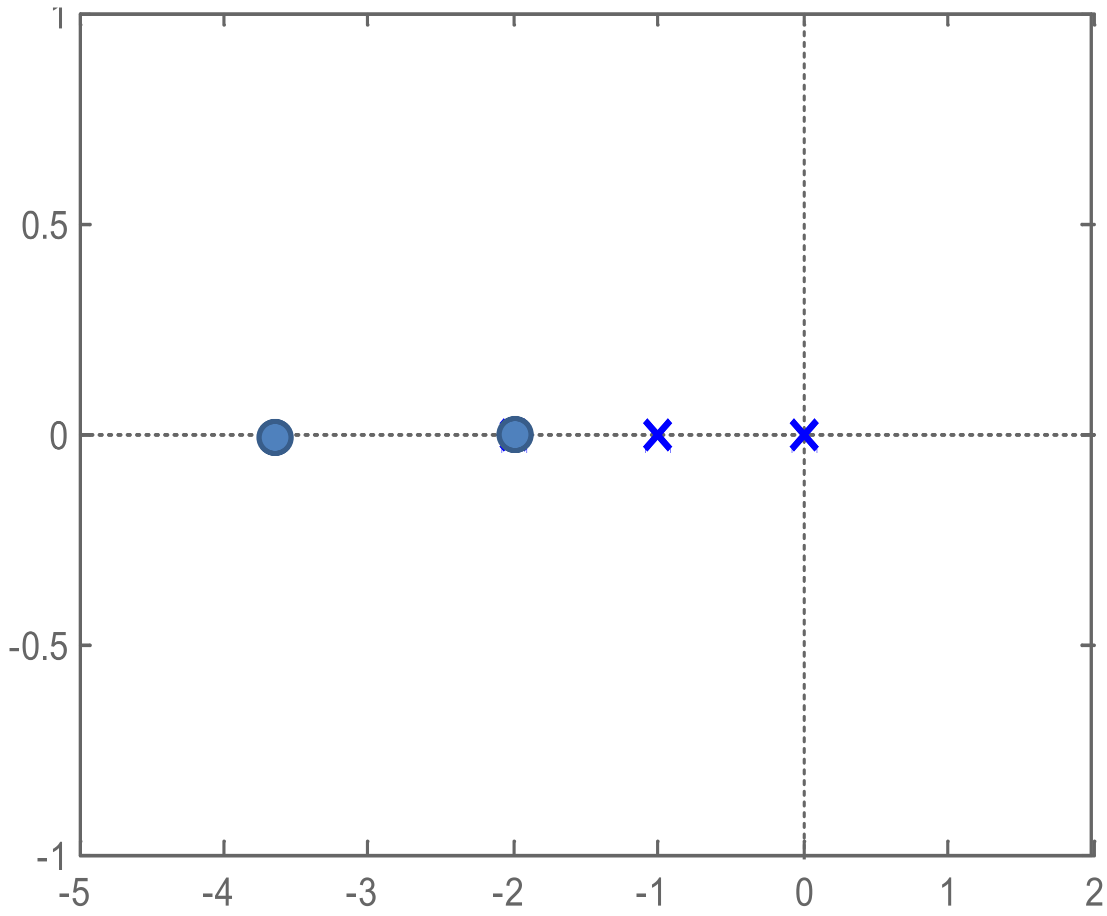
  - It is the point from which the root locus branches leaves real axis and enter in complex plane.



Construction of root loci

- Step-4: Determine the *break-in point*.

  - The break-in point corresponds to a point in the s plane where multiple roots of the characteristic equation occur.
  - It is the point where the root locus branches arrives at real axis.



Construction of root loci

- Step-4: Determine the *breakaway point* or *break-in point*.

- The breakaway or break-in points can be determined from the roots of

$$\frac{dK}{ds} = 0$$

- It should be noted that not all the solutions of dK/ds=0 correspond to actual breakaway points.

- If a point at which dK/ds=0 is on a root locus, it is an actual breakaway or break-in point.

- Stated differently, if at a point at which dK/ds=0 the value of K takes a real positive value, then that point is an actual breakaway or break-in point.

Construction of root loci

- Step-4: Determine the *breakaway point* or *break-in point*.

$$G(s)H(s) = \frac{K}{s(s+1)(s+2)}$$

- The characteristic equation of the system is

$$1 + G(s)H(s) = 1 + \frac{K}{s(s+1)(s+2)} = 0$$

$$\frac{K}{s(s+1)(s+2)} = -1$$

$$K = -\left[s(s+1)(s+2)\right]$$

- The breakaway point can now be determined as

$$\frac{dK}{ds} = -\frac{d}{ds}\left[s(s+1)(s+2)\right]$$

Construction of root loci

- Step-4: Determine the *breakaway point* or *break-in point*.

$$\frac{dK}{ds} = -\frac{d}{ds}\left[s(s+1)(s+2)\right]$$

$$\frac{dK}{ds} = -\frac{d}{ds}\left[s^3 + 3s^2 + 2s\right]$$

$$\frac{dK}{ds} = -3s^2 - 6s - 2$$

- Set *dK/ds=0* in order to determine breakaway point.

$$-3s^2 - 6s - 2 = 0$$

$$3s^2 + 6s + 2 = 0$$

$$s = -0.4226$$

$$= -1.5774$$

Construction of root loci

- Step-4: Determine the *breakaway point* or *break-in point*.

$$s = -0.4226$$
$$= -1.5774$$

- Since the breakaway point must lie on a root locus between 0 and –1, it is clear that *s=–0.4226* corresponds to the actual breakaway point.

- Point s=–1.5774 is not on the root locus. Hence, this point is not an actual breakaway or break-in point.

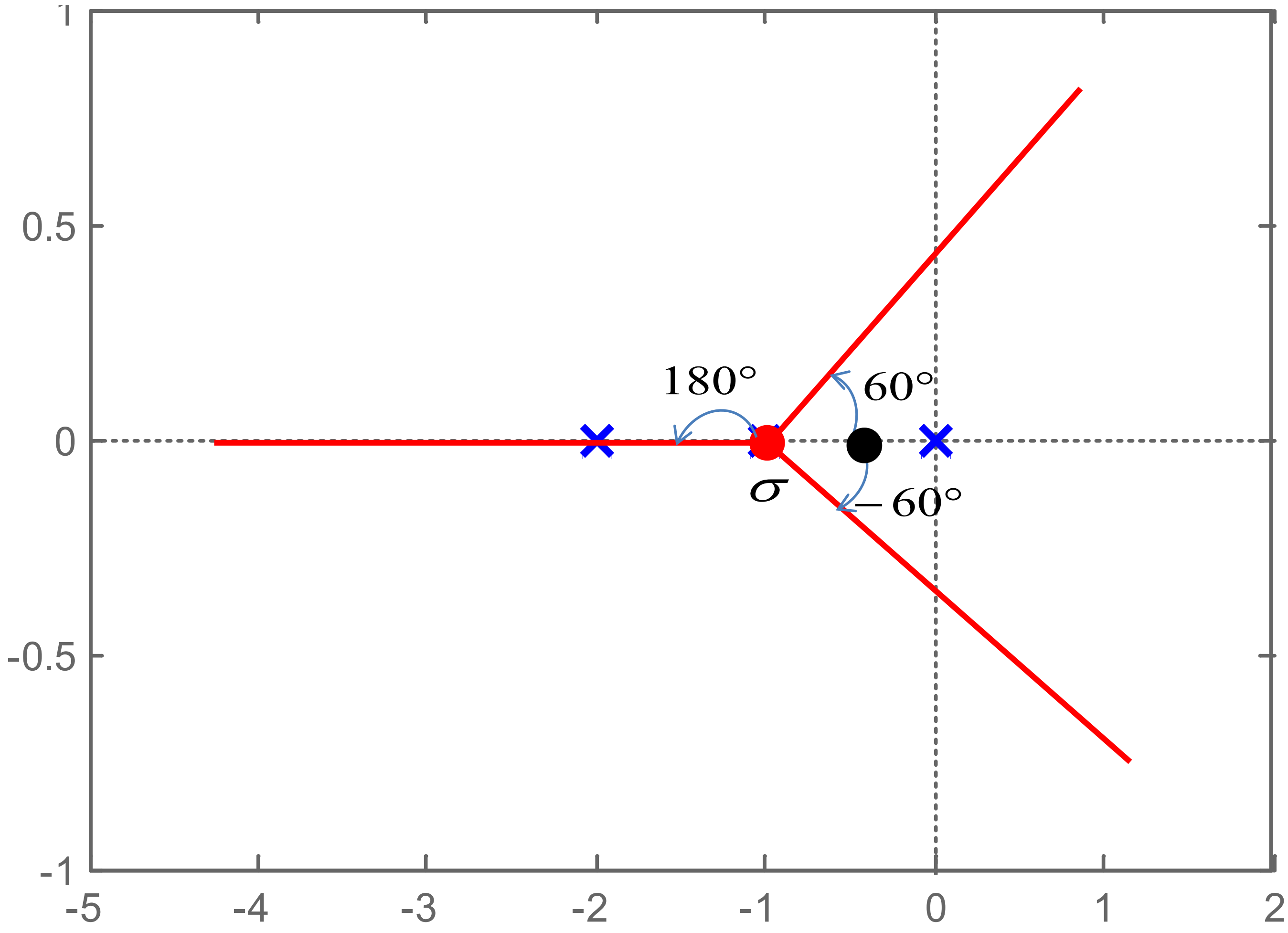
- In fact, evaluation of the values of K corresponding to s=–0.4226 and s=–1.5774 yields

$$K = 0.3849, \qquad \text{for } s = -0.4226$$
$$K = -0.3849, \qquad \text{for } s = -1.5774$$

Construction of root loci

- Step-4: Determine the *breakaway point*.

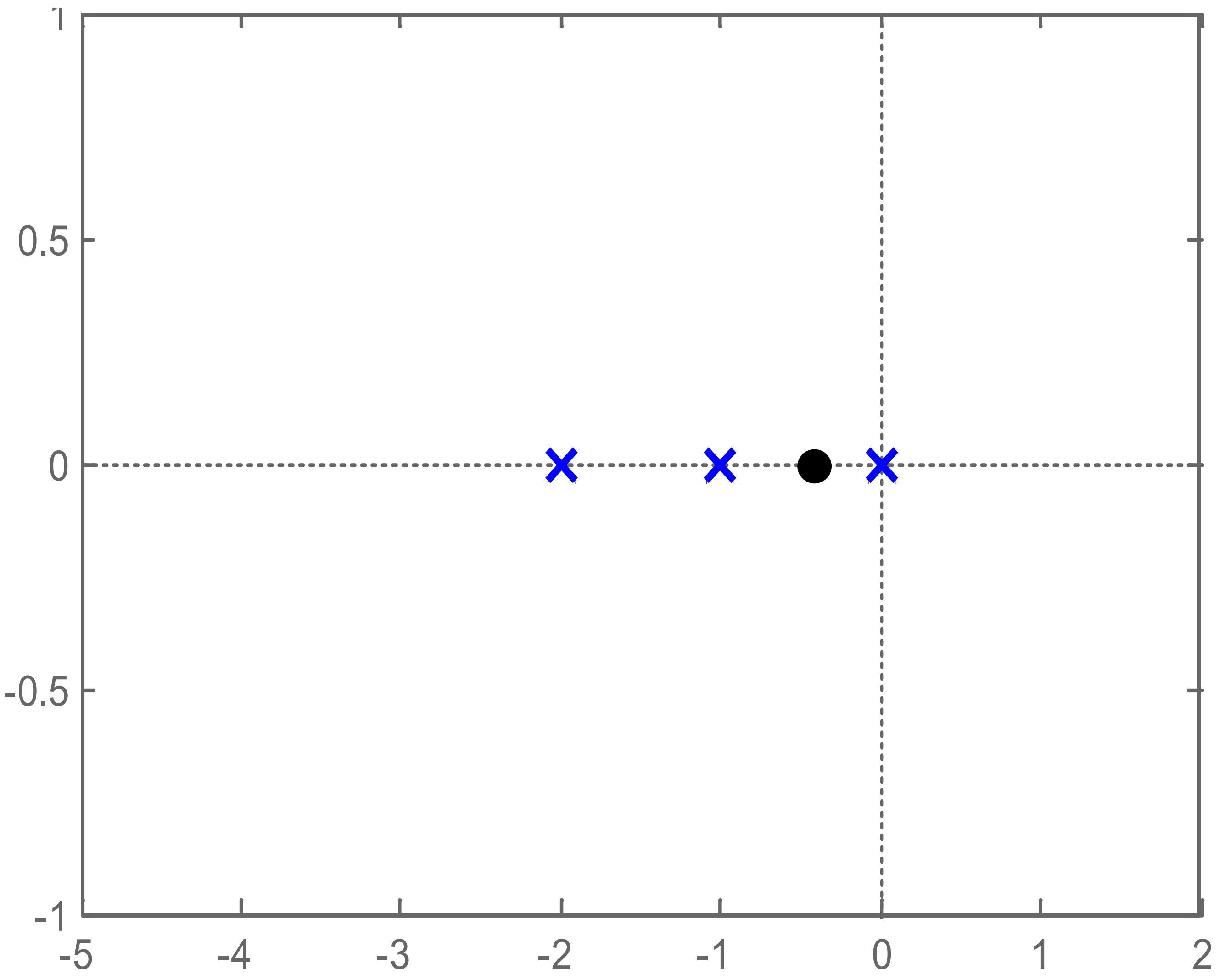
$s = -0.4226$



Construction of root loci

- Step-4: Determine the *breakaway point*.

$$s = -0.4226$$



Home Work

- Determine the Breakaway and break in points

$$KG(s)H(s) = \frac{K(s-3)(s-5)}{(s+1)(s+2)}$$

Solution

$$KG(s)H(s) = \frac{K(s-3)(s-5)}{(s+1)(s+2)} = \frac{K(s^2-8s+15)}{(s^2+3s+2)}$$

$$\frac{K(s^2-8s+15)}{s^2+3s+2} = -1$$

$$K = -\frac{(s^2+3s+2)}{(s^2-8s+15)}$$

- Differentiating ***K*** with respect to **s** and setting the derivative equal to zero yields;

$$\frac{dK}{ds} = -\frac{[(s^2-8s+15)(2s+3)-(s^2+3s+2)(2s-8)]}{(s^2-8s+15)^2} = 0$$

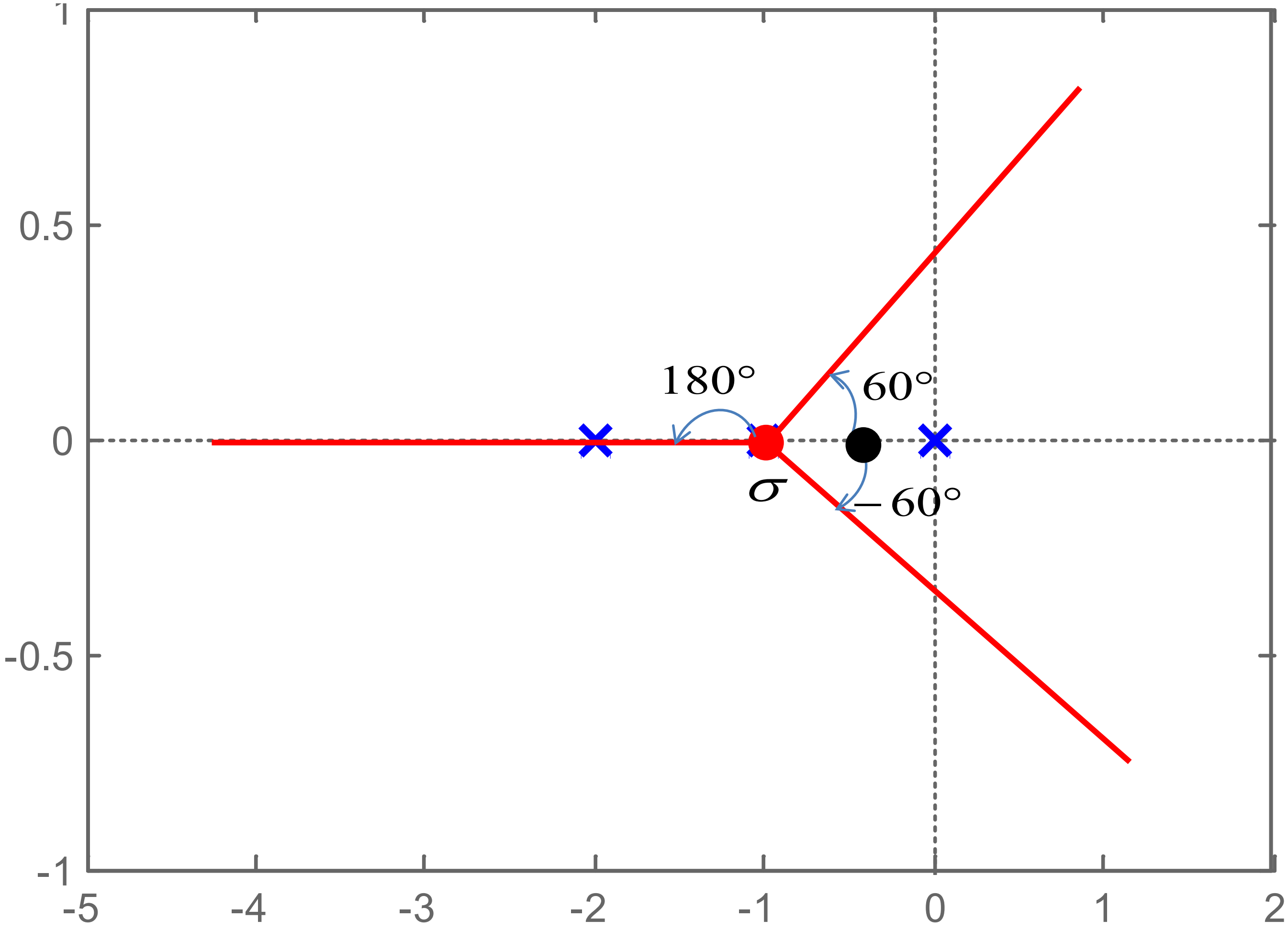
$$11s^2 - 26s - 61 = 0$$

Hence, solving for s, we find the break-away and break-in points;

$$\boxed{s = -1.45 \text{ and } 3.82}$$

Construction of root loci

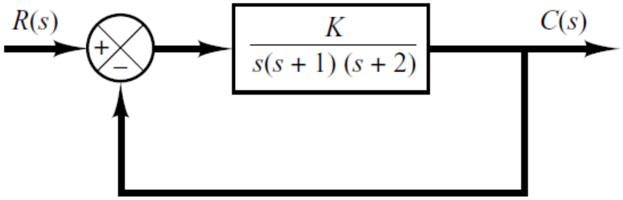
- Step-5: Determine the points where root loci cross the imaginary axis.



Construction of root loci

- Step-5: Determine the points where root loci cross the imaginary axis.
  - These points can be found by use of Routh's stability criterion.
  - Since the characteristic equation for the present system is

$$s^3 + 3s^2 + 2s + K = 0$$



  - The Routh Array Becomes

$$\begin{array}{lll} s^3 & 1 & 2 \\ s^2 & 3 & K \\ s^1 & \dfrac{6-K}{3} & \\ s^0 & K & \end{array}$$

Construction of root loci

- Step-5: Determine the points where root loci cross the imaginary axis.

- The value(s) of K that makes the system marginally stable is 6.

- The crossing points on the imaginary axis can then be found by solving the auxiliary equation obtained from the s^2 row, that is,

$$3s^2 + K = 3s^2 + 6 = 0$$

- Which yields

$$s = \pm j\sqrt{2}$$

s^3	1	2
s^2	3	K
s^1	$\frac{6-K}{3} = 0$	
s^0	K	

$K = 6$

Construction of root loci

- Step-5: Determine the points where root loci cross the imaginary axis.

- An alternative approach is to let s=jω in the characteristic equation, equate both the real part and the imaginary part to zero, and then solve for ω and K.

- For present system the characteristic equation is

$$s^3 + 3s^2 + 2s + K = 0$$

$$(j\omega)^3 + 3(j\omega)^2 + 2j\omega + K = 0$$

$$(K - 3\omega^2) + j(2\omega - \omega^3) = 0$$

Construction of root loci

إلى هنا ينتهي

- **Step-5**: Determine the points where root loci cross the imaginary axis.

$$(K-3\omega^2)+j(2\omega-\omega^3)=0$$

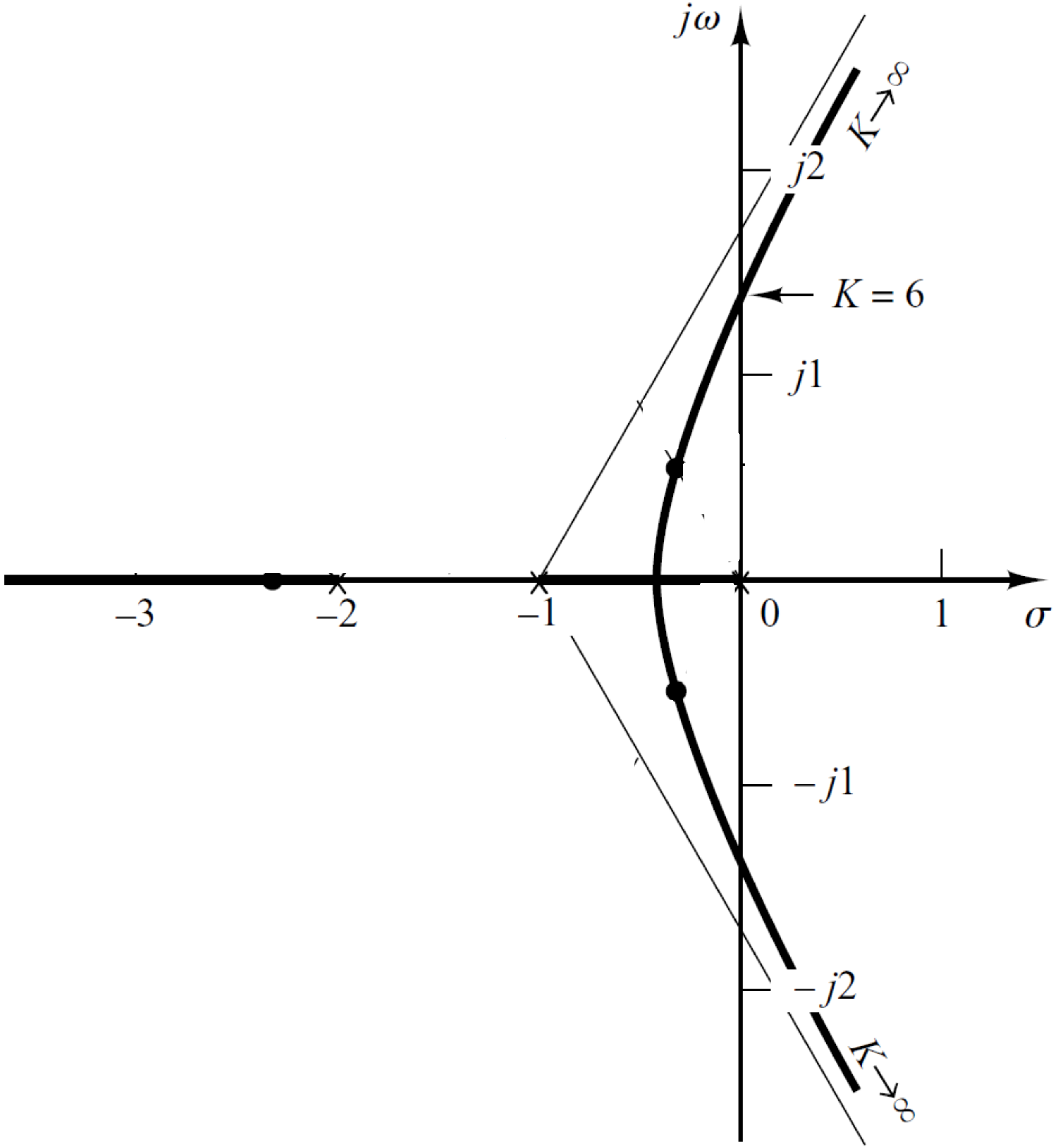
- Equating both real and imaginary parts of this equation to zero

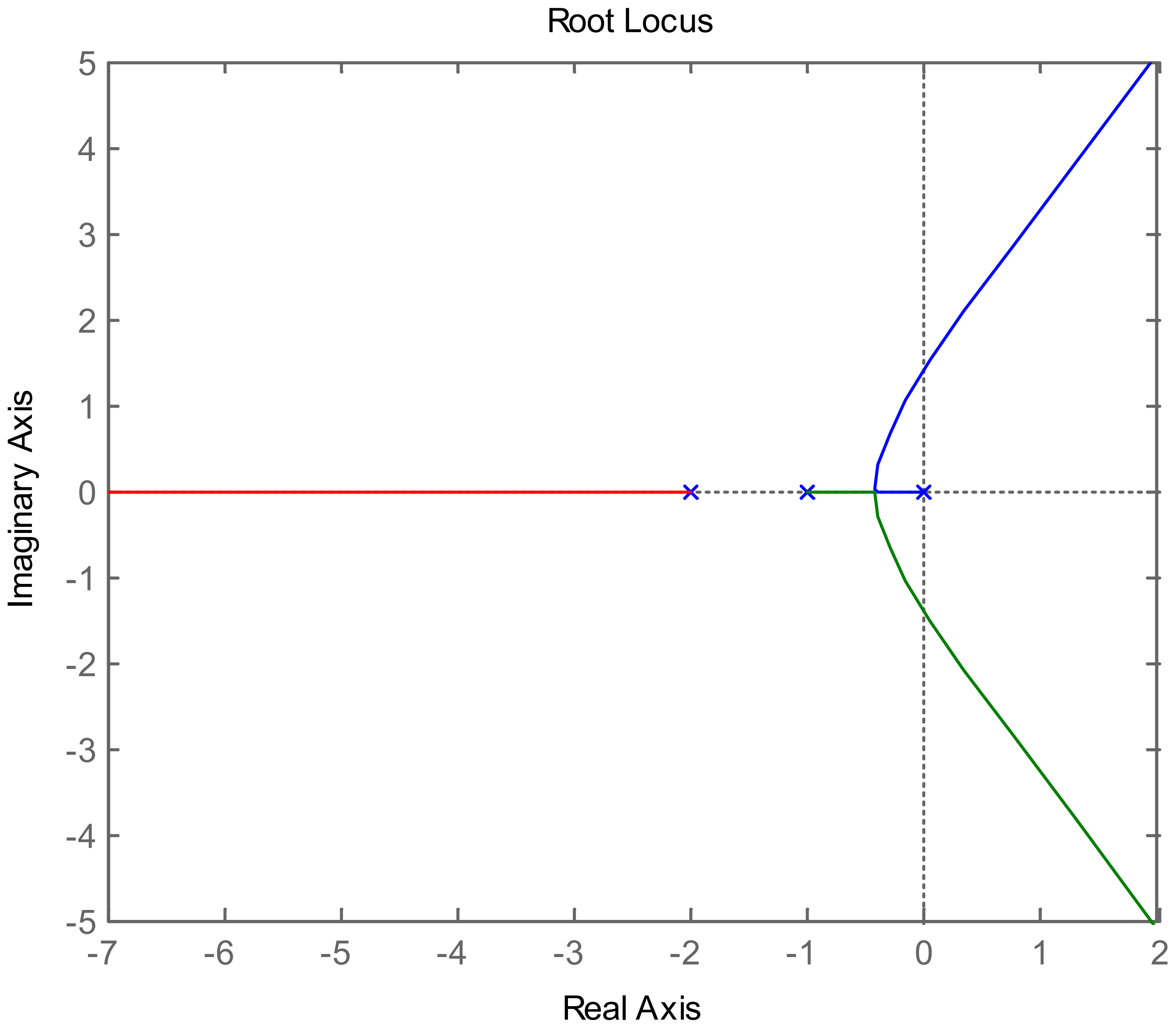
$$(2\omega-\omega^3)=0$$

$$(K-3\omega^2)=0$$

- Which yields

$$\omega = \pm\sqrt{2}, \qquad K = 6 \qquad \text{or} \qquad \omega = 0, \qquad K = 0$$

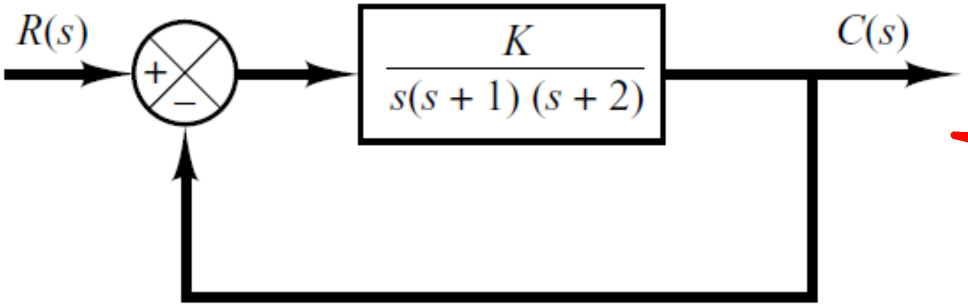

jω
K→∞
j2
K = 6
j1
–3
–2
–1
0
1
σ
– j1
– j2
K→∞


Root Locus
5
4
3
2
1
0
-1
-2
-3
-4
-5
-7
-6
-5
-4
-3
-2
-1
0
1
2
Imaginary Axis
Real Axis

Example#1

من هنا

- Consider following unity feedback system.



- Determine the value of K such that the damping ratio of a pair of dominant complex-conjugate closed-loop poles is 0.5.

$\delta = 0.5 = \cos\theta$

$\theta = \cos^{-1} 0.5$

$\theta = 60^\circ$

$$G(s)H(s) = \frac{K}{s(s+1)(s+2)}$$

Example#1

- The damping ratio of 0.5 corresponds to

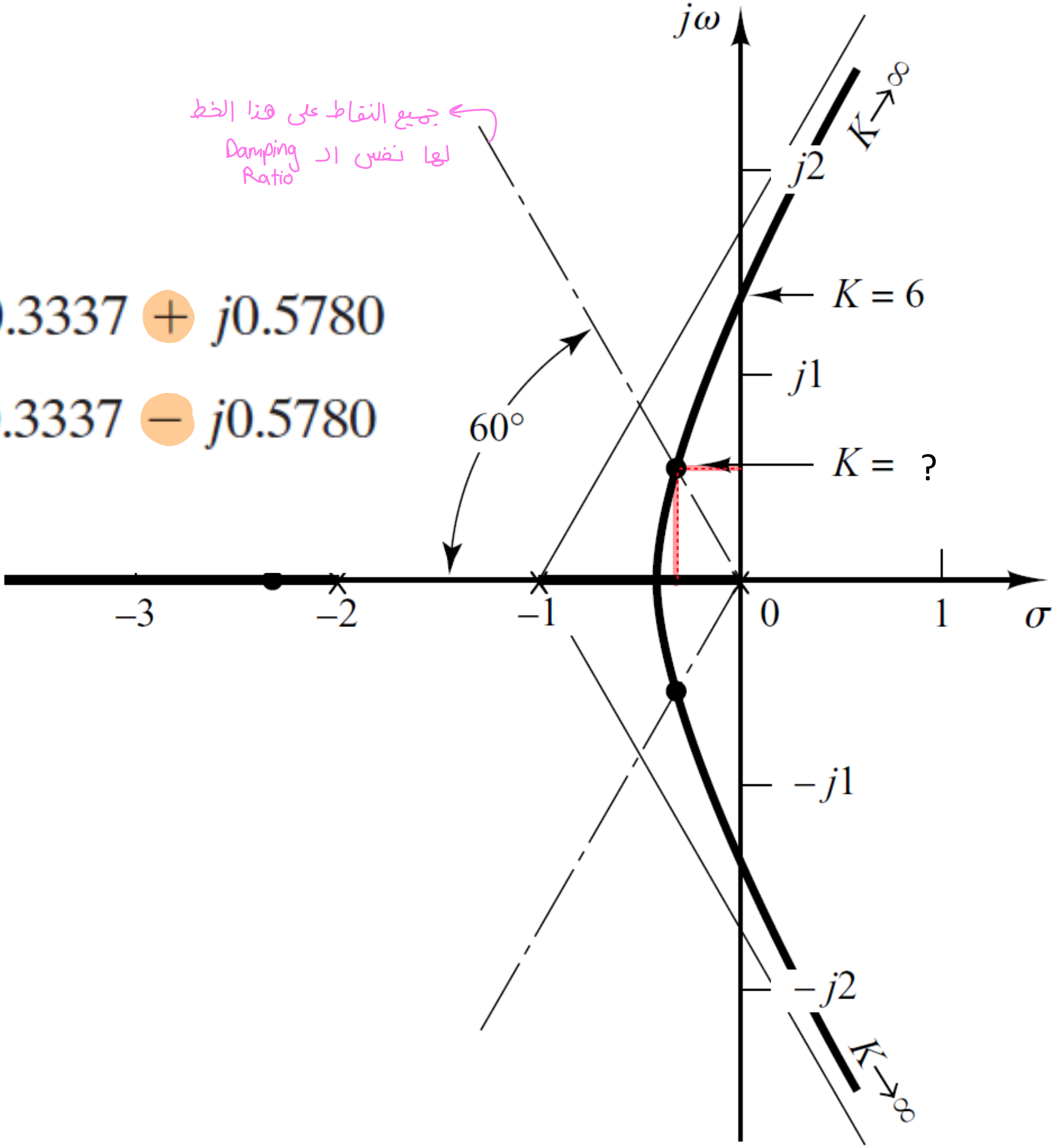
$$\zeta = \cos\theta$$

$$\theta = \cos^{-1}\zeta$$

$$\theta = \cos^{-1}(0.5) = 60^\circ$$

$$s_1 = -0.3337 + j0.5780$$

$$s_2 = -0.3337 - j0.5780$$



جميع النقاط على هذا الخط لها نفس ال Damping Ratio

Example#1

- The value of K that yields such poles is found from the magnitude condition

$$\left|\frac{K}{s(s+1)(s+2)}\right|_{s=-0.3337+j0.5780} = 1$$

$$K = |s(s+1)(s+2)|_{s=-0.3337+j0.5780}$$
$$= 1.0383$$

$$\left|\frac{K}{s(s+1)(s+2)}\right|_{s=-0.3337+j0.5780}=1$$

$$\left|\frac{K}{\underbrace{(-0.3337+j0.5780)}_{}\times(1-0.3337+j0.5780)(2-0.3337+j0.5780)}\right|=1$$

عشان اطلع الmagnitude

on calculator

→ بحطها complex بالأول

↓

$\text{Shift} \rightarrow \overset{\text{Abs}}{\boxed{(}}$

و بدخل

$|-0.3337+0.5780i|$

$=0.66741$

و هكذا لباقي الاجزاء و بينهم ضرب

$$\left|\frac{K}{0.66741\times0.8820\times1.7637}\right|=1$$

$$\frac{K}{1.0382}=1$$

$$\boxed{K=1.0382}$$

Example#1

- The value of K that yields such poles is found from the magnitude condition

$$\left|\frac{K}{s(s+1)(s+2)}\right|_{s=-0.3337+j0.5780} = 1$$

$$K = |s(s+1)(s+2)|_{s=-0.3337+j0.5780}$$
$$= 1.0383$$

Root locus on Matlab

```
S=tf('s')
g= 1/(s*(s+1)*(s+2))
rlocus(g)
```

أو
عن طريق

```
Sisotool(g)
```

تعطيني الResponse

* Root locus تبدأ من جذور المقام عندما K=0

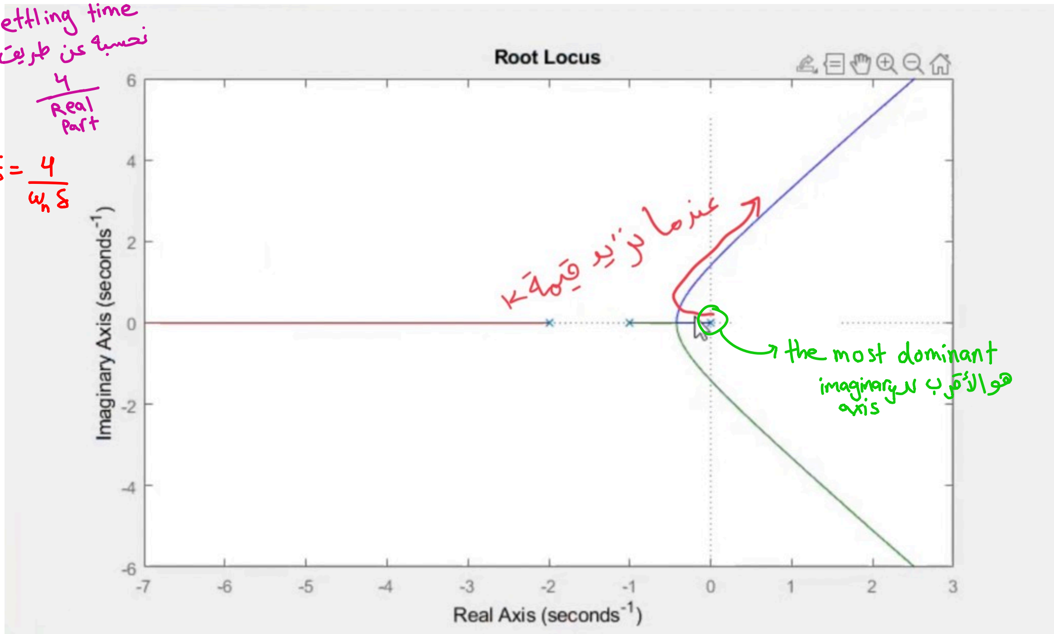
* كلما قلت الزاوية بزيد الـ cos تبعها و بزيد الـ Damping Ratio

* لما نقلل الـ complex part الـ Rise time يزداد والـ Settling time يقل

Settling time
نحسبه عن طريق

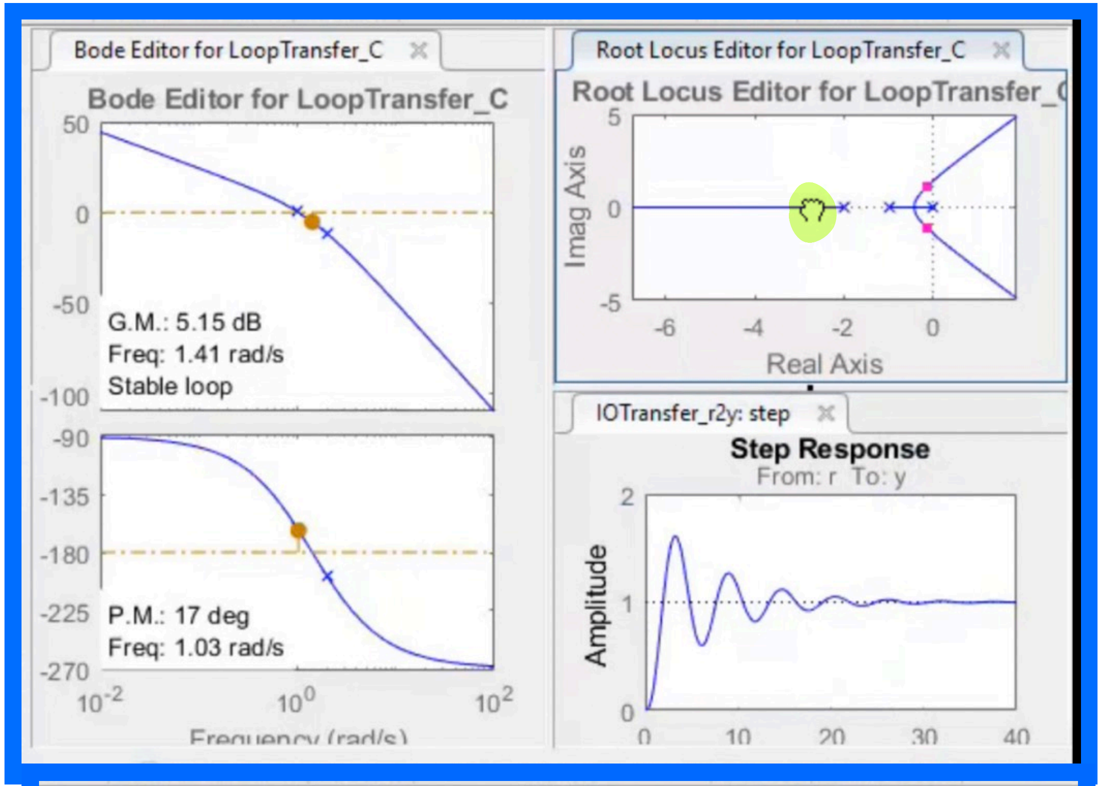
$$\frac{4}{\text{Real part}}$$

$$T_s = \frac{4}{\omega_n \delta}$$



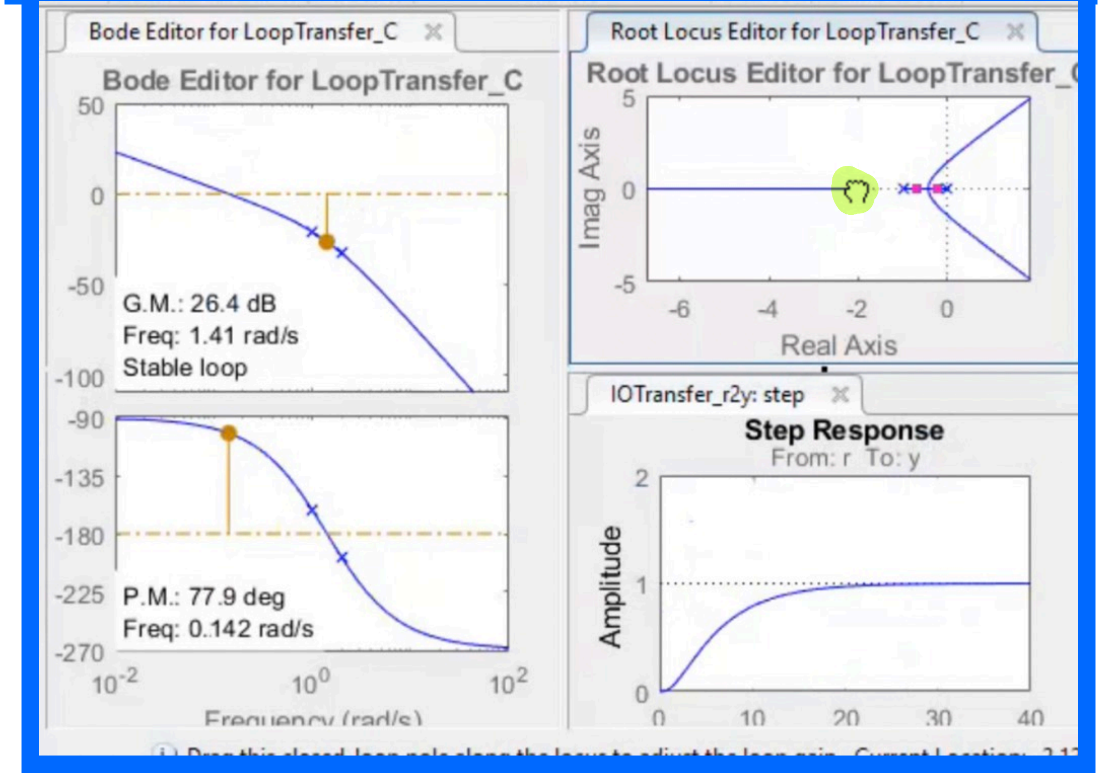
عندما نزيد قيمة K

the most dominant
هو الأقرب لـ imaginary axis



G.M.: 5.15 dB
Freq: 1.41 rad/s
Stable loop

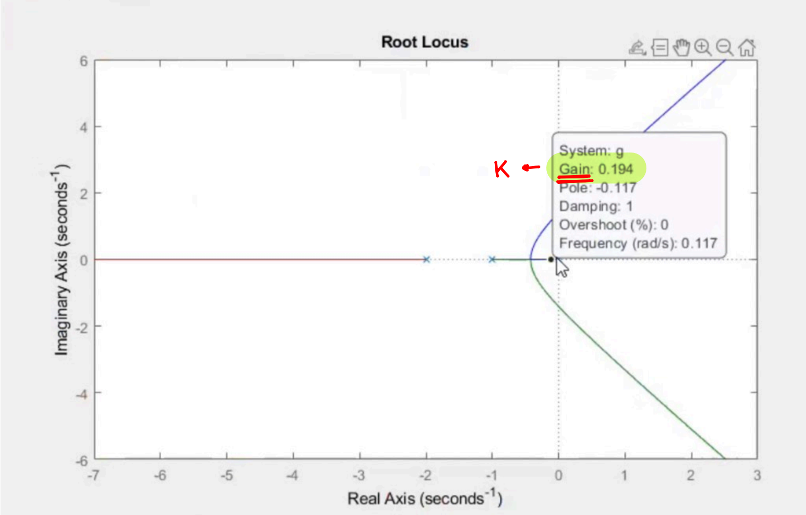
P.M.: 17 deg
Freq: 1.03 rad/s



G.M.: 26.4 dB
Freq: 1.41 rad/s
Stable loop

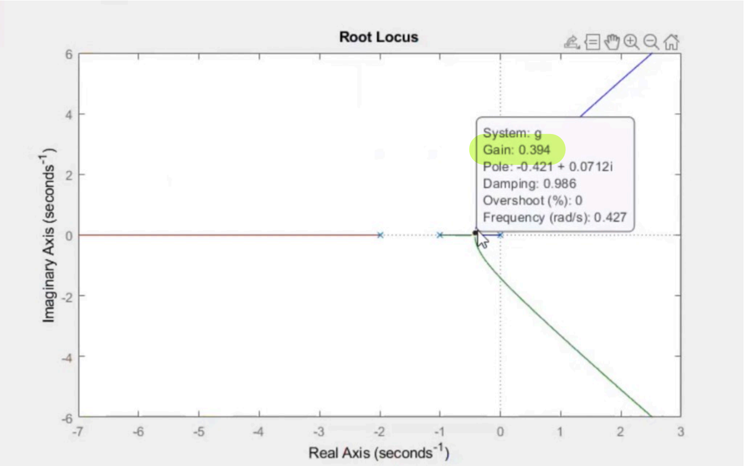
P.M.: 77.9 deg
Freq: 0.142 rad/s

Drag this closed-loop pole along the locus to adjust the loop gain. Current Location: -2.1

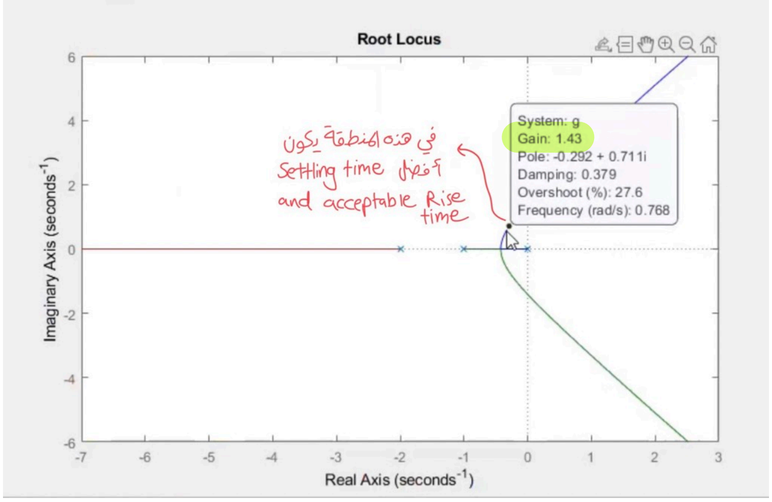


K

System: g
Gain: 0.194
Pole: -0.117
Damping: 1
Overshoot (%): 0
Frequency (rad/s): 0.117

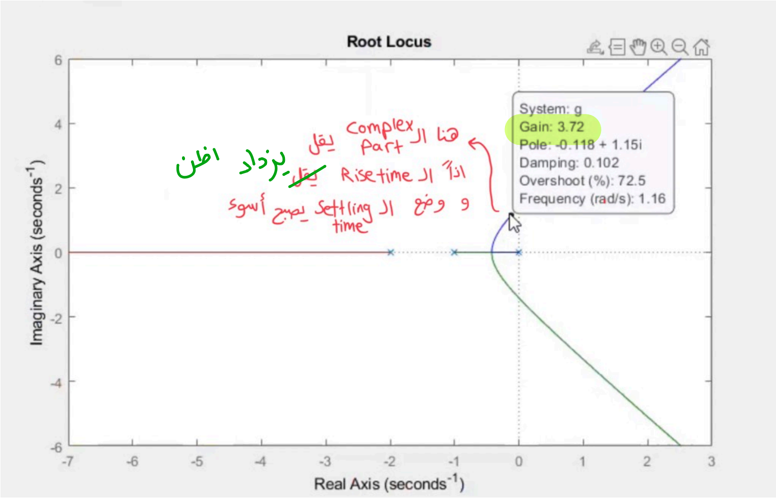


System: g
Gain: 0.394
Pole: -0.421 + 0.0712i
Damping: 0.986
Overshoot (%): 0
Frequency (rad/s): 0.427



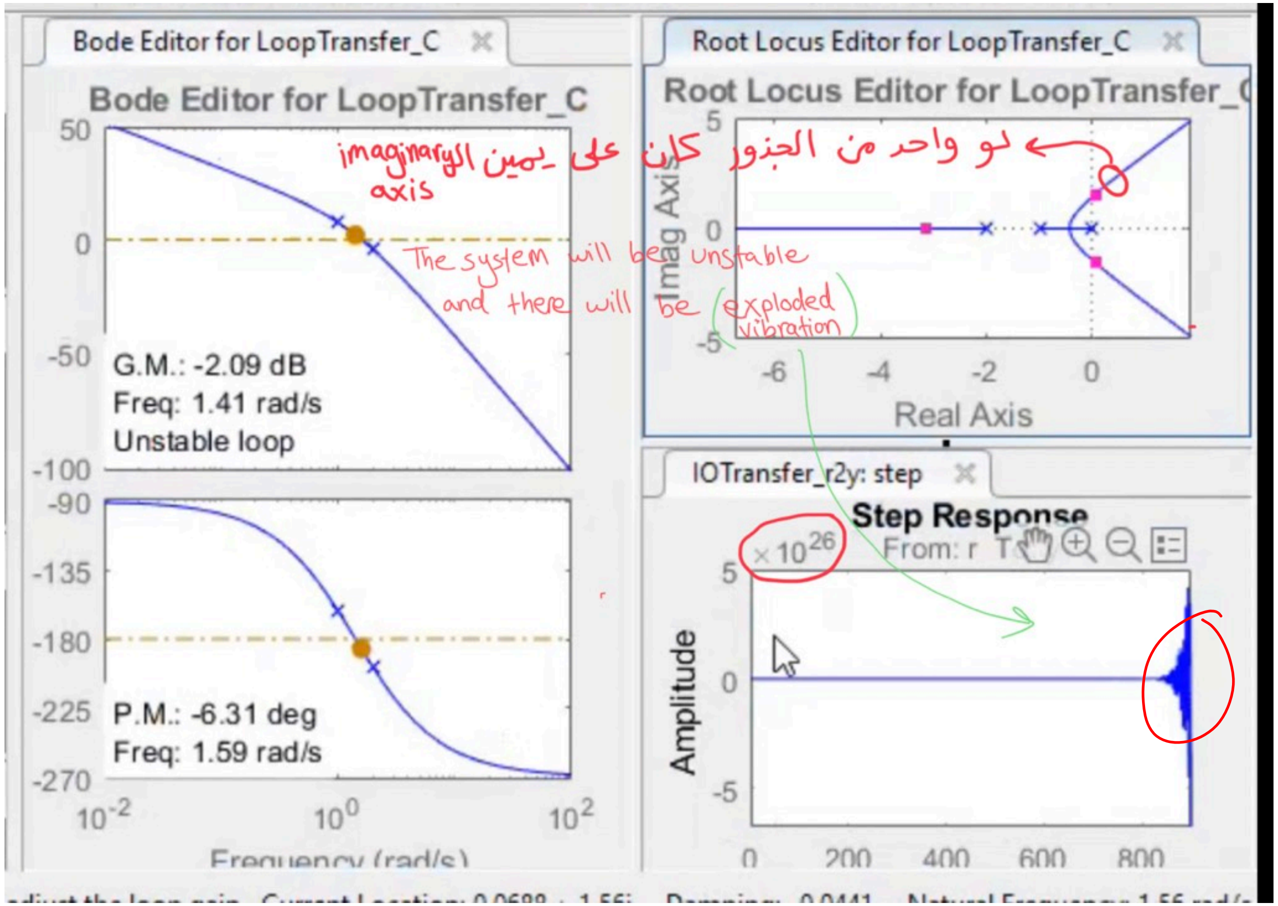
في هذه المنطقة يكون
Settling time أفضل
and acceptable Rise time

System: g
Gain: 1.43
Pole: -0.292 + 0.711i
Damping: 0.379
Overshoot (%): 27.6
Frequency (rad/s): 0.768



هنا ال complex part يقل
اذاً ال Rise time ~~يقل~~ يزداد اظن
و وضع ال Settling time يصبح أسوء

System: g
Gain: 3.72
Pole: -0.118 + 1.15i
Damping: 0.102
Overshoot (%): 72.5
Frequency (rad/s): 1.16

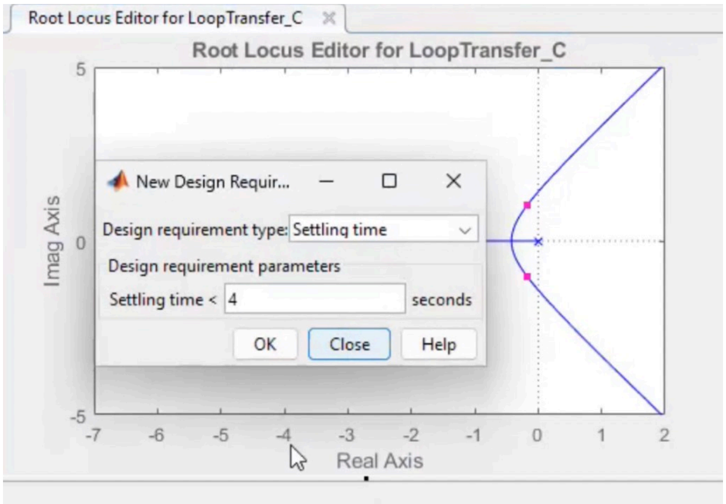


G.M.: -2.09 dB
Freq: 1.41 rad/s
Unstable loop

P.M.: -6.31 deg
Freq: 1.59 rad/s

لو واحد من الجذور كان على يمين ال imaginary axis

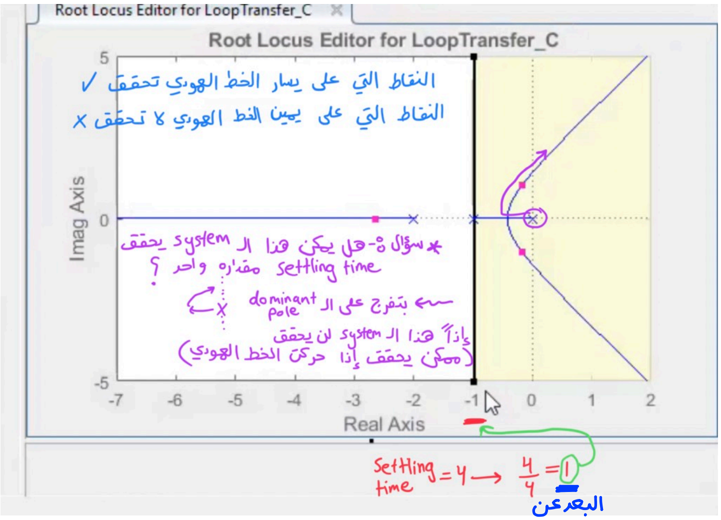
The system will be unstable and there will be (exploded vibration)



Design requirement type: Settling time

Design requirement parameters

Settling time < 4 seconds



النقاط التي على يسار الخط العمودي تحقق ✓

النقاط التي على يمين الخط العمودي لا تحقق X

*سؤال :- هل يمكن هذا الـ system يحقق settling time مقداره واحد ؟

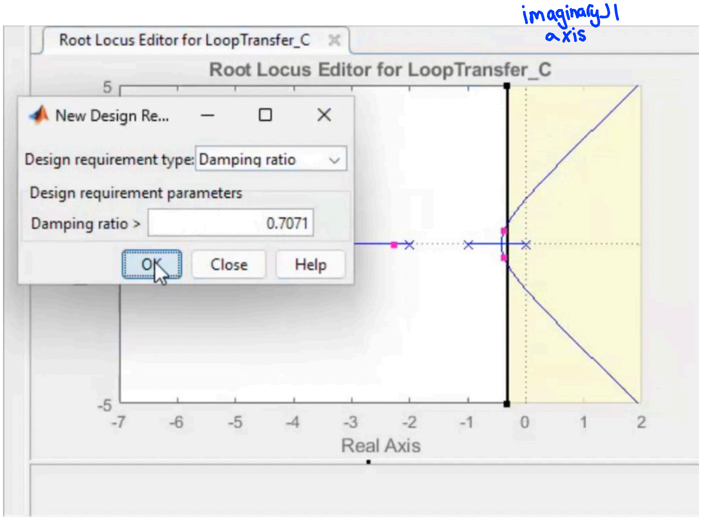
← بتفرج على الـ dominant pole

إذاً هذا الـ system لن يحقق

(ممكن يحقق إذا حركت الخط العمودي)

Settling time $= 4 \rightarrow \frac{4}{4} = \boxed{1}$

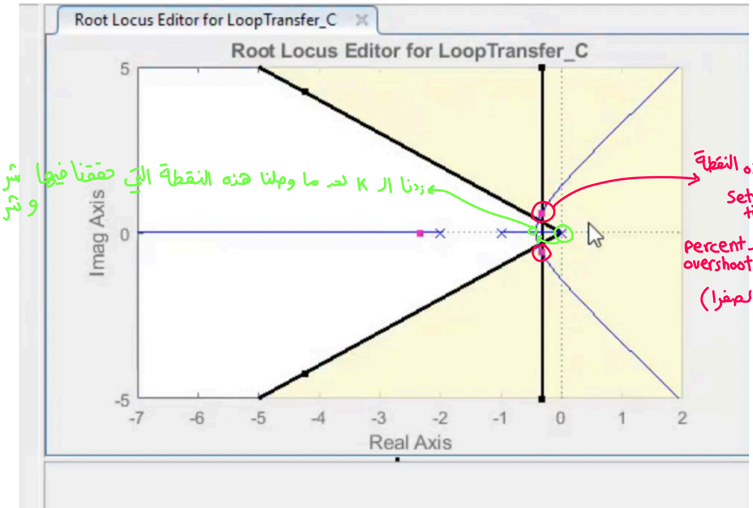
البعد عن الـ imaginary axis



Design requirement type: Damping ratio

Design requirement parameters

Damping ratio > 0.7071



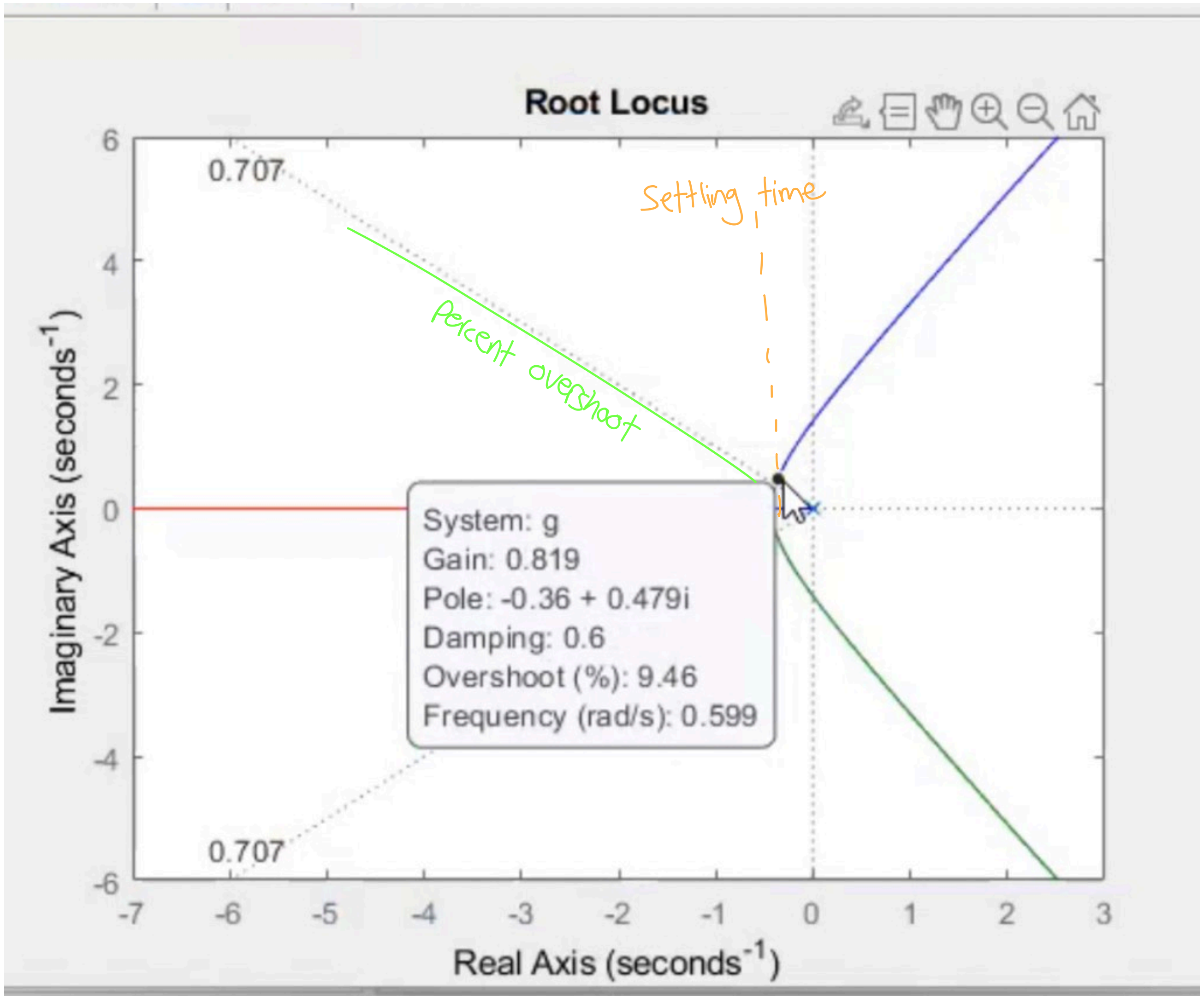
زدنا الـ K لحد ما وصلنا هذه النقطة التي حققنا فيها شرط الـ settling time و شرط الـ percent overshoot

لو زدنا الـ gain إلى هذه النقطة

بصير محقق شرط الـ settling time

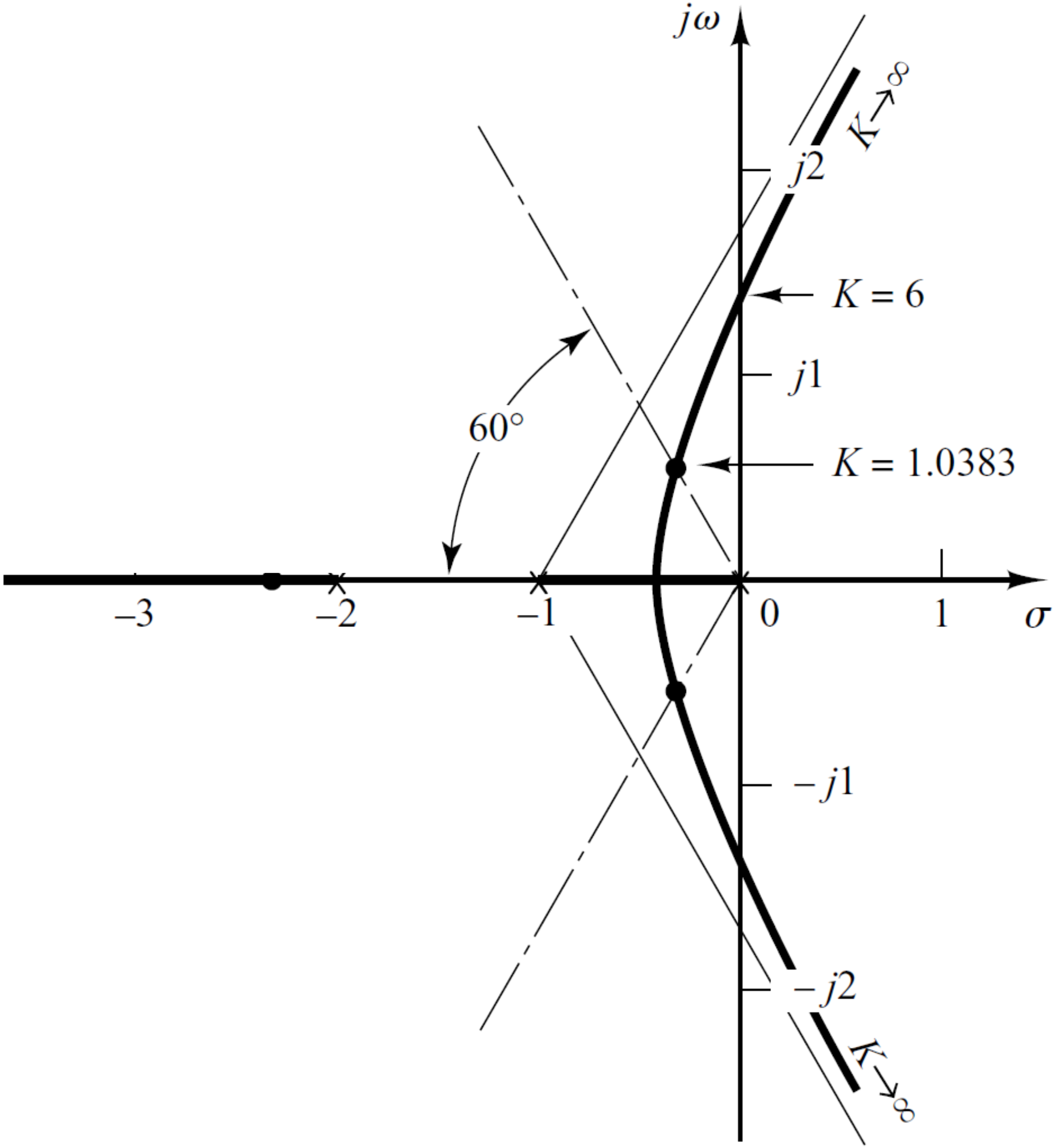
لكن مو محقق شرط الـ percent overshoot

(لأنه جاي في المنطقة الصفرا)



System: g
Gain: 0.819
Pole: -0.36 + 0.479i
Damping: 0.6
Overshoot (%): 9.46
Frequency (rad/s): 0.599

يختار قيمة (K) بين (K_1) و (K_2)


$j\omega$
$K \to \infty$
$j2$
$K = 6$
$j1$
$60°$
$K = 1.0383$
-3
-2
-1
0
1
σ
$-j1$
$-j2$
$K \to \infty$

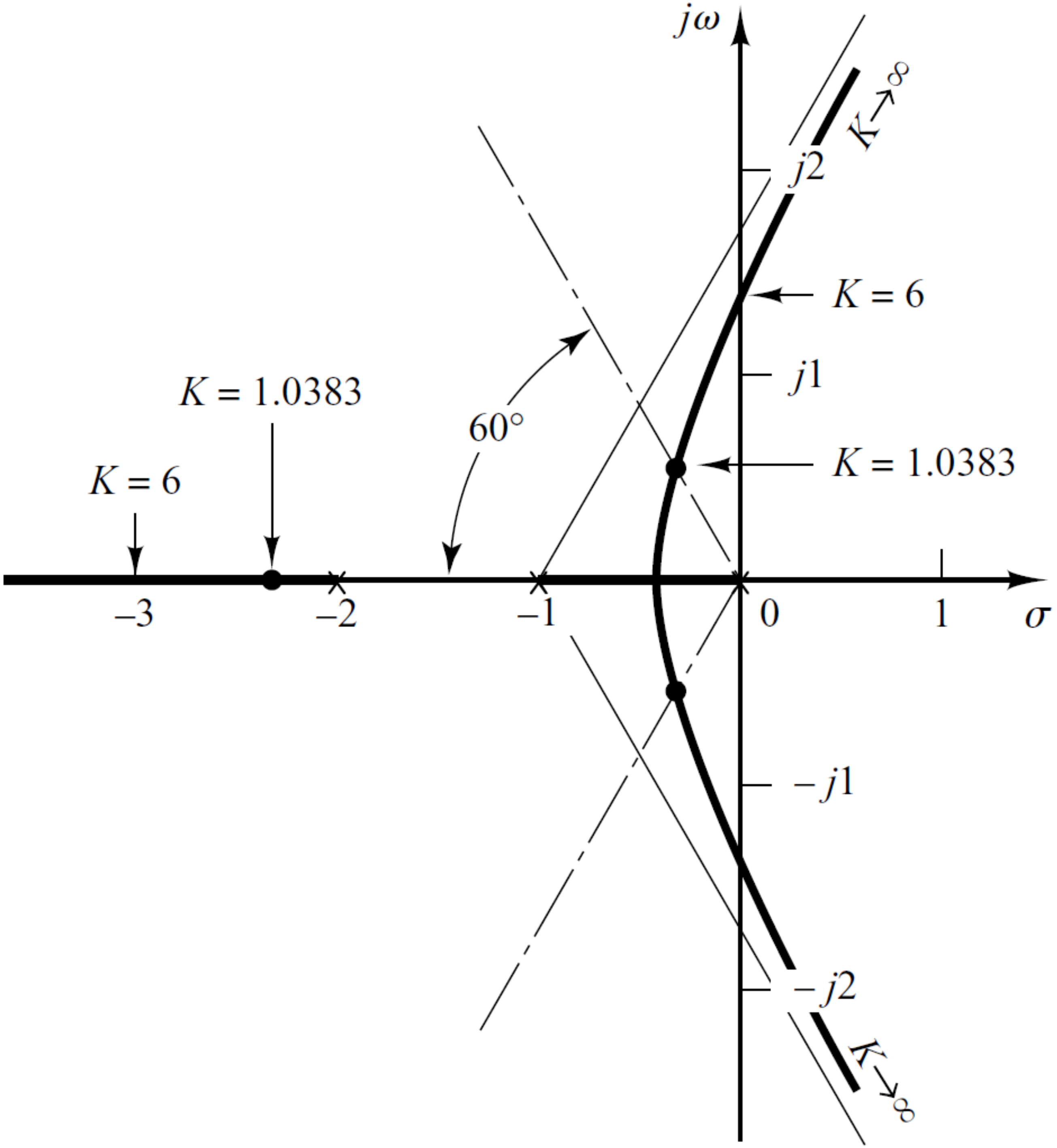
Example#1

- The third closed loop pole at K=1.0383 can be obtained as

$$1+G(s)H(s)=1+\frac{K}{s(s+1)(s+2)}=0$$

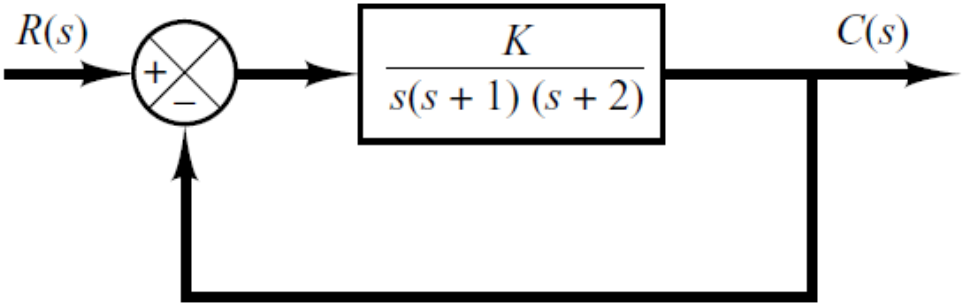
$$1+\frac{1.0383}{s(s+1)(s+2)}=0$$

$$s(s+1)(s+2)+1.0383=0$$


jω
K→∞
j2
K = 6
j1
K = 1.0383
60°
K = 1.0383
K = 6
–3
–2
–1
0
1
σ
–j1
–j2
K→∞

Home Work

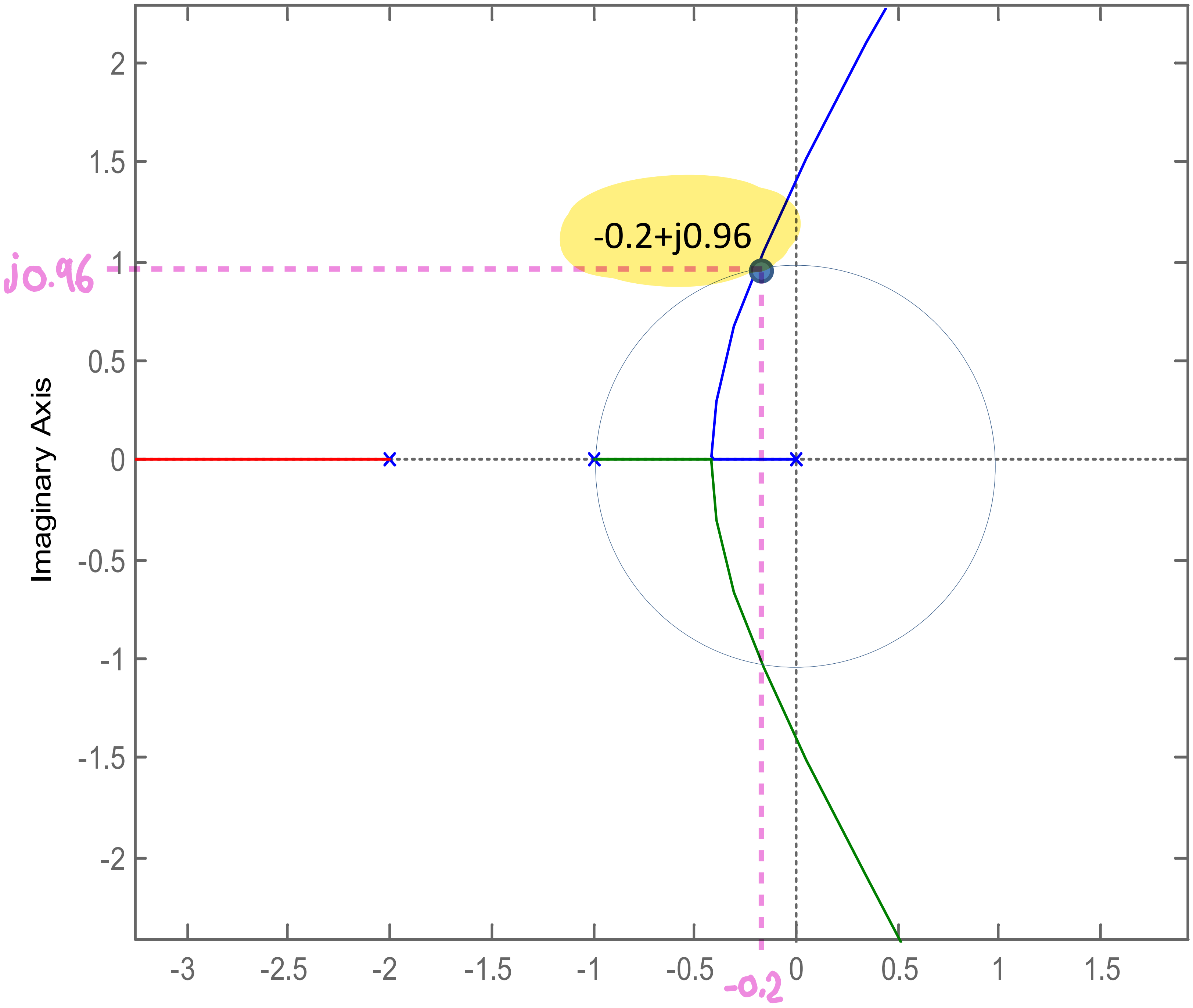
- Consider following unity feedback system.



- Determine the value of K such that the natural undamped frequency of dominant complex-conjugate closed-loop poles is 1 rad/sec.

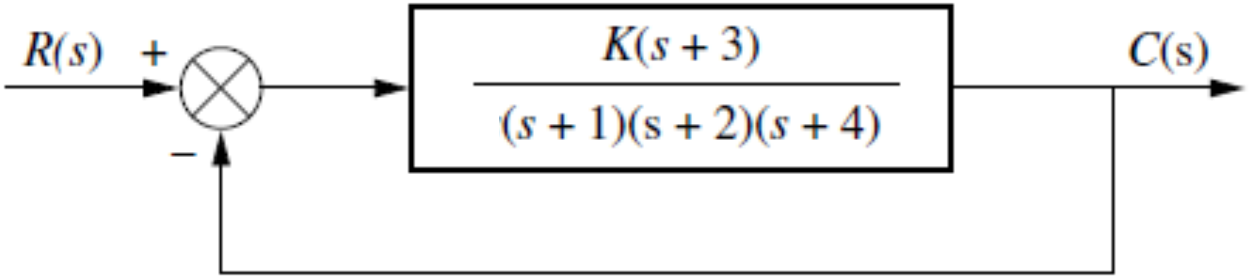
$\omega_n = 1$ rad/sec

$$G(s)H(s) = \frac{K}{s(s+1)(s+2)}$$


-0.2+j0.96
j0.96
-0.2
Imaginary Axis
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5

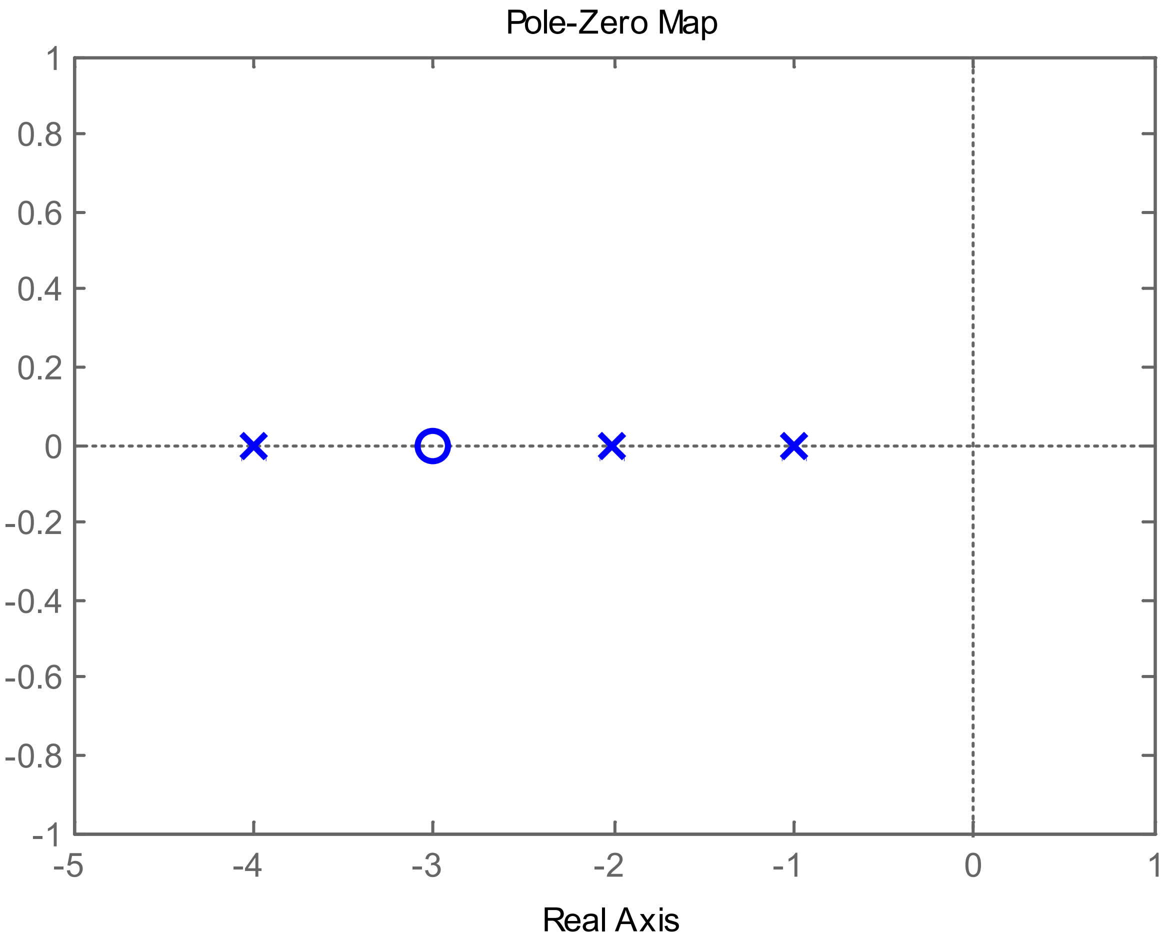
Example#2

- Sketch the root locus of following system and determine the location of dominant closed loop poles to yield maximum overshoot in the step response less than 30%.



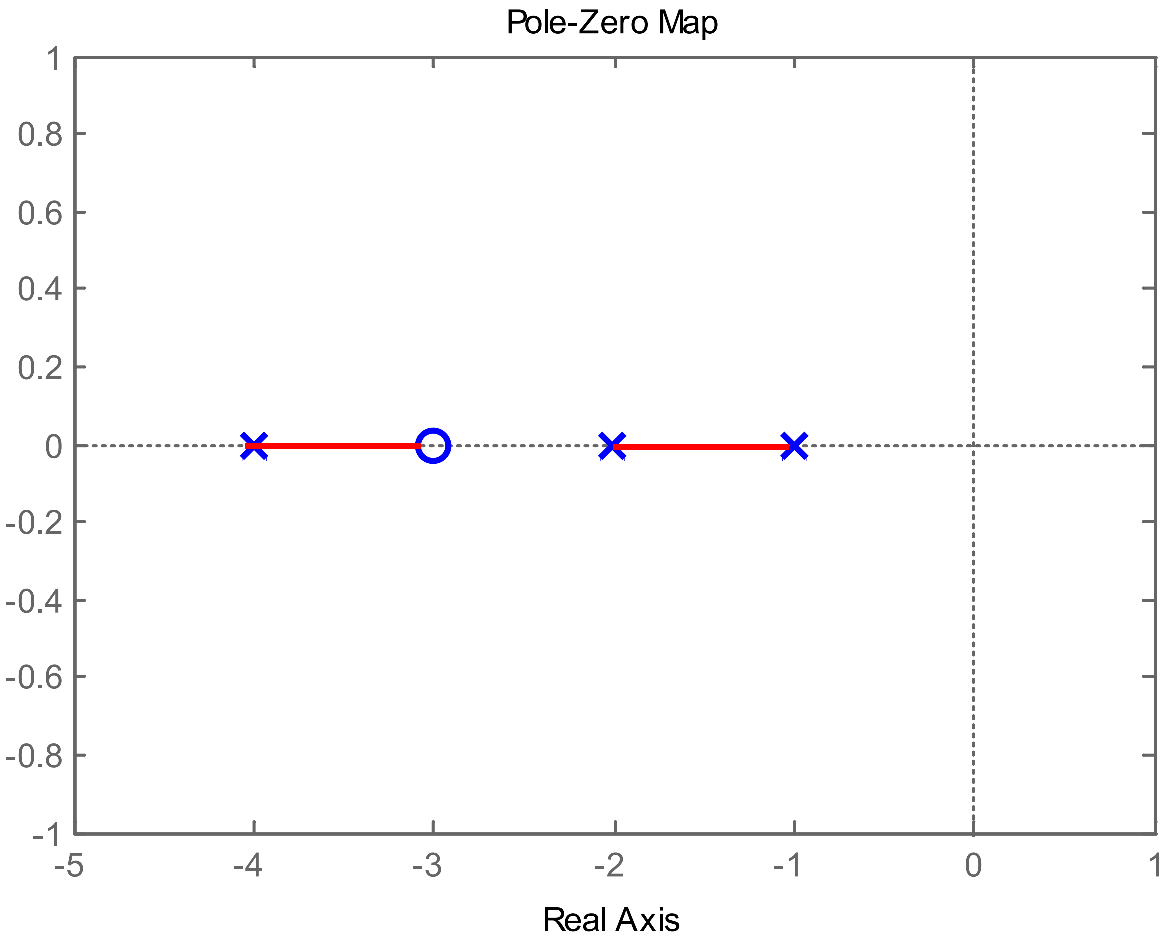
Example#2

- Step-1: Pole-Zero Map



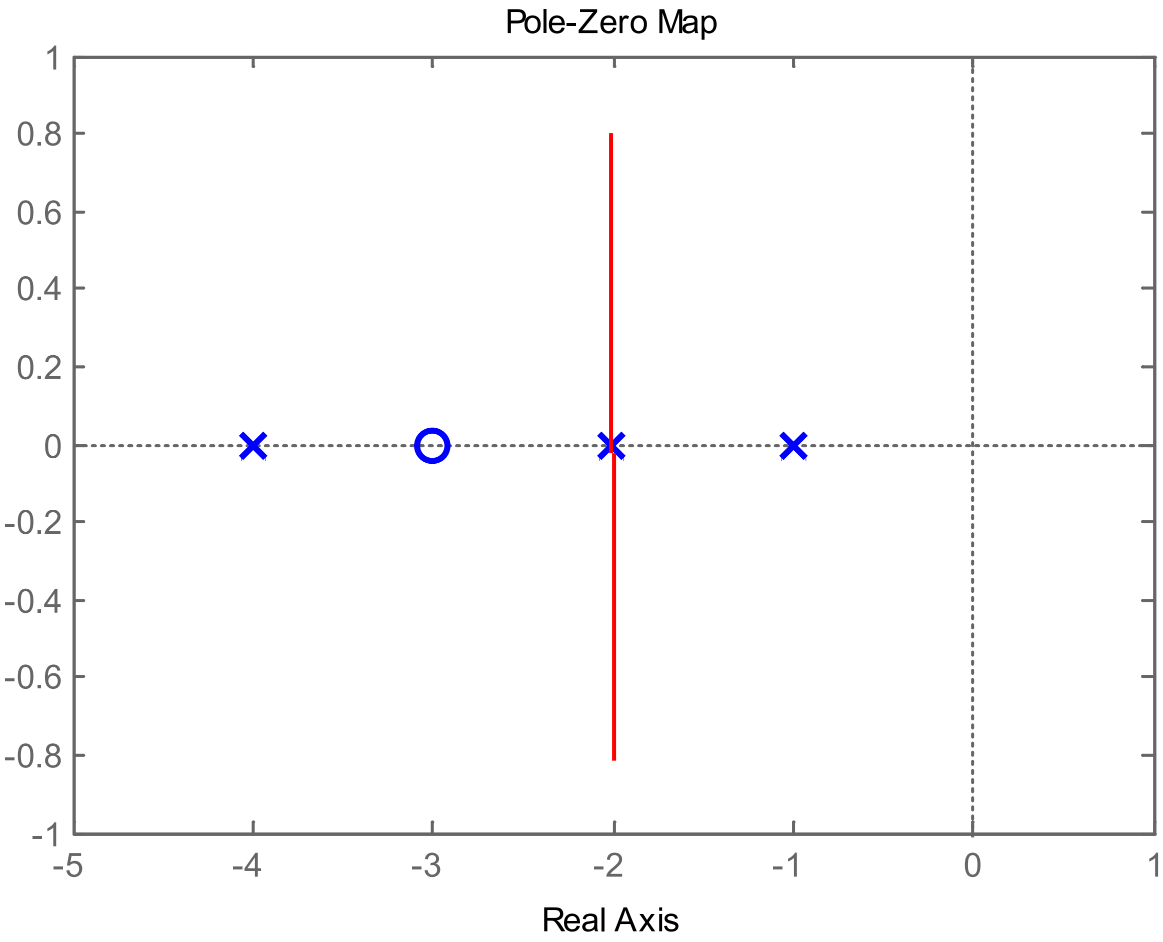
Example#2

- Step-2: Root Loci on Real axis



Example#2

- Step-3: Asymptotes

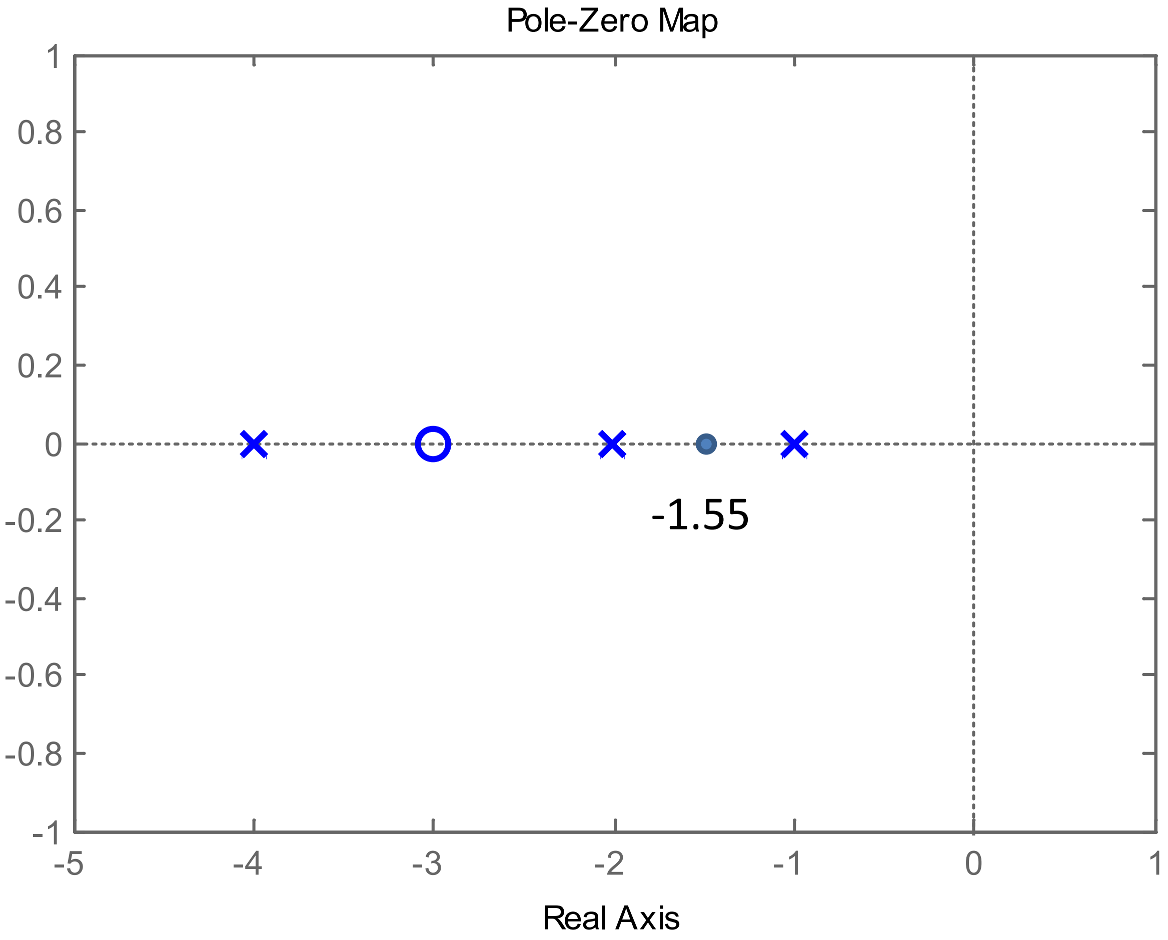


$$\psi = \pm 90°$$

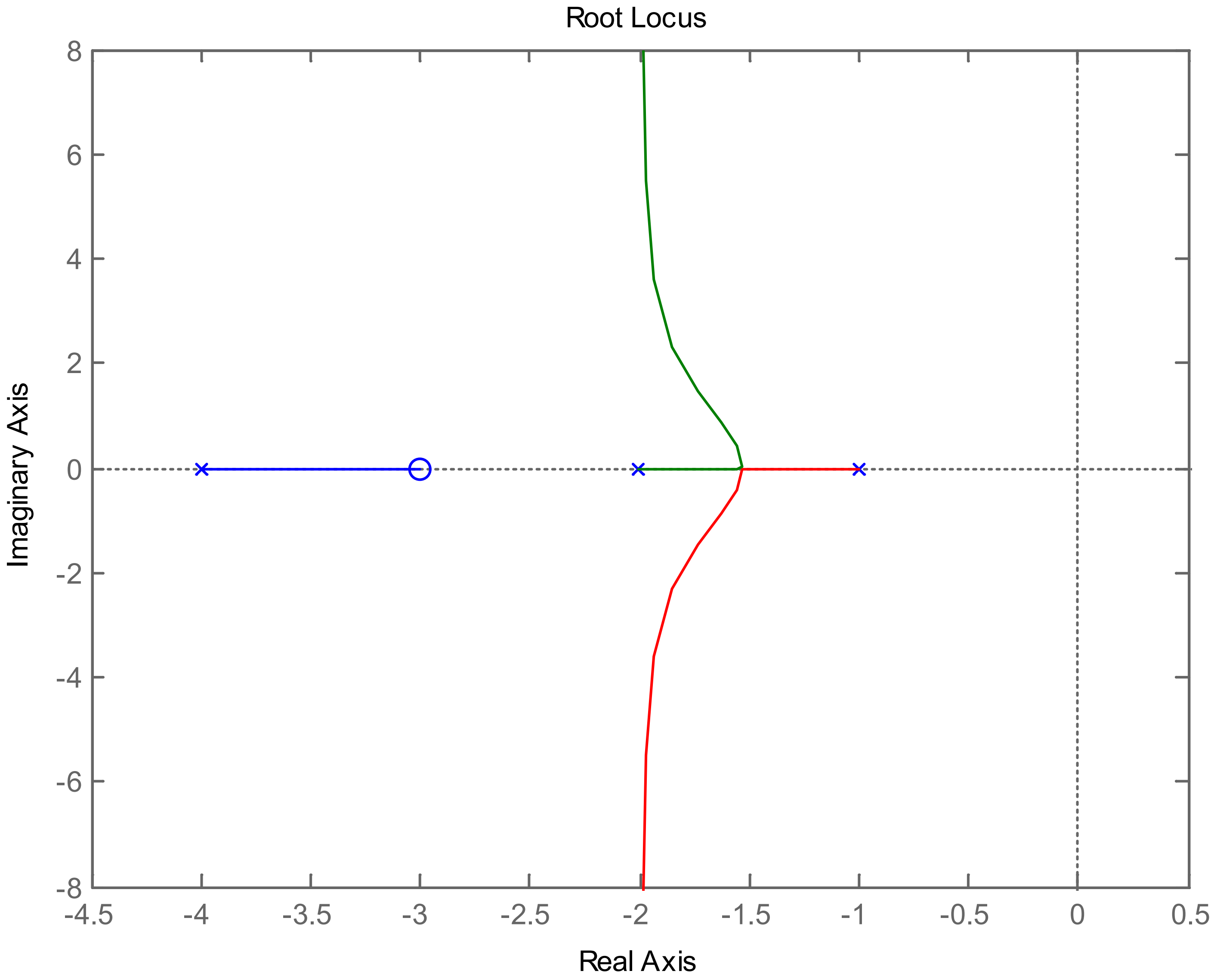
$$\sigma = -2$$

Example#2

- Step-4: breakaway point



Example#2


Root Locus
Imaginary Axis
Real Axis
8
6
4
2
0
-2
-4
-6
-8
-4.5
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5

Example#2

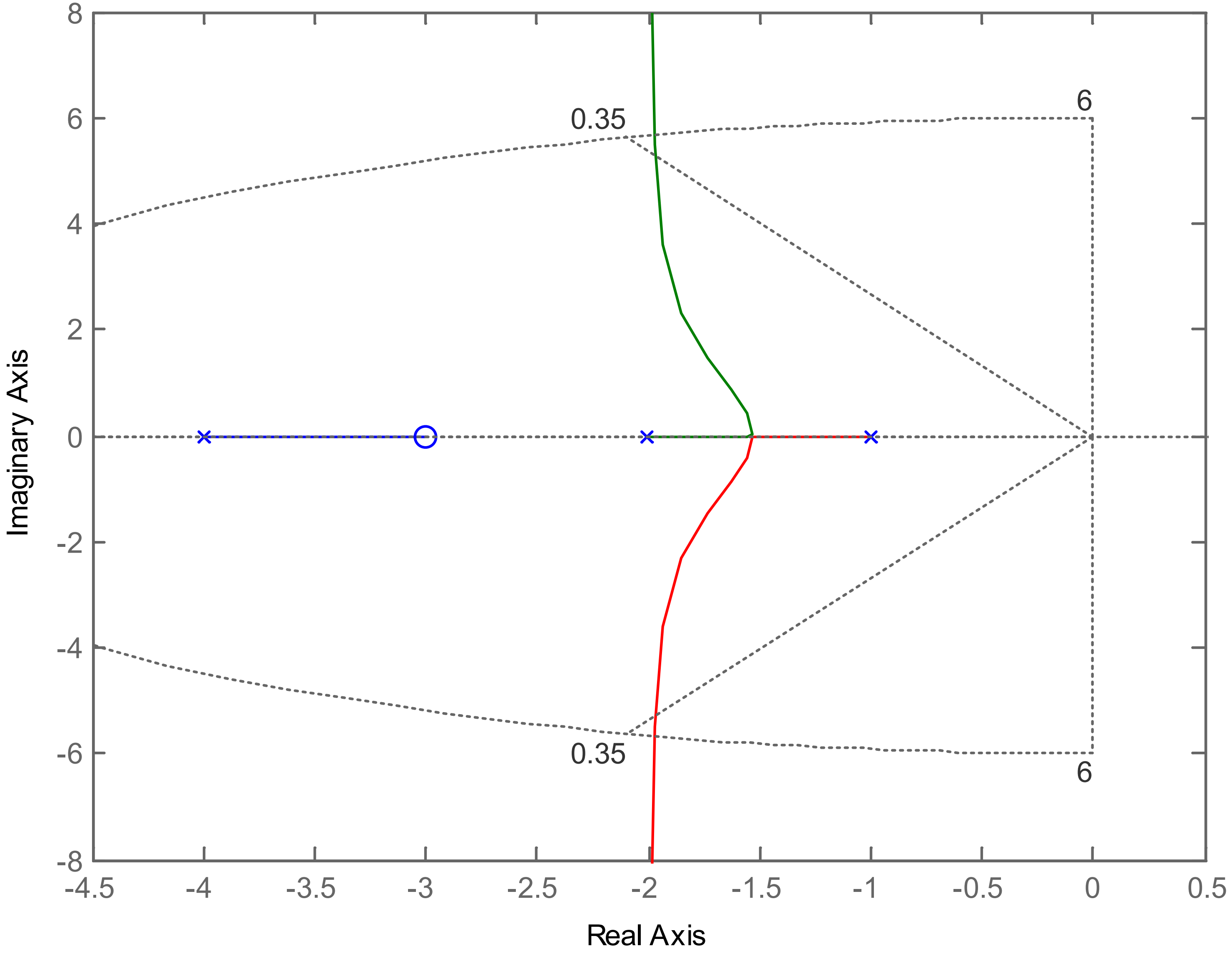
- Mp<30% corresponds to

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100$$

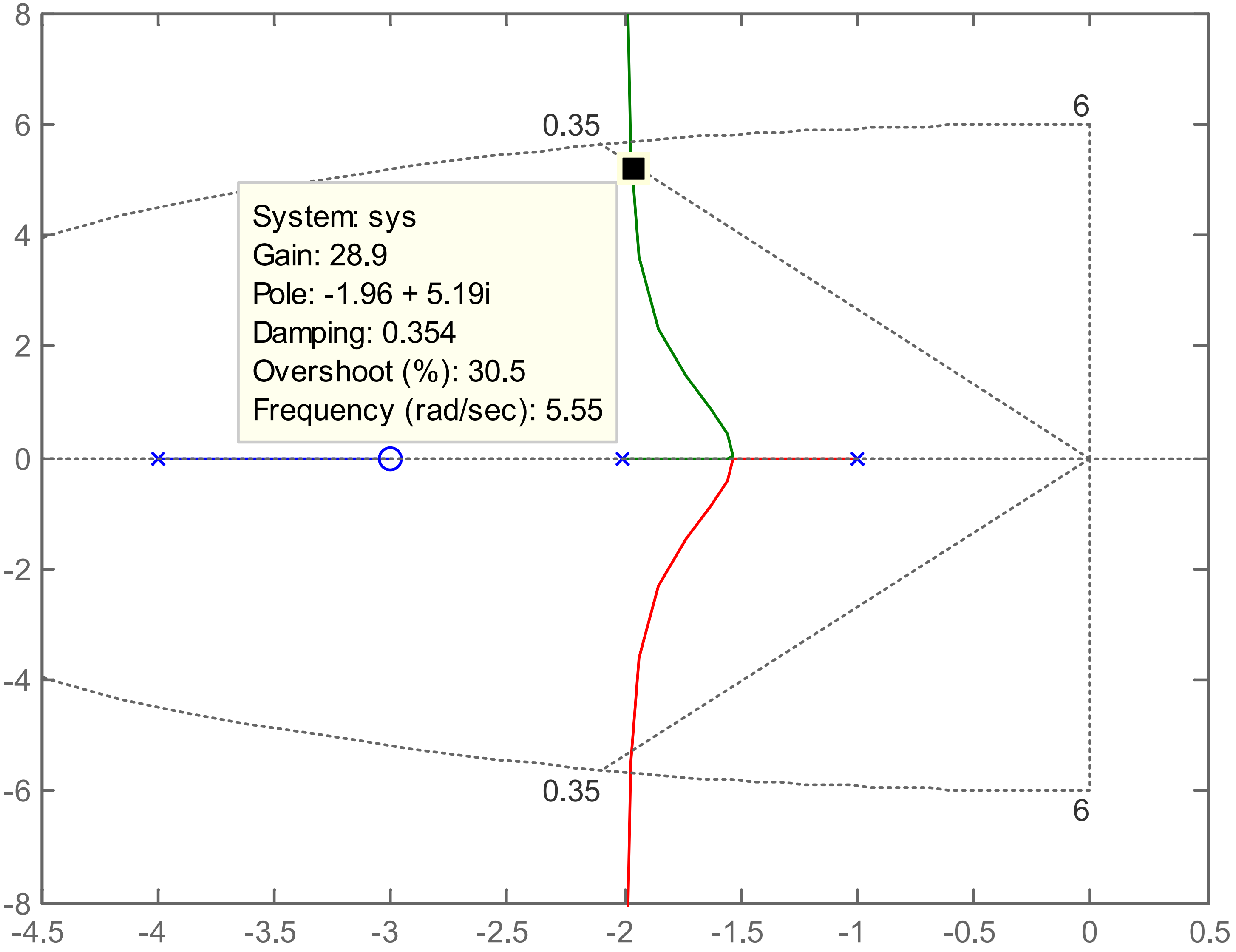
$$30\% = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100$$

$$\zeta > 0.35$$

Example#2


0.35
6
0.35
6
Imaginary Axis
Real Axis
8
6
4
2
0
-2
-4
-6
-8
-4.5
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5

Example#2



System: sys
Gain: 28.9
Pole: -1.96 + 5.19i
Damping: 0.354
Overshoot (%): 30.5
Frequency (rad/sec): 5.55

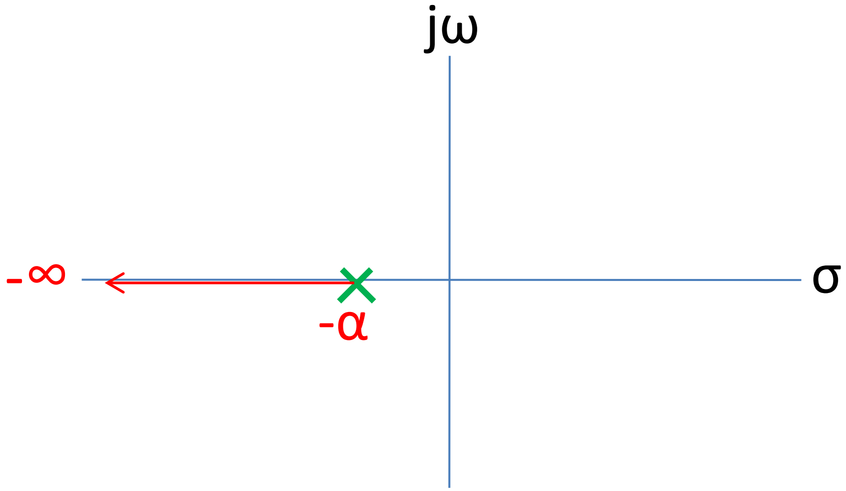
Root Locus of 1^{st} Order System

اظن من هون لنهاية ال Root locus مو داخل

- 1^{st} order systems (without zero) are represented by following transfer function.

$$G(s)H(s) = \frac{K}{s+\alpha}$$

- Root locus of such systems is a horizontal line starting from -α and moves towards -∞ as K reaches infinity.



Home Work

- Draw the Root Locus of the following systems.

1) $$G(s)H(s) = \frac{K}{s+2}$$

2) $$G(s)H(s) = \frac{K}{s-1}$$

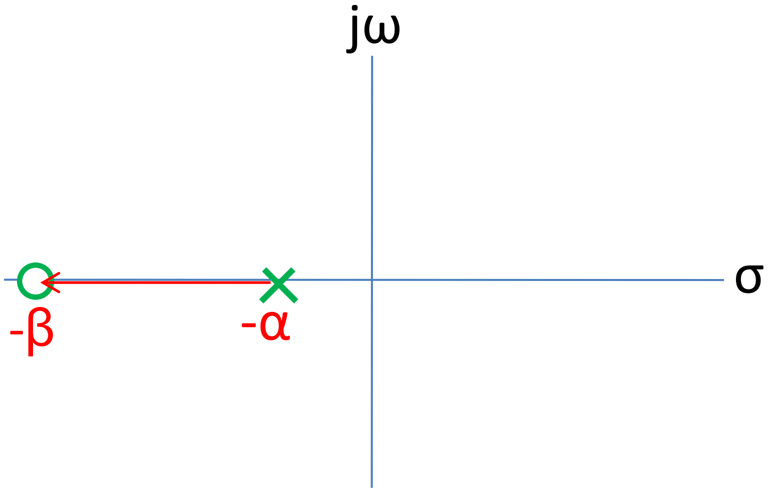
3) $$G(s)H(s) = \frac{K}{s}$$

Root Locus of 1^{st} Order System

- 1^{st} order systems with zero are represented by following transfer function.

$$G(s)H(s) = \frac{K(s+\beta)}{s+\alpha}$$

- Root locus of such systems is a horizontal line starting from -α and moves towards -β as K reaches infinity.



Home Work

- Draw the Root Locus of the following systems.

1) $$G(s)H(s) = \frac{Ks}{s+2}$$

2) $$G(s)H(s) = \frac{K(s+5)}{s-1}$$

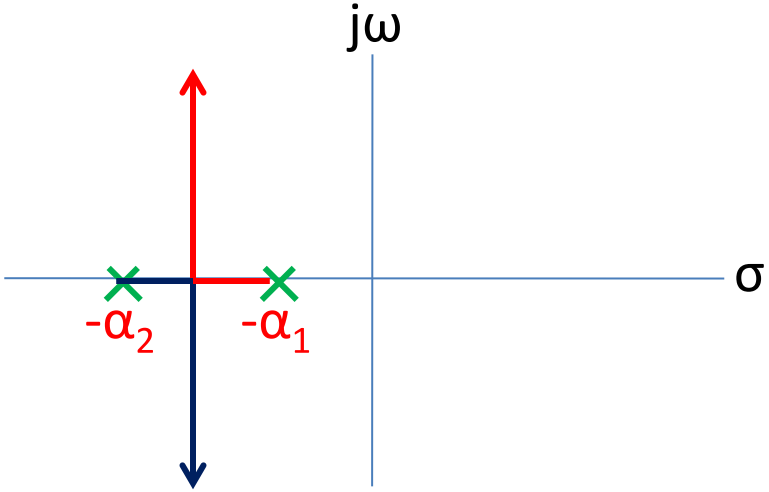
3) $$G(s)H(s) = \frac{K(s+3)}{s}$$

Root Locus of 2nd Order System

- Second order systems (without zeros) have two poles and the transfer function is given

$$G(s)H(s) = \frac{K}{(s+\alpha_1)(s+\alpha_2)}$$

- Root loci of such systems are vertical lines.



Home Work

- Draw the Root Locus of the following systems.

1) $$G(s)H(s) = \frac{K}{s(s+2)}$$

2) $$G(s)H(s) = \frac{K}{s^2}$$

3) $$G(s)H(s) = \frac{K}{(s+1)(s-3)}$$

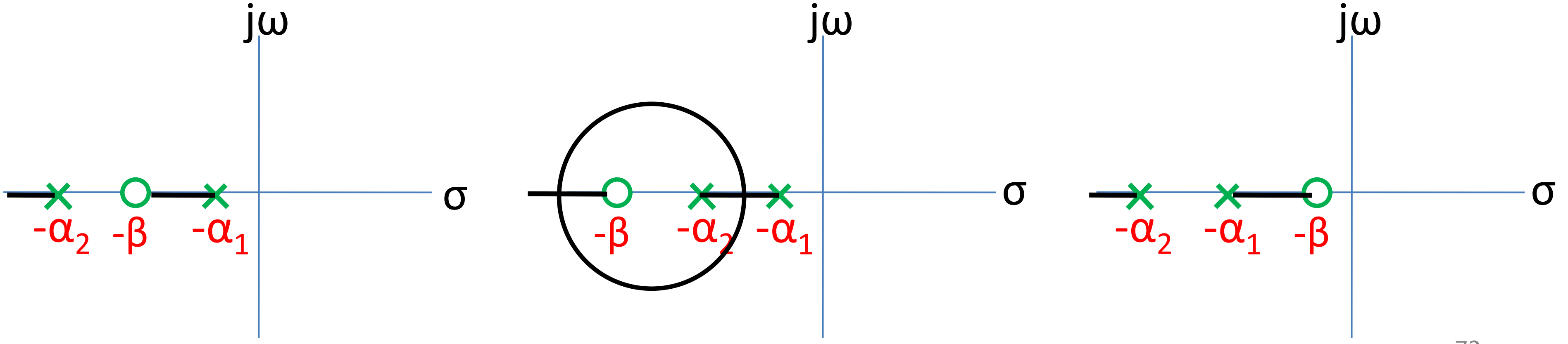
4) $$G(s)H(s) = \frac{K}{s^2 + 3s + 10}$$

Root Locus of 2nd Order System

- Second order systems (with one zero) have two poles and the transfer function is given

$$G(s)H(s) = \frac{K(s+\beta)}{(s+\alpha_1)(s+\alpha_2)}$$

- Root loci of such systems are either horizontal lines or circular depending upon pole-zero configuration.



Home Work

- Draw the Root Locus of the following systems.

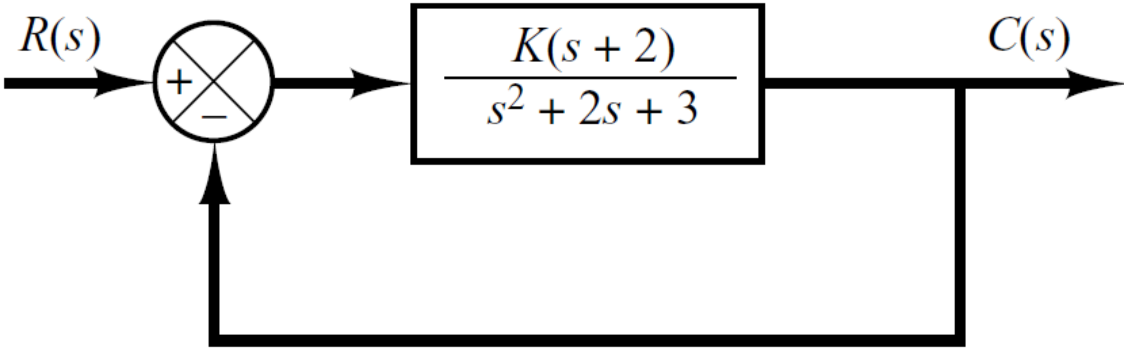
1) $$G(s)H(s) = \frac{K(s+1)}{s(s+2)}$$

2) $$G(s)H(s) = \frac{K(s-2)}{s^2}$$

3) $$G(s)H(s) = \frac{K(s+5)}{(s+1)(s-3)}$$

Example

- Sketch the root-locus plot of following system with complex-conjugate open loop poles.



$$G(s) = \frac{K(s+2)}{s^2+2s+3}, \qquad H(s) = 1$$

$G(s)$ has a pair of complex-conjugate poles at

$$s = -1 + j\sqrt{2}, \qquad s = -1 - j\sqrt{2}$$

Example

- ***Step-1:*** Pole-Zero Mao

- ***Step-2:*** Determine the root loci on real axis

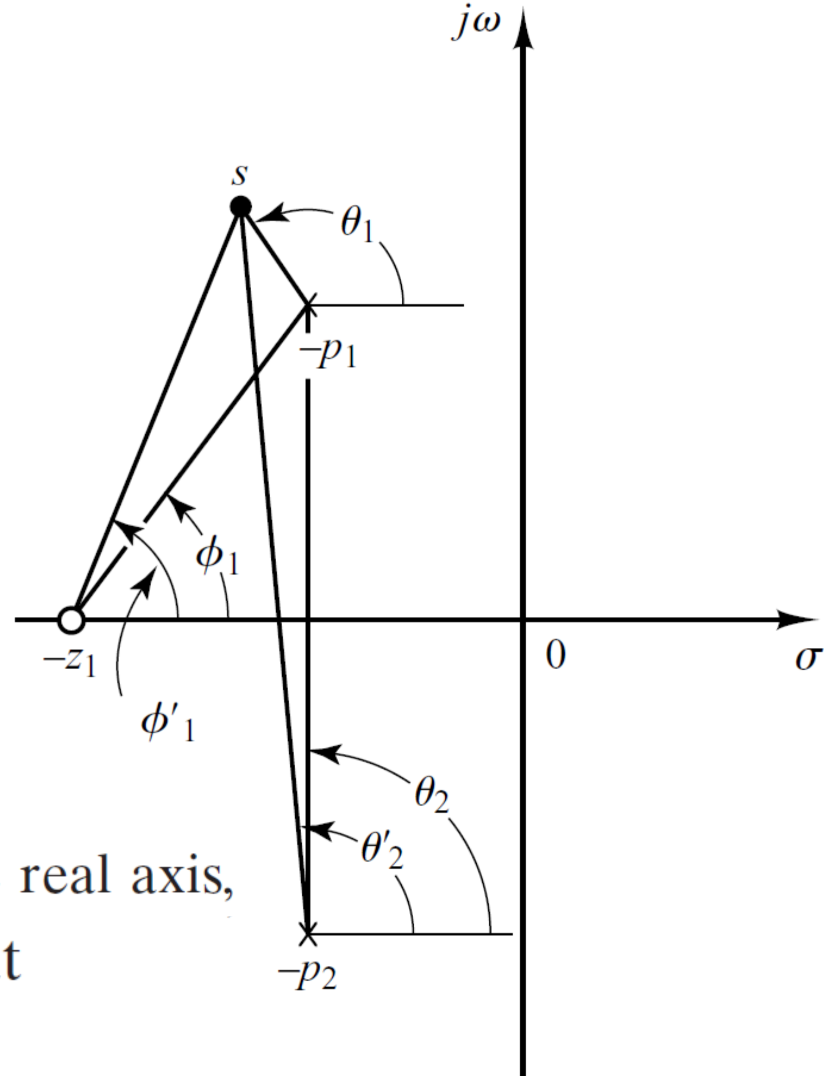
- ***Step-3***: Asymptotes

Example

- *Step-4:* Determine the angle of departure from the complex-conjugate open-loop poles.
  - The presence of a pair of complex-conjugate open-loop poles requires the determination of the angle of departure from these poles.
  - Knowledge of this angle is important, since the root locus near a complex pole yields information as to whether the locus originating from the complex pole migrates toward the real axis or extends toward the asymptote.

Example

- *Step-4:* Determine the angle of departure from the complex-conjugate open-loop poles.



The angle of departure is then

$$\theta_1 = 180^\circ - \theta_2 + \phi_1$$
$$= 180^\circ - 90^\circ + 55^\circ = 145^\circ$$

Since the root locus is symmetric about the real axis, the angle of departure from the pole at

$$s = -p_2 \text{ is } -145^\circ$$

Example

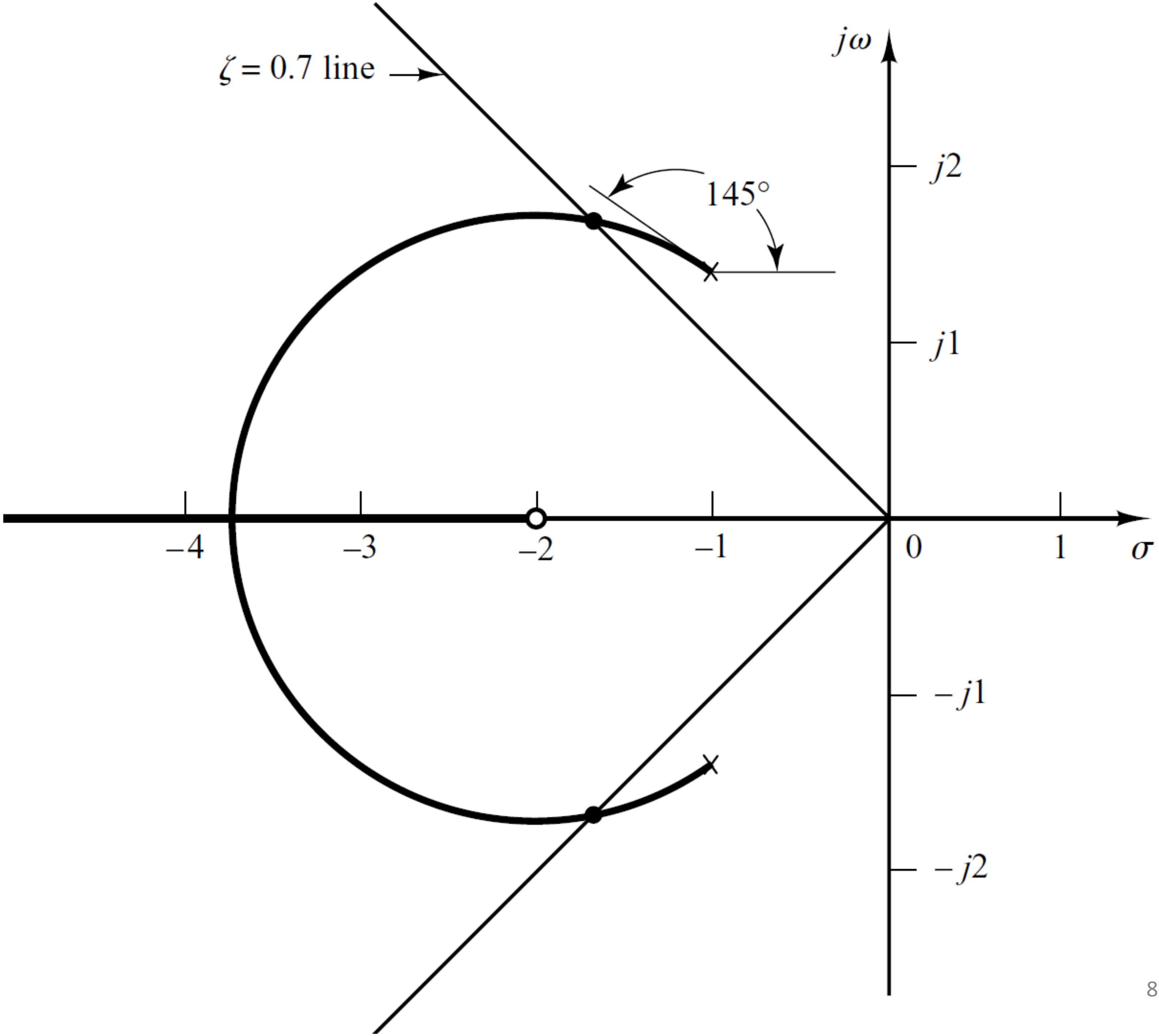
- *Step-5:* Break-in point

$$K = -\frac{s^2 + 2s + 3}{s + 2}$$

$$\frac{dK}{ds} = -\frac{(2s + 2)(s + 2) - (s^2 + 2s + 3)}{(s + 2)^2} = 0$$

$$s^2 + 4s + 1 = 0$$

$$s = -3.7320 \qquad \text{or} \qquad s = -0.2680$$



ζ = 0.7 line
jω
j2
145°
j1
−4
−3
−2
−1
0
1
σ
−j1
−j2

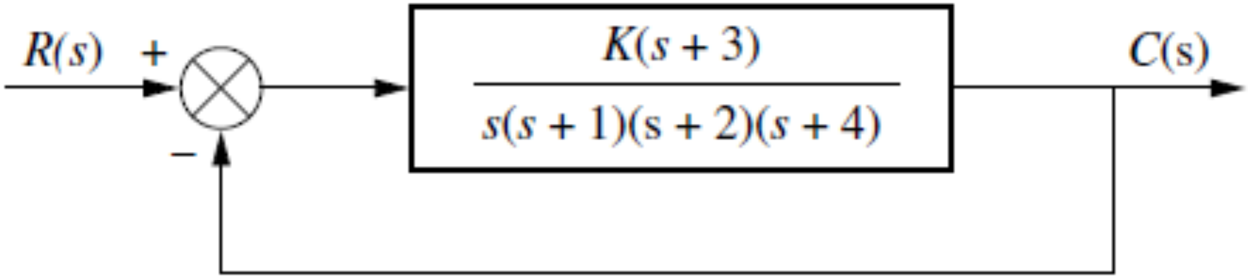
Root Locus of Higher Order System

- Third order System without zero

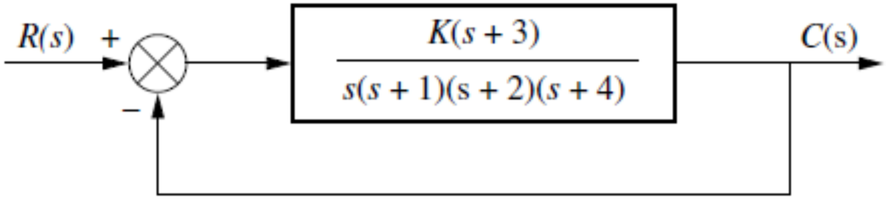
$$G(s)H(s) = \frac{K}{(s+\alpha_1)(s+\alpha_2)(s+\alpha_3)}$$

Root Locus of Higher Order System

- Sketch the Root Loci of following unity feedback system



$$G(s)H(s)=\frac{K(s+3)}{s(s+1)(s+2)(s+4)}$$



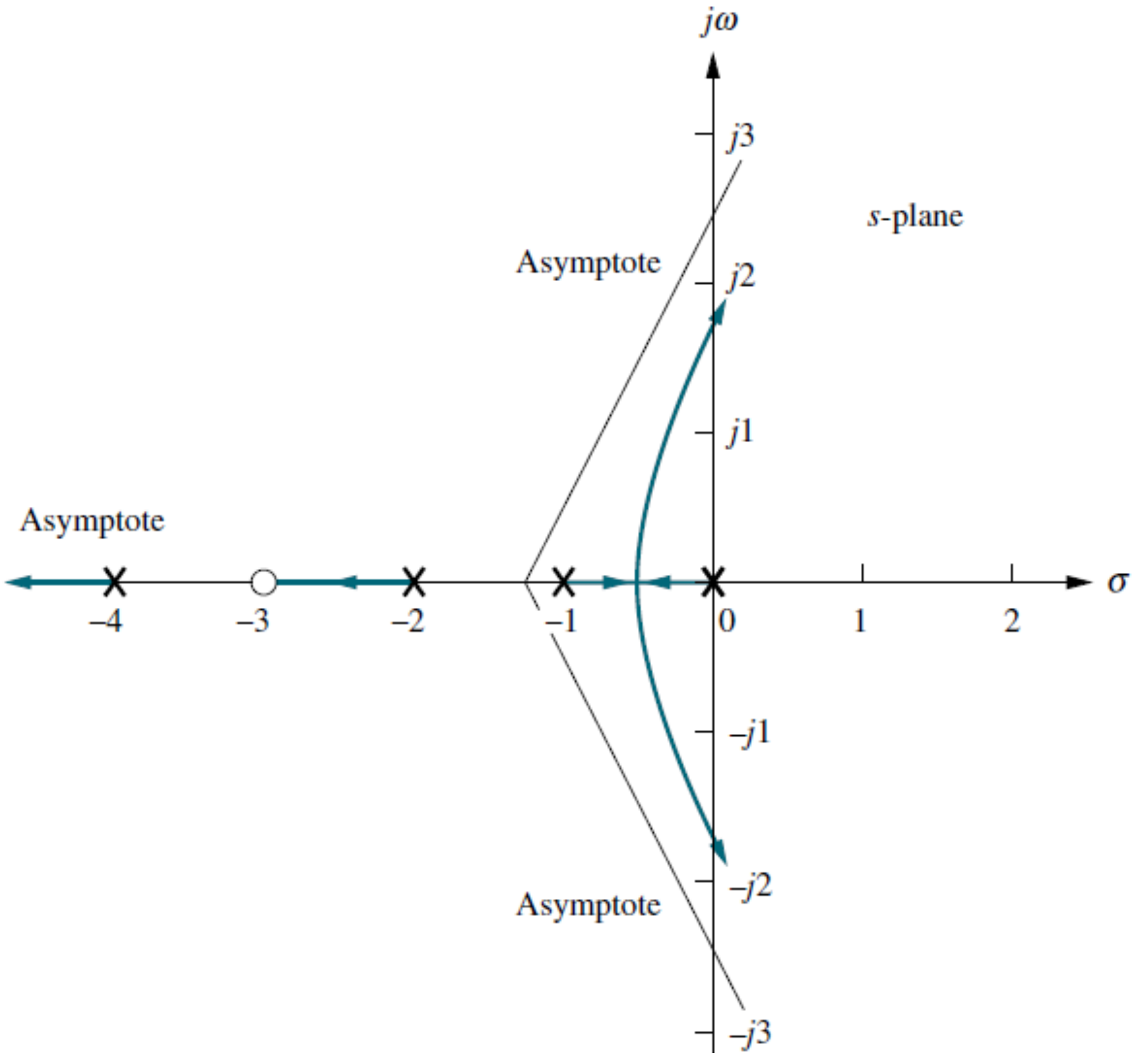
- Let us begin by calculating the asymptotes. The real-axis intercept is evaluated as;

$$\sigma_a = \frac{(-1-2-4)-(-3)}{4-1} = -\frac{4}{3}$$

- The angles of the lines that intersect at - 4/3, given by

$$\begin{aligned}\theta_a &= \frac{(2k+1)\pi}{\#\text{finite poles} - \#\text{finite zeros}} \\ &= \pi/3 \qquad \text{for } k = 0 \\ &= \pi \qquad \text{for } k = 1 \\ &= 5\pi/3 \qquad \text{for } k = 2\end{aligned}$$

- The Figure shows the complete root locus as well as the asymptotes that were just calculated.



Example: Sketch the root locus for the system with the characteristic equation of;

$$1 + GH(s) = 1 + \frac{K(s + 1)}{s(s + 2)(s + 4)^2}$$

a) Number of finite poles = n = 4.
b) Number of finite zeros = m = 1.
c) Number of asymptotes = n - m = 3.
d) Number of branches or loci equals to the number of finite poles (n) = 4.
e) The portion of the real-axis between, 0 and -2, and between, -4 and -∞, lie on the root locus for *K* > 0.

- Using Eq. (v), the real-axis asymptotes intercept is evaluated as;

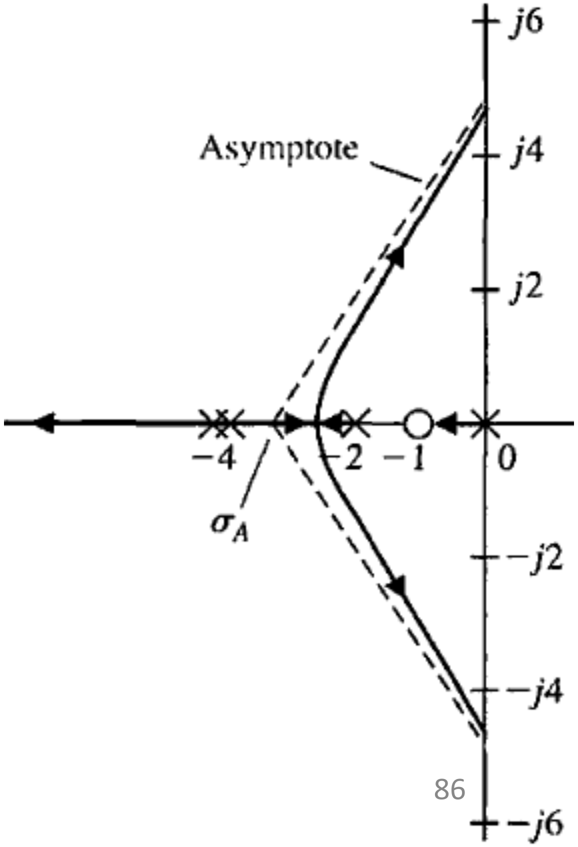
$$\sigma_a = \frac{(-2) + 2(-4) - (-1)}{n - m} = \frac{-10 + 1}{4 - 1} = -3$$

- The angles of the asymptotes that intersect at - 3, given by Eq. (vi), are;

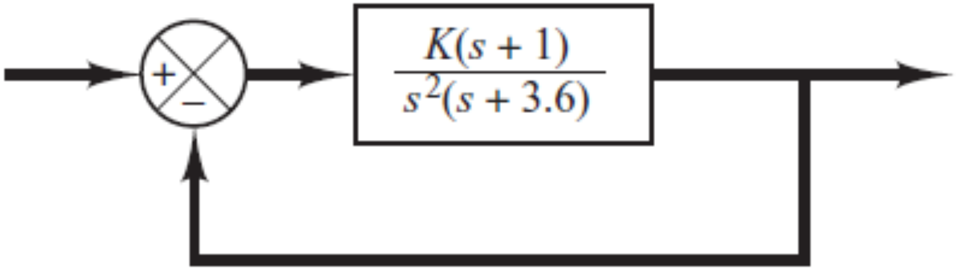
$$\theta_a = \frac{(2k + 1)\pi}{n - m} = \frac{(2k + 1)\pi}{4 - 1}$$

For $K = 0$, $\theta a = 60^o$
For $K = 1$, $\theta a = 180^o$
For $K = 2$, $\theta a = 300^o$

- The root-locus plot of the system is shown in the figure below.
- It is noted that there are three asymptotes. Since n – m = 3.
- The root loci must begin at the poles; two loci (or branches) must leave the double pole at s = -4.
- Using Eq. (vii), the breakaway point, σ, can be determine as;
- The solution of the above equation is $\sigma = -2.59$.



Example: Sketch the root loci for the system.



- A root locus exists on the real axis between points **$s = –1$** and **$s = –3.6$**.
- The intersection of the asymptotes and the real axis is determined as,

$$\sigma_a = \frac{0+0+3.6-1}{n\ -m} = \frac{2.6}{3\ -1} = -1.3$$

- The angles of the asymptotes that intersect at – 1.3, given by Eq. (vi), are;

$$\theta_a = \frac{(2k+1)\pi}{n\ -m} = \frac{(2k+1)\pi}{3-1}$$

For $K = 0$, $\theta a = 90^o$
For $K = 1$, $\theta a = -90^o$ or 270^o

- Since the characteristic equation is $s^3 + 3.6s^2 + K(s+1) = 0$
- We have $K = -\frac{s^3 + 3.6s^2}{s+1} \longrightarrow$ (a)

- The breakaway and break-in points are found from Eq. (a) as,

$$\frac{dK}{ds} = -\frac{(3s^2 + 7.2s)(s + 1) - (s^3 + 3.6s^2)}{(s + 1)^2} = 0$$

$$\text{or} \qquad s^3 + 3.3s^2 + 3.6s = 0$$

From which we get,

$$s = 0, \qquad s = -1.65 + j0.9367, \qquad s = -1.65 - j0.9367$$

- Point s = 0 corresponds to the actual breakaway point. But points $s = 1.65 \pm j0.9367$ neither breakaway nor break-in points, because the corresponding gain values *K* become complex quantities.

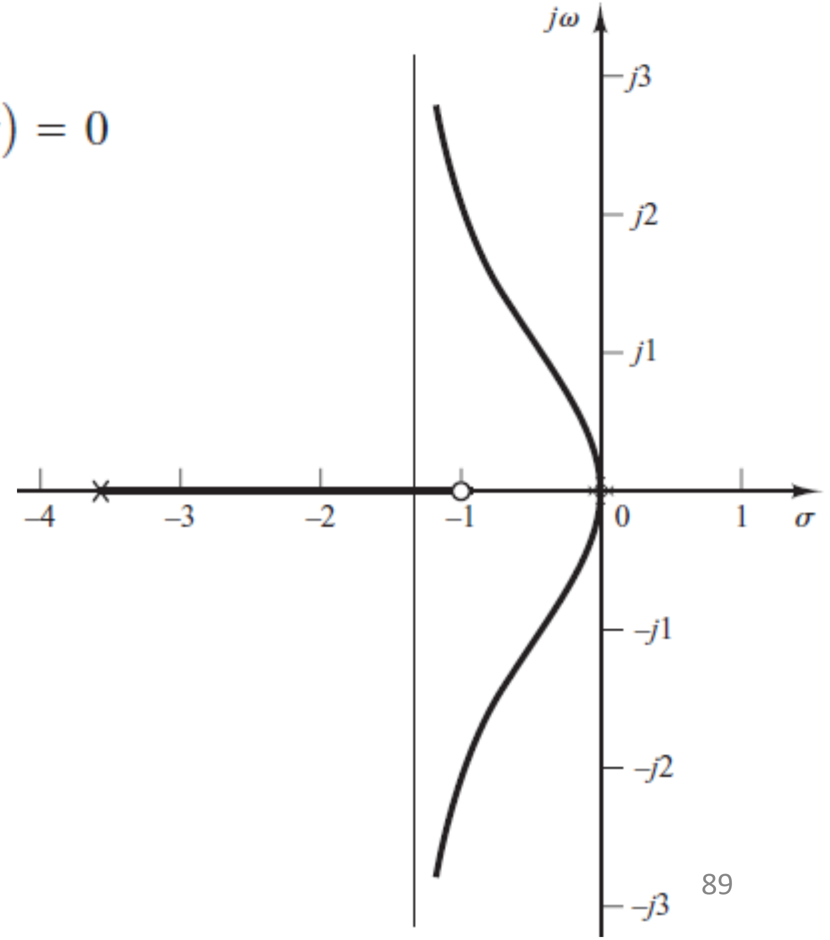
- To check the points where root-locus branches may cross the imaginary axis, substitute $s = j\omega$ into the characteristic equation, yielding.

$$(j\omega)^3 + 3.6(j\omega)^2 + Kj\omega + K = 0$$

or

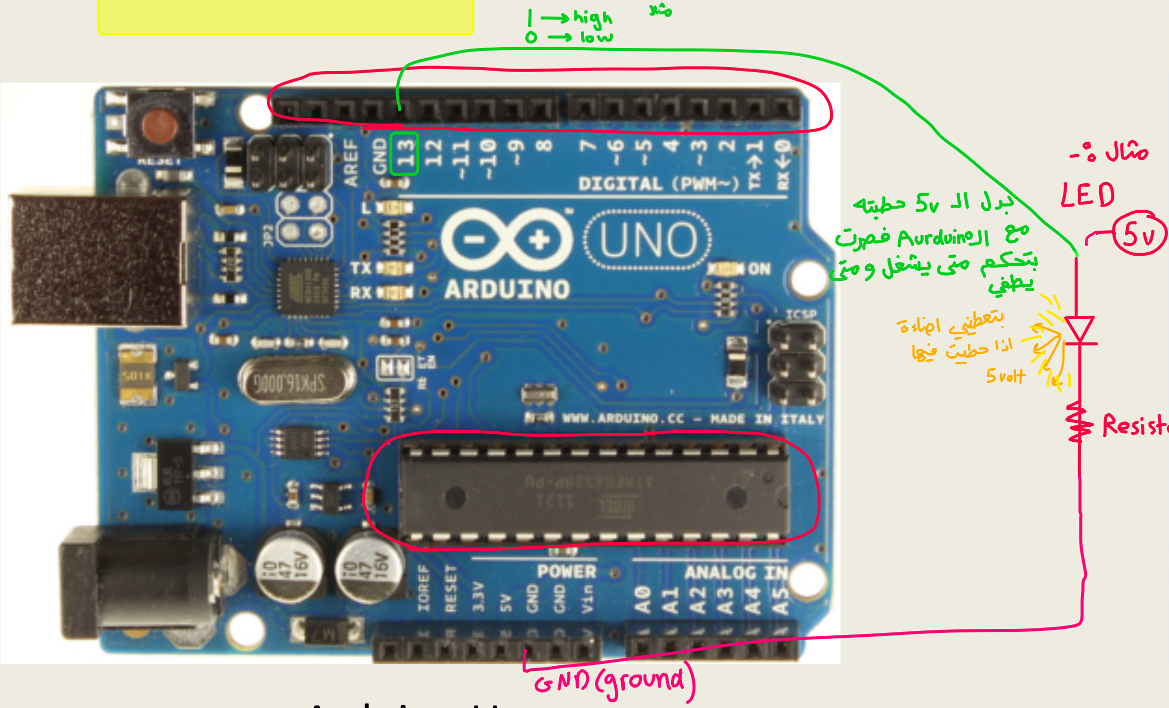
$$(K - 3.6\omega^2) + j\omega(K - \omega^2) = 0$$

- Notice that this equation can be satisfied only if $\omega = 0, K = 0$.
- Because of the presence of a double pole at the origin, the root locus is tangent to the $j\omega$axis at $k = 0$.
- The root-locus branches do not cross the $j\omega$axis.
- The root loci of this system is shown in the Figure.

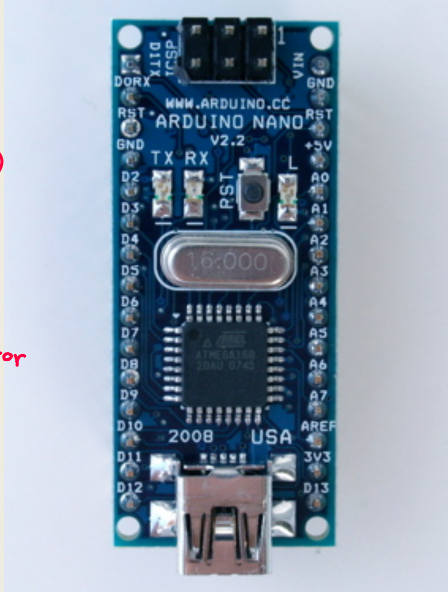


Arduino

Arduino: This is our Brain in Phys120B

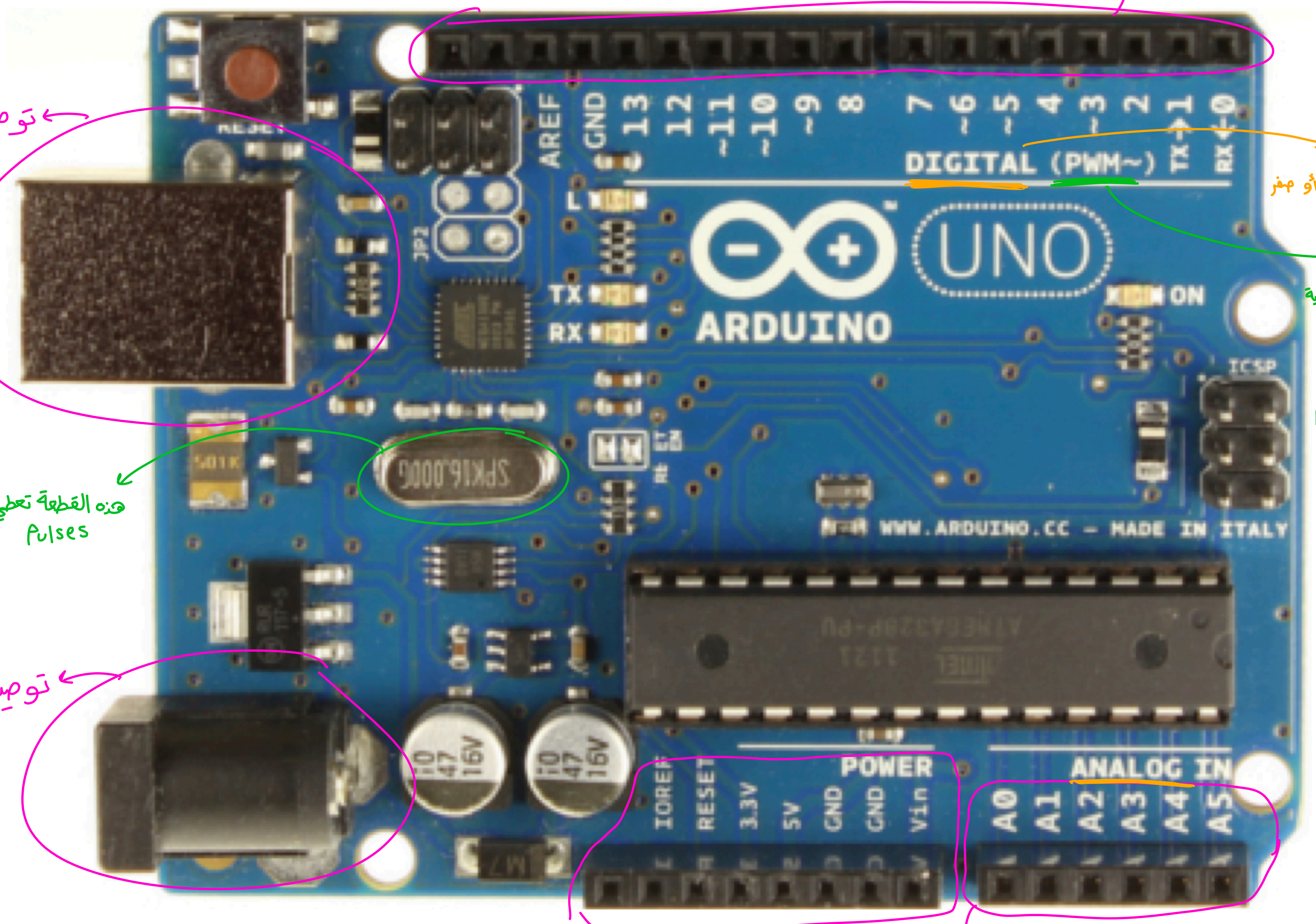


Arduino Uno



Arduino Nano

- Packaged Microcontroller (ATMega 328)
  - lots of varieties; we'll primarily use Uno and Nano
  - USB interface; breakout to pins for easy connections
  - Cross-platform, Java-based IDE, C-based language
  - Provides higher-level interface to guts of device



Digital
sensors توصيل

USB توصيل

(Digital)
يأخذ قيم واحد أو صفر

(Pulse with Modulation)
يعطي Pulses بوحدة الثانية

اذا زاد عدد
الـPulses
بقدر اعتبره
High

اذا قلت ال
Pulses في جزء
من الثانية
بقدر اعتبره
(semi analog)

هذه القطعة تعطي
Pulses

توصيل شاحن

توصيل ال
Power supply

توصيل
sensors

(Analog)
يأخذ قيم ليست صفر و واحد

Board
جاهز للاستعمال

Every Arduino “Sketch”

- Each “sketch” (code) has these common elements

```
// variable declarations, like
const int LED=13;
```

بنعرفهم constants
في البداية

يعني استبدل كلمة LED
ب 13

```
void setup()
{
  // configuration of pins, etc.
}
```

setup يُنفذ مرة واحدة

```
void loop()
{
  // what the program does, in a continuous loop
}
```

loop تتكرر

- Other subroutines can be added, and the internals can get pretty big/complex

Rudimentary C Syntax

- Things to immediately know
  - anything after // on a line is ignored as a comment
  - braces { } encapsulate blocks
  - semicolons ; must appear after every command
    - exceptions are conditionals, loop invocations, subroutine titles, precompiler things like `#include`, `#define`, and a few others
  - every variable used in the program needs to be declared
    - common options are `int`, `float`, `char`, `long`, `unsigned long`, `void`
    - conventionally happens at the top of the program, or within subroutine if confined to { } block
  - Formatting (spaces, indentation) are irrelevant in C
    - but it is to your great benefit to adopt a rigid, readable format
    - much easier to read if indentation follows consistent rules

Example Arduino Code

شي يضوي و يطفي

13 الرقم يلي موصين معه

```
// blink_LED. . . . . . . . slow blink of LED on pin 13
const int LED = 13;      // LED connected to pin 13
                         // const: will not change in prog.
```

يعني كل كلمة LED نستبدلها بـ 13

مهم

```
void setup()             // obligatory; void->returns nada
{
  pinMode(LED, OUTPUT); // pin 13 as output (Arduino cmd)
}
```

!!

ينفذ مرة واحدة فقط

13

عم نحكي عن ال LED

اذاً output

```
void loop()              // obligatory; returns nothing
{
  digitalWrite(LED, HIGH); // turn LED ON (Arduino cmd)
  delay(1000);             // wait 1000 ms (Arduino cmd)
  digitalWrite(LED, LOW); // turn LED OFF
  delay(1000);             // wait another second
}
```

13

اشتغل لمدة one second

5 volt

ويعني delay ثانية واحدة (1000 ms)

13

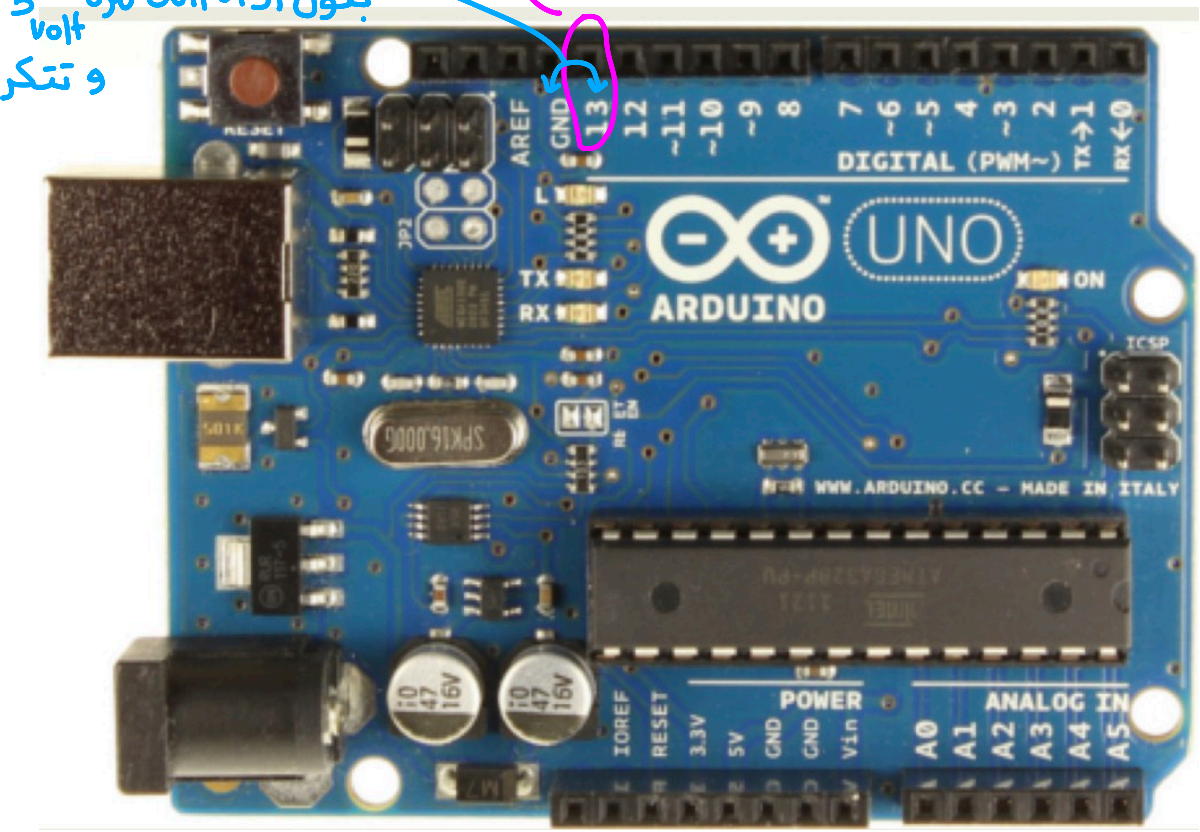
0 volt

طفي لمدة one second

و تتكرر ......

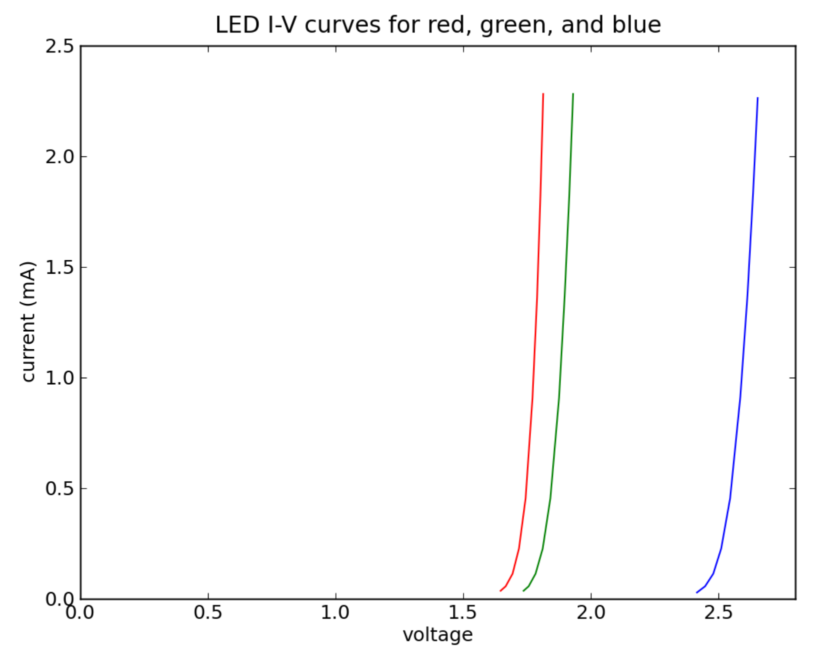
Pin 13 as output

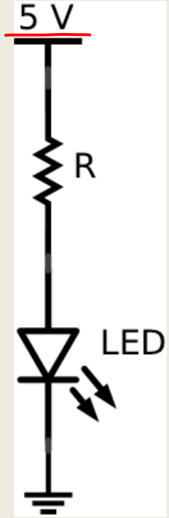
لو قست بين ال 13 وال ground بالـVoltmeter
بكون الoutput مرة 5 volt لمدة ثانية ومرة 0 volt لمدة ثانية
و تتكرر



LED hookup

- The output of Arduino digital I/O pins will be either 0 or 5 volts
- An LED has a diode-like I-V curve
- Can’t just put 5 V across
  - it’ll blow, unless current is limited
- Put resistor in series, so ~2.5 V drop across each
  - 250 Ω would mean 10 mA
  - 10 mA is pretty bright





لما نعطيه 5volt بضوي لمدة 1second
لما نعطيه 0volt بطفي 1second

مرة سوف يضوي
ومرة سوف يطفي

✓

Comments on Code

- Good practice to start code with descriptive comment
  - include name of sketch so easy to relate print-out to source
- Most lines commented: also great practice
- Only one integer variable used, and does not vary
  - so can declare as `const`
- `pinMode()`, `digitalWrite()`, and `delay()` are Arduino commands
- `OUTPUT`, `HIGH`, `LOW` are Arduino-defined constants
  - just map to integers: 1, 1, 0, respectively
- Could have hard-coded `digitalWrite(13,1)`
  - but less human-readable than `digitalWrite(LED, HIGH)`
  - also makes harder to change output pins (have to hunt for each instance of 13 and replace, while maybe not every 13 should be)

Capital Letters →

✓

Arduino-Specific Commands

- Command reference: http://arduino.cc/en/Reference/HomePage
  - Also abbr. version in Appendix C of *Getting Started* book (2nd ed.)
- In first week, we'll see:
  - ✓ `pinMode`(*pin*, [INPUT $\overset{\text{or}}{|}$ OUTPUT])
  - ✓ `digitalWrite`(*pin*, [LOW | HIGH])
  - ✓ `digitalRead`(*pin*) → `int`
  - ✗ `analogWrite`(*pin*, [0…255])
  - ✓ `analogRead`(*pin*) → `int` in range [0..1023]
  - `delay`(*integer milliseconds*)
  - `millis()` → `unsigned long` (ms elapsed since reset)

Arduino Serial Commands

- Also we'll use serial communications in week 1:
  - `Serial.begin`(*baud*): in `setup`; 9600 is common choice
  - `Serial.print`(*string*): *string* → `"example text "`
  - `Serial.print`(*data*): prints *data* value (default encoding)
  - `Serial.print`(*data,encoding*)
    - *encoding* is DEC, HEX, OCT, BIN, BYTE for format
  - `Serial.println()`: just like `print`, but CR & LF (`\r\n`) appended تنتقل إلى سطر جديد
  - `Serial.available()` → `int` (how many bytes waiting)
  - `Serial.read()` → `char` (one byte of serial buffer)
  - `Serial.flush()`: empty out pending serial buffer

Types in C

- We are likely to deal with the following types

```
char c;             // single byte
int i;              // typical integer
unsigned long j;    // long positive integer
float x;            // floating point (single precision)
double y;           // double precision
```

حرف واحد

short

long

8 digits بعد الفاصلة العشرية

16 digits بعد الفاصلة العشرية

```
c = 'A';
i = 356;
j = 230948935;
x = 3.1415927;
y = 3.14159265358979;
```

→ 8 digits بعد الفاصلة العشرية

→ 16 بعد الفاصلة

- Note that the variable `c='A'` is just an 8-bit value, which happens to be 65 in decimal, 0x41 in hex, 01000001
  - could say `c = 65;` or `c = 0x41;` with equivalent results
- Not much call for double precision in Arduino, but good to know about for other C endeavors

Changing Types (Casting)

- Don't try to send float values to pins, and watch out when dividing integers for unexpected results
- Sometimes, we need to compute something as a floating point, then change it to an integer
  - `ival = (int) fval;`
  - `ival = int(fval);   // works in Arduino, anyhow`
- Beware of integer math:
  - 1/4 = 0; 8/9 = 0; 37/19 = 1
  - so sometimes want `fval = ((float) ival1)/ival2`
  - or `fval = float(ival1)/ival2 //okay in Arduino`

Conditionals

- The if statement is a workhorse of coding
  - `if (i < 2)`
  - `if (i <= 2)`
  - `if (i >= -1)`
  - `if (i == 4) // note difference between == and =`
  - `if (x == 1.0)`
  - `if (fabs(x) < 10.0)` <> not equall
  - `if (i < 8 && i > -5)` `// && = and`
  - `if (x > 10.0 || x < -10.0)` `// || = or`
- Don't use assignment (=) in test clauses ! = not
  - Remember to double up ==, &&, ||
- Will execute single following command, or next { } block
  - wise to form { } block even if only one line, for readability/expansion
- Can combine with else statements for more complex behavior

If..else construction

* في منطقة التعريف يجب أن نُعرِّف مثلاً

```
Const int BUTTON = 12 ;
Const int LED    = 13 ;
```

اذا كان عندي 2 pins شغالين لازم اعرفهم الثنين

- Snippet from code to switch LED ON/OFF by listening to a button

الان منطقة ال setup

```
Void setup ()
{
  PinMode (BUTTON , INPUT) ;
  PinMode ( LED , OUTPUT) ;
}
```

```
void loop()
{
  val = digitalRead(BUTTON);
  if (val == HIGH){
    digitalWrite(LED, HIGH);
  } else {
    digitalWrite(LED, LOW);
  }
}
```

(1 0)

capital letter

كلمة واحدة

- BUTTON and LED are simply constant integers defined at the program start
- Note the use of braces
  - exact placement/arrangement unnec., but be consistent

For loops

- Most common form of loop in C
  - also `while`, `do..while` loops
  - associated action encapsulated by braces

```
int k,count;

count = 0;
for (k=0; k < 10; k++)
{
  count += 1;
  count %= 4;
}
```

من zero إلى 10
لفات

- `k` is iterated
  - assigned to zero at beginning
  - confined to be less than 10
  - incremented by one after each loop (could do `k += 1`)
- `for(;;)` makes infinite loop (no conditions)
- `x += 1` means `x = x + 1`; `x %= 4` means `x = x % 4`
  - `count` will go 1, 2, 3, 0, 1, 2, 3, 0, 1, 2 then end loop

`#define` to ease the coding

```
#define NPOINTS 10
#define HIGHSTATE 1
```

الاسلوب القديم

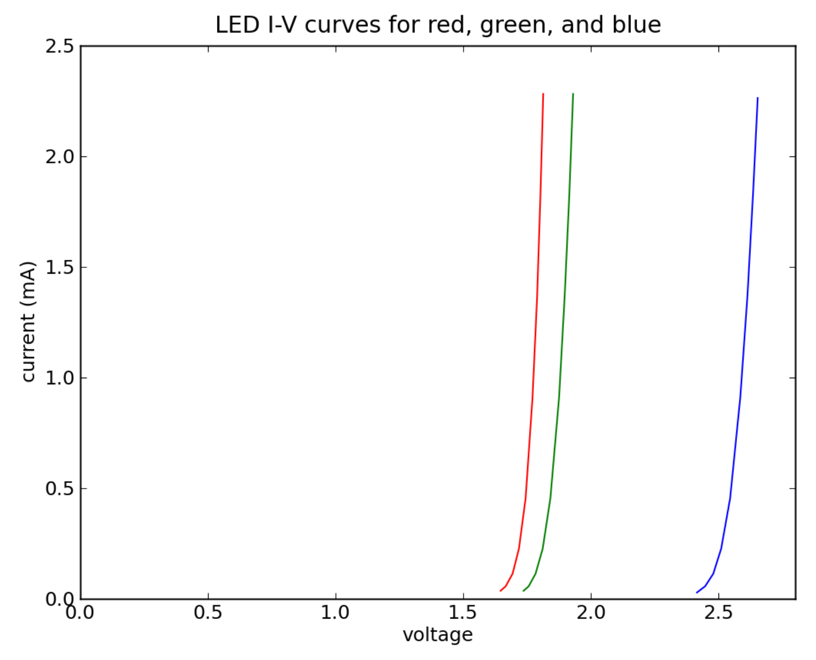
constant
integer

Const int Npoints = 10

- `#define` comes in the "preamble" of the code
  - note no semi-colons
  - just a text replacement process: any appearance of NPOINTS in the source code is replaced by 10
  - Convention to use all CAPs to differentiate from normal variables or commands
  - Now to change the number of points processed by that program, only have to modify one line
  - Arduino.h defines handy things like HIGH = 0x1, LOW = 0x0, INPUT = 0x0, OUTPUT = 0x1, INPUT_PULLUP = 0x2, PI, HALF_PI, TWO_PI, DEG_TO_RAD, RAD_TO_DEG, etc. to make programming easier to read/code

LED hookup

- The output of Arduino digital I/O pins will be either 0 or 5 volts
- An LED has a diode-like I-V curve
- Can’t just put 5 V across
    - it’ll blow, unless current is limited
- Put resistor in series, so ~2.5 V drop across each
    - 250 Ω would mean 10 mA
    - 10 mA is pretty bright



Blink Function (Subroutine)

- For complex blink patterns, it pays to consolidate blink operation into a function

```
void blink(int ontime, int offtime)
{
   // turns on LED (externally defined) for ontime ms
   // then off for offtime ms before returning
   digitalWrite(LED, HIGH);
   delay(ontime);
   digitalWrite(LED, LOW);
   delay(offtime);
}
```

- Now call with, e.g., `blink(600,300)`

بضوي لفترة 600 ms

وبطفي لفترة 300 ms

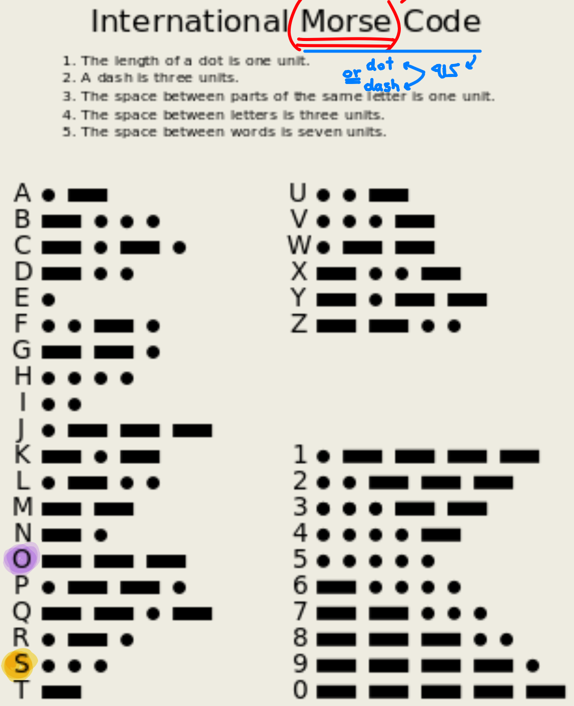
milli second

- Note function definition expects two integer arguments
- LED is assumed to be global variable (defined outside of loop)

Blink Constructs

- For something like Morse Code, could imagine building functions on functions, like

تقلل عدد الاسطر التي نحتاجها



International Morse Code

كله dot or dash

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

بنفذ هاد ٣ مرات
بضوي 200 ms
وبطفي 200 ms
ويكررها ٣ مرات

```
void dot()
{ blink(200,200); }
```

ontime — offtime

فترة قصيرة — فترة قصيرة

```
void dash()
{ blink(600,200); }
```

فترة طويلة — فترة قصيرة

معناها بين الحرف والحرف انتظر 400 (ms) ملي ثانية

```
void letterspace()
{ delay(400); }
```

ينفذ ال delay 400 ms

معناها بين الكلمة والكلمة انتظر 800 (ms)

```
void wordspace()
{ delay(800); }
```

- And then perhaps letter functions:

```
void morse_s()
{ dot(); dot(); dot(); letterspace(); }

void morse_o()
{ dash(); dash(); dash(); letterspace(); }
```

حياخذ 2800 ms

Morse, continued

- You could then spell out a word pretty easily like:

```
  void loop()
{
  morse_s();        → 1200 ms
  morse_o();        → 2800 ms
  morse_s();        → 1200 ms
  wordspace();      → 800 ms
}
```

حياخذ

معناه هذا البرنامج
سوف يأخذ للتنفيذ

6000 ms

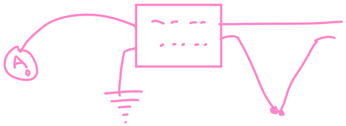
- Once you have a library of all the letters, it would be very simple to blink out anything you wanted

Temperature Measurements

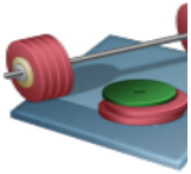
Theory

Different methods for measuring the Tempertature:

- Thermocouples
- Thermistors
- RTD (Resistance Temperature Detector)
  - e.g. Pt100
- Infrared
- Thermometers



Temperature Sensors



Make the following Temperature Sensors work with Arduino:

NTC Thermistor

TMP36

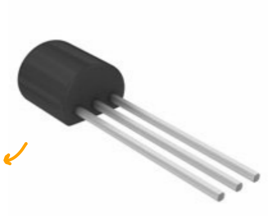
↗ الأشهر

* ممكن يجيب النا واحد منهم و يطلب طريقة التوصيل على ال Aurduino board

أو انو نكتب code لقراءة ال temprature باستخدام واحد منهم

Small-scale Temperature Sensors

TMP36



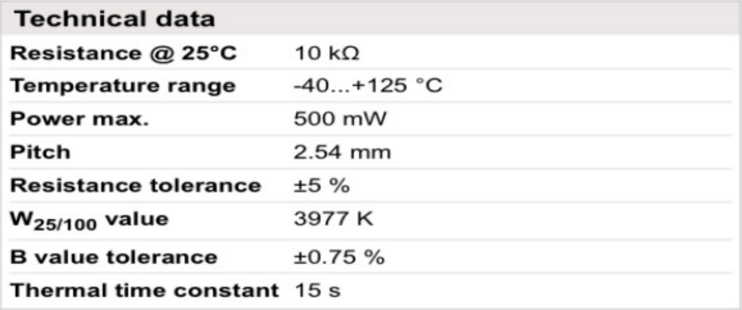
مقصوص cylinder ↙

Technical data	
Temperature measurement range	-40...+125 °C
Accuracy	±2 °C (0...70 °C)
Power supply	2.3...5.5 V
Package	TO-92
Temperature sensitivity, voltage	10 mV/°C

https://www.sparkfun.com/products/10988

https://www.elfa.se/elfa3~eu_en/elfa/init.do?item=73-889-29&toc=0&q=73-889-29

NTC Thermistor



Technical data	
Resistance @ 25°C	10 kΩ
Temperature range	-40...+125 °C
Power max.	500 mW
Pitch	2.54 mm
Resistance tolerance	±5 %
$W_{25/100}$ value	3977 K
B value tolerance	±0.75 %
Thermal time constant	15 s

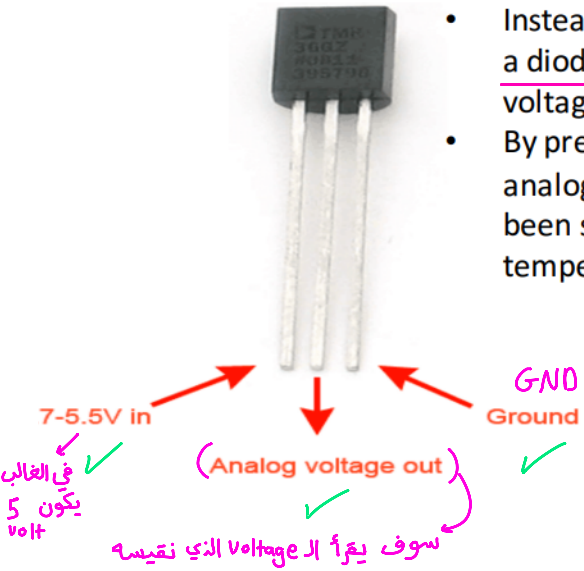
https://www.elfa.se/elfa3~eu_en/elfa/init.do?item=60-260-41&toc=0&q=60-260-41

Tutorial: http://garagelab.com/profiles/blogs/tutorial-using-ntc-thermistors-with-arduino

TMP36

Theory

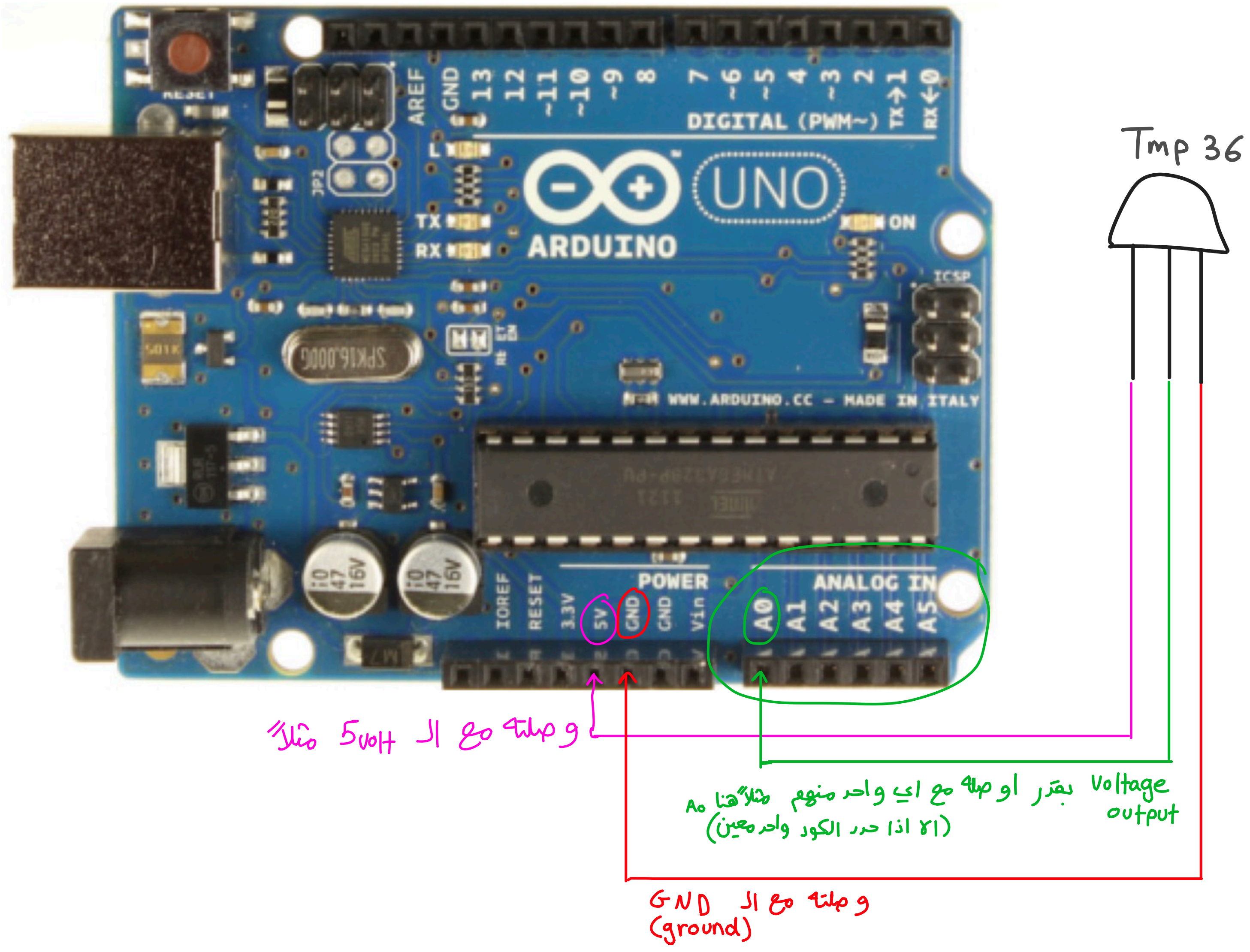
- These sensors use a solid-state technique to determine the temperature. That is to say, they don't use mercury (like old thermometers), bimetalic strips (like in some home thermometers or stoves), nor do they use thermistors (temperature sensitive resistors).
- Instead, they use the fact as temperature increases, the voltage across a diode increases at a known rate. (Technically, this is actually the voltage drop between the base and emitter - the Vbe - of a transistor.)
- By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature. There have been some improvements on the technique but, essentially that is how temperature is measured.



Because these sensors have no moving parts, they are precise, never wear out, don't need calibration, work under many environmental conditions, and are consistant between sensors and readings. Moreover they are very inexpensive and quite easy to use.

ttps://learn.adafruit.com/tmp36-temperature-sensor

* مثلاً اذا طلب السؤال توصيل



وصلته مع ال 5volt مثلاً

Voltage output بقدر اوصله مع اي واحد منهم مثلاً هنا A0 (الا اذا حدد الكود واحد معين)

وصلته مع ال GND (ground)

Datasheet Calculations

Theory

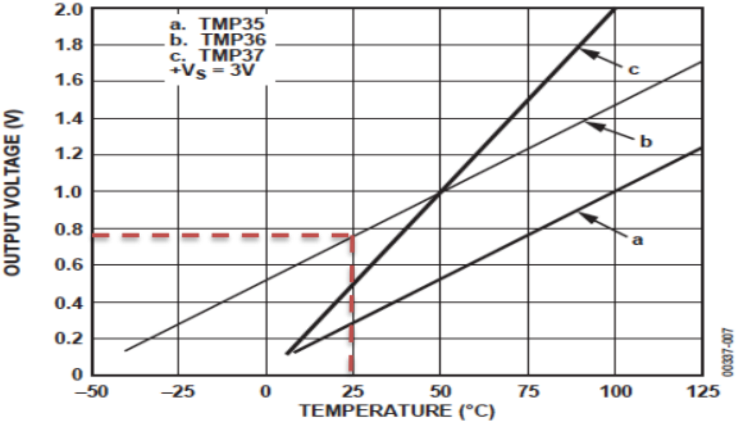


Figure 6. Output Voltage vs. Temperature

From the plot we have:

(x1, y1) = (750mV, 25°C)

(x2, y2) = (1000mV, 50°C)

Linear relationship: $y = ax + b$

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

You have to find a (slope) and b (intercept):

y-25°C = ((50°C-25°C)/(1000mV-750mV)) * (x-750mV)

This gives: y[°C] = (1/10)*x[mv]-50

Voltage-based Sensors

According to the TMP36 datasheet, the relation of the output voltage to the actual temperature uses this equation:

$$y[°C] = (1/10)*x[mv]-50$$

Where the voltage value is specified in millivolts.

However, before you use that equation, you must convert the integer value that the analogRead function returns into a millivolt value.

10-bit analog to digital converter

You know that for a 5000mV (5V) value span the analogRead function will return 1024 possible values:

$$voltage = (5000 / 1024) * output$$

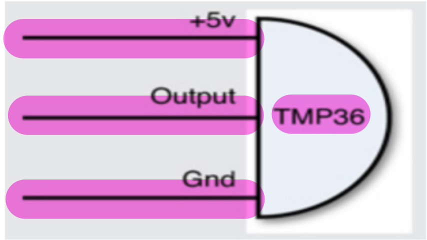
mV

Where

$$output = analogRead(aichannel)$$

0-1023

A0-A5



Code For TMP36

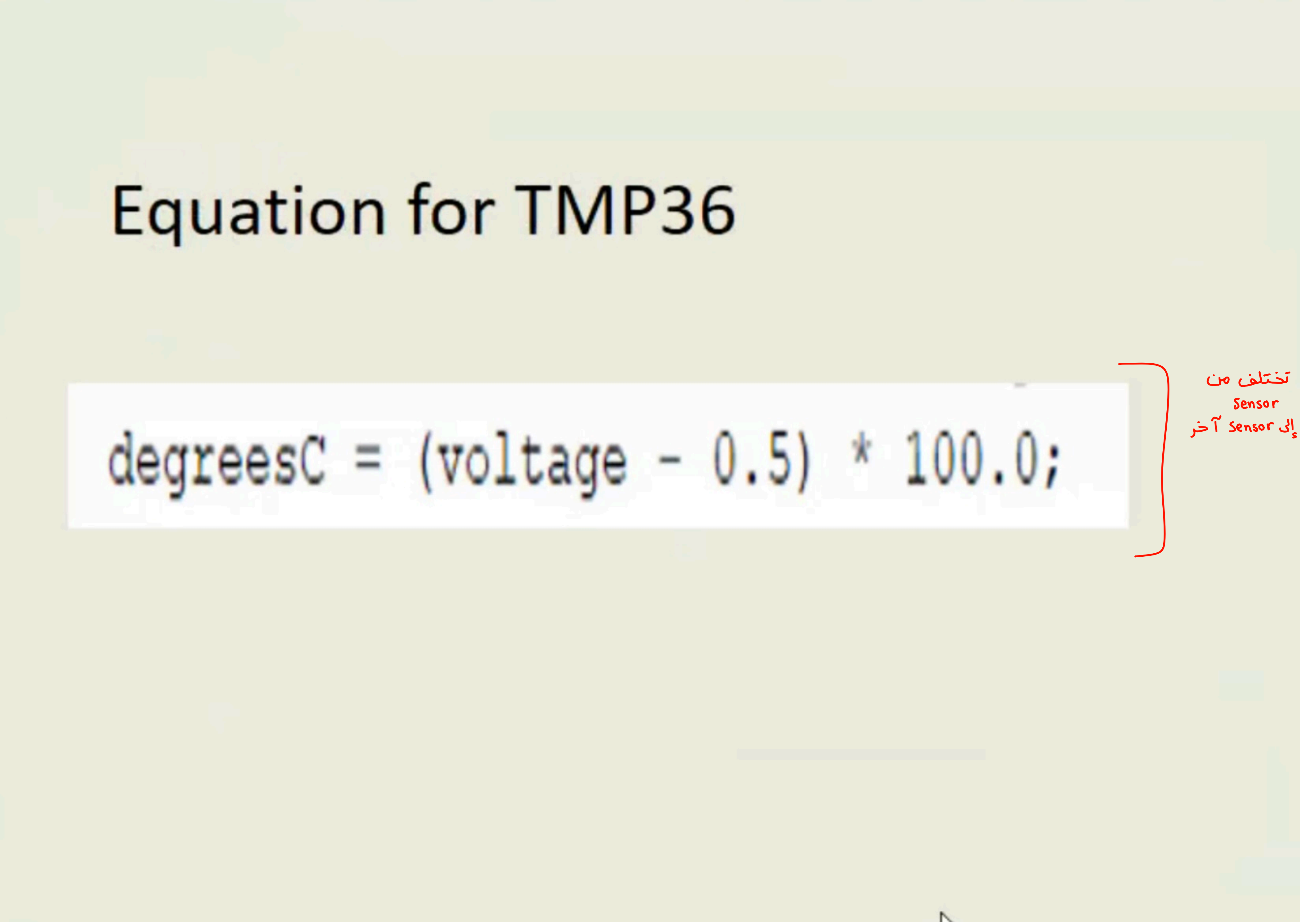
ال code
حفظ

CODE For temperature measurement

```
const int temperaturePin = 0 ;
```

Note يعني
(ليس من ضمن الكود)

```
//We'll use analog input 0 to read Temperature Data const int temperaturePin = 0;
void setup()
{ Serial.begin(9600); }
void loop()
{ float voltage, degreesC, degreesF;
voltage = getVoltage(temperaturePin);
// Now we'll convert the voltage to degrees Celsius.
// This formula comes from the temperature sensor datasheet:
degreesC = (voltage - 0.5) * 100.0;
// Send data from the Arduino to the serial monitor window
Serial.print("voltage: ");
Serial.print(voltage);
Serial.print(" deg C: ");
Serial.println(degreesC);
delay(1000);
// repeat once per second (change as you wish!) }
float getVoltage(int pin)
{ return (analogRead(pin) * 0.004882814); }
// This equation converts the 0 to 1023 value that analogRead()
// returns, into a 0.0 to 5.0 value that is the true voltage
// being read at that pin.
```

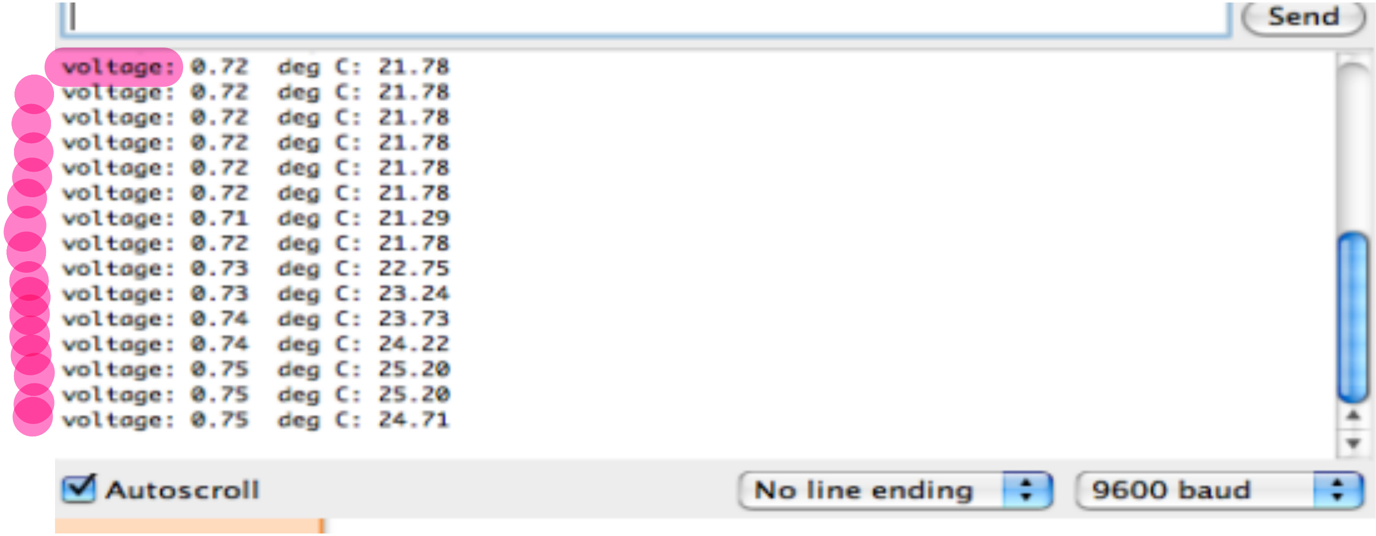


Equation for TMP36

```
degreesC = (voltage - 0.5) * 100.0;
```

تختلف من
sensor
إلى sensor آخر

In the Computer



```
voltage: 0.72  deg C: 21.78
voltage: 0.72  deg C: 21.78
voltage: 0.72  deg C: 21.78
voltage: 0.72  deg C: 21.78
voltage: 0.72  deg C: 21.78
voltage: 0.72  deg C: 21.78
voltage: 0.71  deg C: 21.29
voltage: 0.72  deg C: 21.78
voltage: 0.73  deg C: 22.75
voltage: 0.73  deg C: 23.24
voltage: 0.74  deg C: 23.73
voltage: 0.74  deg C: 24.22
voltage: 0.75  deg C: 25.20
voltage: 0.75  deg C: 25.20
voltage: 0.75  deg C: 24.71
```

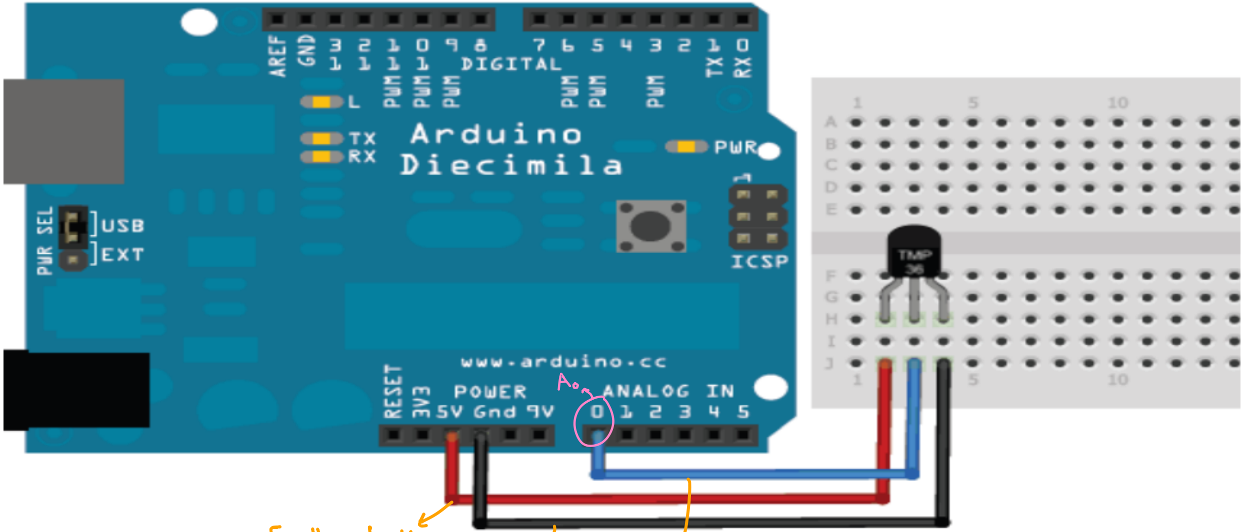
☑ Autoscroll

No line ending

9600 baud

Wiring

TMP36 Temperature Wiring



اليسار مع ال 5v

اليمين مع الground

النص مع ال 0 (A0)

```
// We'll use analog input 0 to read Temperature Data const int t
void setup()
 { Serial.begin(9600); }
void loop()
{ float voltage, degreesC, degreesF;
voltage = getVoltage(temperaturePin);
 // Now we'll convert the voltage to degrees Celsius.
// This formula comes from the temperature sensor datasheet:
 degreesC = (voltage - 0.5) * 100.0;
 // Send data from the Arduino to the serial monitor window
Serial.print("voltage: ");
Serial.print(voltage);
Serial.print(" deg C: ");
Serial.println(degreesC);
delay(1000);
// repeat once per second (change as you wish!) }
float getVoltage(int pin)
 { return (analogRead(pin) * 0.004882814); }
// This equation converts the 0 to 1023 value that analogRead()
// returns, into a 0.0 to 5.0 value that is the true voltage
// being read at that pin.
```

يعني comment

[ليس من الcode]

variable

" "
بتنكتب زي ما هي
بال computer

variable

نتقل لسطر جديد

انتظار
(تأخير)

$\frac{5}{1024}$

Arduino
يقرأ ال voltage

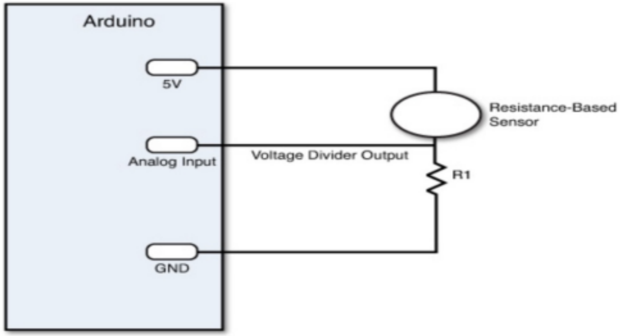
Resistance-based Sensors

Theory

The problem with resistance sensors is that the Arduino analog interfaces can't directly detect resistance changes.

This will require some extra electronic components. The easiest way to detect a change in resistance is to convert that change to a voltage change. You do that using a **voltage divider**, as shown below.

Thermistor



By keeping the power source output constant, as the resistance of the sensor changes, the voltage divider circuit changes, and the output voltage changes. The size of resistor you need for the R1 resistor depends on the resistance range generated by the sensor and how sensitive you want the output voltage to change.

Generally, a value between 1K and 10K ohms works just fine to create a meaningful output voltage that you can detect in your Arduino analog input interface.

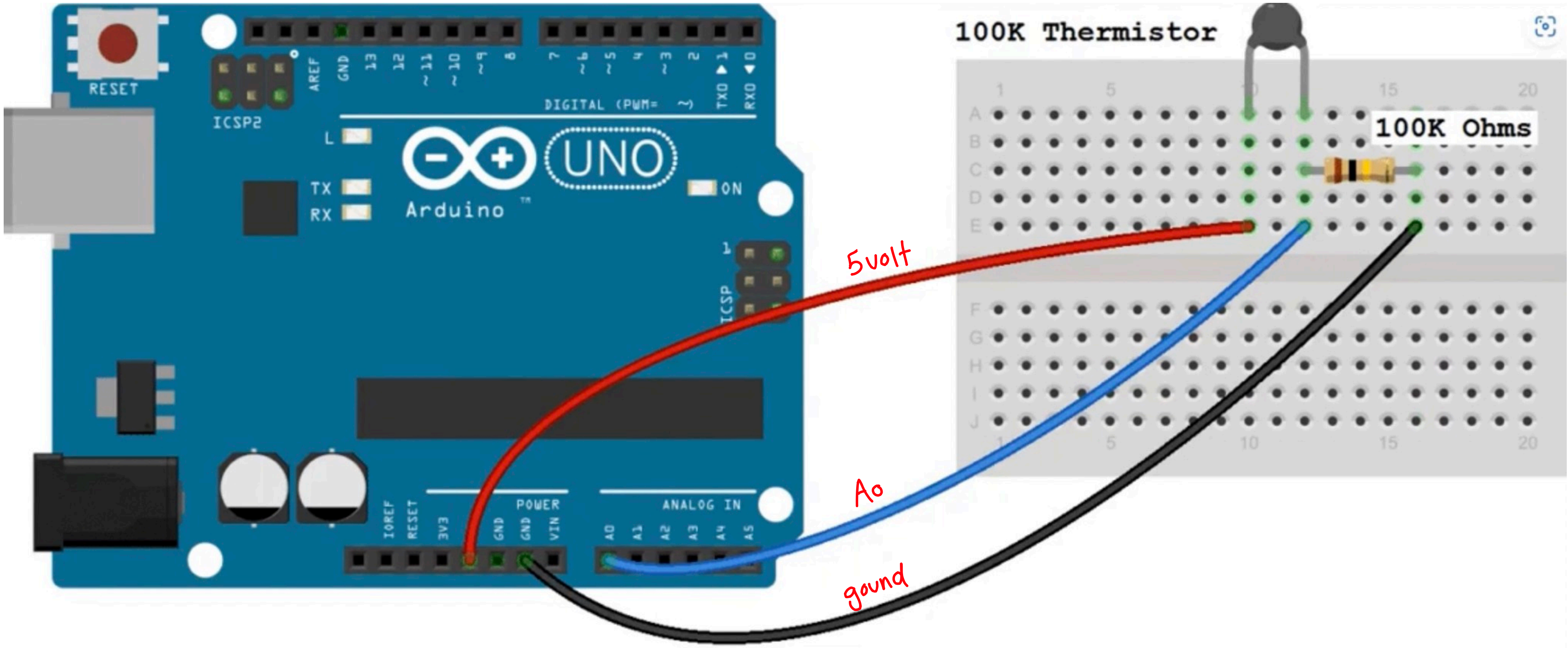
E.g., the Steinhart-Hart Equation can be used to find the Temperature:

$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3$$

معادلة تربط المقاومة بالحرارة

Thermistor wiring

(Extra slide)



Thermistor Code حفظ

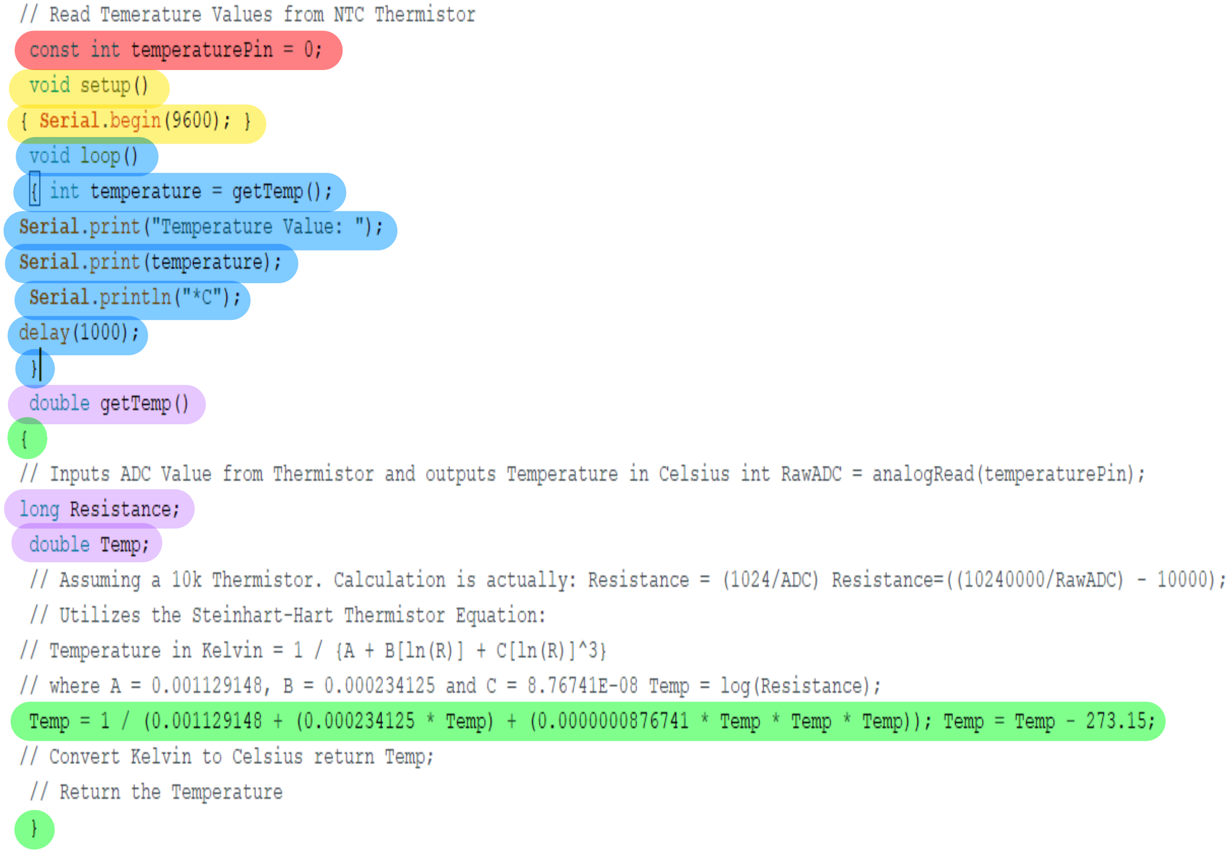
```
// Read Temerature Values from NTC Thermistor
const int temperaturePin = 0;
void setup()
{ Serial.begin(9600); }
void loop()
{ int temperature = getTemp();
Serial.print("Temperature Value: ");
Serial.print(temperature);
Serial.println("*C");
delay(1000);
}
double getTemp()
{
// Inputs ADC Value from Thermistor and outputs Temperature in Celsius int RawADC =
analogRead(temperaturePin);
long Resistance;
double Temp;
// Assuming a 10k Thermistor. Calculation is actually: Resistance = (1024/ADC)
Resistance=((10240000/RawADC) - 10000);
// Utilizes the Steinhart-Hart Thermistor Equation:
// Temperature in Kelvin = 1 / {A + B[ln(R)] + C[ln(R)]^3}
// where A = 0.001129148, B = 0.000234125 and C = 8.76741E-08 Temp = log(Resistance);
                                    ln(Resistance)                           ln(Resistance) * ln(Resistance)
Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp *
Temp)); Temp = Temp - 273.15;
// Convert Kelvin to Celsius return Temp;
// Return the Temperature
```

* حتى الارقام العشرية حفظ

Read Temerature Values from NTC Thermistor

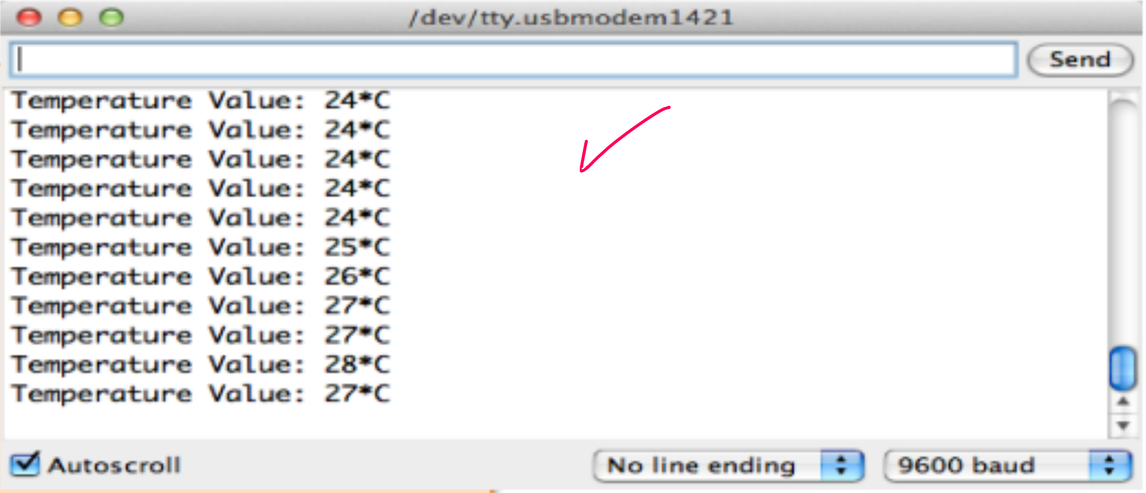
```
const int temperaturePin = 0;
void setup()
{ Serial.begin(9600); }
void loop()
{ int temperature = getTemp();
Serial.print("Temperature Value: ");
Serial.print(temperature);
Serial.println("*C");
delay(1000);
}
double getTemp()
{
long Resistance;
double Temp;
Resistance=((10240000/RawADC) - 10000);
Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp * Temp));
 Temp = Temp - 273.15;
```

In Arduino



```
// Read Temerature Values from NTC Thermistor
 const int temperaturePin = 0;
 void setup()
{ Serial.begin(9600); }
 void loop()
 { int temperature = getTemp();
Serial.print("Temperature Value: ");
Serial.print(temperature);
 Serial.println("*C");
delay(1000);
 }
 double getTemp()
{
// Inputs ADC Value from Thermistor and outputs Temperature in Celsius int RawADC = analogRead(temperaturePin);
long Resistance;
 double Temp;
 // Assuming a 10k Thermistor. Calculation is actually: Resistance = (1024/ADC) Resistance=((10240000/RawADC) - 10000);
 // Utilizes the Steinhart-Hart Thermistor Equation:
// Temperature in Kelvin = 1 / {A + B[ln(R)] + C[ln(R)]^3}
// where A = 0.001129148, B = 0.000234125 and C = 8.76741E-08 Temp = log(Resistance);
 Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp * Temp)); Temp = Temp - 273.15;
// Convert Kelvin to Celsius return Temp;
 // Return the Temperature
 }
```

In Computer + wiring



/dev/tty.usbmodem1421

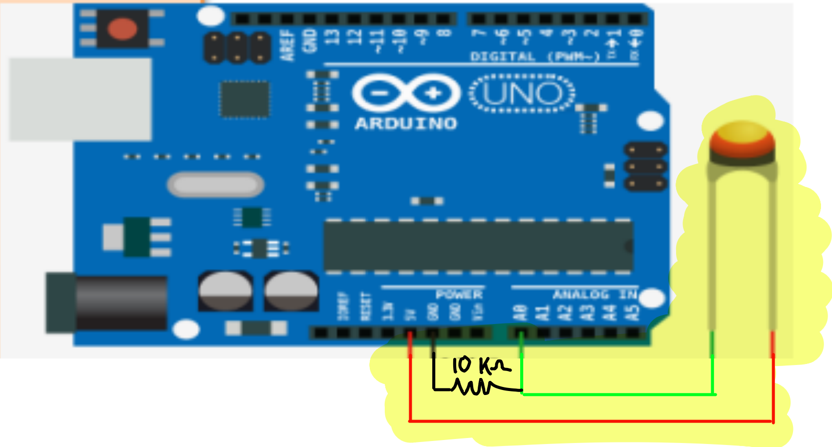
```
Temperature Value: 24*C
Temperature Value: 24*C
Temperature Value: 24*C
Temperature Value: 24*C
Temperature Value: 24*C
Temperature Value: 25*C
Temperature Value: 26*C
Temperature Value: 27*C
Temperature Value: 27*C
Temperature Value: 28*C
Temperature Value: 27*C
```

☑ Autoscroll

No line ending

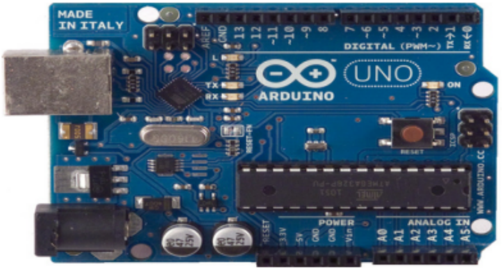
9600 baud

Serial Monitor



Temperature Data Logger/Embedded DAQ System

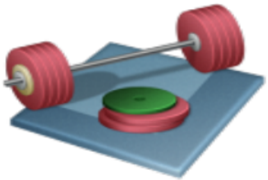
You use the PC when creating the software, then you download the software to the Arduino and disconnect the USB cable. Use e.g., a 9V battery or an external Power Supply.



NTC Thermistor

Use different Temperature sensors for comparison, i.e log data from 2 different sensors at the same time.

Temperature Data Logger/ Embedded DAQ System



Create a **Temperature Logger**/Embedded DAQ System. Suggested Tasks:

- Create and use a **Lowpass Filter/Average Filter**
- **Alarm** functionality: Use LEDs with different colors when Temperature is above/below the Limits
- Use e.g., Arduino **Wi- Fi/Ethernet Shield** for Communication over a network - or use the microSD card on these Shields
- Save the data to a microSD card located on the Wi- Fi/Ethernet Shield - or connect e.g., to **xively.com** or **temboo.com** - which are free datalogging sites.
- Log Temperature Data for e.g., 24 hours and import Data into Excel, LabVIEW or MATLAB for Analysis and Visualization
- Use e.g. a 9V battery or an external power source to make it portable and small

إلى هنا المطلوب من الـArduino في الـ Final

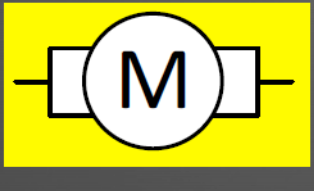
فكرة ال motor
⟶ to get full rotation

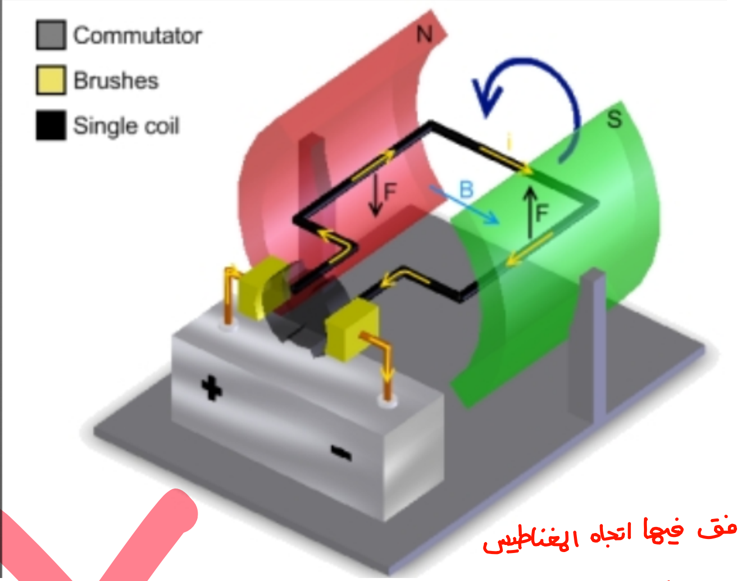
(Not required in Final Exam)
فقط للفهم

ما راح يجي اسئلة
عليه بالفاينل (فقط للفهم)

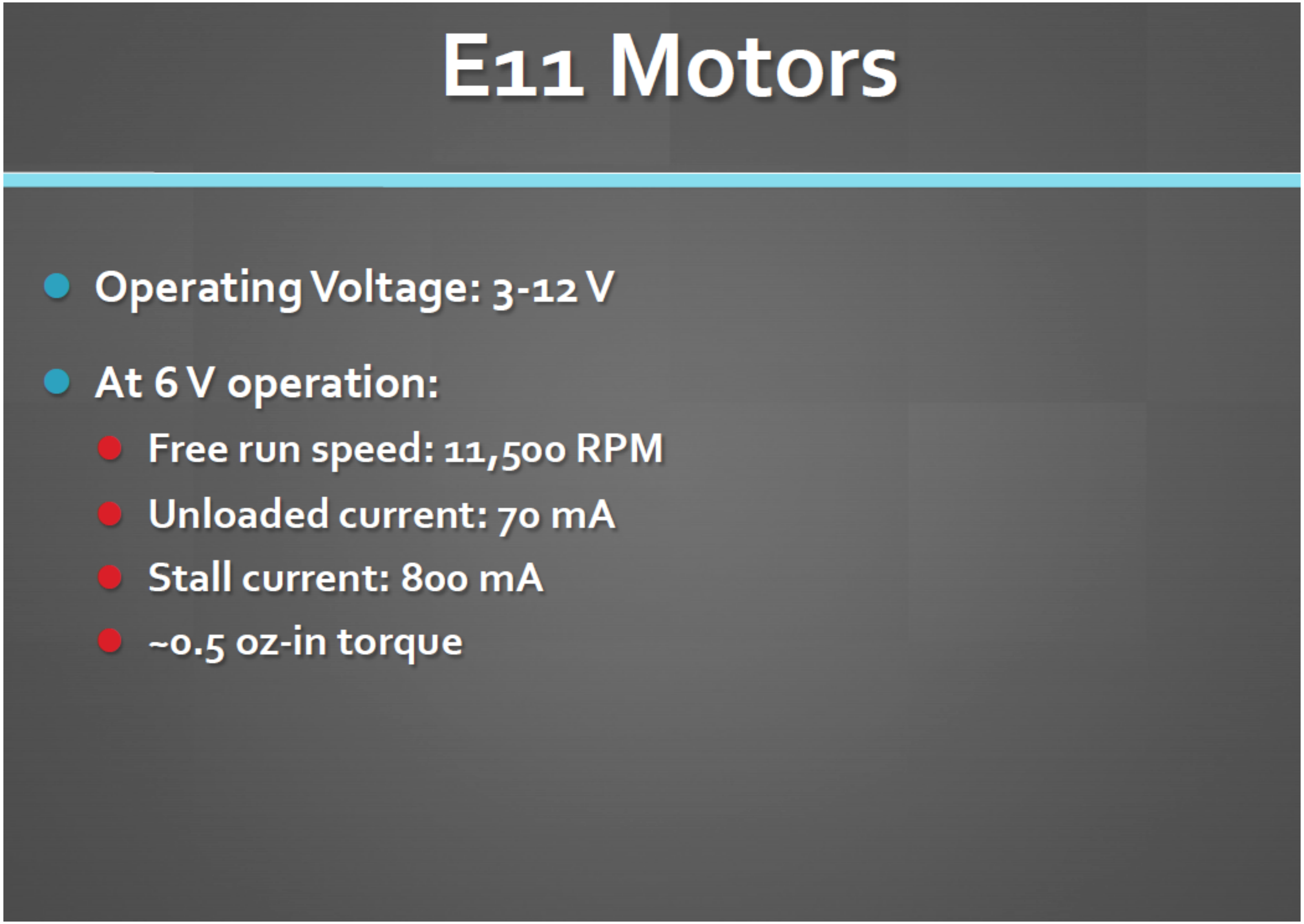
DC Motor

- DC motors spin when a steady voltage is applied
  - Can draw significant current (~ 1A or more)
- Fixed permanent magnet
- Rotating coil
- Brushes





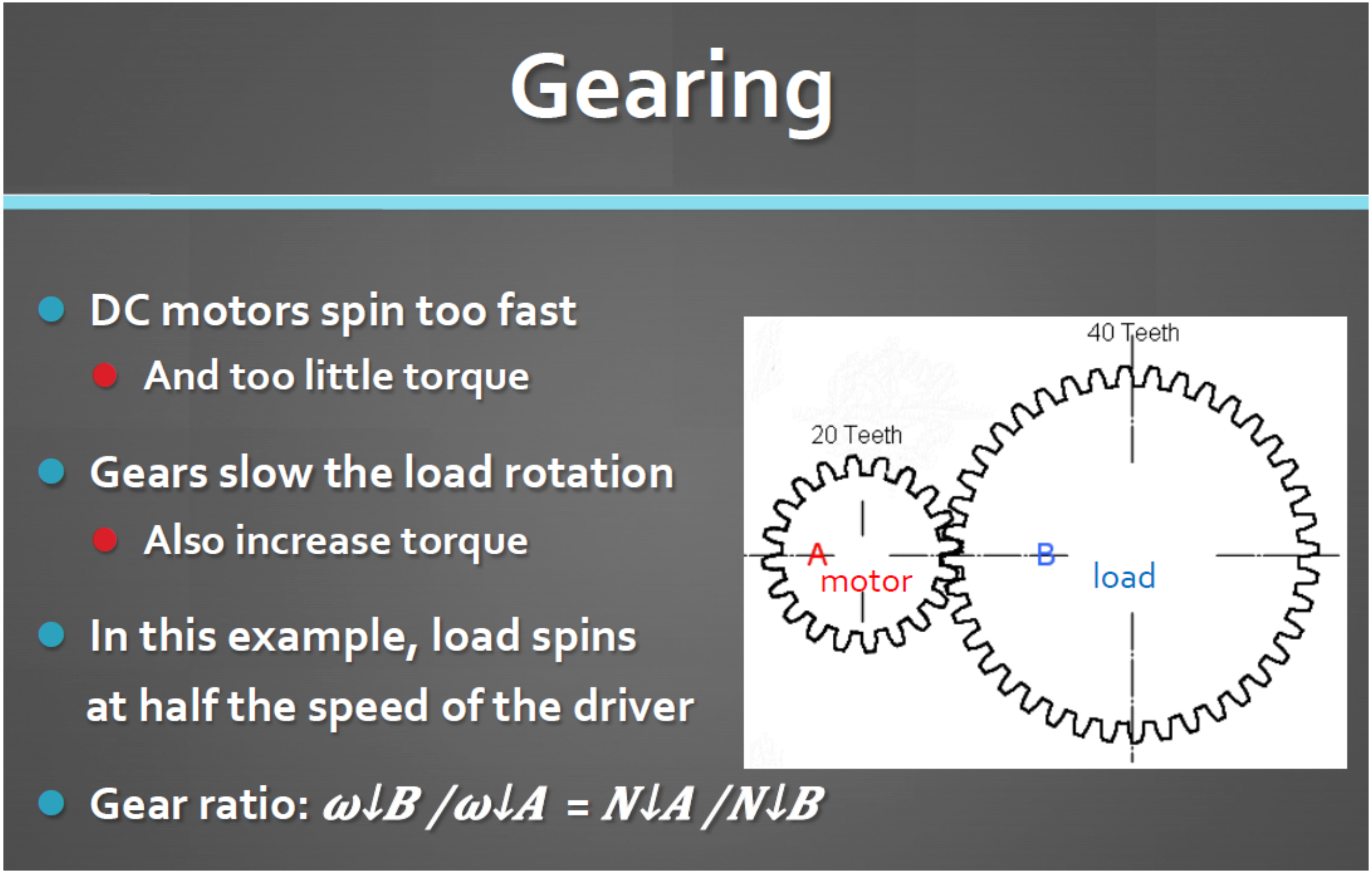
في اللحظة التي يتوافق فيها اتجاه المغناطيس
الداخلي مع الخارجي ← ينعكس اتجاه التيار الكهربائي



E11 Motors

- Operating Voltage: 3-12 V
- At 6 V operation:
  - Free run speed: 11,500 RPM
  - Unloaded current: 70 mA
  - Stall current: 800 mA
  - ~0.5 oz-in torque

Gears → تقلل السرعة
وتزيد الـTorque
و تزيد قوة الـ motor

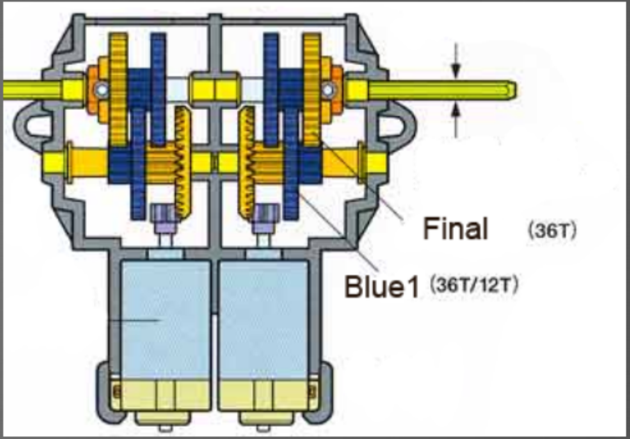


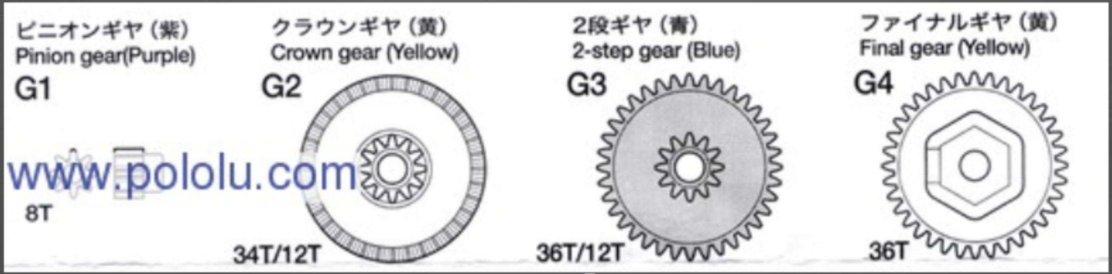
Gearing

- DC motors spin too fast
  - And too little torque
- Gears slow the load rotation
  - Also increase torque
- In this example, load spins at half the speed of the driver
- Gear ratio: $\omega\downarrow B / \omega\downarrow A = N\downarrow A / N\downarrow B$

Example: Tamiya Gear Box

- **Gear Ratio:**
  - Final to Blue1
  - Blue1 to Blue2
  - Blue2 to Crown
  - Crown to Pinion
  - Total:



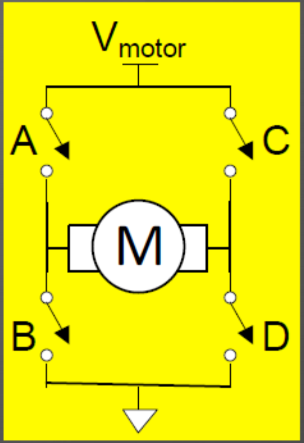


pololu.com

H-Bridge

- Motors require large current to operate
  - But Arduino outputs only offer 40 mA
- H-Bridges are used to drive the large current

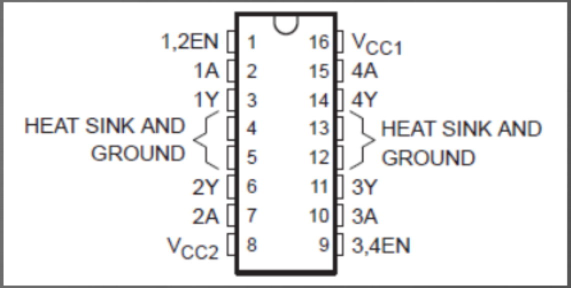
A	B	C	D	Motor
ON	OFF	OFF	ON	
OFF	ON	ON	OFF	
ON	OFF	ON	OFF	
OFF	OFF	OFF	OFF	
ON	ON	OFF	OFF	

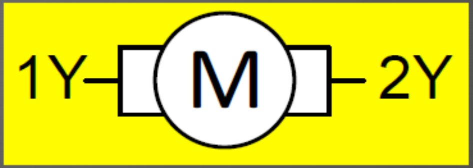


SN754410 H-Bridge

- 754410 Dual H-Bridge is easy to control with digital logic
  - V_{CC1} = Logic Supply (5V)
  - V_{CC2} = Motor Supply (4.5-36 V)

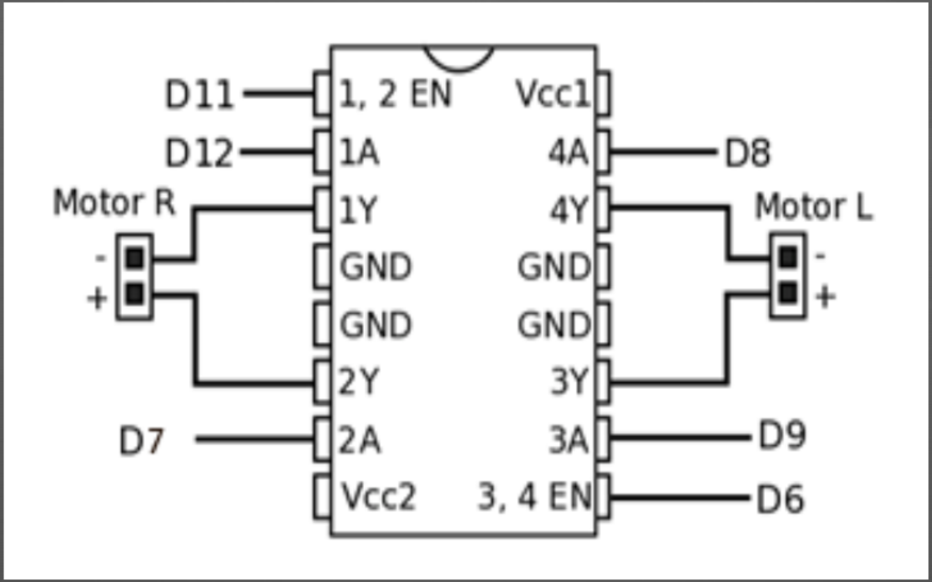
12En	1A	2A	Motor
0	X	X	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

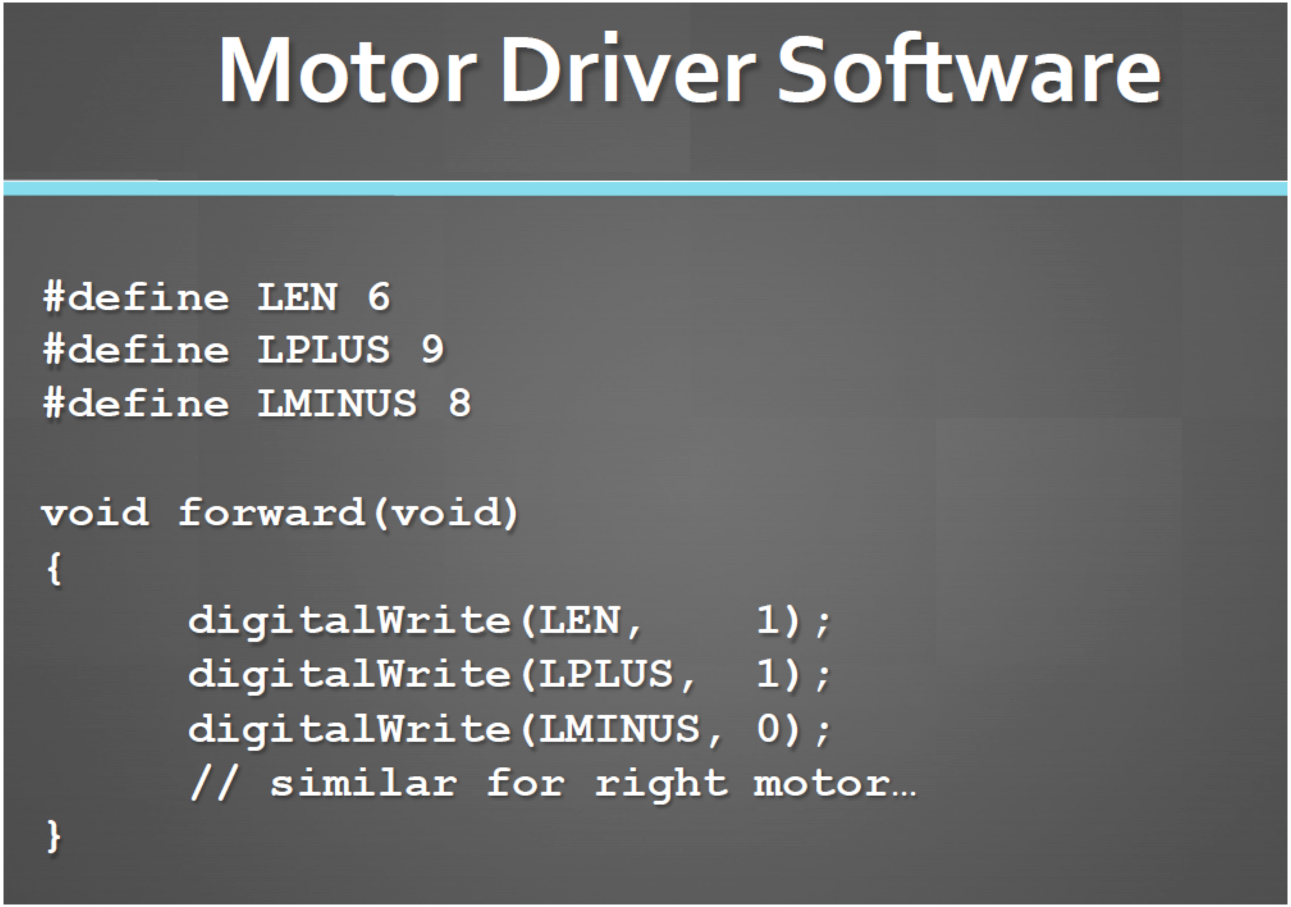




- Contains two H-Bridges to drive two motors

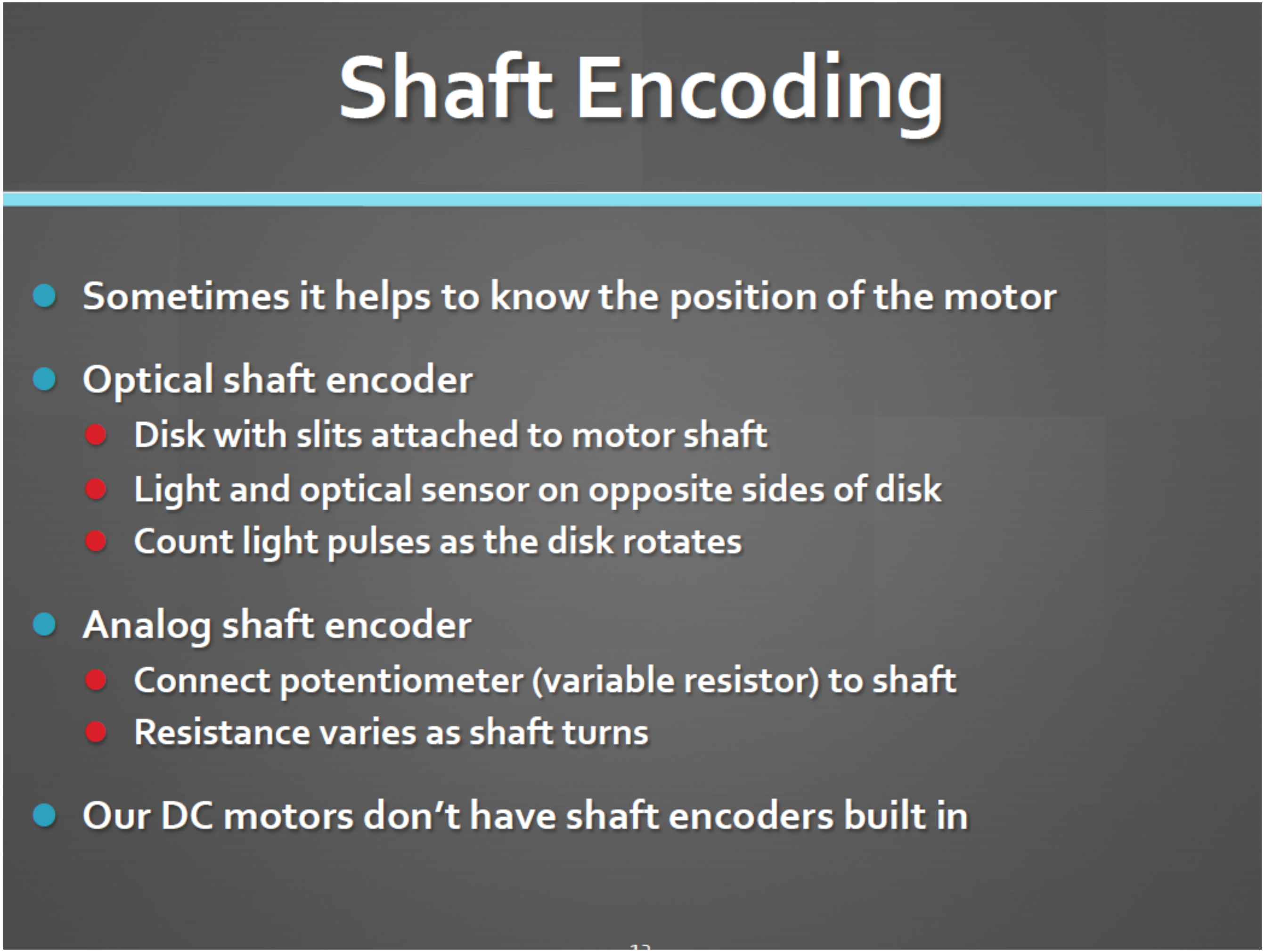
Mudduino H-Bridge Interface





Motor Driver Software

```
#define LEN 6
#define LPLUS 9
#define LMINUS 8

void forward(void)
{
      digitalWrite(LEN,    1);
      digitalWrite(LPLUS,  1);
      digitalWrite(LMINUS, 0);
      // similar for right motor…
}
```

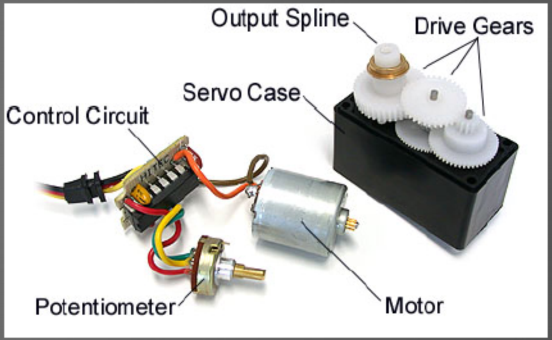


Shaft Encoding

- Sometimes it helps to know the position of the motor
- Optical shaft encoder
  - Disk with slits attached to motor shaft
  - Light and optical sensor on opposite sides of disk
  - Count light pulses as the disk rotates
- Analog shaft encoder
  - Connect potentiometer (variable resistor) to shaft
  - Resistance varies as shaft turns
- Our DC motors don’t have shaft encoders built in

Servo Motor

- Servo motors are designed to be easy to use
  - DC motor
  - Gearing
  - Analog shaft encoder
  - Control circuitry
  - High-current driver

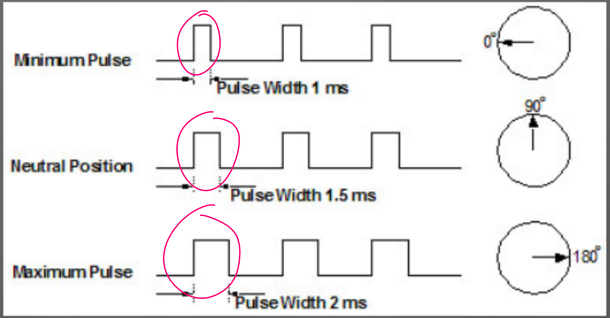


servocity.com

- Three wires: 5V, GND, Control
- Turn from 0 to 180 degrees
  - Position determined by pulses on control wire

Servo Pulse Width Modulation

- Control position with 50 Hz (20 ms) pulses
- Pulse width modulation (PWM)
  - 1 ms = 0°
  - 1.5 ms = 90°
  - 2 ms = 180°

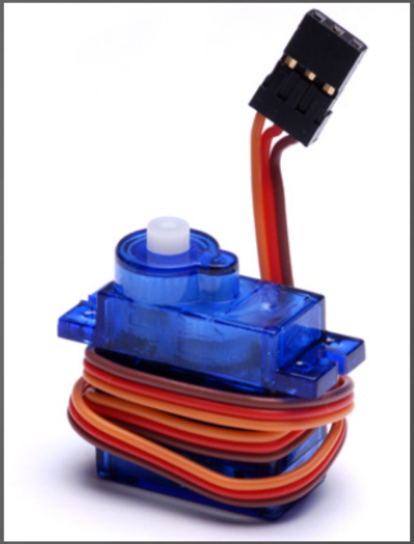


servocity.com

* مساحة ال Pulses
تختلف

SG90 Servo

- 4.0 – 7.2 V Operation
- At 4.8 V
  - Speed: 0.12 sec / 60 degrees (83 RPM)
  - Stall Torque: 16.7 oz-in

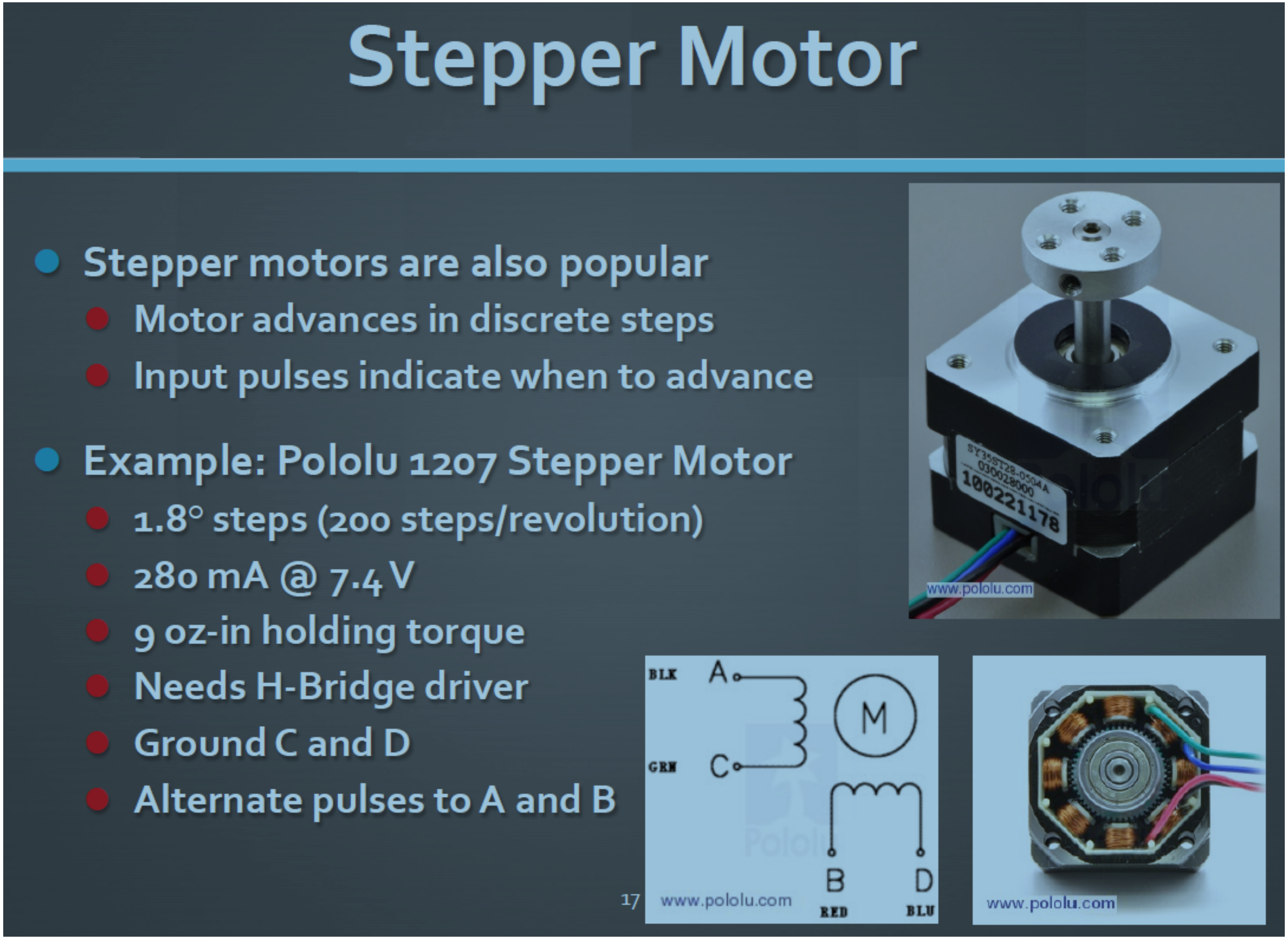


hobbypartz.com

Arduino Servo Library

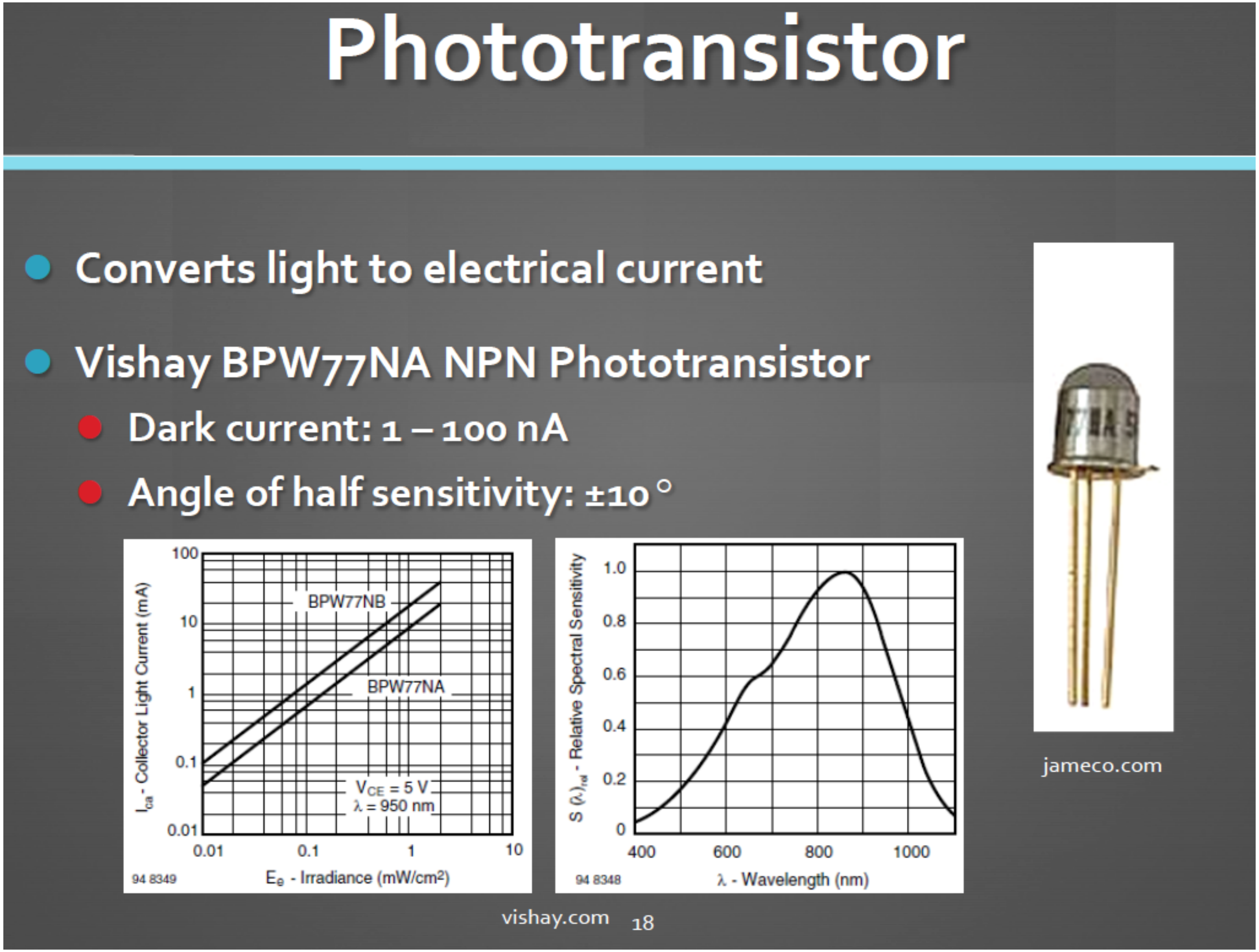
- Arduino offers a servo library for controlling servos

```
// servotest.pde
// David_Harris@hmc.edu 1 October 2011

#include <Servo.h>

// pins
#define SERVOPIN 10

// Global variable for the servo information
Servo servo;

void testServo()
{
  initServo();
  servo.write(90);  // set angle between 0 and 180 degrees
}

void initServo()
{
  pinMode(SERVOPIN, OUTPUT);
  servo.attach(SERVOPIN);
}
```



Stepper Motor

- Stepper motors are also popular
  - Motor advances in discrete steps
  - Input pulses indicate when to advance
- Example: Pololu 1207 Stepper Motor
  - 1.8° steps (200 steps/revolution)
  - 280 mA @ 7.4 V
  - 9 oz-in holding torque
  - Needs H-Bridge driver
  - Ground C and D
  - Alternate pulses to A and B



Phototransistor

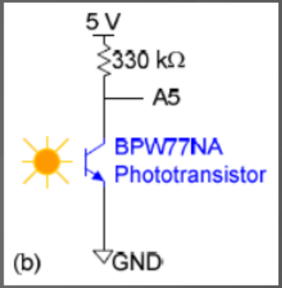
- Converts light to electrical current
- Vishay BPW77NA NPN Phototransistor
  - Dark current: 1 – 100 nA
  - Angle of half sensitivity: ±10 °

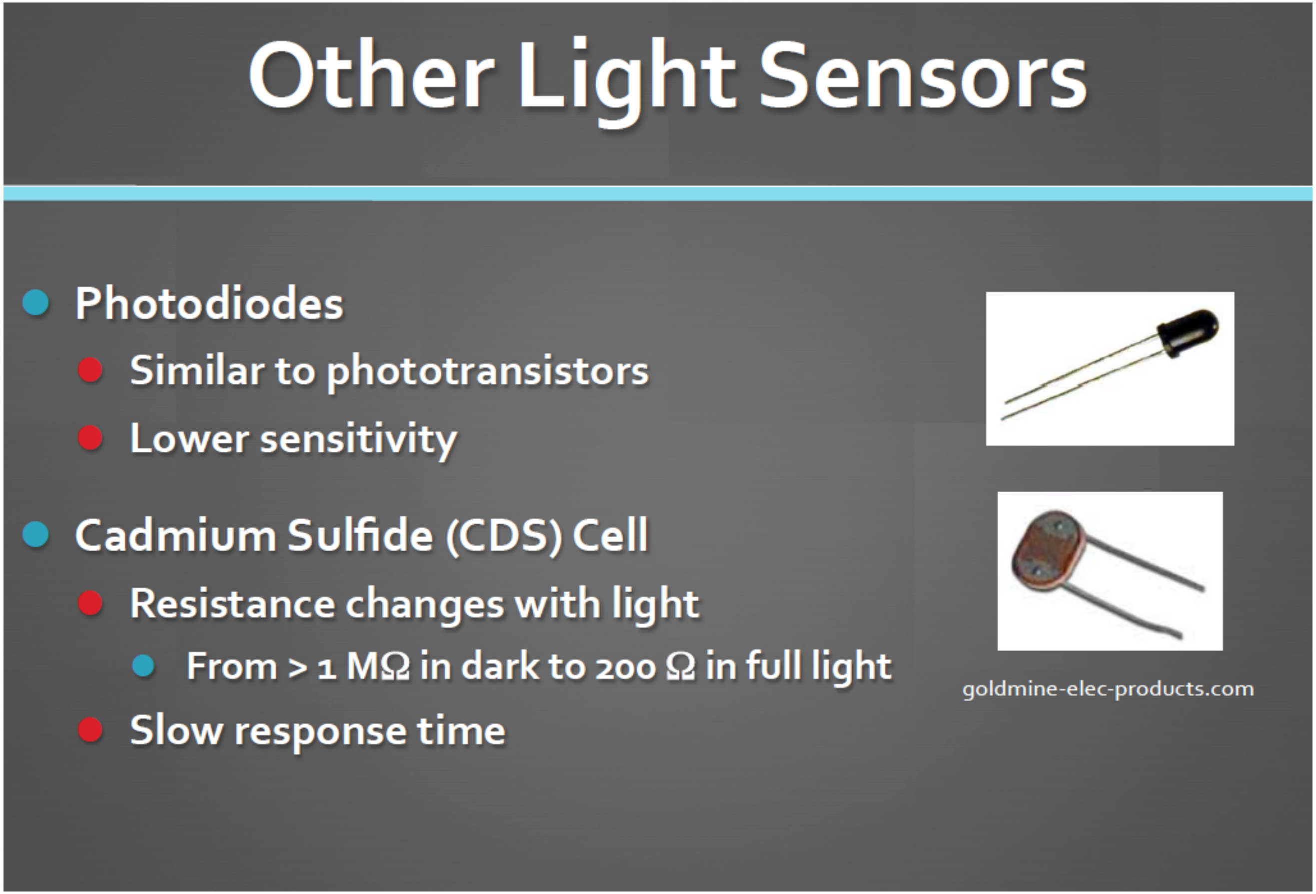
jameco.com

vishay.com

Phototransistor Circuit

- Leave base terminal unconnected
- $V_{out} = 5 - I_{photo} \times 330\ k\Omega$
  - In dark, $V_{out} \approx 5$ V
  - For $I_{photo} > 15\ \mu A$, V_{out} drops to ~0
- Large resistor gives sensitivity to weak light

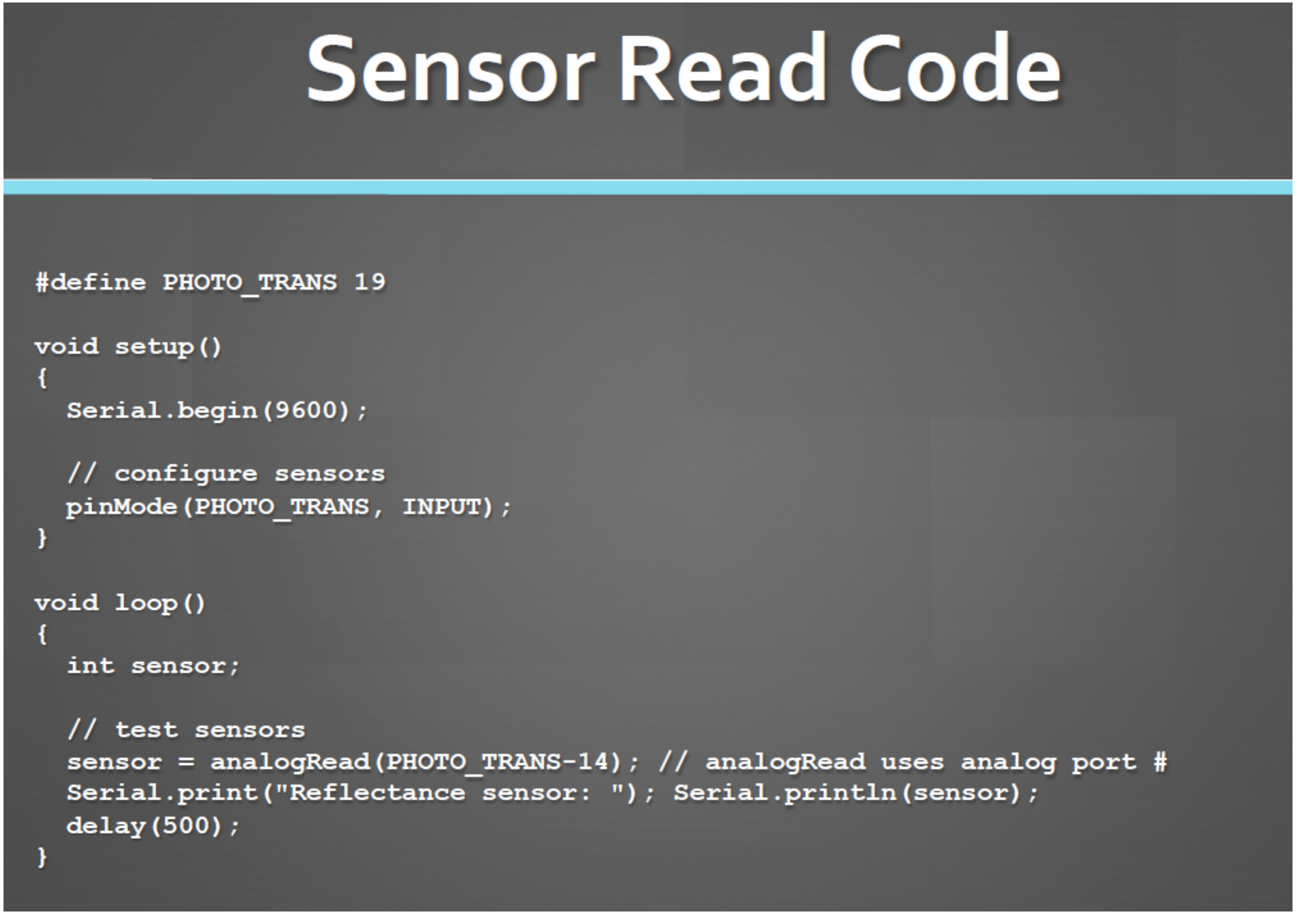




Other Light Sensors

- Photodiodes
  - Similar to phototransistors
  - Lower sensitivity
- Cadmium Sulfide (CDS) Cell
  - Resistance changes with light
    - From > 1 MΩ in dark to 200 Ω in full light
  - Slow response time

goldmine-elec-products.com



Sensor Read Code

```
#define PHOTO_TRANS 19

void setup()
{
  Serial.begin(9600);

  // configure sensors
  pinMode(PHOTO_TRANS, INPUT);
}

void loop()
{
  int sensor;

  // test sensors
  sensor = analogRead(PHOTO_TRANS-14); // analogRead uses analog port #
  Serial.print("Reflectance sensor: "); Serial.println(sensor);
  delay(500);
}
```

Sensor Averaging

- Sensors are subject to noise
- Average multiple readings for more stable results

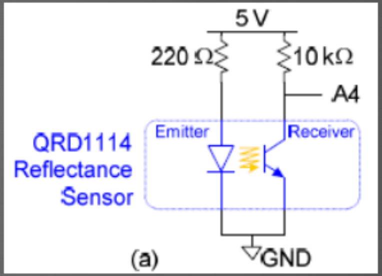
Reflectance Sensor

- Infrared LED and phototransistor pair
  - LED illuminates surface
  - Phototransistor receives reflected light
  - Daylight filter on sensor reduces interference
  - Sensitive to distance, color, reflectivity
- Fairchild QRD1114 Reflectance Sensor
  - ~20 mA LED current
  - 1.7 V LED ON voltage
  - 940 nm wavelength (near infrared)

fairchild.com

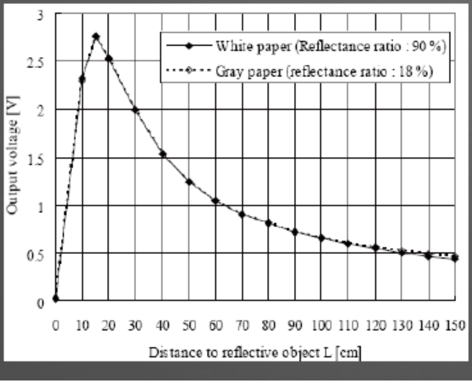
Reflectance Sensor Circuit

- I_{LED} = (5-1.7 V) / 220 Ω = 15 mA
- $V_{out} = 5 - I_{photo} \times 10\ k\Omega$
- Resistor was selected to give a good range of response



IR Distance Sensor

- Sharp GP2Y0A21YK0F
- Range of 8 to 60”
- Triangulates with linear CCD array
- Three terminals: 5V, GND, Signal



25

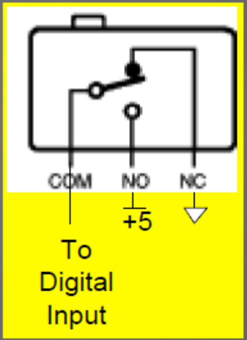
Ultrasonic Distance Sensor

- Measure flight time of ultrasonic pulse
  - Less sensitive to ambient light
  - More precise
  - More expensive
- Example: LV-MaxSonar-EZ
  - 42 KHz ultrasonic beam
  - Range of 254" with resolution of 1"
  - 2.5 – 5.5 V operation
  - Analog voltage output

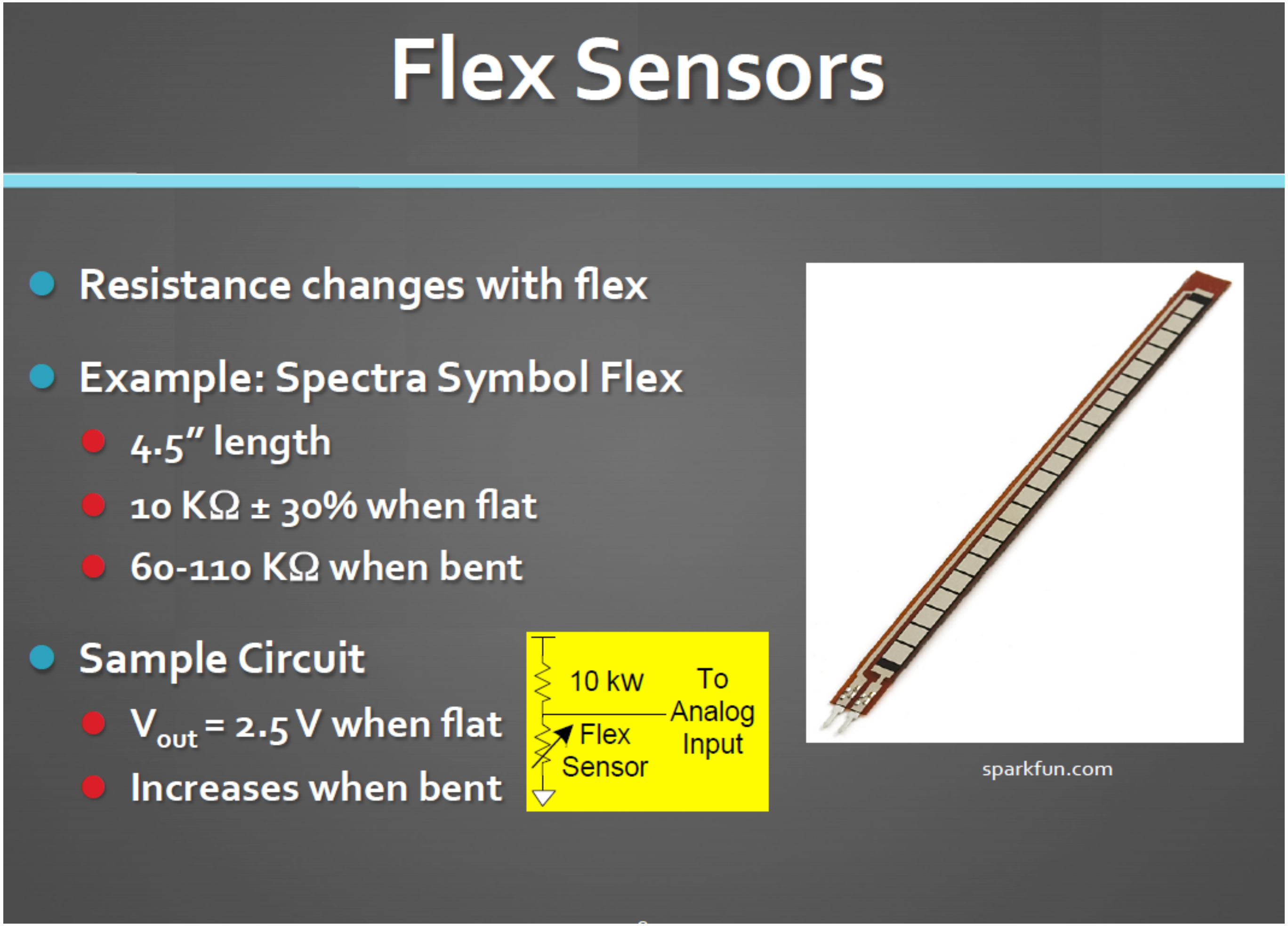
maxbotix.com

Switches

- Switches are useful for proximity detection
- Three terminals
  - COM: Common
  - NO: Normally Open
  - NC: Normally Closed
- Mounting issues
  - Good supporting surface
  - Gang 2 or more with plate between



sparkfun.com



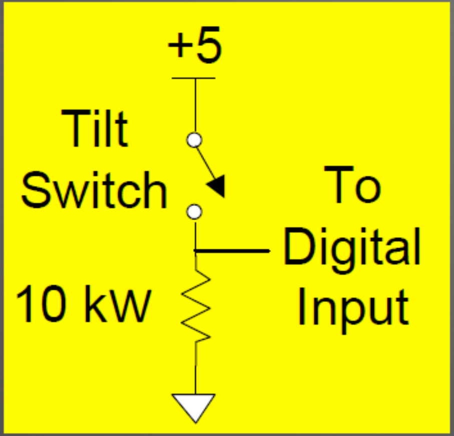
Flex Sensors

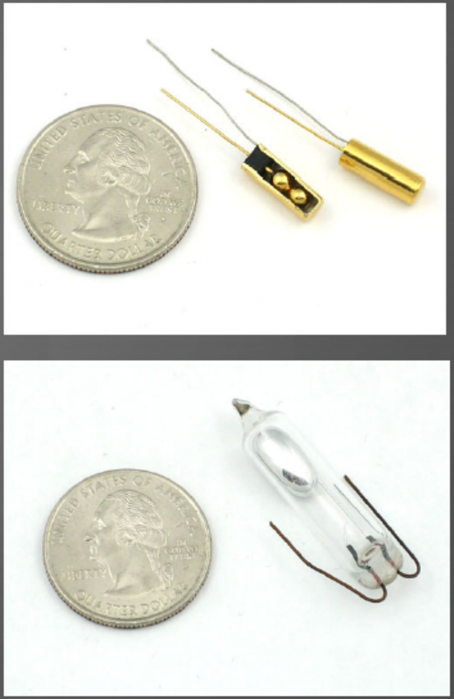
- Resistance changes with flex
- Example: Spectra Symbol Flex
  - 4.5” length
  - 10 KΩ ± 30% when flat
  - 60-110 KΩ when bent
- Sample Circuit
  - V_{out} = 2.5 V when flat
  - Increases when bent

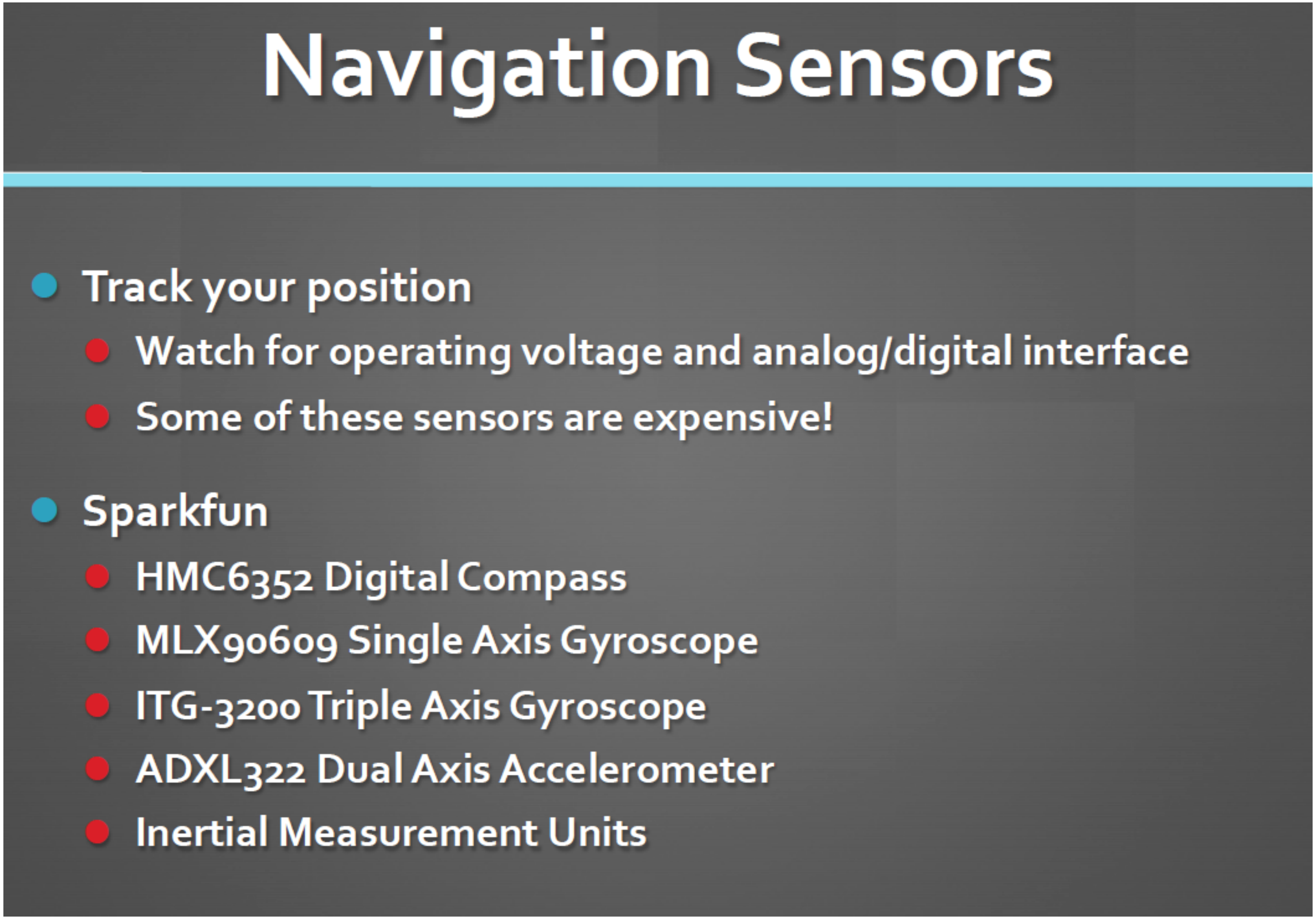
sparkfun.com

Tilt Switches

- Mercury or Ball
- Warn if your bot is about to topple!

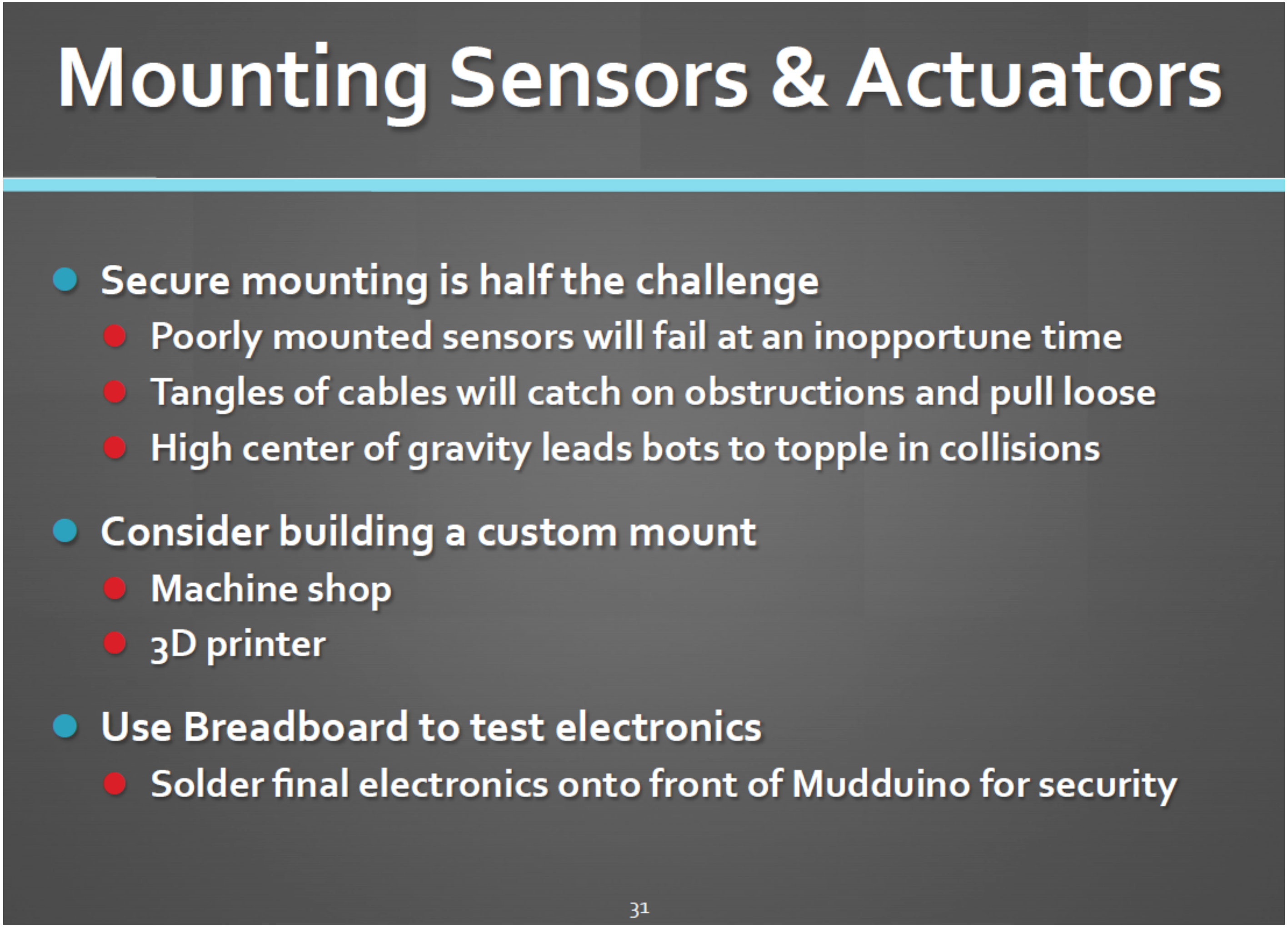






Navigation Sensors

- Track your position
  - Watch for operating voltage and analog/digital interface
  - Some of these sensors are expensive!
- Sparkfun
  - HMC6352 Digital Compass
  - MLX90609 Single Axis Gyroscope
  - ITG-3200 Triple Axis Gyroscope
  - ADXL322 Dual Axis Accelerometer
  - Inertial Measurement Units



Mounting Sensors & Actuators

- Secure mounting is half the challenge
  - Poorly mounted sensors will fail at an inopportune time
  - Tangles of cables will catch on obstructions and pull loose
  - High center of gravity leads bots to topple in collisions
- Consider building a custom mount
  - Machine shop
  - 3D printer
- Use Breadboard to test electronics
  - Solder final electronics onto front of Mudduino for security

Adhesives

- Cynoacrylate (CA) Glue (aka Super Glue)
  - Fast drying, good for bonding plastic
  - Low shear strength
  - Don’t bond your fingers – wear gloves
- Hot Glue
- Electrical Tape
  - Insulator, low strength
- Gaffer’s Tape
  - Like duct tape, but stronger and removes cleanly

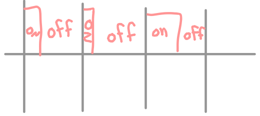
من (DC motor) إلى هنا المادة التي ليست داخلة بالفاينل

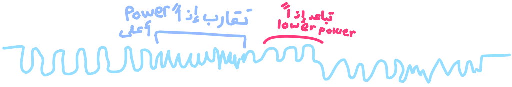
PID

Lecture Outline

- ☐ Introduction to PID
- ☐ Modes of Control
  - ☐ On-Off Control
  - ☐ Proportional Control
  - ☐ Proportional + Integral Control
  - ☐ Proportional + Derivative Control
  - ☐ Proportional + Integral + Derivative Control
- ☐ PID Tuning Rules
  - ☐ Zeigler-Nichol's Tuning Rules
    - ☐ 1st Method
    - ☐ 2nd Method

Introduction





* السرعة تتناسب مع الـfrequency

* في الـ Setting يجب الانتباه على الـ Quality

* setting ارتفاع الحرارة يكون على شكل (exponential)
Not Linear

في الـ control اذا وضعنا الـSetting غلط لن نستفيد

* التداخل بين الـ Design ← setting ← control ← يجعل جميعهم مهمين حتى نحقق الهدف بشكل عام

- PID Stands for

① نستعمله في البداية
و هو بعطيني سرعة النهوض
— P → Proportional (K)

* مقدار الـ Power يلي داخله لحالة
Power = constant x Error
طردي مع الPower

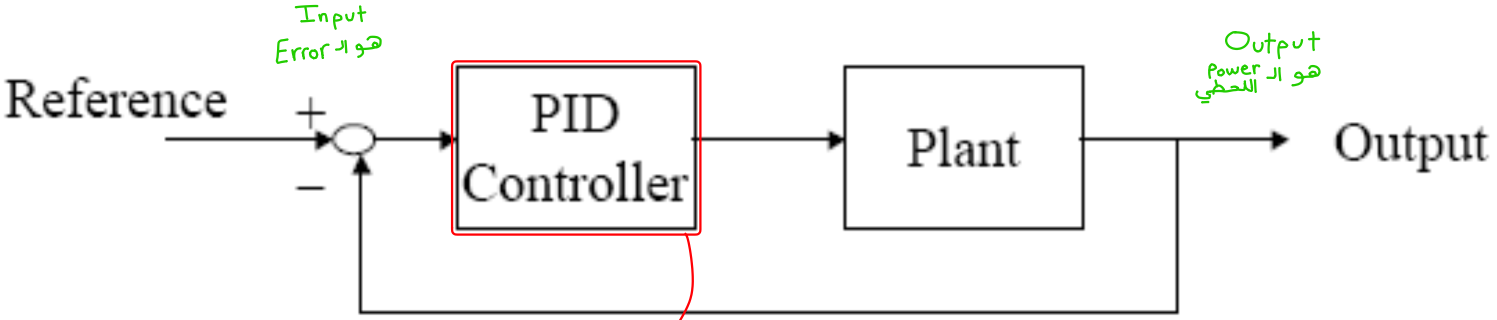
③ نستعمله آخر اشي في الـsteady state (آخر مرحلة)
— I → Integral
دوره يكون في مرحلة الـ steady state (أخر فترة)

* لو كان لحالة يكون :-
constant x Error المساحة تحت منحنى الـ
وهو ضعيف
في اول الفترة الزمنية يكون صفر أو قريب من الصفر

② يقلل الـ overshoot
(معاكس للتغير)
(حرف الـ S في المقام يكافئ تماماً اضافة damper بالتالي يزداد الـdamping Ratio و يقلل الـ over shoot)
— D → Derivative
(Fictual damper)

يكون مقدار الـPower معاكسة للتغير يلي بصير بالـ Error
و كل ما زاد التغير زادت المقاومة

* لما يكون عندي very high overshoot بلجأ لـ Fictual damper لتقليل الـ %OS



الهدف منه هو تنظيم دخول الطاقة
بحيث نوصل لدرجة الحرارة المطلوبة بأقل فترة زمنية (أقل T_R Rise time)
لكن يجب أن يكون هناك overshoot بسيط نوعاً ما
(اذا بدي good Rise time لازم يكون عندي slightly overshoot و إلا بكون الـ system ضعيف)

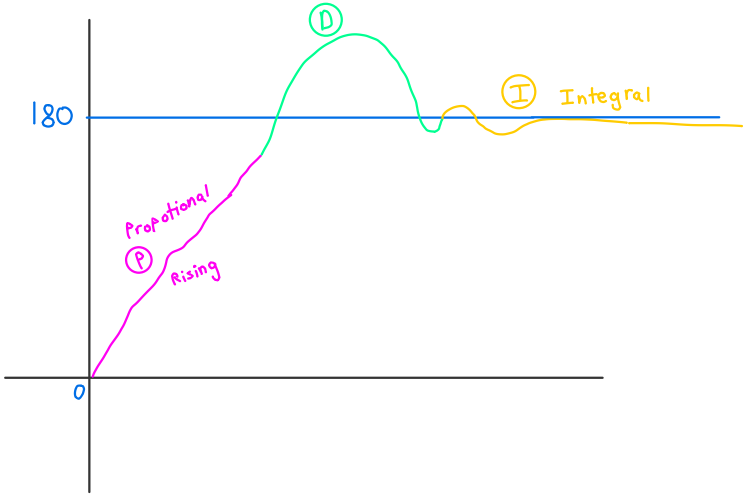
* إذاً // الـcontroler Pid بحاول يعطينا (الهدف) → Strong T_R, less %OS, less T_s

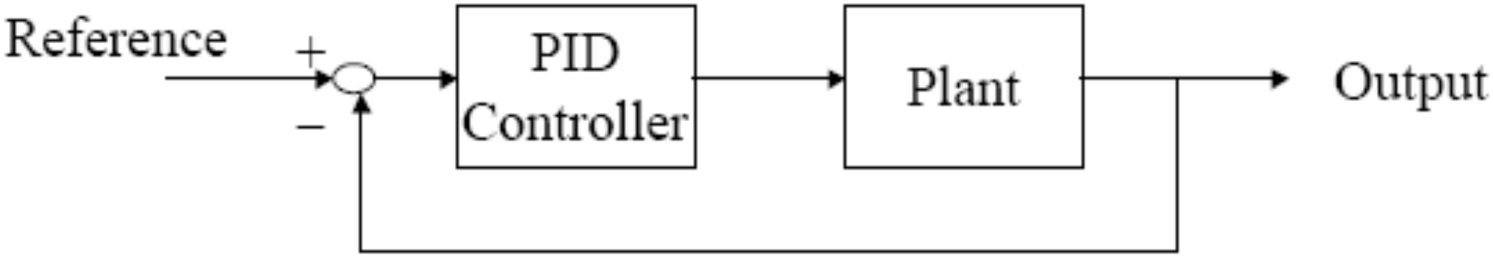
الـ Power تتغير بشكل لحظي

Introduction

- PID Stands for
  - P → Proportional
  - I → Integral
  - D → Derivative

(s)





Introduction

- The usefulness of PID controls lies in their general applicability to most control systems.

- In particular, when the mathematical model of the plant is not known and therefore analytical design methods cannot be used, PID controls prove to be most useful.

- In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control.

Inverter ⟷ Pid Controller

Introduction

- It is interesting to note that more than half of the industrial controllers in use today are PID controllers or modified PID controllers.
- Because most PID controllers are adjusted on-site, many different types of tuning rules have been proposed in the literature.
- Using these tuning rules, delicate and fine tuning of PID controllers can be made on-site.

* المشكلة في الـ Pid controller انه فيه ثلاث Parameters for Errors

K_p
K_I
K_D

← تم تطوير طريقة لحل هذه المشكلة و هي (Autotunning)

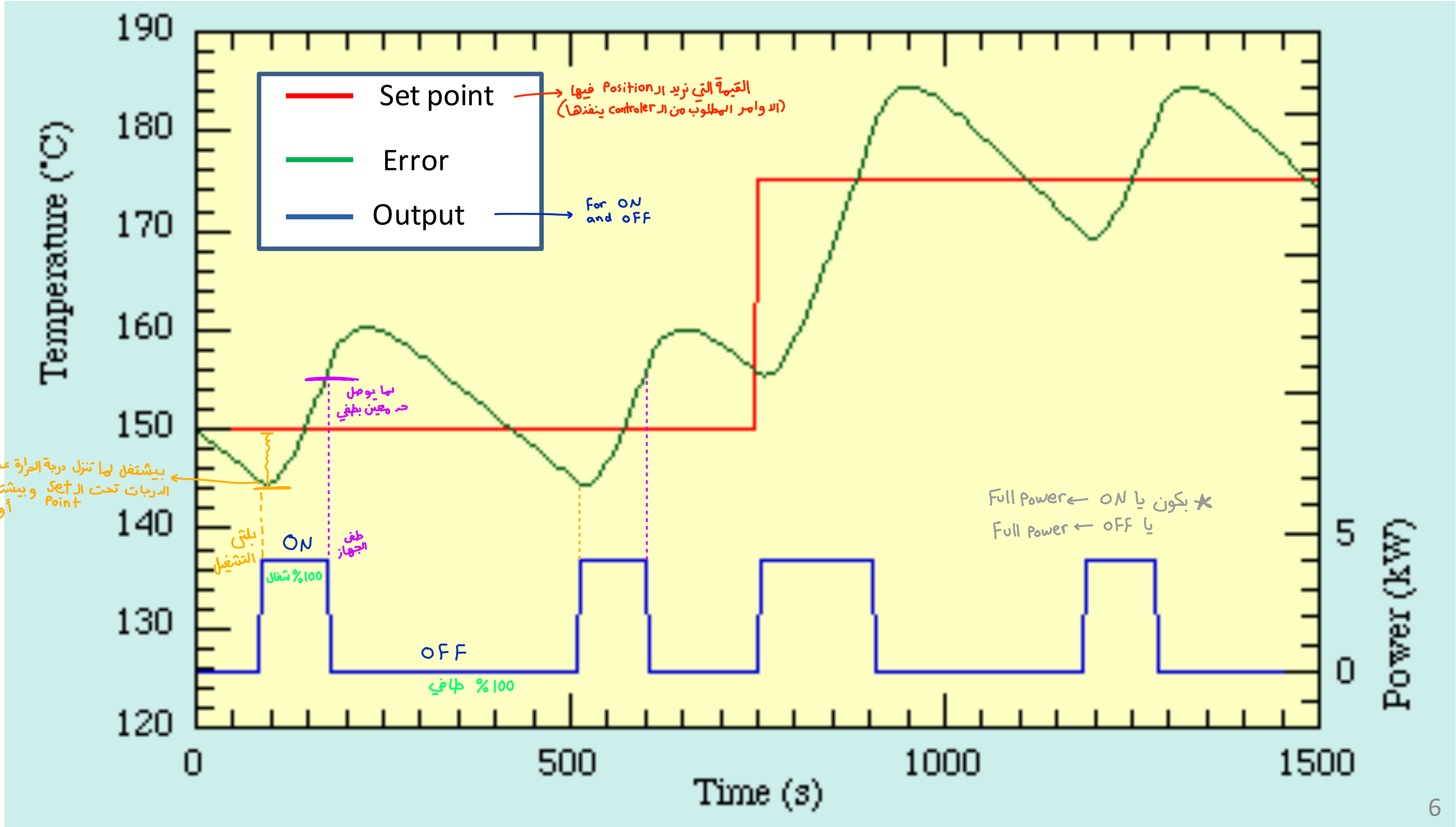
Four Modes of Controllers

- Each mode of control has specific advantages and limitations.

  - On-Off (Bang Bang) Control
  - Proportional (P)
  - Proportional plus Integral (PI)
  - Proportional plus Derivative (PD)
  - Proportional plus Integral plus Derivative (PID)

On-Off Control

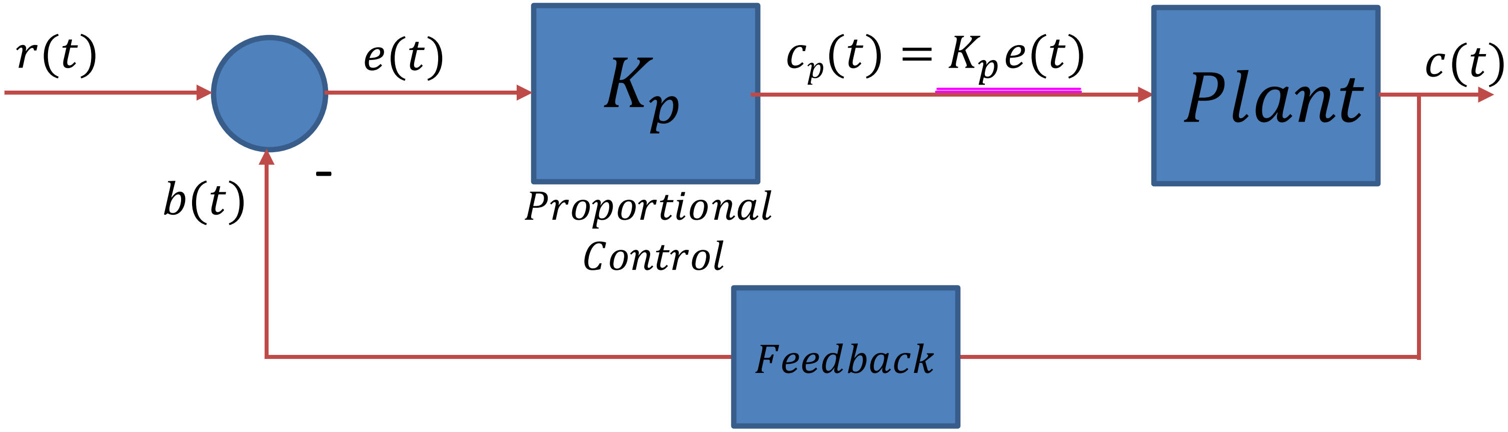
- This is the simplest form of control.



تكون مقدار ال Power يلي داخلة
تساوي constant x Error

Proportional Control (P)

- In *proportional mode*, there is a continuous linear relation between value of the controlled variable and position of the final control element.



- Output of proportional controller is

$$c_p(t) = K_p e(t)$$

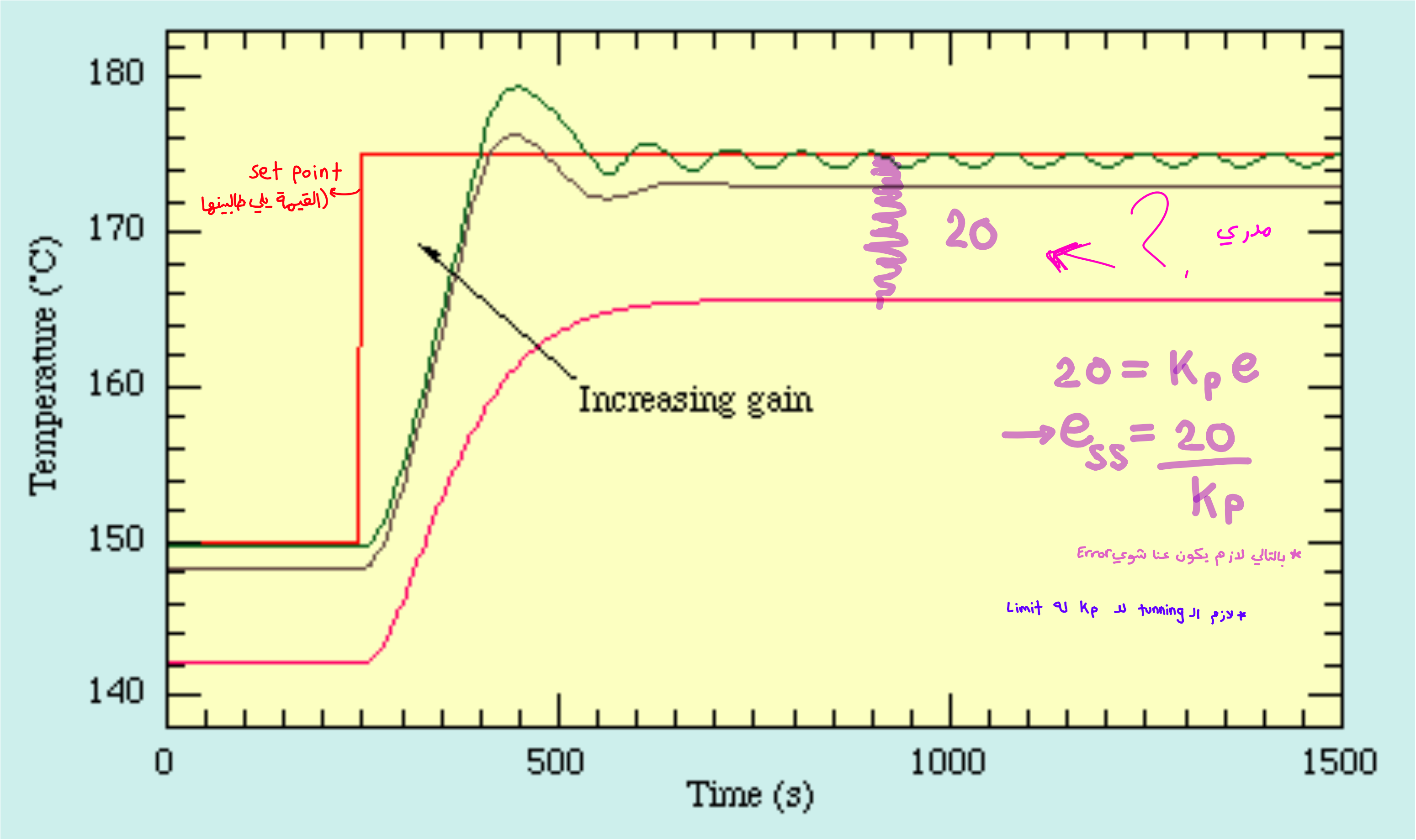
- The transfer function can be written as

$$\frac{C_p(s)}{E(s)} = K_p$$

Proportional Controllers (P)

- As the gain is increased the system responds faster to changes in set-point but becomes progressively underdamped and eventually unstable.

(Kp) are usually subjected to Steady State Error



set point (القيمة يلي طالبينها)

20

?

مدري

$$20 = K_p e$$

$$\rightarrow e_{ss} = \frac{20}{K_p}$$

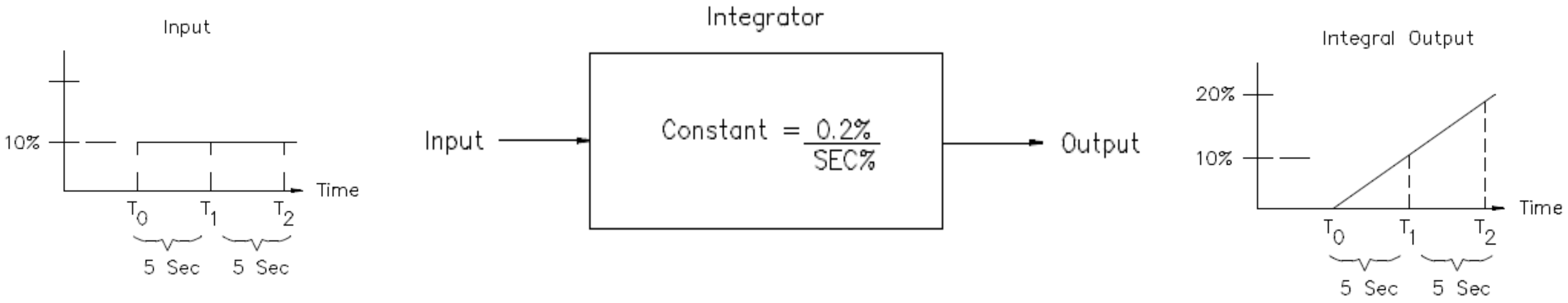
* بالتالي لازم يكون عنا شوي Error

* لازم ال tunning لل Kp ول Limit

Proportional Plus Integral Controllers (PI)

- Integral control describes a controller in which the output rate of change is dependent on the magnitude of the input.

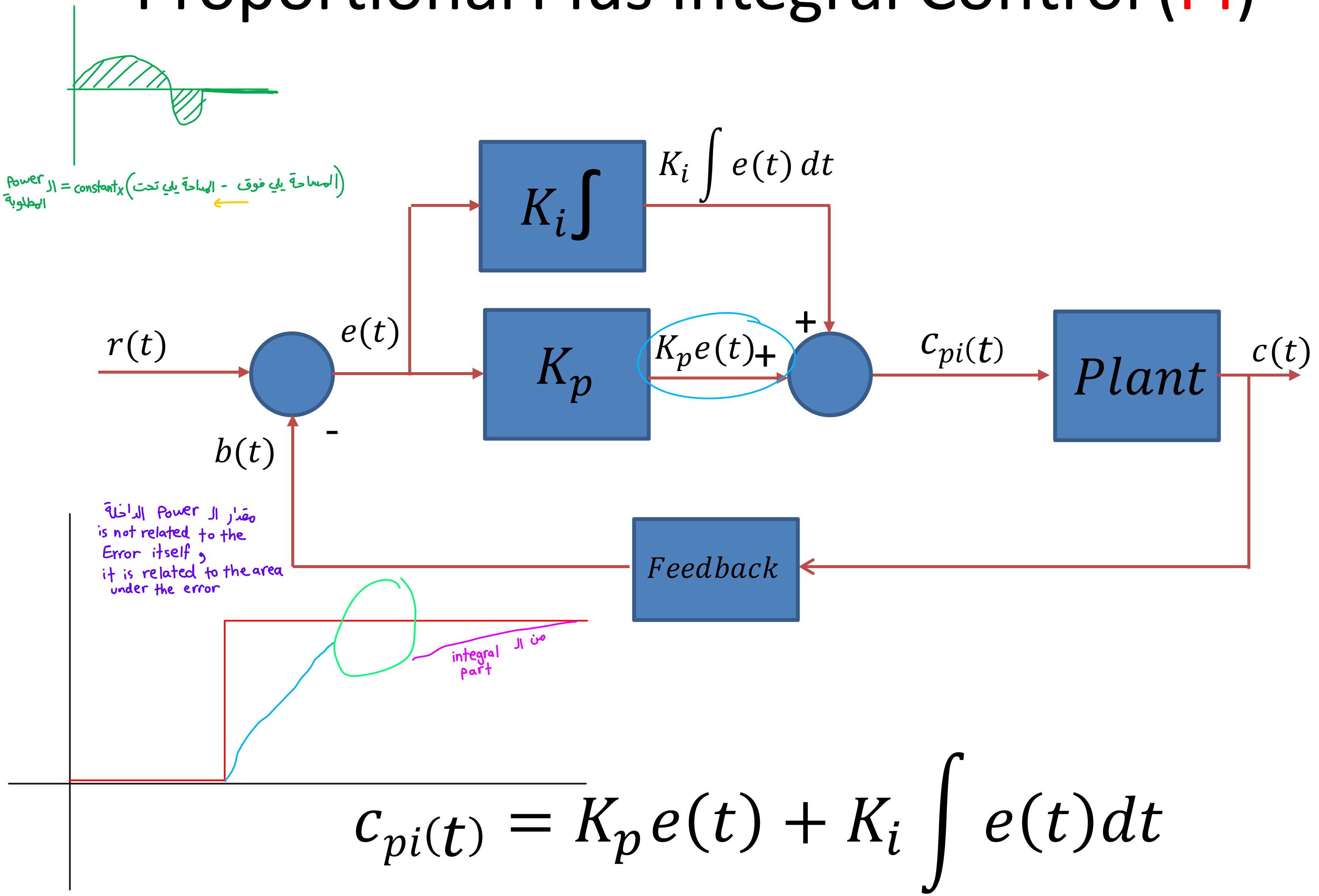
- Specifically, a smaller amplitude input causes a slower rate of change of the output.



Proportional Plus Integral Controllers (PI)

- The major advantage of integral controllers is that they have the unique ability to return the controlled variable back to the exact set point following a disturbance.
- Disadvantages of the integral control mode are that it responds relatively slowly to an error signal and that it can initially allow a large deviation at the instant the error is produced.
- This can lead to system instability and cyclic operation. For this reason, the integral control mode is not normally used alone, but is combined with another control mode.

Proportional Plus Integral Control (PI)



ال Power = constant x (المساحة يلي فوق - المساحة يلي تحت) المطلوبة

مقدار ال Power الداخلة is not related to the Error itself , it is related to the area under the error

$$c_{pi}(t) = K_pe(t) + K_i\int e(t)dt$$

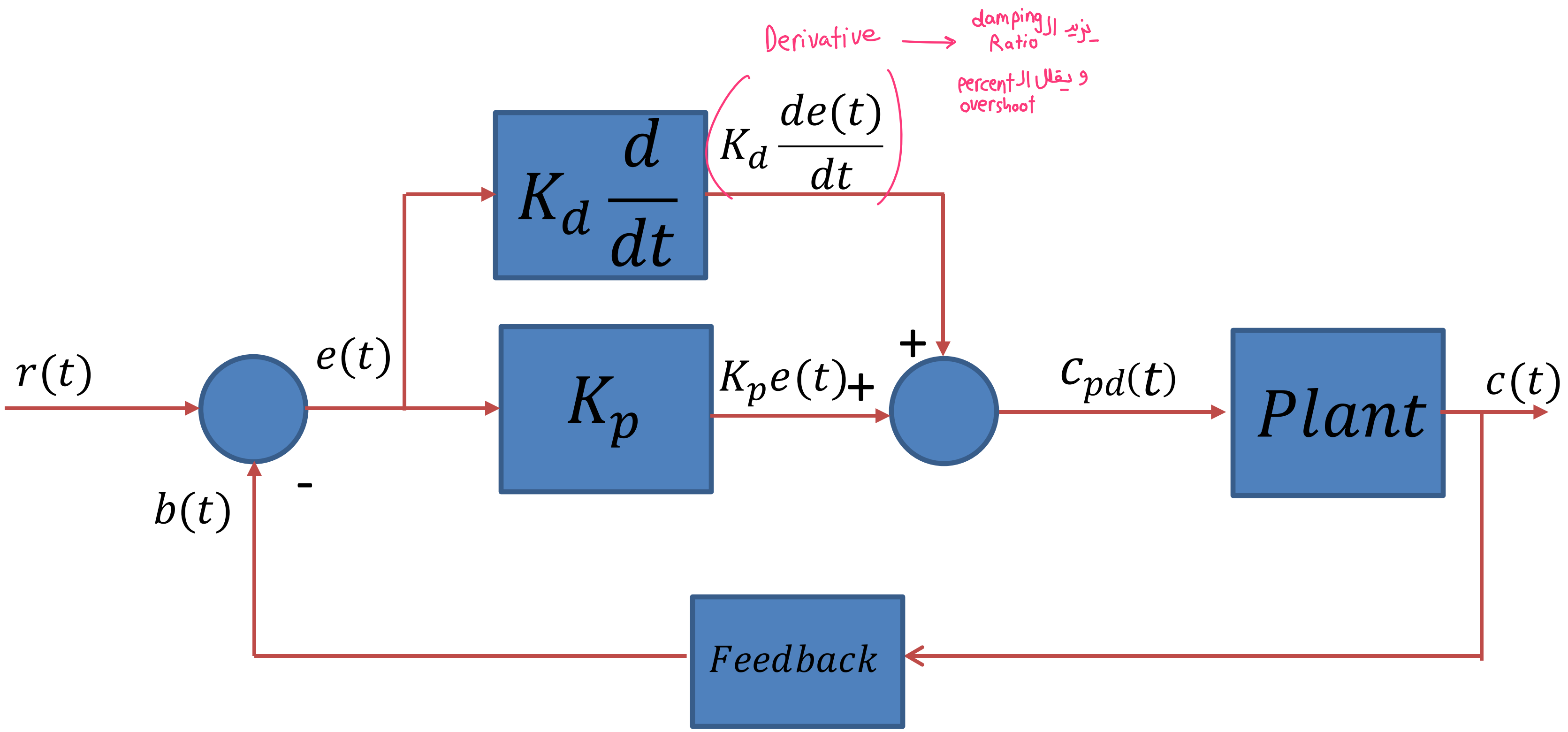
Proportional Plus Integral Control (PI)

$$c_{pi(t)} = K_p e(t) + K_i \int e(t)dt$$

- The transfer function can be written as

$$\frac{C_{pi}(s)}{E(s)} = K_p + K_i \frac{1}{s}$$

Proportional Plus derivative Control (PD)



Derivative → يزيد الـ damping Ratio

و يقلل الـ percent overshoot

$$c_{pd(t)} = K_p e(t) + K_d \frac{de(t)}{dt}$$

Proportional Plus derivative Control (PD)

$$c_{pd(t)} = K_p e(t) + K_d \frac{de(t)}{dt}$$

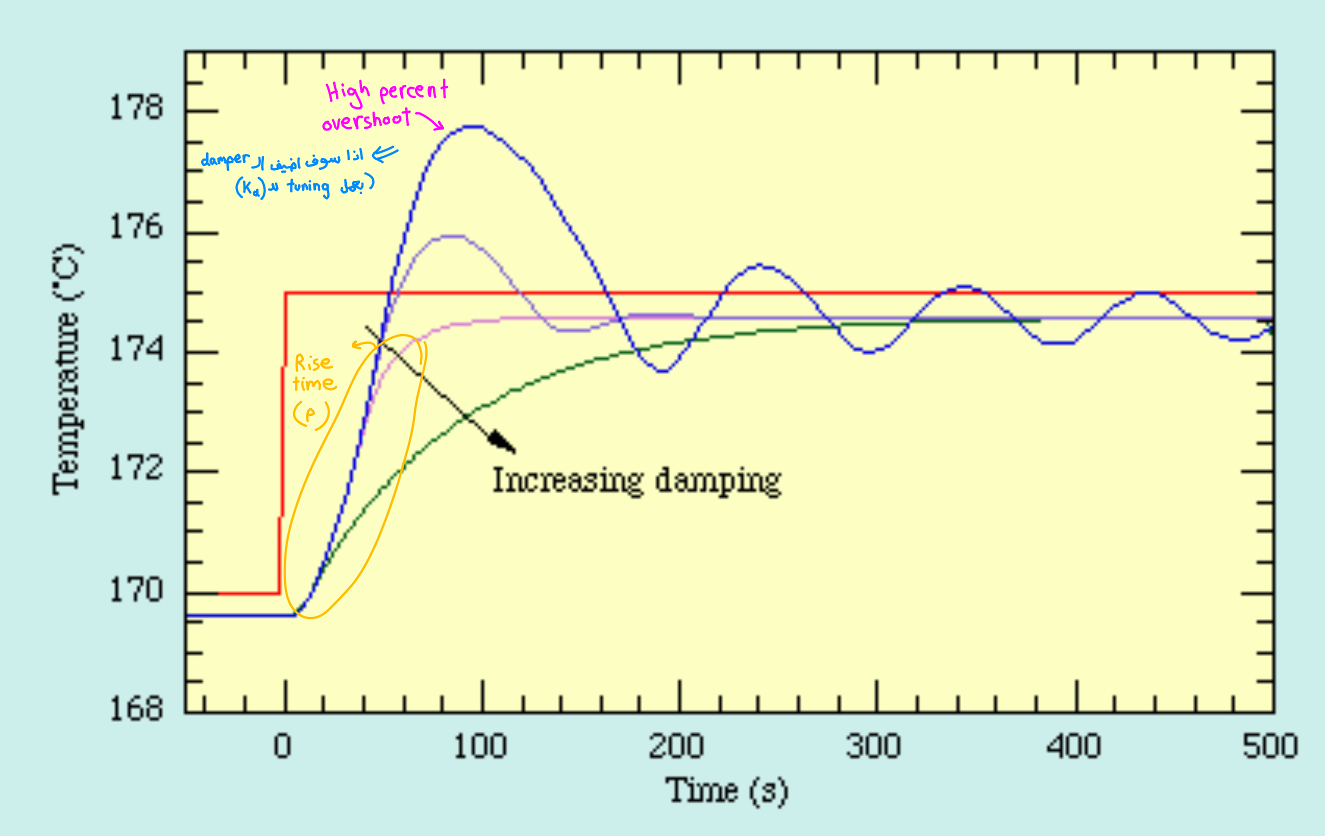
- The transfer function can be written as

$$\frac{C_{pd}(s)}{E(s)} = K_p + K_d s$$

Proportional Plus derivative Control (PD)

integral غير موجود

- The stability and overshoot problems that arise when a proportional controller is used at high gain can be mitigated by adding a term proportional to the time-derivative of the error signal. The value of the damping can be adjusted to achieve a critically damped response.

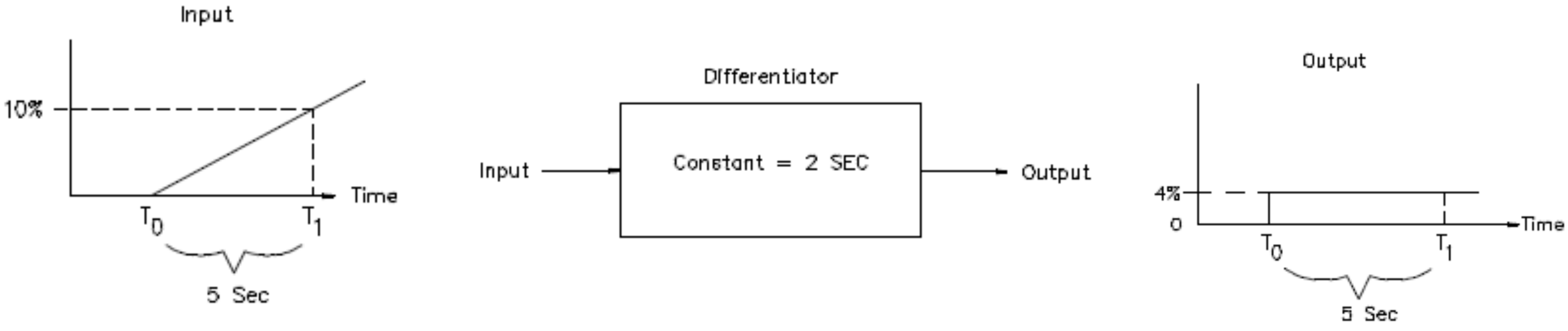


Proportional Plus derivative Control (PD)

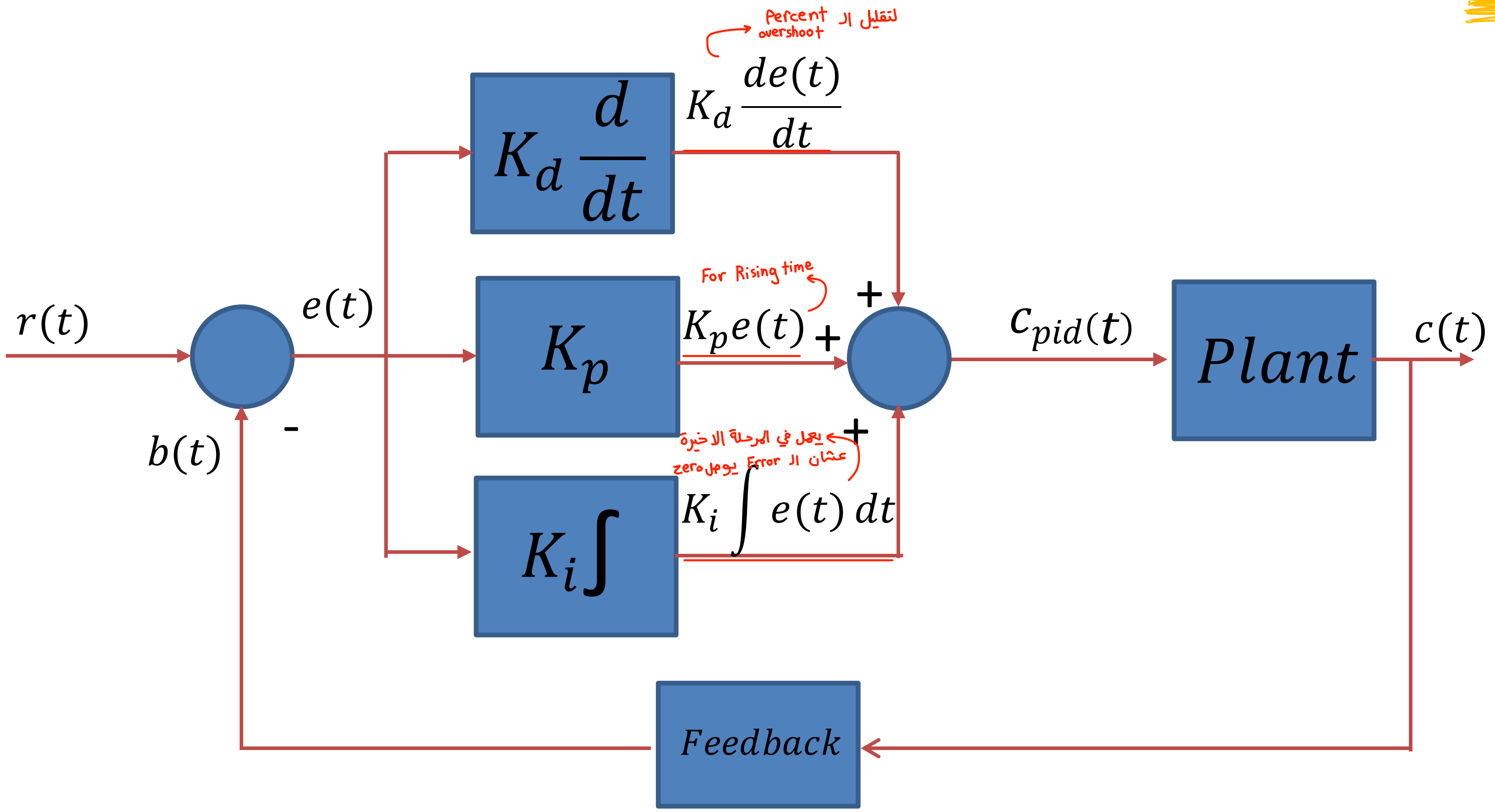
- The higher the error signal rate of change, the sooner the final control element is positioned to the desired value.

- The added derivative action reduces initial overshoot of the measured variable, and therefore aids in stabilizing the process sooner.

- This control mode is called proportional plus derivative (PD) control because the derivative section responds to the rate of change of the error signal



Proportional Plus Integral Plus Derivative Control (PID)



$$c_{pid}(t) = K_p e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt}$$

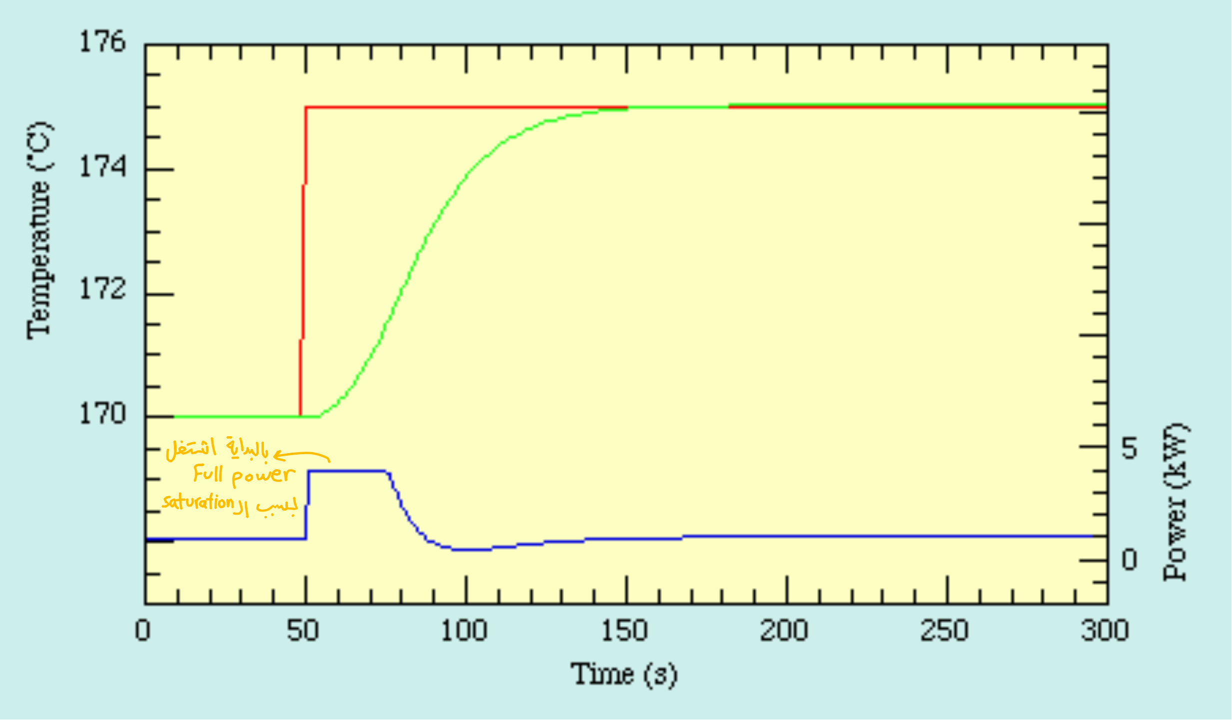
Proportional Plus Integral Plus Derivative Control (PID)

$$c_{pid(t)} = K_p e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt}$$

$$\frac{C_{pid}(s)}{E(s)} = K_p + K_i \frac{1}{s} + K_d\, s$$

Proportional Plus Integral Plus Derivative Control (PID)

- Although PD control deals neatly with the overshoot and ringing problems associated with proportional control it does not cure the problem with the steady-state error. Fortunately it is possible to eliminate this while using relatively low gain by adding an integral term to the control function which becomes



بالبداية اشتغل
Full power
بسبب ال saturation

The Characteristics of P, I, and D controllers

Summary

يفضل الحفظ !!

	CL RESPONSE	RISE TIME T_R	OVERSHOOT OS%	SETTLING TIME T_s	S-S ERROR e_{ss}
P	Kp ↑	Decrease ↓	Increase ↑	Small Change	Decrease ↓
I	Ki	Decrease ↓	Increase ↑	Increase ↑	Eliminate 0
D	Kd	Small Change	Decrease ↓	Decrease ↓	Small Change

Tips for Designing a PID Controller

1. Obtain an open-loop response and determine what needs to be improved
2. Add a proportional control to improve the rise time * نضل نزيد ال K_p لحد ما يصير ال Rise time مقبول
3. Add a derivative control to improve the overshoot * نضيف ال K_d لحد ما يصير ال over shoot مقبول
4. Add an integral control to eliminate the steady-state error
5. Adjust each of K_p, K_i, and K_d until you obtain a desired overall response.

- Lastly, please keep in mind that you do not need to implement all three controllers (proportional, derivative, and integral) into a single system, if not necessary. For example, if a PI controller gives a good enough response (like the above example), then you don't need to implement derivative controller to the system. Keep the controller as simple as possible.

Auto Tunning

Part-II

PID TUNING RULES

PID Tuning

- The transfer function of PID controller is given as

$$\frac{C_{pid}(s)}{E(s)} = K_p + K_i \frac{1}{s} + K_d\, s$$

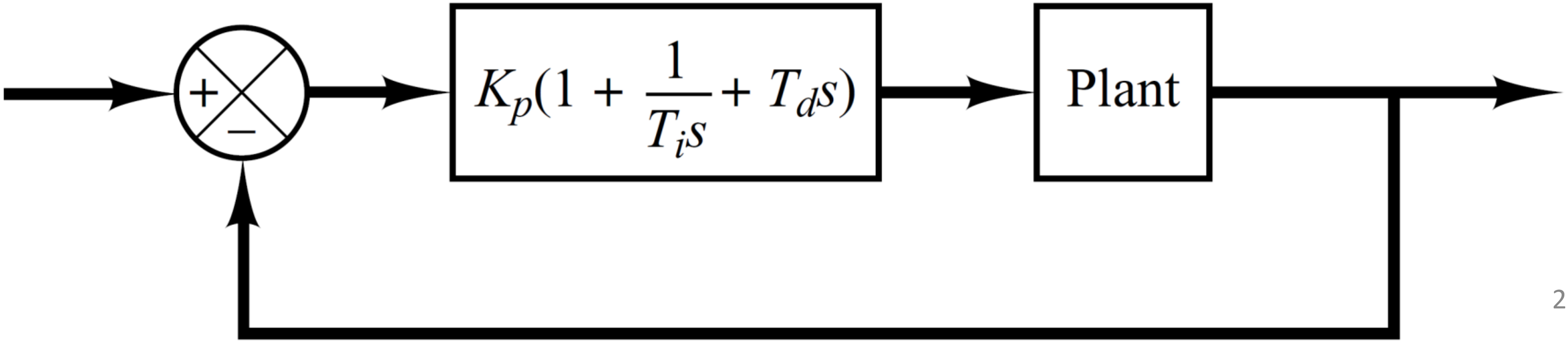
- It can be simplified as

$$\boxed{\frac{C_{pid}(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d\, s\right)}$$

← هذا الشكل
مشهورأكتر
من الفوق

- Where

$$\boxed{T_i = \frac{K_p}{K_i}} \qquad \boxed{T_d = \frac{K_d}{K_p}}$$



PID Tuning

- ✓ The process of selecting the controller parameters (K_p, T_i and T_d) to meet given performance specifications is known as controller tuning.

- Ziegler and Nichols suggested rules for tuning PID controllers experimentally.

- Which are useful when mathematical models of plants are not known.

- These rules can, of course, be applied to the design of systems with known mathematical models.

PID Tuning

- Such rules suggest a set of values of K_p, T_i and T_d that will give a stable operation of the system.

- However, the resulting system may exhibit a large maximum overshoot in the step response, which is unacceptable.

- In such a case we need series of fine tunings until an acceptable result is obtained.

- In fact, the Ziegler–Nichols tuning rules give an educated guess for the parameter values and provide a starting point for fine tuning, rather than giving the final settings for K_p, T_i and T_d in a single shot.

Zeigler-Nichol's PID Tuning Methods

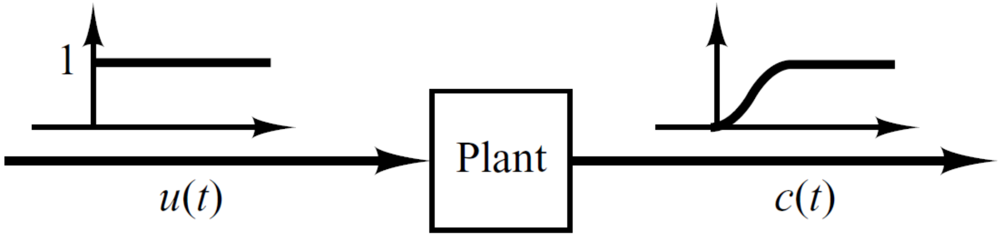
- Ziegler and Nichols proposed rules for determining values of the K_p, T_i and T_d based on the transient response characteristics of a given plant.

- Such determination of the parameters of PID controllers or tuning of PID controllers can be made by engineers on-site by experiments on the plant.

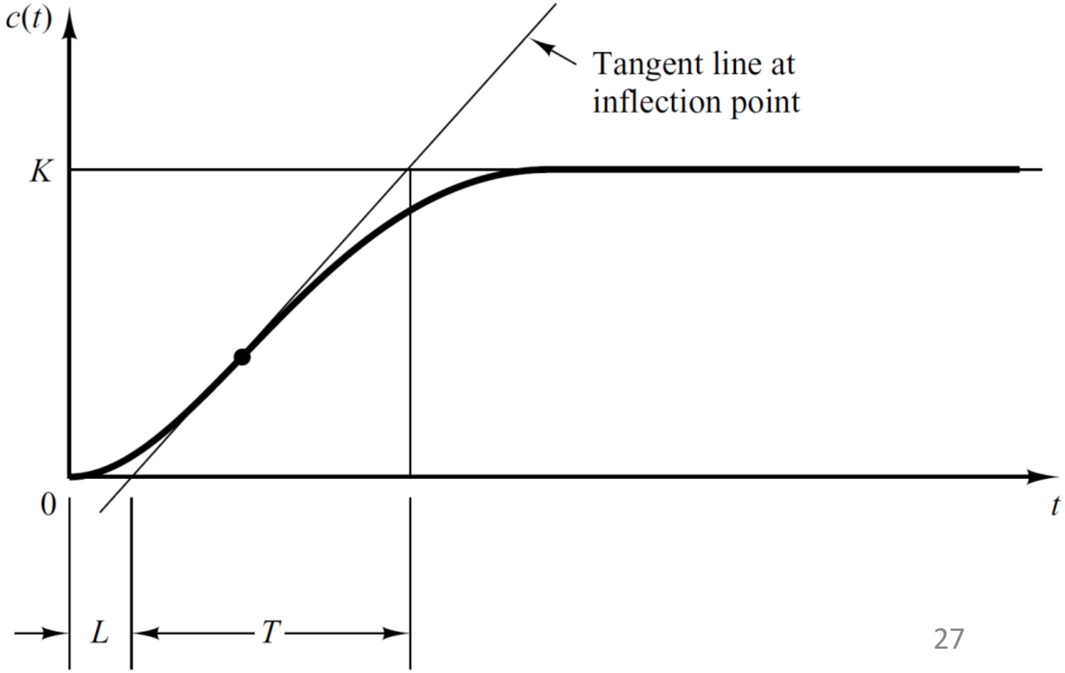
- There are two methods called Ziegler–Nichols tuning rules:
  - First method (open loop Method)
  - Second method (Closed Loop Method)

Zeigler-Nichol’s First Method

- In the first method, we obtain experimentally the response of the plant to a unit-step input.



- If the plant involves neither integrator(s) nor dominant complex-conjugate poles, then such a unit-step response curve may look S-shaped



Zeigler-Nichol's First Method

- This method applies if the response to a step input exhibits an S-shaped curve.

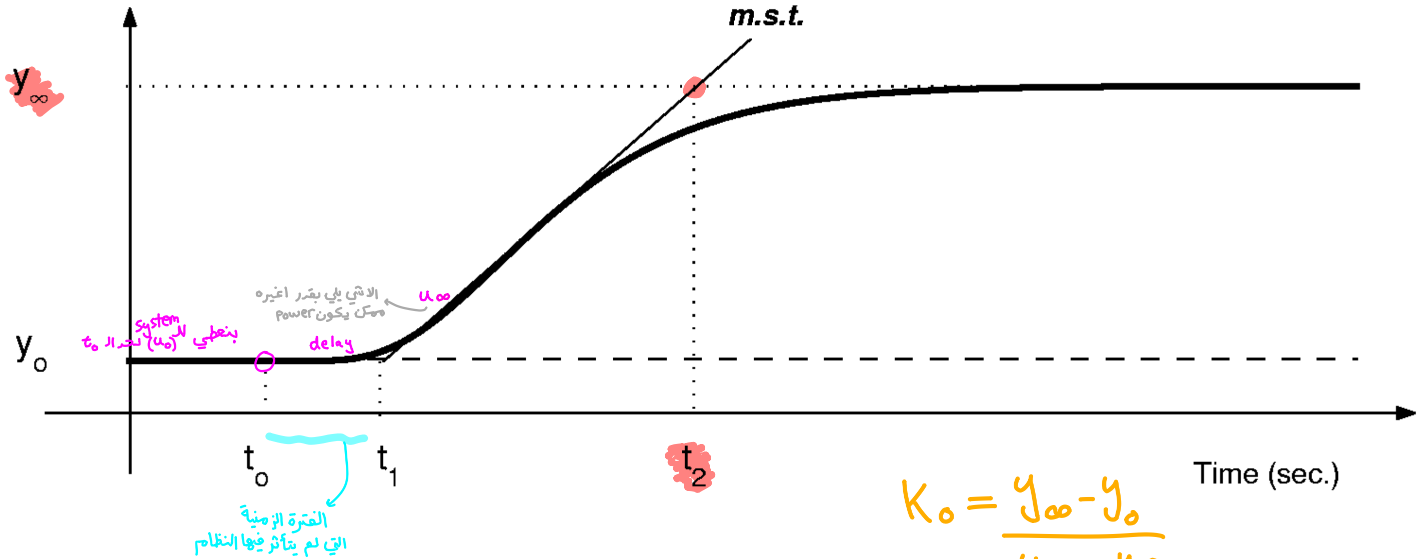
- Such step-response curves may be generated experimentally or from a dynamic simulation of the plant.

Table-1

Type of Controller	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

Figure 6.6: *Plant step response*

The suggested parameters are shown in Table 6.2.



$$K_o = \frac{y_\infty - y_o}{u_\infty - u_o}$$

$$\tau_o = t_1 - t_o$$

$$v_o = t_2 - t_1$$

First Method Ziegler Nichols

A linearized quantitative version of a simple plant can be obtained with an open loop experiment, using the following procedure:

1. With the plant in open loop, take the plant manually to a normal operating point. Say that the plant output settles at $y(t) = y_0$ for a constant plant input $u(t) = u_0$.
2. At an initial time, t_0, apply a step change to the plant input, from u_0 to u_∞ (*this should be in the range of 10 to 20% of full scale*).

Cont/...

3. Record the plant output until it settles to the new operating point. Assume you obtain the curve shown on the next slide. This curve is known as the *process reaction curve*.

   In Figure 6.6, m.s.t. stands for *maximum slope tangent*.

4. Compute the parameter model as follows

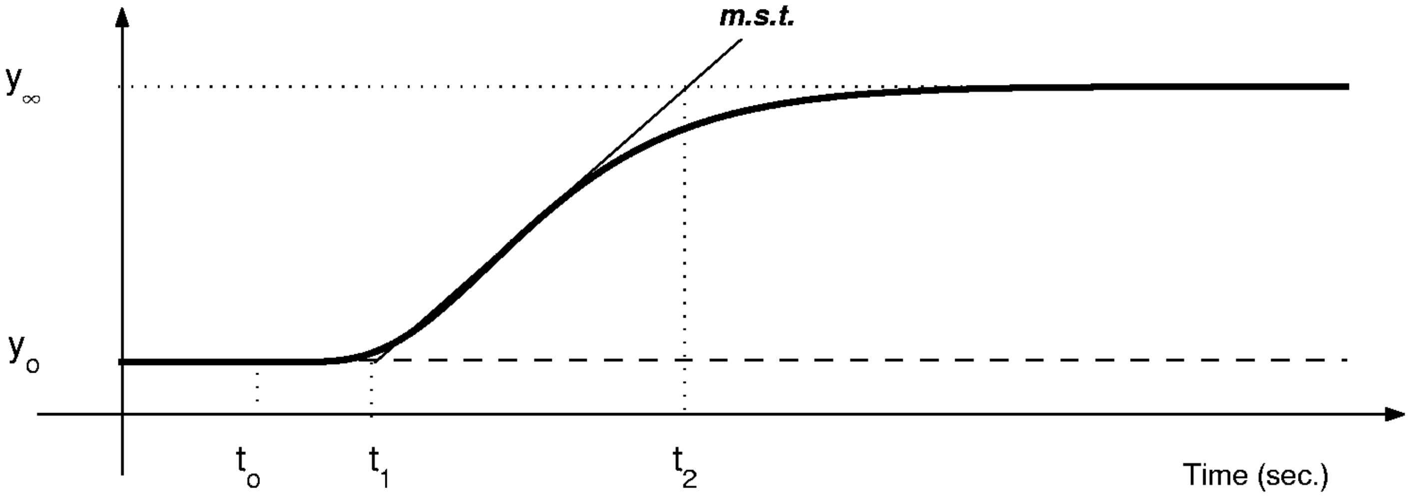
$$\boxed{K_o = \frac{y_\infty - y_o}{u_\infty - u_o};} \qquad \boxed{\tau_o = t_1 - t_o;} \qquad \boxed{\nu_o = t_2 - t_1}$$

حفظ ←

(اما الجدول بيجي بالسؤال)

Figure 6.6: *Plant step response*

The suggested parameters are shown in Table 6.2.

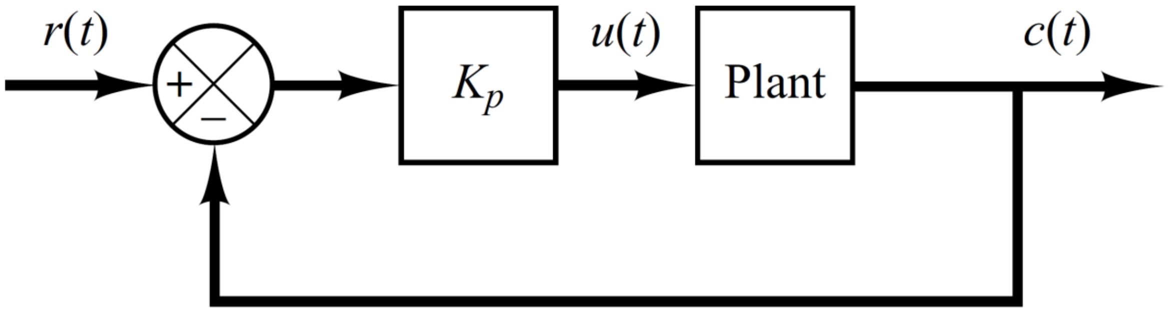


Zeigler-Nichol’s Second Method

هنا نتكلم عن الـclosed loop

- In the second method, we first set $T_i = \infty$ and $T_d = 0$.

- Using the proportional control action only (as shown in figure), increase K_p from 0 to a critical value K_{cr} at which the output first exhibits sustained oscillations.



- If the output does not exhibit sustained oscillations for whatever value K_p may take, then this method does not apply.

Zeigler-Nichol’s Second Method

- Thus, the critical gain K_{cr} and the corresponding period P_{cr} are determined.

بنضل نزيد ال Kp حتى نوصل
الsinusoidal و هنا نكون بال critical value
و لو زدنا ال Kp أكتر نصبح في مرحلة الinstability

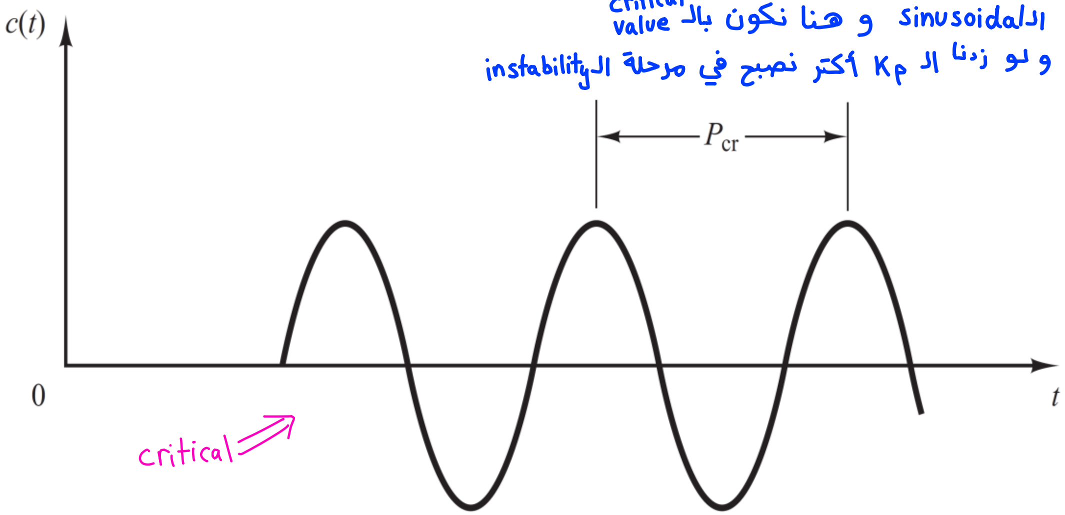


Table-2

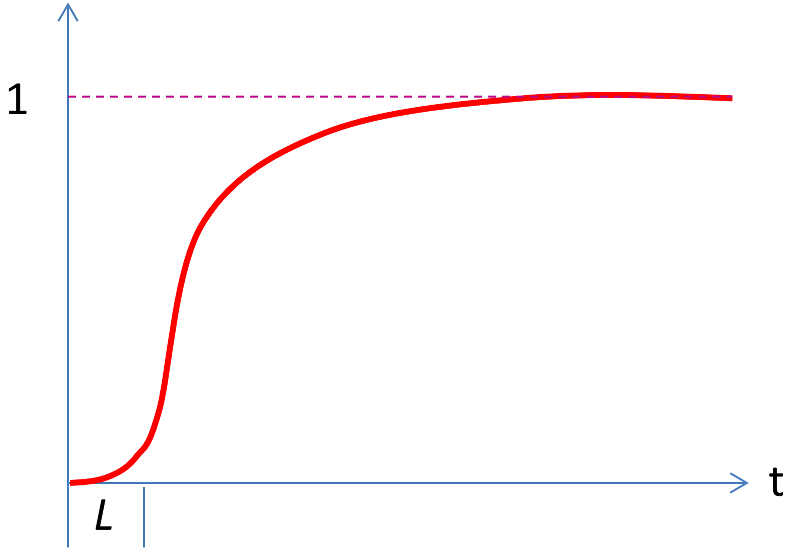
Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Example-1

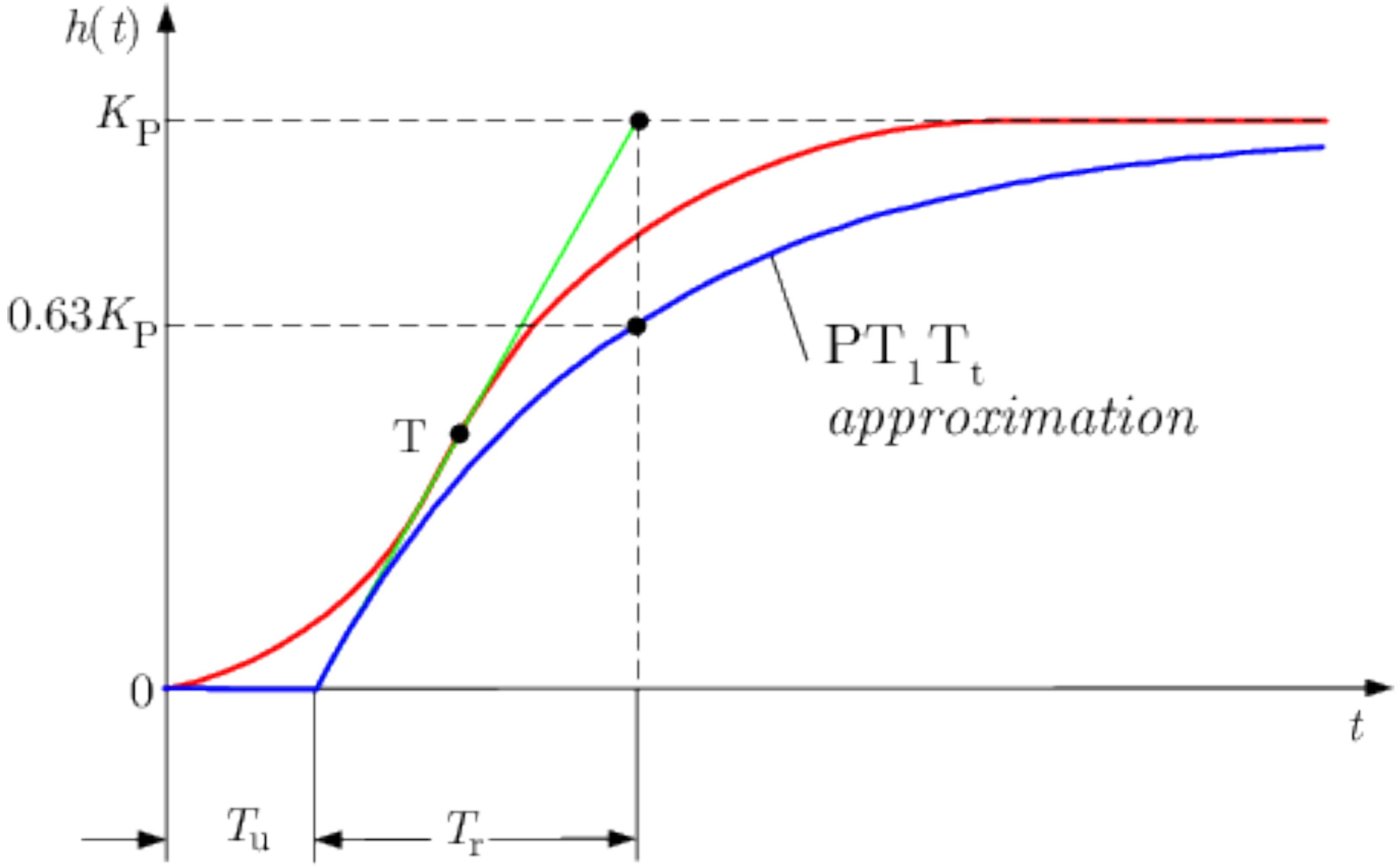
يغير الوحدات

$$\frac{C(s)}{R(s)} = \frac{K}{Ts+1}\boxed{e^{-sL}}$$

يدل على
ال delay

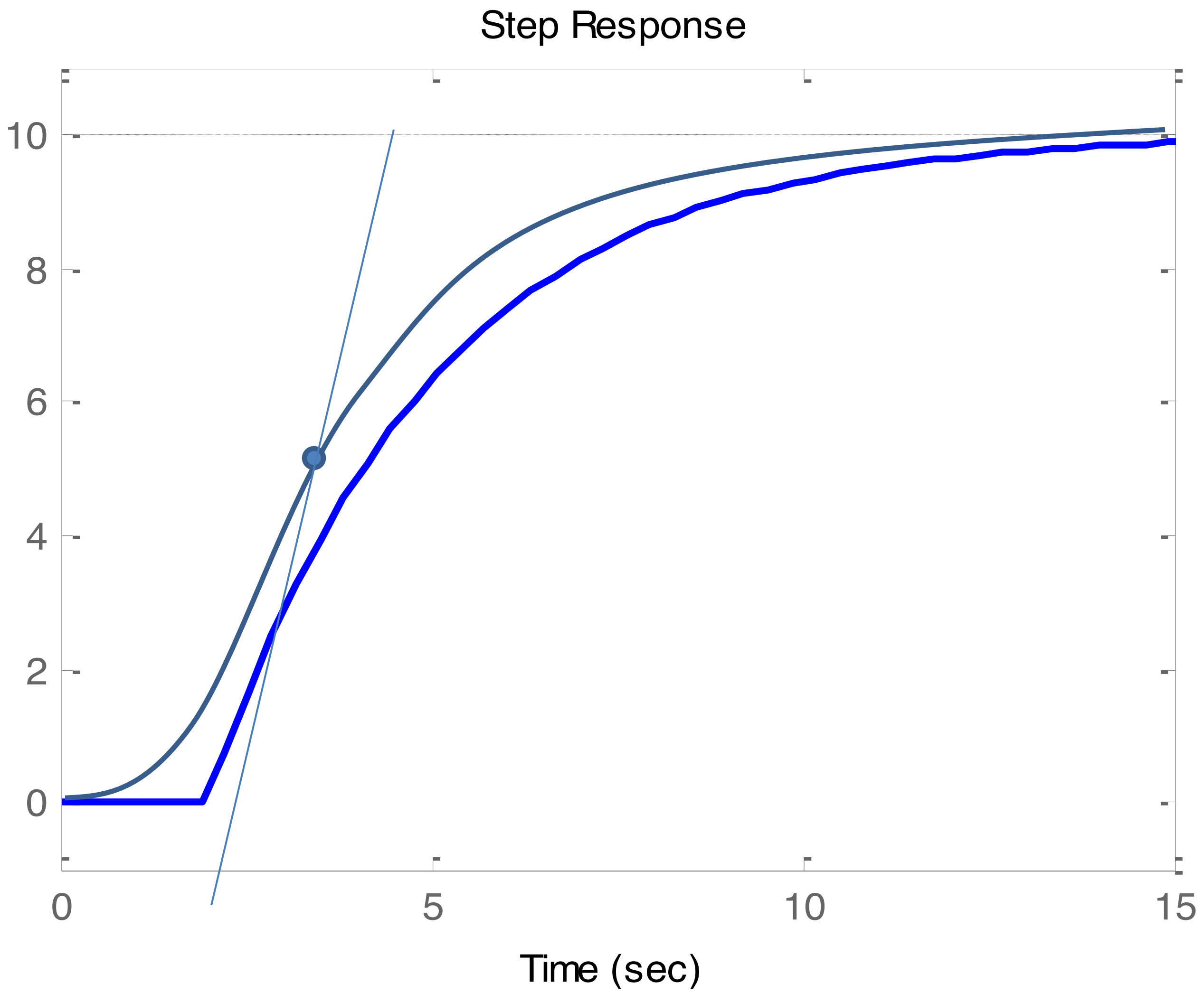


Example-1



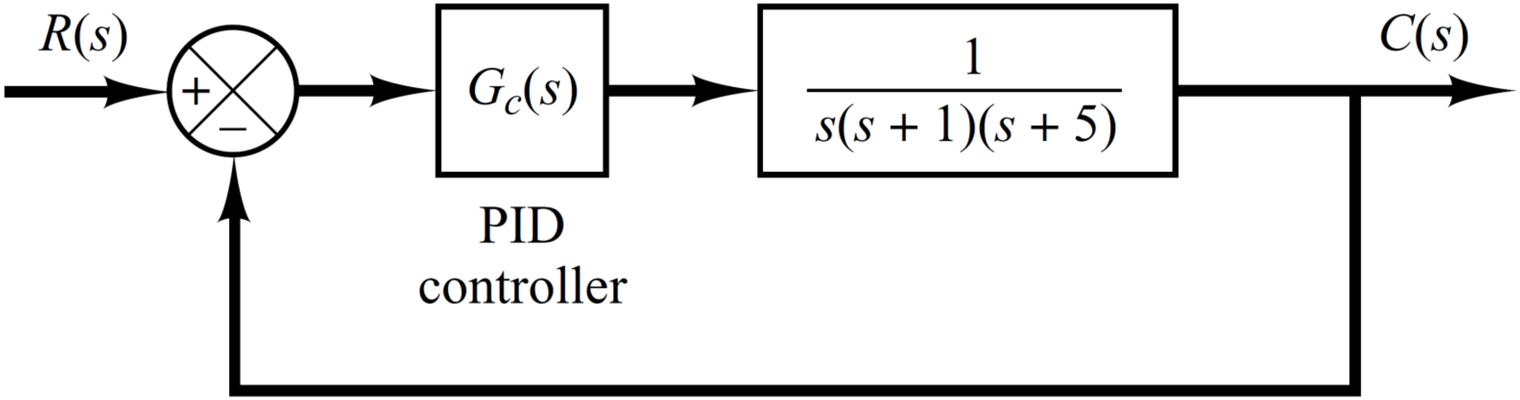
Example-1

$$\frac{C(s)}{R(s)} = \frac{10}{3s+1}e^{-2s}$$



Example-2

- Consider the control system shown in following figure.



- Apply a Ziegler–Nichols tuning rule for the determination of the values of parameters K_p, T_i and T_d.

Example-2

- Transfer function of the plant is

$$G(s) = \frac{1}{s(s+1)(s+5)}$$

- Since plant has an integrator therefore Ziegler-Nichol's first method is not applicable.

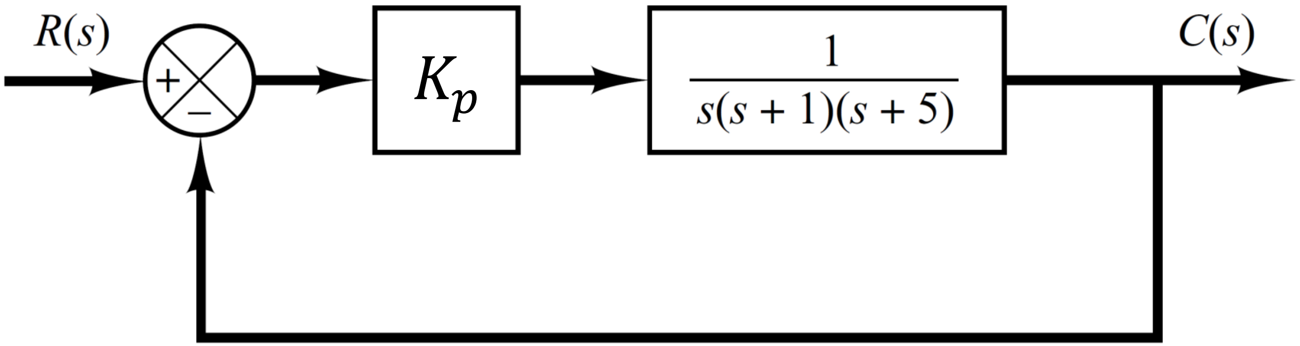
- According to second method proportional gain is varied till sustained oscillations are produced.

- That value of K_c is referred as K_{cr}.

Example-2

- Here, since the transfer function of the plant is known we can find K_{cr} using
  - Root Locus
  - Routh-Herwitz Stability Criterion

- By setting $T_i = \infty$ and $T_d = 0$ closed loop transfer function is obtained as follows.



$$\frac{C(s)}{R(s)} = \left(\frac{K_p}{s(s+1)(s+5)+K_p}\right)$$

بعد فك
ال closed
Loop transfer
function

Example-2

- The value of K_p that makes the system marginally unstable so that sustained oscillation occurs can be obtained as

$$s^3 + 6s^2 + 5s + K_p = 0$$

- The Routh array is obtained as

$$\begin{array}{lcc} s^3 & 1 & 5 \\ s^2 & 6 & K_p \\ s^1 & \left(\dfrac{30 - K_p}{6} = 0 \right) & \\ s^0 & K_p & \end{array}$$

- Examining the coefficients of first column of the Routh array we find that sustained oscillations will occur if $K_p = 30$.

- Thus the critical gain K_{cr} is

$$K_{cr} = 30$$

$30 - K_p = 0$

$K_p = 30$

Example-2

- With gain K_p set equal to 30, the characteristic equation becomes

$$s^3 + 6s^2 + 5s + 30 = 0$$

- To find the frequency of sustained oscillations, we substitute $s = j\omega$ into the characteristic equation.

$$(j\omega)^3 + 6(j\omega)^2 + 5j\omega + 30 = 0$$

- Further simplification leads to

$$6(5 - \omega^2) + j\omega(5 - \omega^2) = 0$$

$$6(5 - \omega^2) = 0$$

$$\omega = \sqrt{5}\, rad/sec$$

Example-2

$$\omega = \sqrt{5}\, rad/sec$$

- Hence the period of sustained oscillations P_{cr} is

$$\boxed{P_{cr} = \frac{2\pi}{\omega}}$$

$$P_{cr} = \frac{2\pi}{\sqrt{5}} = 2.8099\ sec$$

- Referring to Table-2

$$K_p = 0.6K_{cr} = 18$$

$$T_i = 0.5P_{cr} = 1.405$$

$$T_d = 0.125P_{cr} = 0.35124$$

Example-2

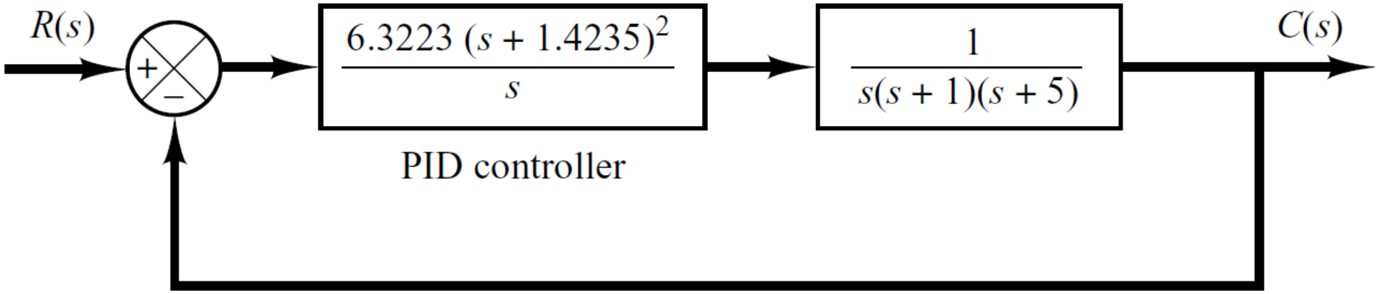
$K_p = 18$ $\qquad T_i = 1.405$ $\qquad T_d = 0.35124$

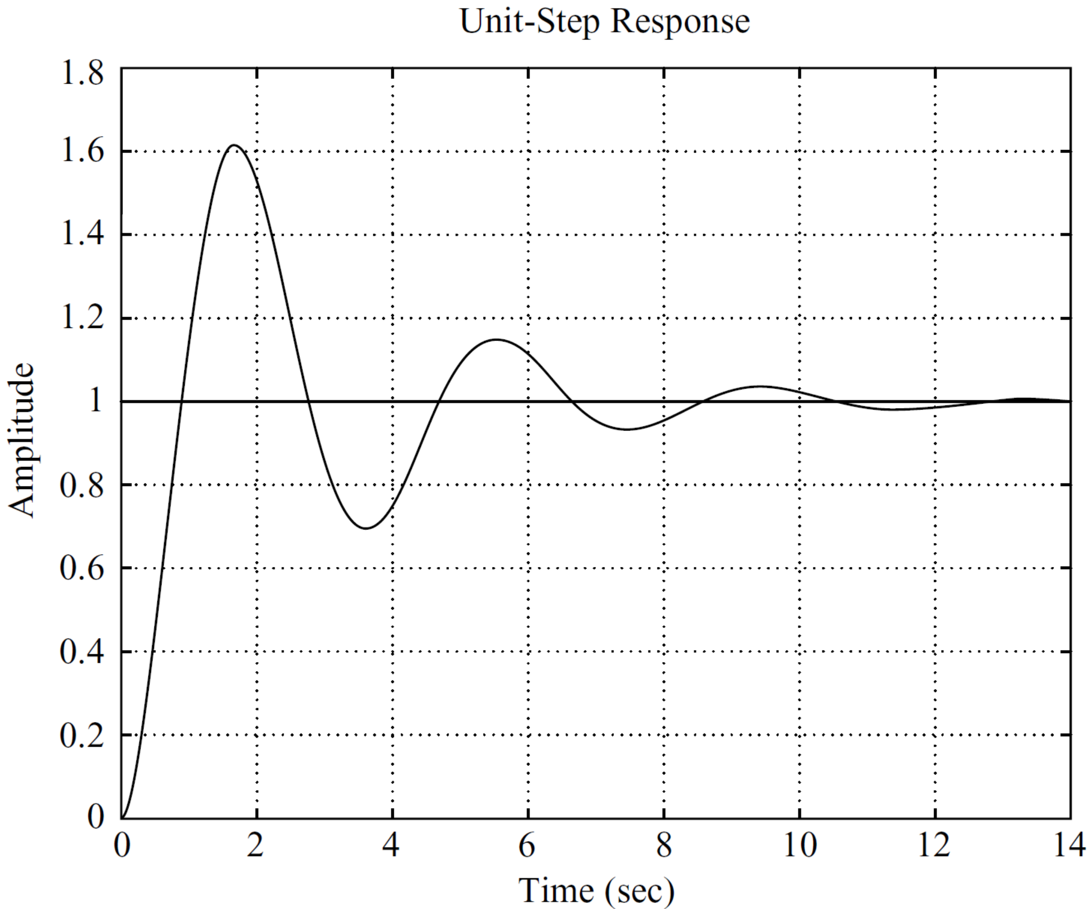
- Transfer function of PID controller is thus obtained as

$$G_c(s) = K_p(1 + \frac{1}{T_i s} + T_d\ s)$$

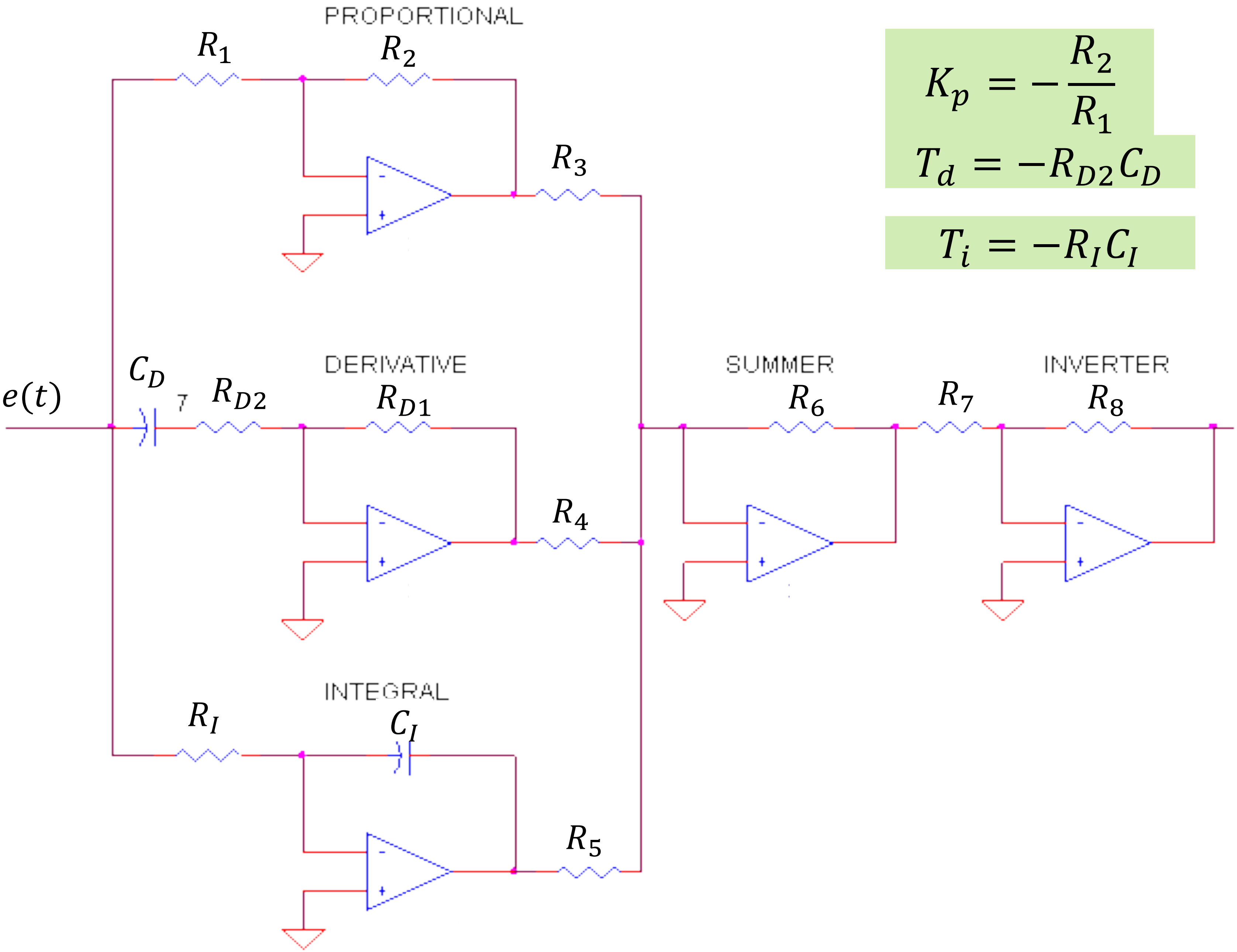
$$G_c(s) = 18(1 + \frac{1}{1.405s} + 0.35124s)$$

Example-2


$R(s)$
+
−
$\frac{6.3223\,(s+1.4235)^2}{s}$
PID controller
$\frac{1}{s(s+1)(s+5)}$
$C(s)$


Unit-Step Response
Amplitude
Time (sec)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
0
2
4
6
8
10
12
14

Electronic PID Controller

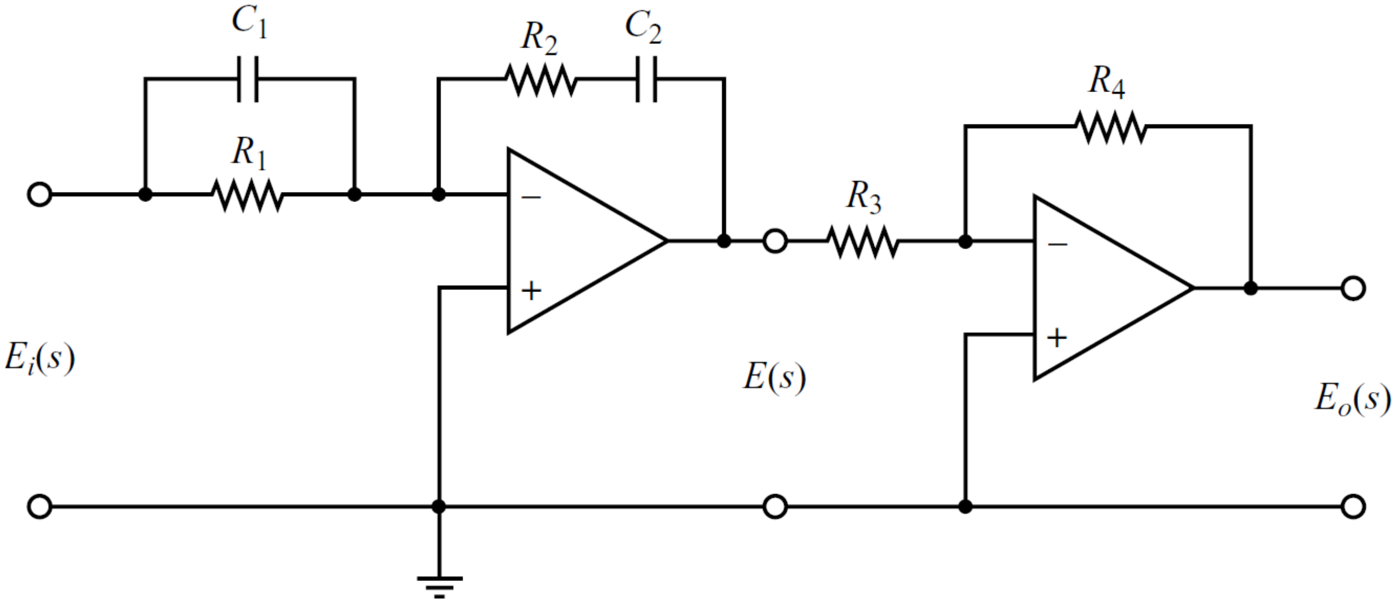


$$K_p = -\frac{R_2}{R_1}$$

$$T_d = -R_{D2}C_D$$

$$T_i = -R_I C_I$$

Electronic PID Controller



$$\frac{E_o(s)}{E_i(s)} = \frac{R_4}{R_3}\,\frac{(R_1C_1s+1)(R_2C_2s+1)}{R_2C_2s}$$

$$\frac{E_o(s)}{E_i(s)} = \frac{R_4R_2}{R_3R_1}\left(\frac{R_1C_1+R_2C_2}{R_2C_2}+\frac{1}{R_2C_2s}+R_1C_1s\right)$$

Electronic PID Controller

$$\frac{E_o(s)}{E_i(s)} = \frac{R_4 R_2}{R_3 R_1}\left(\frac{R_1 C_1 + R_2 C_2}{R_2 C_2} + \frac{1}{R_2 C_2 s} + R_1 C_1 s\right)$$

$$\frac{E_o(s)}{E_i(s)} = \frac{R_4(R_1 C_1 + R_2 C_2)}{R_3 R_1 C_2}\left[1 + \frac{1}{(R_1 C_1 + R_2 C_2)s} + \frac{R_1 C_1 R_2 C_2}{R_1 C_1 + R_2 C_2} s\right]$$

$$K_p = \frac{R_4(R_1 C_1 + R_2 C_2)}{R_3 R_1 C_2}$$

$$T_i = R_1 C_1 + R_2 C_2$$

$$T_d = \frac{R_1 C_1 R_2 C_2}{R_1 C_1 + R_2 C_2}$$

- In terms of K_p, K_i, K_d we have

$$K_p = \frac{R_4(R_1 C_1 + R_2 C_2)}{R_3 R_1 C_2}$$

$$K_i = \frac{R_4}{R_3 R_1 C_2}$$

$$K_d = \frac{R_4 R_2 C_1}{R_3}$$

PID implementation using Arduino: Method 1

In the s-domain the PID controller has the following form

$$U(s) = K\big(1 + \frac{1}{sT_i} + sT_d\big)E(s) \qquad (1)$$

where $U(s)$ is the control action that is sent to the actuator, E(s) is the control error defined by

$$E(s) = Y_r(s) - Y(s) \qquad (2)$$

$$u(t) = K\Big(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\dot{e}(t)\Big) \qquad (3)$$

$$e(t) = y_r(t) - y(t) \qquad (4)$$

$$u(t) = K\big(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\dot{e}(t)\big) \qquad (3)$$

Take derivative of both sides

$$\dot{u}(t) = K\dot{e}(t) + \frac{K}{T_i}e(t) + KT_d\ddot{e}(t) \qquad (6)$$

$$\dot{u}(t) \approx \frac{u_k - u_{k-1}}{h} \qquad (7)$$

تم شرحهم
بشكل سريع

Similarly, we approximate the first derivative of the control error

$$\dot{e}(t) \approx \frac{e_k - e_{k-1}}{h} \tag{8}$$

The second derivative of the control error is approximated as follows

$$\ddot{e}(t) \approx \frac{\dot{e}_k - \dot{e}_{k-1}}{h} \tag{9}$$

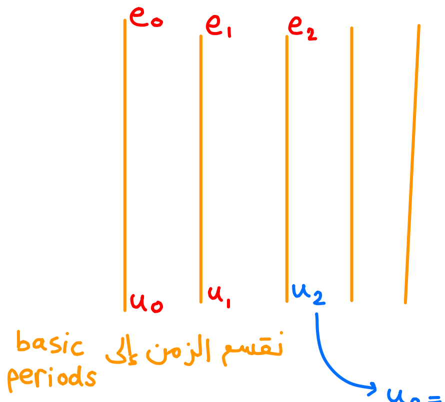
By substituting \eqref{firstDerivativeApproximationError} for the time indices k and $k-1$, we obntain

$$\ddot{e}(t) \approx \frac{e_k - 2e_{k-1} + e_{k-2}}{h^2} \tag{10}$$

$$u_k = u_{k-1} + K_0 e_k + K_1 e_{k-1} + K_2 e_{k-2} \qquad (11)$$

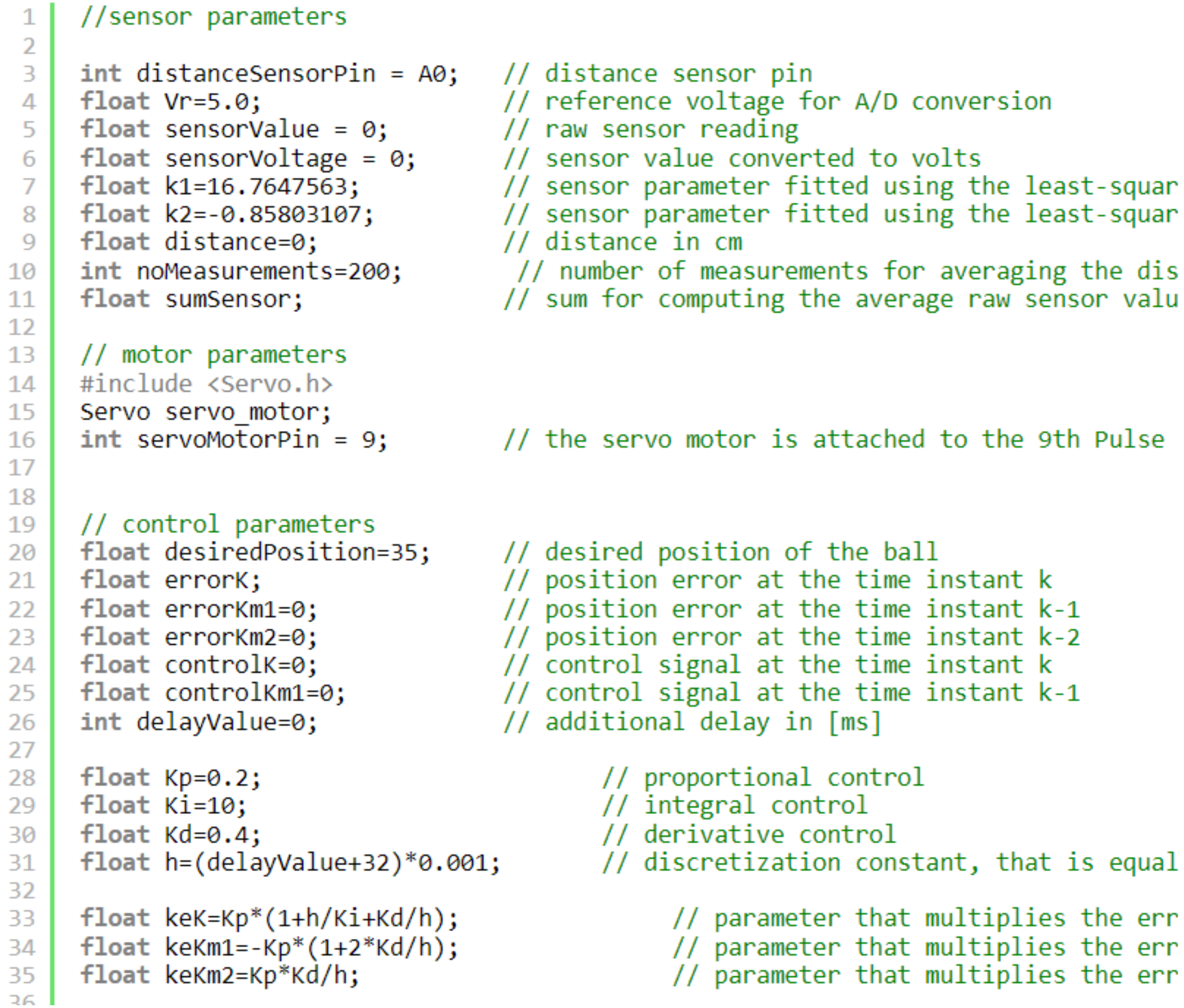
where the constants K_0, K_1, and K_2 are determined as follows

$$K_0 = K\left(1 + \frac{h}{T_i} + \frac{T_d}{h}\right)$$
$$K_1 = -K(1 + \frac{2T_d}{h}) \qquad (12)$$
$$K_2 = \frac{KT_d}{h}$$

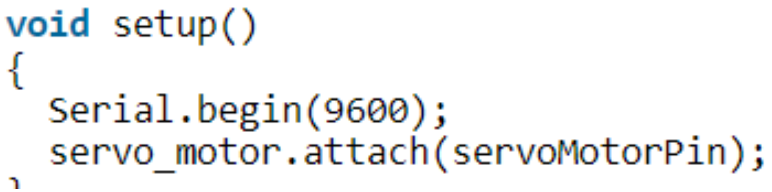


$u_2 = u_1 + K_0 e_2 + K_1 e_1 + K_2 e_0$

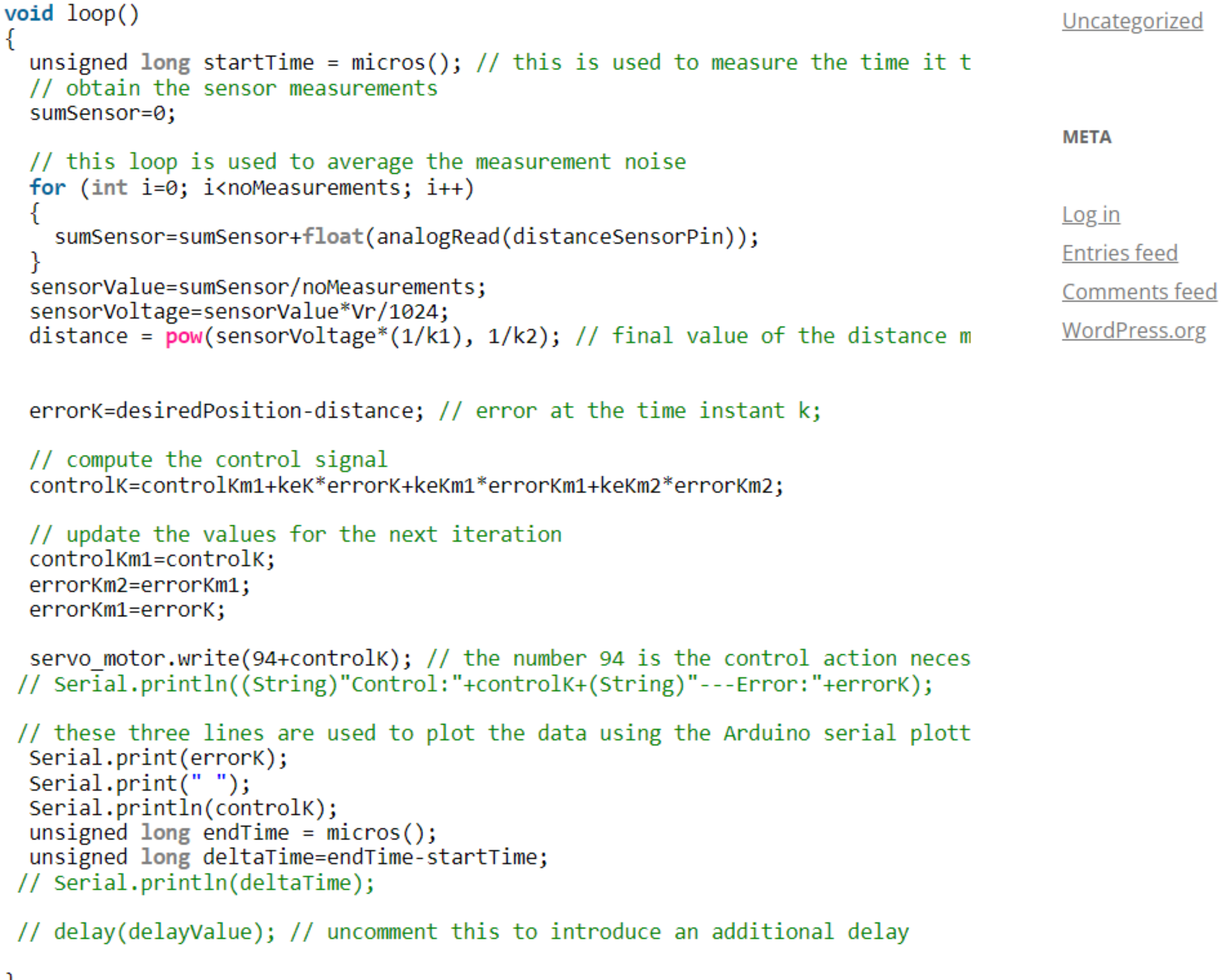
$u_3 = u_2 + K_0 e_3 + K_1 e_2 + K_2 e_1$ ⇐ حسب المعادلة التي في بداية السلايد



```
//sensor parameters

int distanceSensorPin = A0;   // distance sensor pin
float Vr=5.0;                 // reference voltage for A/D conversion
float sensorValue = 0;        // raw sensor reading
float sensorVoltage = 0;      // sensor value converted to volts
float k1=16.7647563;          // sensor parameter fitted using the least-squar
float k2=-0.85803107;         // sensor parameter fitted using the least-squar
float distance=0;             // distance in cm
int noMeasurements=200;        // number of measurements for averaging the dis
float sumSensor;              // sum for computing the average raw sensor valu

// motor parameters
#include <Servo.h>
Servo servo_motor;
int servoMotorPin = 9;        // the servo motor is attached to the 9th Pulse


// control parameters
float desiredPosition=35;     // desired position of the ball
float errorK;                 // position error at the time instant k
float errorKm1=0;             // position error at the time instant k-1
float errorKm2=0;             // position error at the time instant k-2
float controlK=0;             // control signal at the time instant k
float controlKm1=0;           // control signal at the time instant k-1
int delayValue=0;             // additional delay in [ms]

float Kp=0.2;                        // proportional control
float Ki=10;                         // integral control
float Kd=0.4;                        // derivative control
float h=(delayValue+32)*0.001;       // discretization constant, that is equal

float keK=Kp*(1+h/Ki+Kd/h);               // parameter that multiplies the err
float keKm1=-Kp*(1+2*Kd/h);               // parameter that multiplies the err
float keKm2=Kp*Kd/h;                      // parameter that multiplies the err
```

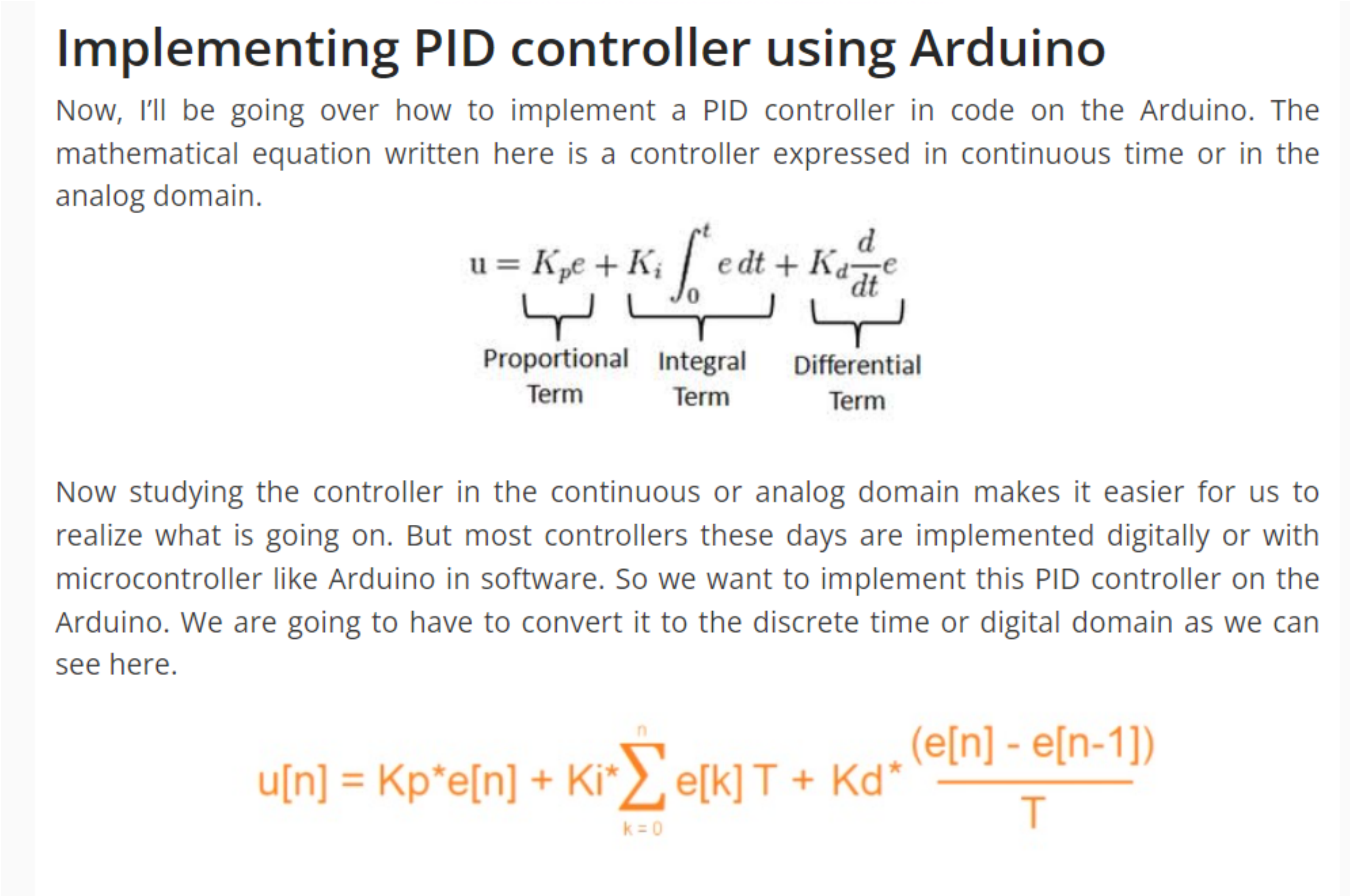


```
void setup()
{
  Serial.begin(9600);
  servo_motor.attach(servoMotorPin);
}
```



```
void loop()
{
  unsigned long startTime = micros(); // this is used to measure the time it t
  // obtain the sensor measurements
  sumSensor=0;

  // this loop is used to average the measurement noise
  for (int i=0; i<noMeasurements; i++)
  {
    sumSensor=sumSensor+float(analogRead(distanceSensorPin));
  }
  sensorValue=sumSensor/noMeasurements;
  sensorVoltage=sensorValue*Vr/1024;
  distance = pow(sensorVoltage*(1/k1), 1/k2); // final value of the distance m


  errorK=desiredPosition-distance; // error at the time instant k;

  // compute the control signal
  controlK=controlKm1+keK*errorK+keKm1*errorKm1+keKm2*errorKm2;

  // update the values for the next iteration
  controlKm1=controlK;
  errorKm2=errorKm1;
  errorKm1=errorK;

  servo_motor.write(94+controlK); // the number 94 is the control action neces
 // Serial.println((String)"Control:"+controlK+(String)"---Error:"+errorK);

 // these three lines are used to plot the data using the Arduino serial plott
  Serial.print(errorK);
  Serial.print(" ");
  Serial.println(controlK);
  unsigned long endTime = micros();
  unsigned long deltaTime=endTime-startTime;
 // Serial.println(deltaTime);

 // delay(delayValue); // uncomment this to introduce an additional delay

}
```

Method II



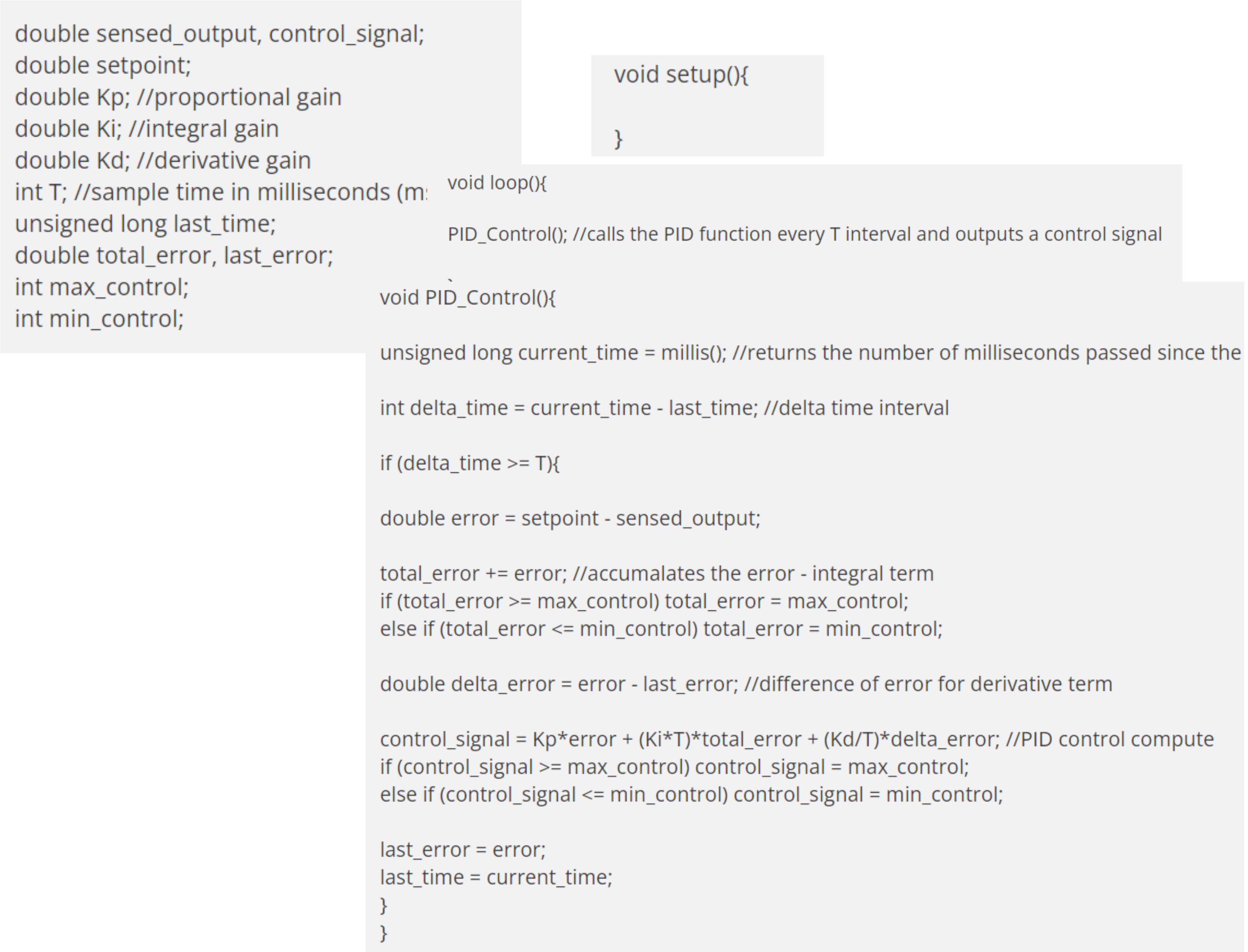
Implementing PID controller using Arduino

Now, I'll be going over how to implement a PID controller in code on the Arduino. The mathematical equation written here is a controller expressed in continuous time or in the analog domain.

$$u = \underbrace{K_p e}_{\text{Proportional Term}} + \underbrace{K_i \int_0^t e\,dt}_{\text{Integral Term}} + \underbrace{K_d \frac{d}{dt} e}_{\text{Differential Term}}$$

Now studying the controller in the continuous or analog domain makes it easier for us to realize what is going on. But most controllers these days are implemented digitally or with microcontroller like Arduino in software. So we want to implement this PID controller on the Arduino. We are going to have to convert it to the discrete time or digital domain as we can see here.

$$u[n] = Kp^{*}e[n] + Ki^{*}\sum_{k=0}^{n} e[k]\,T + Kd^{*}\,\frac{(e[n] - e[n-1])}{T}$$



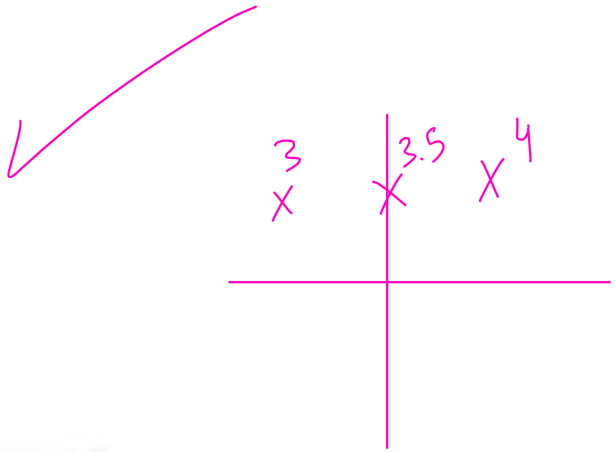
```
double sensed_output, control_signal;
double setpoint;
double Kp; //proportional gain
double Ki; //integral gain
double Kd; //derivative gain
int T; //sample time in milliseconds (m
unsigned long last_time;
double total_error, last_error;
int max_control;
int min_control;
```

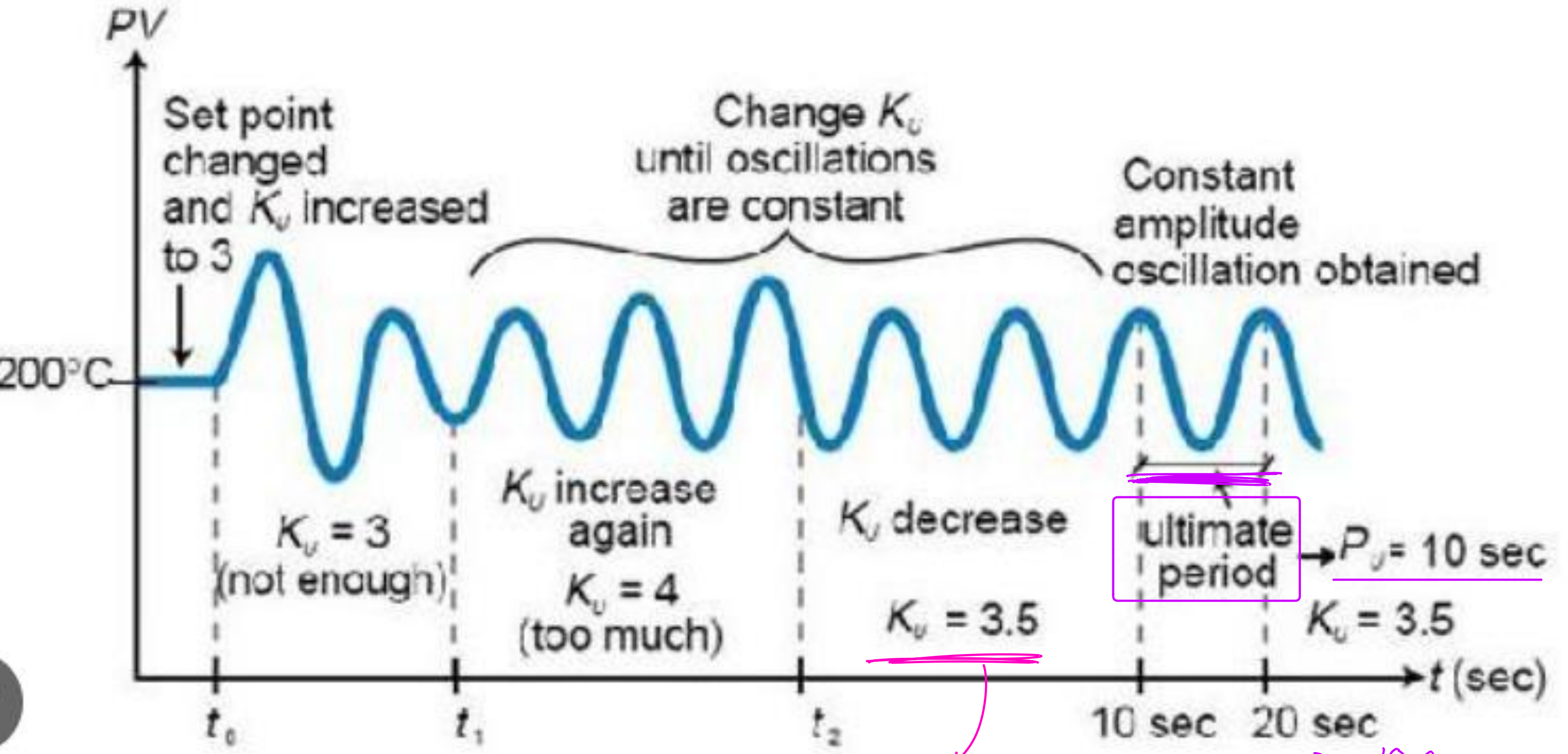
```
void setup(){

}
```

```
void loop(){

PID_Control(); //calls the PID function every T interval and outputs a control signal
```

```
void PID_Control(){

unsigned long current_time = millis(); //returns the number of milliseconds passed since the

int delta_time = current_time - last_time; //delta time interval

if (delta_time >= T){

double error = setpoint - sensed_output;

total_error += error; //accumalates the error - integral term
if (total_error >= max_control) total_error = max_control;
else if (total_error <= min_control) total_error = min_control;

double delta_error = error - last_error; //difference of error for derivative term

control_signal = Kp*error + (Ki*T)*total_error + (Kd/T)*delta_error; //PID control compute
if (control_signal >= max_control) control_signal = max_control;
else if (control_signal <= min_control) control_signal = min_control;

last_error = error;
last_time = current_time;
}
}
```

Tuning example:





$K_{critical}$

$p = 20 - 10 = 10$ s

بعد ما لقيتها برجع للجدول

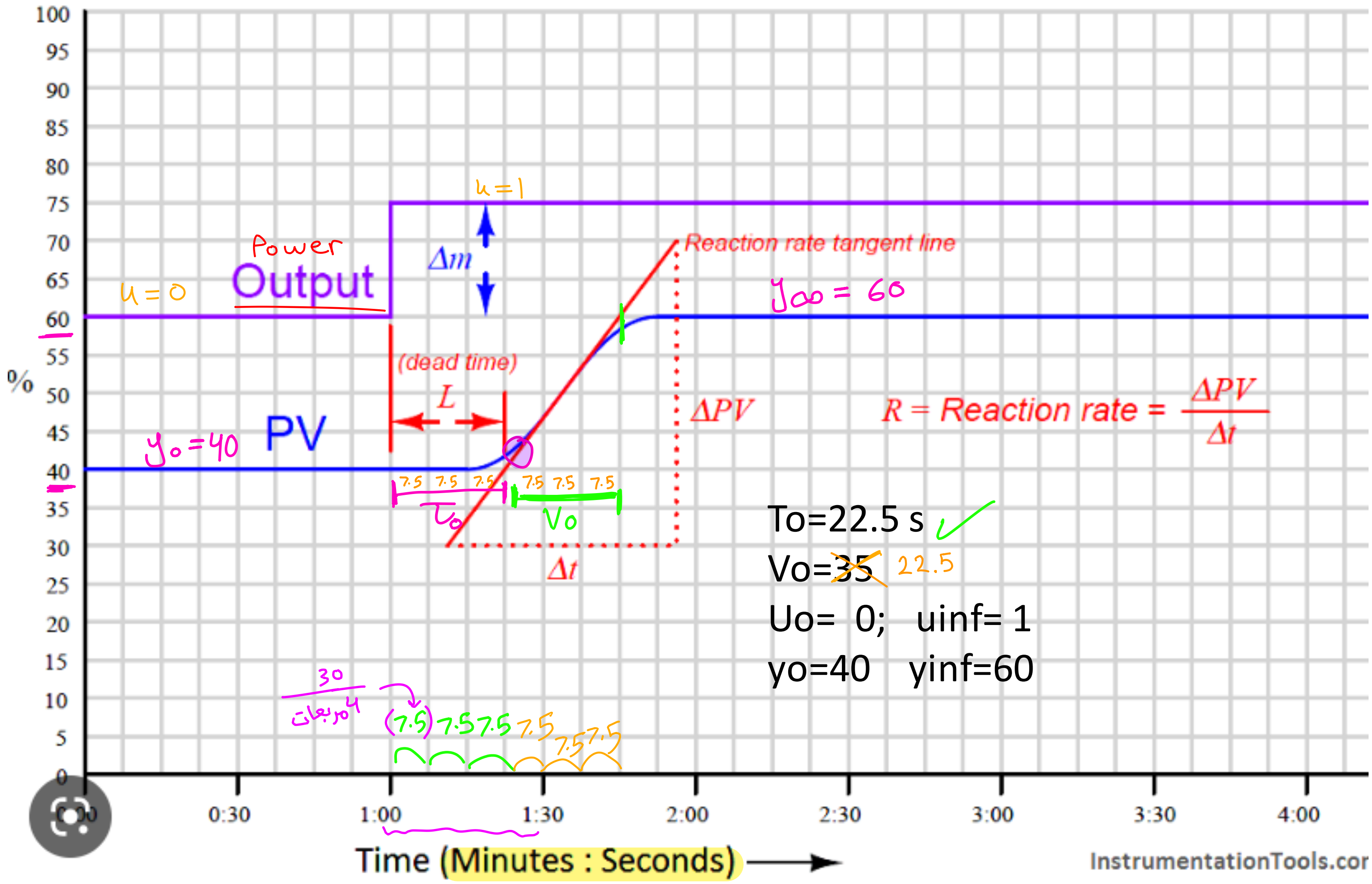
Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$ 3.5	$0.5P_{cr}$ 10	$0.125P_{cr}$ 10

Now select the required controller from table based on the question.
For example if the required is PI then we select the second row

Kp=0.45 *Kcr =0.45 *3.5
Ti=1/1.2 * Pcr= 1/1.2 * 10

By yourself solve the same example if PID is required not PI

Tuning example II



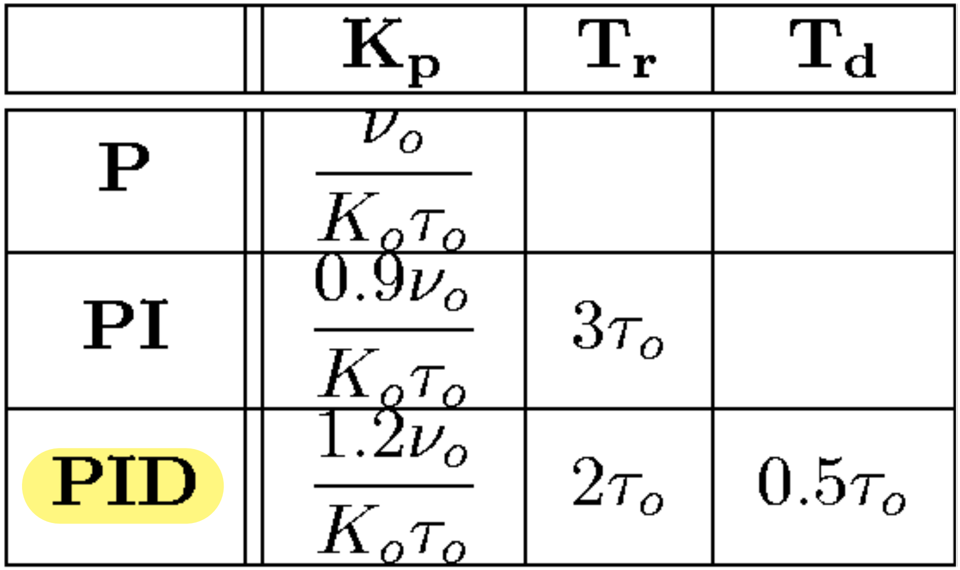
To=22.5 s ✓

Vo=~~35~~ 22.5

Uo= 0; uinf= 1

yo=40 yinf=60

Kcr = 20



	$\mathbf{K_p}$	$\mathbf{T_r}$	$\mathbf{T_d}$
P	$\frac{\nu_o}{K_o\tau_o}$		
PI	$\frac{0.9\nu_o}{K_o\tau_o}$	$3\tau_o$	
PID	$\frac{1.2\nu_o}{K_o\tau_o}$	$2\tau_o$	$0.5\tau_o$

Ko=(60-40)/(1-0)=20

If we select PID to implement

Kp=1.2 * ~~35~~ 22.5/(20*22.5) ; Tr=2 *22.5;

Td=0.5*22.5

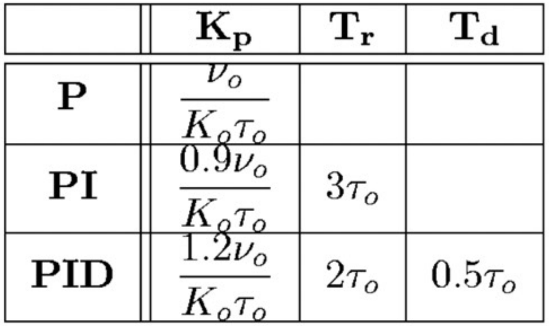
* Summary :-

Response curve

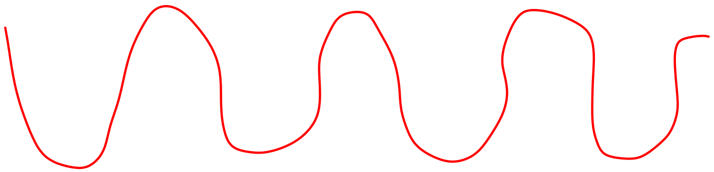
في هذا الشكل نستعمل

First method

و هذا الجدول



	K_p	T_r	T_d
P	$\frac{\nu_o}{K_o\tau_o}$		
PI	$\frac{0.9\nu_o}{K_o\tau_o}$	$3\tau_o$	
PID	$\frac{1.2\nu_o}{K_o\tau_o}$	$2\tau_o$	$0.5\tau_o$

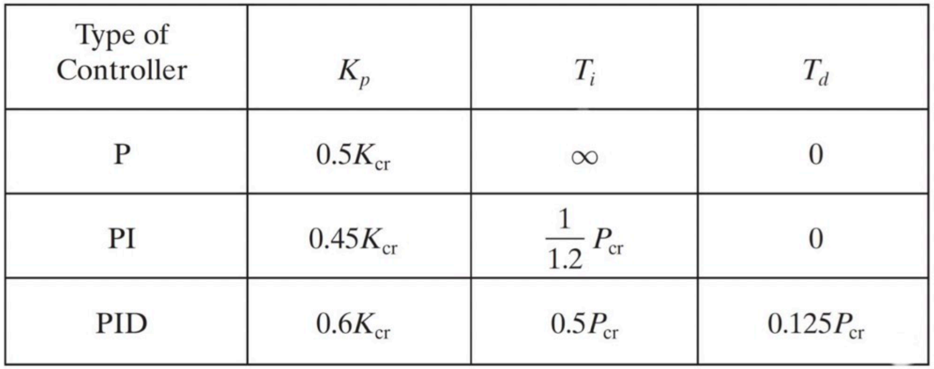


Sinusoidal

في هذا الشكل نستعمل

Second Method

و هذا الجدول



Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

From Differential Equation to Difference Equation:

*ما اظن احدناه

- Definition of Derivative:

$$\frac{dU}{dt} = \lim_{\Delta t \to 0} \frac{U(t + \Delta t) - U(t)}{\Delta t}$$

- Algebraically Manipulate to Difference Eq:

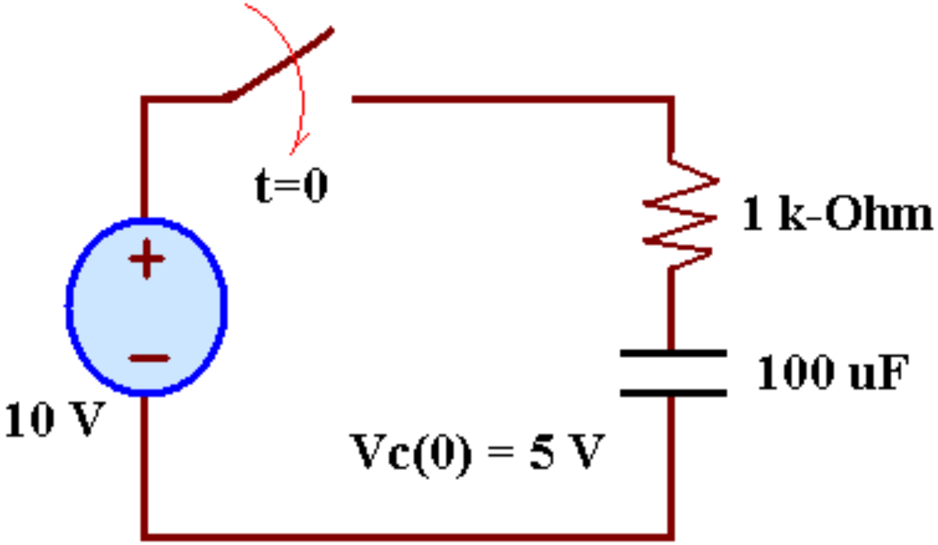
$$U(t + \Delta t) = U(t) + \Delta t * \frac{dU}{dt}$$

(for sufficiently small Δt)

- Apply this to Iteratively Solve First Order Linear Differential Equations (hold for applause)

Implementing Difference Eqs:

- Consider the following RC Circuit, with 5 Volts of initial charge on the capacitor:



- KVL around the loop:

-Vs + Ic*R + Vc = 0,   Ic = C*dVc/dt

OR dVc/dt = (Vs –Vc)/RC

Differential to Difference with Time-Step, T:

- Differential Equation:

$$dVc/dt = (Vs - Vc)/RC$$

- Difference Equation by Definition:

$$Vc(kT+T) = Vc(kT) + T*dVc/dt$$

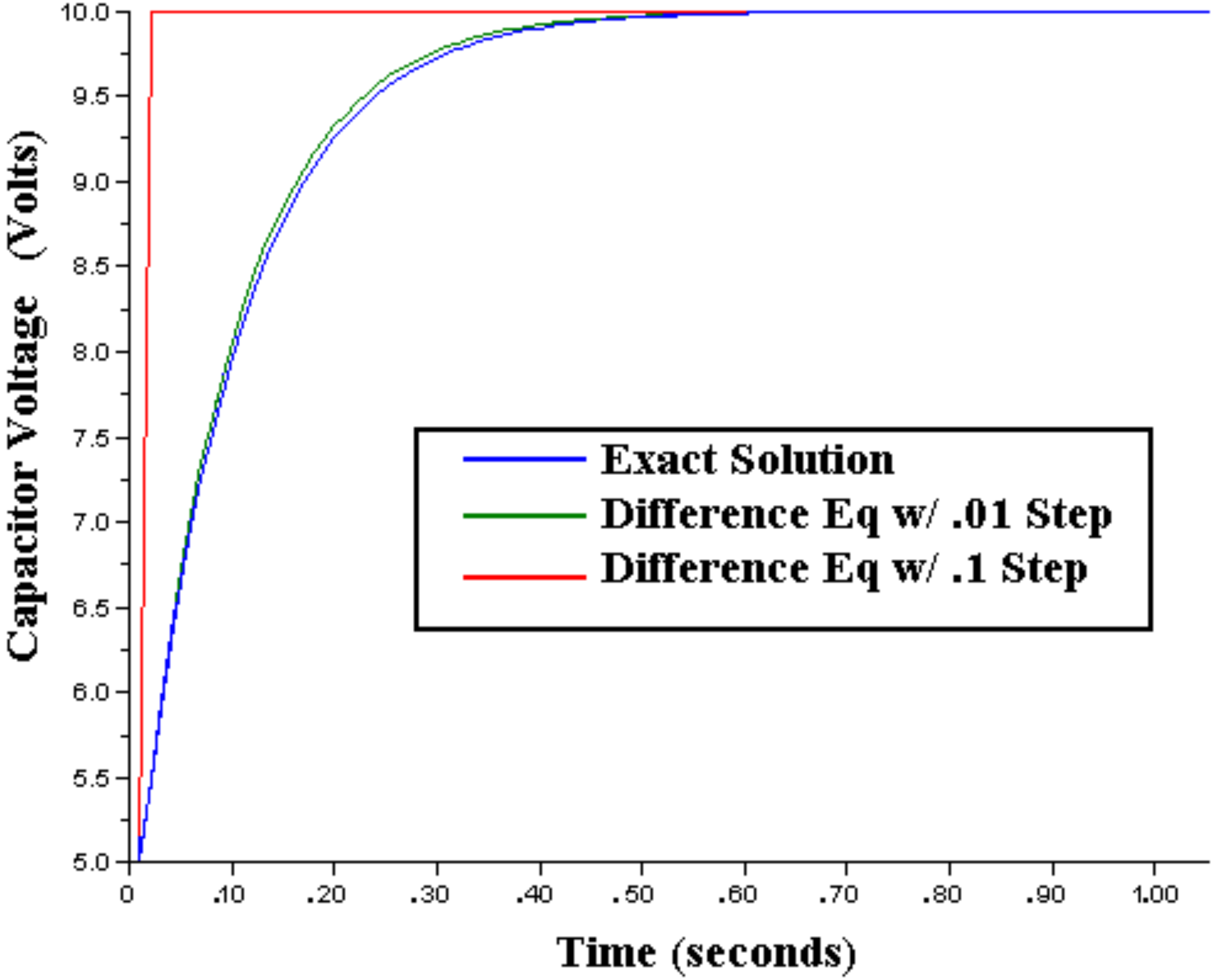
- Substituting:

$$Vc(kT+T) = Vc(kT) + T*(Vs - Vc(kT))/RC$$

Coding in SciLab:

```
R=1000
C=1e-4
Vs=10
Vo=5
//Initial Value of Difference Equation (same as Vo)
Vx(1)=5
//Time Step
dt=.01
//Initialize counter and time variable
i=1
t=0
//While loop to calculate exact solution and difference equation
while i<101, Vc(i)=Vs+(Vo-Vs)*exp(-t/(R*C)),
Vx(i+1)=Vx(i)+dt*(Vs-Vx(i))/(R*C),
t=t+dt,
i=i+1,
end
```

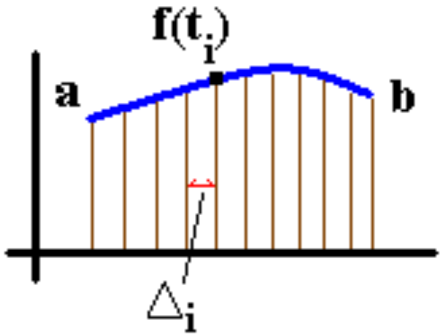
Results:


Capacitor Voltage (Volts)
10.0
9.5
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
0
.10
.20
.30
.40
.50
.60
.70
.80
.90
1.00
Time (seconds)
Exact Solution
Difference Eq w/ .01 Step
Difference Eq w/ .1 Step

Integration by Trapezoidal Approximation:

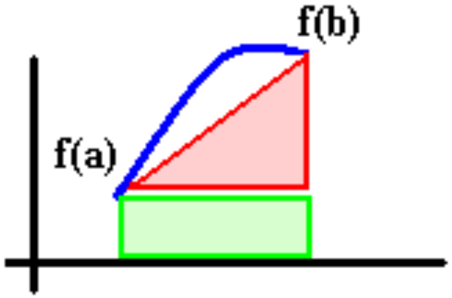
- Definition of Integration (area under curve):

$$F(b) = \int_{a}^{b} f(t)dt = \sum_{i=1}^{n} f(t_i)\Delta_i$$



- Approximation by Trapezoidal Areas

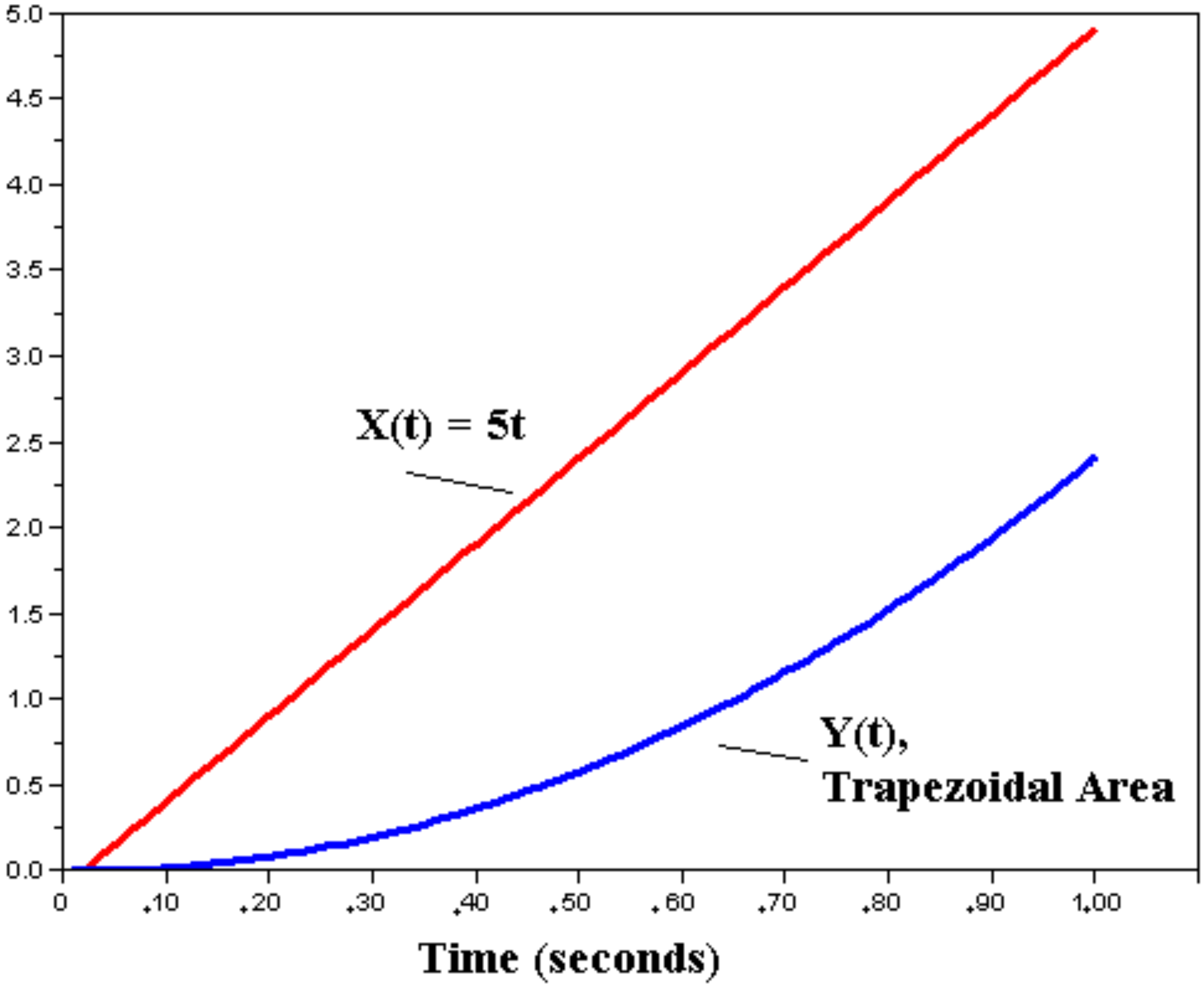
$$F(b) = (b - a)f(a) + \frac{1}{2}(b-a)(f(b) - f(a))$$



Trapezoidal Approximate Integration in SciLab:

```
//Calculate and plot X=5t and integrate it with a Trapezoidal approx.
//Time Step
dt=.01
//Initialize time and counter
t=0
i=2
//Initialize function and its trapezoidal integration function
X(1)=0
Y(1)=0
//Perform time step calculation of function and trapezoidal integral
while i<101,X(i)=5*t,Y(i)=Y(i-1)+dt*X(i-1)+0.5*dt*(X(i)-X(i-1)),
t=t+dt,
i=i+1,
end
//Plot the results
plot(X)
plot(Y)
```

Results:


5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
.10
.20
.30
.40
.50
.60
.70
.80
.90
1.00
Time (seconds)
X(t) = 5t
Y(t),
Trapezoidal Area

Coding the PID

- Using Difference Equations, it is possible now to code the PID algorithm in a high level language

$$p(t) = Kp*e(t) \rightarrow P(kT) = Kp*E(kT)$$

$$i(t) = Ki*\int e(t)dt \rightarrow$$

$$I(kT+T) = Ki*[I(kT)+T*E(kT+T)+.5(E(kT+T)-E(kT))]$$

$$d(t) = Kd* de(t)/dt \rightarrow D(kT+T) = Kd*[E(kT+T)-E(kT)]/T$$

Example: Permanent Magnet DC Motor

State-Space Description of the DC Motor:

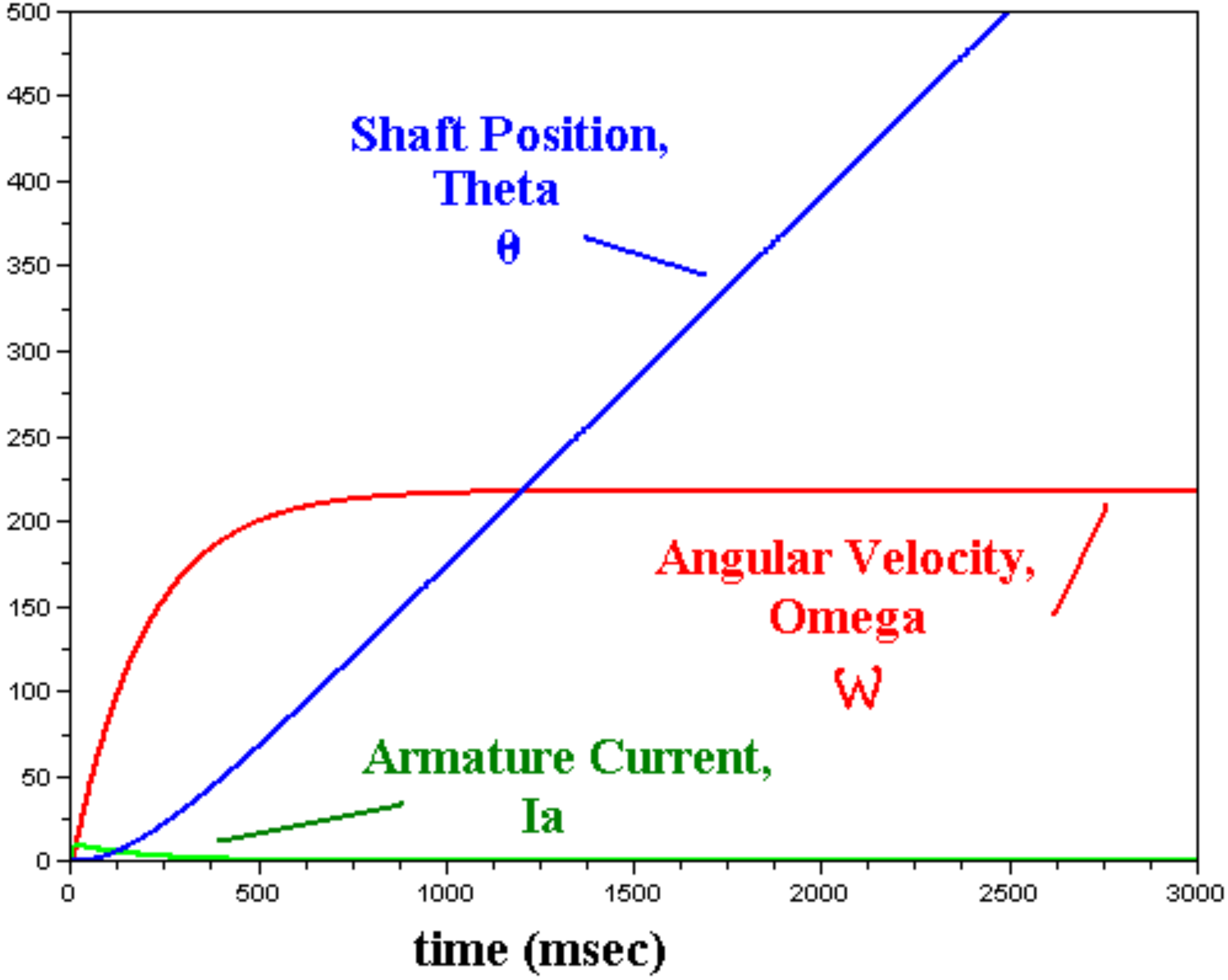
0. $\theta' = \omega$ (angular frequency)
1. $J\theta'' + B\theta' = KtIa$ → $\omega' = -B\omega/J + KtIa/J$
2. $LaIa' + RaIa = Vdc - Ka\theta'$ →

$$Ia' = -Ka\omega/La - RaIa/La + Vdc/La$$

In Matrix Form:

$$\begin{bmatrix} \dot{\theta} \\ \dot{W} \\ \dot{Ia} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{-B}{J} & \frac{Kt}{J} \\ 0 & -\frac{Ka}{La} & -\frac{Ra}{La} \end{bmatrix} * \begin{bmatrix} \theta \\ W \\ Ia \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{La} \end{bmatrix} * Vdc$$

Scilab Emulation of PM DC Motor using State Space Equations



DC Motor with PID control

```
//PID position control of permanent magnet DC motor
//Constants
Ra=1.2;La=1.4e-3;Ka=.055;Kt=Ka;J=.0005;B=.01*J;Ref=0;Kp=5;Ki=1;Kd=1
//Initial Conditions
Vdc(1)=0;Theta(1)=0;Omega(1)=0;Ia(1)=0;P(1)=0;I(1)=0;D(1)=0;E(1)=0
//Time Step (Seconds)
dt=.001
//Initialize Counter and time
i=1;t(1)=0
//While loop to simulate motor and PID difference equation approximation
while i<1500, Theta(i+1)=Theta(i)+dt*Omega(i),
    Omega(i+1)=Omega(i)+dt*(-B*Omega(i)+Kt*Ia(i))/J,
    Ia(i+1)=Ia(i)+dt*(-Ka*Omega(i)-Ra*Ia(i)+Vdc(i))/La,
    E(i+1)=Ref-Theta(i+1),
    P(i+1)=Kp*E(i+1),
    I(i+1)=Ki*(I(i)+dt*E(i)+0.5*dt*(E(i+1)-E(i))),
    D(i+1)=Kd*(E(i+1)-E(i))/dt,
    Vdc(i+1)=P(i+1)+I(i+1)+D(i+1),
            //Check to see if Vdc has hit power supply limit
            if Vdc(i+1)>12 then Vdc(i+1)=12
            end
     t(i+1)=t(i)+dt,
     i=i+1,
            //Call for a new shaft position
            if i>5 then Ref=10
            end
end
```

*إلى هنا

الجزء يلي ما اظن اخذناه

Chapter 6

Classical PID Control

الجداول والمعادلات في هذه المحاضرة
ليست للحفظ وسيتم توفيرها بالامتحان

المطلوب بالامتحان القواعد

This chapter examines a particular control structure that has become almost universally used in industrial control. It is based on a particular fixed structure controller family, the so-called PID controller family. These controllers have proven to be robust and extremely beneficial in the control of many important applications.

PID stands for:

- **P** (*Proportional*)
- **I** (*Integral*)
- **D** (*Derivative*)

Historical Note

Early feedback control devices implicitly or explicitly used the ideas of proportional, integral and derivative action in their structures. However, it was probably not until Minorsky's work on ship steering* published in 1922, that rigorous theoretical consideration was given to PID control.

This was the first mathematical treatment of the type of controller that is now used to control almost all industrial processes.

* Minorsky (1922) "Directional stability of automatically steered bodies", *J. Am. Soc. Naval Eng*., 34, p.284.

PID Structure

Consider the simple SISO control loop shown in Figure 6.1:

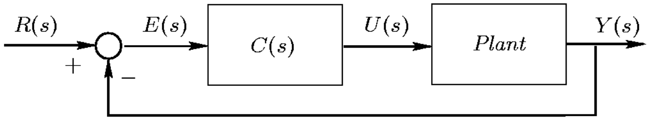


Figure 6.1: *Basic feedback control loop*

ليست داخلة في الامتحان

(ليست حفظ) لكنهم فهم

Error x Kp = قيمة الPower → Maximum Power < قيمة الPower

The *standard form* PID are:

Proportional only: $C_P(s) = K_p$

Proportional plus Integral: $C_{PI}(s) = K_p\left(1 + \frac{1}{T_r s}\right)$

Proportional plus derivative: $C_{PD}(s) = K_p\left(1 + \frac{T_d s}{\tau_D s + 1}\right)$

(مشتقة الError) ← يلي هو ميل المماس

Proportional, integral and derivative: $C_{PID}(s) = K_p\left(1 + \frac{1}{T_r s} + \frac{T_d s}{\tau_D s + 1}\right)$

$\frac{1}{s}$ → تكامل

MATLAB

Control System Toolbox

Transfer Function

$$H(s)=\frac{p_1 s^n + p_2 s^{n-1} + \ldots + p_{n+1}}{q_1 s^m + q_1 s^{m-1} + \ldots + q_{m+1}}$$

where

$p_1, p_2 \ldots p_{n+1}$ numerator coefficients

$q_1, q_1 \ldots q_{m+1}$ denominator coefficients

Copyright © 2017 by Pearson Education, Ltd. All Rights Reserved PEARSON

Control System Toolbox

Transfer Function

- Consider a linear time invariant (LTI) single-input/single-output system

$$y''+6y'+5y=4u'+3u$$

كل اشتقاق عبارة عن s

$\ddot{y} = s^2$

$\dot{y} = s$

وهكذا

- Applying Laplace Transform to both sides with zero initial conditions

$$G(s)=\frac{Y(s)}{U(s)}=\frac{4s+3}{s^2+6s+5}$$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Transfer Function

small

```
>> num = [4 3];
>> den = [1 6 5];
>> sys = tf(num,den)
   Transfer function:
     4 s + 3
   -----------------
   s^2 + 6 s + 5
```

```
>> [num,den] =
      tfdata(sys,'v')
 num =
       0   4    3
 den =
       1   6    5
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Zero-pole-gain model (ZPK)

zeros

$$H(s) = \underbrace{K}_{\text{gain}} \frac{(s-p_1)(s-p_2)+\ldots+(s-p_n)}{(s-q_1)(s-q_2)+\ldots+(s-q_m)}$$

Poles

(ZPK)

where

$p_1, p_2 \ldots p_{n+1}$ the zeros of H(s)

$q_1, q_1 \ldots q_{m+1}$ the poles of H(s)

Copyright © 2017 by Pearson Education, Ltd. All Rights Reserved

Control System Toolbox

Zero-pole-gain model (ZPK)

Matlab is case sensitive
zpk small letter

- Consider a Linear time invariant (LTI) single-input/single-output system

$$y''+6y'+5y=4u'+3u$$

- Applying Laplace Transform to both sides with zero initial conditions

اخذت 4 عامل مشترك

$$G(s)=\frac{Y(s)}{U(s)}=\frac{4s+3}{s^2+6s+5}=\frac{4(s+0.75)}{(s+1)(s+5)}$$

تحليل المقام

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Zero-pole-gain model (ZPK)

```
>> sys1 =
 zpk(-0.75,[-1 -5],4)
```

لو أكثر من جذر
فبحط أقواس مربع

```
Zero/pole/gain:
4 (s+0.75)
-----------
(s+1) (s+5)
```

```
>> [ze,po,k] = zpkdata(sys1,'v')
ze =
   -0.7500
po =
    -1
    -5
k =
     4
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

```
>> s=tf{‘s’}
>>G=(4*s+3)/(s^2+6*s+5)
```

Easiest way to write a transfer function

أي منطقة فيها ضرب لازم نحط *

This is the way suggested for the new versions in matlab

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved
PEARSON

On Matlab :-

(In the Command Window)

```
>> num =[4  3]
num =
        4   3
>> den =[1  6  5]
den =
        1   6   5
>> g=tf(num,den)
g =
       4s + 3
    -------------
    s^2+6s+5
>> g2=tf([1 3],[1 5 7])
g2 =   s + 3
     -----------
     s^2 + 5s+7
>> g2=tf([1 3],[1 7])                    ← modify g2
g2=  S+3
    -------
     S+7
>> g2=tf([1 3],[1 0 7])                  ← modify g2
g2 =   s+3
     --------
     S^2+7
>> series (g,g2)
ans =
             4S^2+15S+9
    ------------------------------
    S^4+6S^3+12S^2+42S+35
>> parralel (g,g2)
ans =        5S^3+12S^2+51S+36
    ------------------------------
    S^4+6S^3+12S^2+42S+35
```

```
>> feedback (g,-1)
```

← unity feedback

```
ans =      4s + 3
       -------------
       s^2 + 2s + 2
```

```
>> feedback (g , g2 , -1)
```

← not unity feedback

g: followed function

g2: backward (feedback) function

```
ans =
        4s^3 + 3s^2 + 28s + 21
   ----------------------------------
   s^4 + 6s^3 + 16s^2 + 57s + 44
```

```
>> f =tf data (g)
f =
   1x1 cell array
    {[0  4  3]}
```

← للترجيع للمعاملات الأصلية

انتباه للأقواس

```
>> [num , den] = tf data (g)
num =
      1x1 cell array
      {[0  4  3]}
den =
      1x1 cell array
      {[1  6  5]}
>> num {1}
ans =
        0   4   3
>> num {1} (1)
ans =
        0
```

هذه الأقواس لـ objects

الأقواس الدائرية لـ Arrays or functions

بده العنصر الأول اظن

* يوجد فرق بينهم *

```
>> num1 = [ 0   4   3]
   num1 =
          0   4   3
```

← array

```
>> num
 num= 1x1  cell array
       {[ 0  4  3 ]}
```

← object

```
>> num {1}
  ans =
         0   4   3
>> num 2  = num
  num2 =
       1 x 1  cell array
      {[0  4  3]}
>> num2  = num {1}
  num2 =
          0  4  3
```

```
>> num{2} = 5
num =
    1x2  cell  array
  {[0 4 3]}   {[5]}
```

العنصرالأول ↗ ↖ العنصرالثاني

```
>> num{3} = 'd'
num =
    1x3  cell  array
  {[0 4 3]}   {[5]}   {'d'}
```

* مومتأكدة [array للتحويل إلى

First Method

```
>> num2 = num{1}
num2 =
        0   4   3
```

Second Method

```
>> g3 = zpk(-0.75, [-1 -5], 4)
g3 =   4 (s+0.75)
      ------------
      (s+1) (s+5)
```

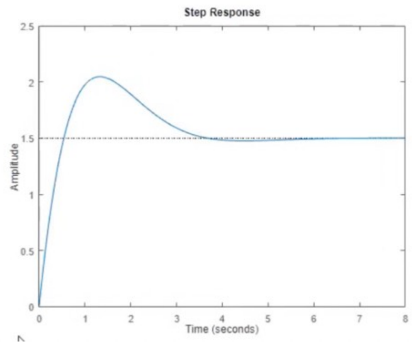
or

```
>> s=tf('s')
s =
     s
```

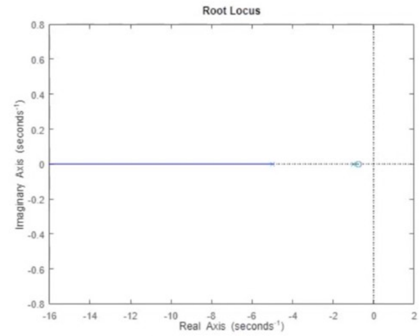
* اظن الاقواس غلط { }

```
>> g3= (4*s +3)/(s^2 +6*s + 5)
g3 =    4s+3
      ----------
      s^2 +6 s +5
```

```
>> step (feedback (g, -1))
```



```
>> rlocus (g)
```

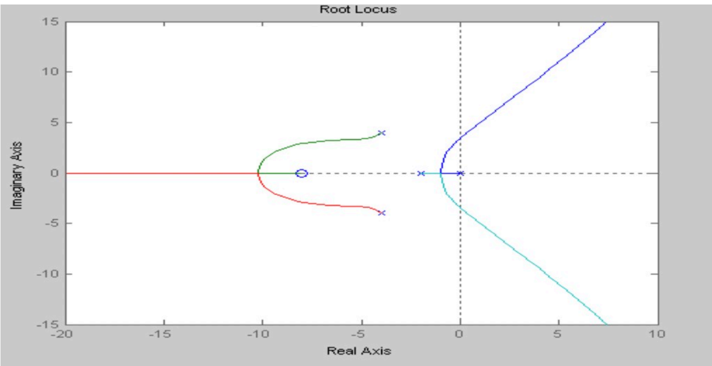


Plot the root locus of the following system

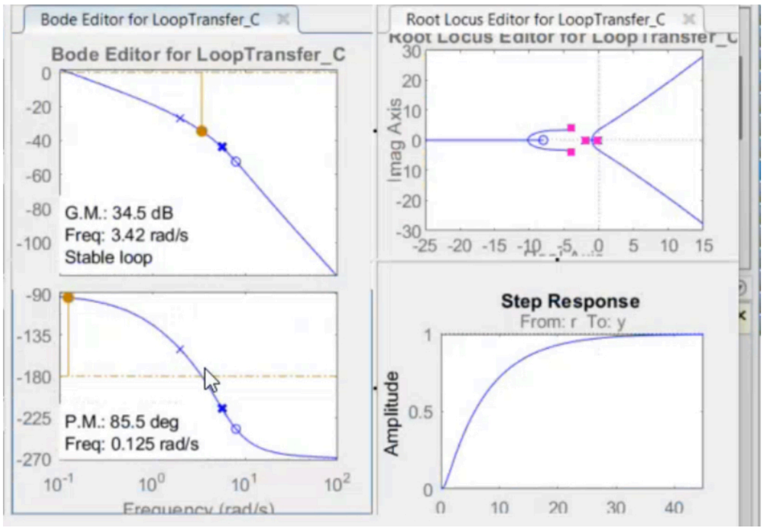
$$G(s) = \frac{K(s+8)}{s(s+2)(s^2+8s+32)}$$

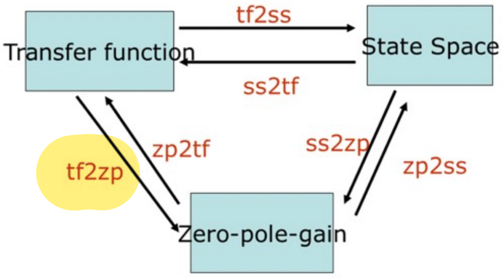
* in Matlab

```
>> num = [1 8]
num =
      1   8
>> den = conv(conv([1 0],[1 2]),[1 8 32])
den =
 1  10  48  64  0
>> sys = (num, den)
          s+8
 ----------------------
 s^4 +10s^3 + 48s^2 +64s
>> rlocus (sys)   OR   >> rlocus(tf([1 8],conv(conv([1 0],[1 2]),[1 8 32])))
```



```
>> sisotool(tf([1 8],conv(conv([1 0],[1 2]),[1 8 32])))
```





```
≫tf2zpk([4 3],[1 6 5])
ans =  0
       0
   -0.7500
≫[z,p,k]=tf2zpk([4 3],[1 6 5])
```

*State Space Model

slides

```
≫tf2ss([4 3],[1 6 5])
ans =
        -6   -5
         1    0

≫[A,B,C,D]=tf2ss([4 3],[1 6 5])
A = -6   -5
     1    0

B =  1
     0

C =  4  3

D =  0

≫[num,den]=ss2tf(A,B,C,D)
num =
      0   4   3
den = 1   6   5
```

Control System Toolbox

State-Space Model (SS)

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where

x	state vector
u *and* y	input and output vectors
A, B, C *and* D	state-space matrices

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

y'' → يعني راح امثلها ب two states

double prime

Control System Toolbox

State-Space Models

- Consider a Linear time invariant (LTI) single-input/single-output system

$$y''+6y'+5y=4u'\cancel{'}+3u$$

- State-space model for this system is

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ -5 & -6 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}+\begin{bmatrix} 0 \\ 1 \end{bmatrix}u \qquad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$y=\begin{bmatrix} 3 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Copyright © 2017 by Pearson Education, Ltd. All Rights Reserved PEARSON

$$\frac{Y}{R}=\frac{4s+3}{s^2+6s+5}=\frac{1}{s^2+6s+5}*(4s+3)=\frac{X}{R}*\frac{Y}{X}$$

$$\frac{1}{s^2+6s+5}=\frac{X}{R}$$

→ هنا بجعل موضوع القانون (R)

$$(4s+3)=\frac{Y}{X}$$

→ هنا بجعل موضوع القانون (Y)

To differential eq.

ثم نشتق

$$R=X''+6X'+5X; \qquad \underline{Y=4X'+3X}$$

X1,X2; X1'=X2; X2'=(-5X1-6X2) +R

In Matrix form

$$\begin{bmatrix} X1' \\ X2' \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ -5 & -6 \end{bmatrix}*\begin{bmatrix} X1 \\ X2 \end{bmatrix}+\begin{bmatrix} 0 \\ 1 \end{bmatrix}R$$

$$Y=\begin{bmatrix} 3 & 4 \end{bmatrix}*\begin{bmatrix} X1 \\ X2 \end{bmatrix}+[0]R$$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

State-Space Models

```
>> sys = ss([0 1; -5 -6],[0;1],[3,4],0)
 a =                                          c =
       x1  x2                                       x1  x2
   x1   0   1                                   y1   3   4
   x2  -5  -6


 b =                                          d =
       u1                                           u1
   x1   0                                       y1   0
   x2   1
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

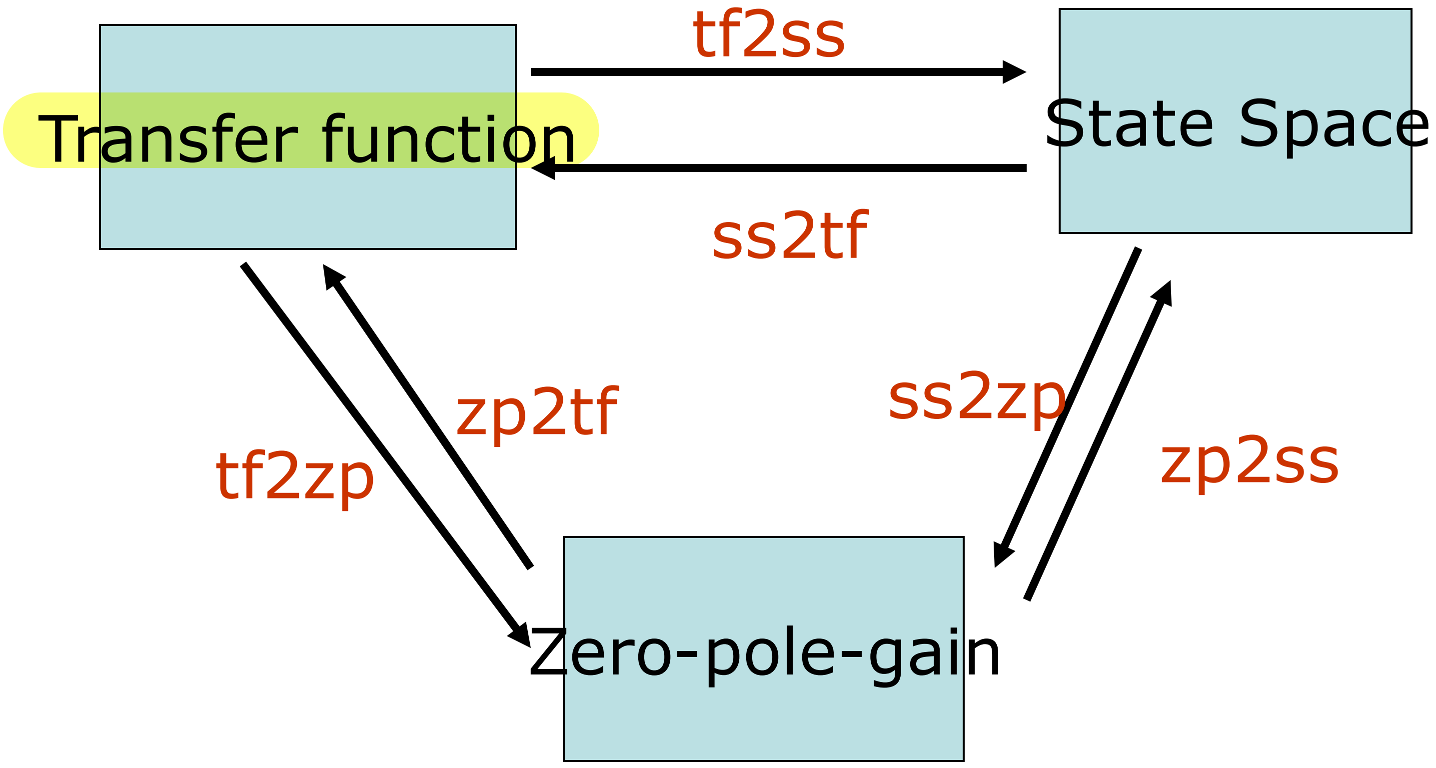
Control System Toolbox

State Space Models

- **rss, drss** - **Random stable state-space models.**
- **ss2ss** - **State coordinate transformation.**
- **canon** - **State-space canonical forms.**
- **ctrb** - **Controllability matrix.**
- **obsv** - **Observability matrix.**
- **gram** - **Controllability and observability gramians.**
- **ssbal** - **Diagonal balancing of state-space realizations.**
- **balreal** - **Gramian-based input/output balancing.**
- **modred** - **Model state reduction.**
- **minreal** - **Minimal realization and pole/zero cancellation.**
- **sminreal** - **Structurally minimal realization.**

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Conversion between different models



Transfer function

State Space

Zero-pole-gain

tf2ss

ss2tf

zp2tf

tf2zp

ss2zp

zp2ss

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Time Response of Systems

- Impulse Response *(impulse)*
- Step Response *(step)*
- General Time Response *(lsim)*
- Polynomial multiplication (*conv)*
- Polynomial division *(deconv)*
- Partial Fraction Expansion *(residue)*
- gensig - Generate input signal for lsim.

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Time Response of Systems

- The impulse *response* of a system is its output when the *input is a unit impulse*.
- The step *response* of a system is its output when the *input is a unit step*.
- The general *response* of a system to *any input* can be computed using the lsim command.

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Time Response of Systems

Problem Given the LTI system

$$G(s) = \frac{3s+2}{2s^3+4s^2+5s+1}$$

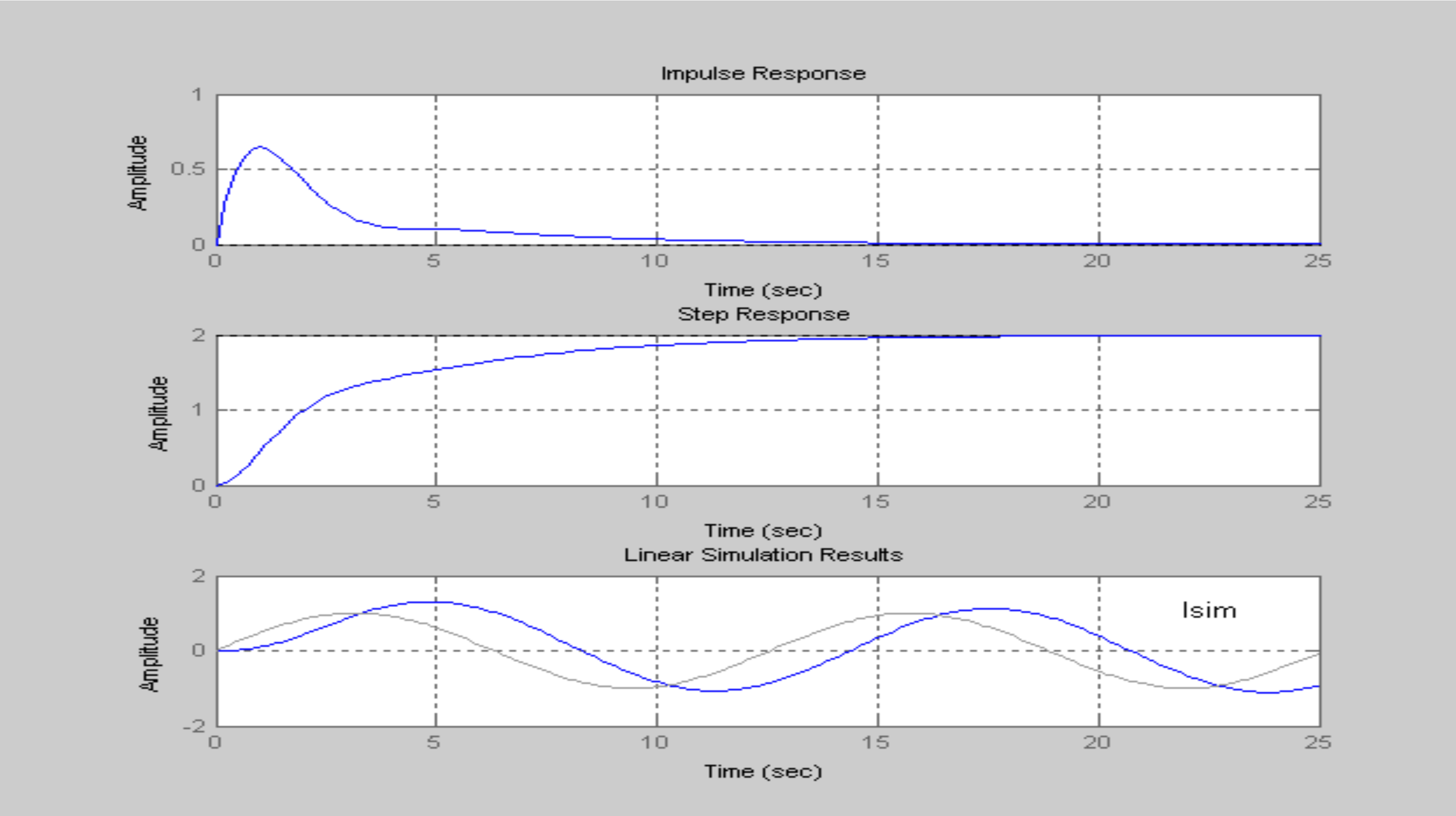
Plot the following responses for:

- The impulse response using the `impulse` command.
- The step response using the `step` command.
- The response to the input $u(t) = sin(0.5t)$ calculated using both the `lsim` commands

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Time Response of Systems



Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Frequency Domain Analysis and Design

Root Locus

- Plot the root locus of the following system

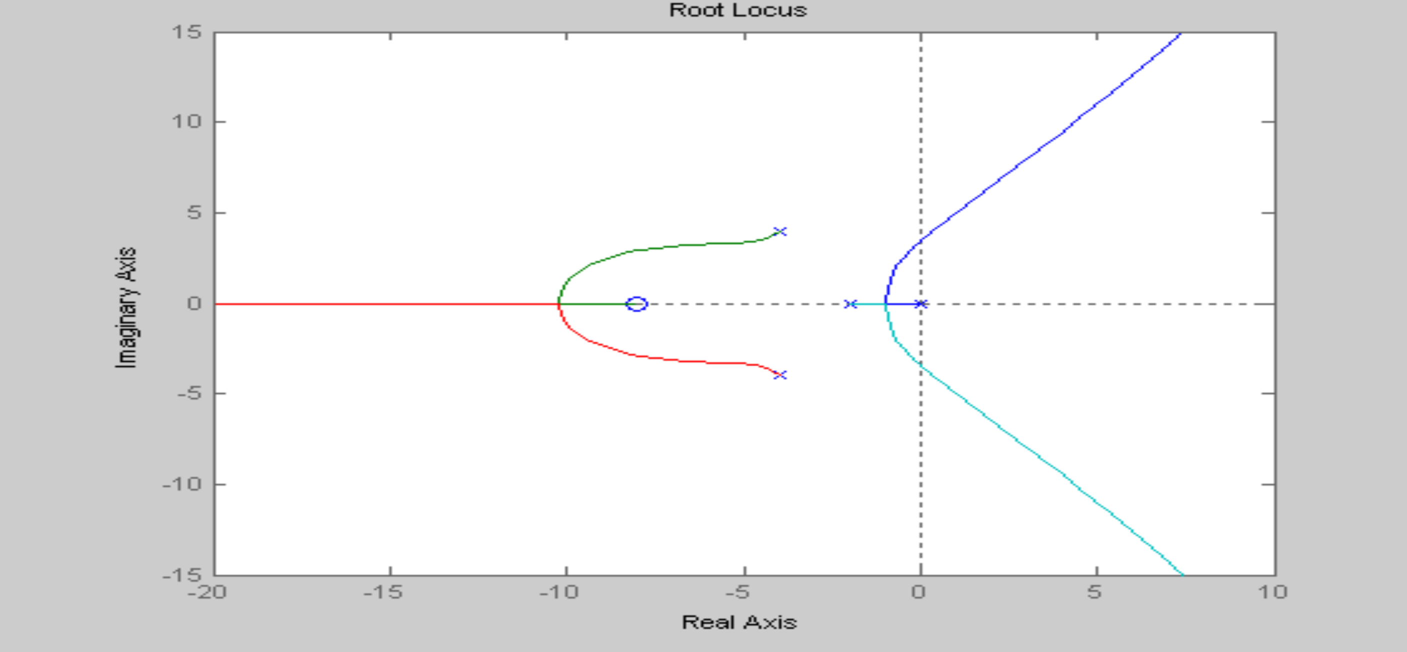
$$G(s) = \frac{K(s+8)}{s(s+2)(s^2+8s+32)}$$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Frequency Domain Analysis and Design

Root Locus

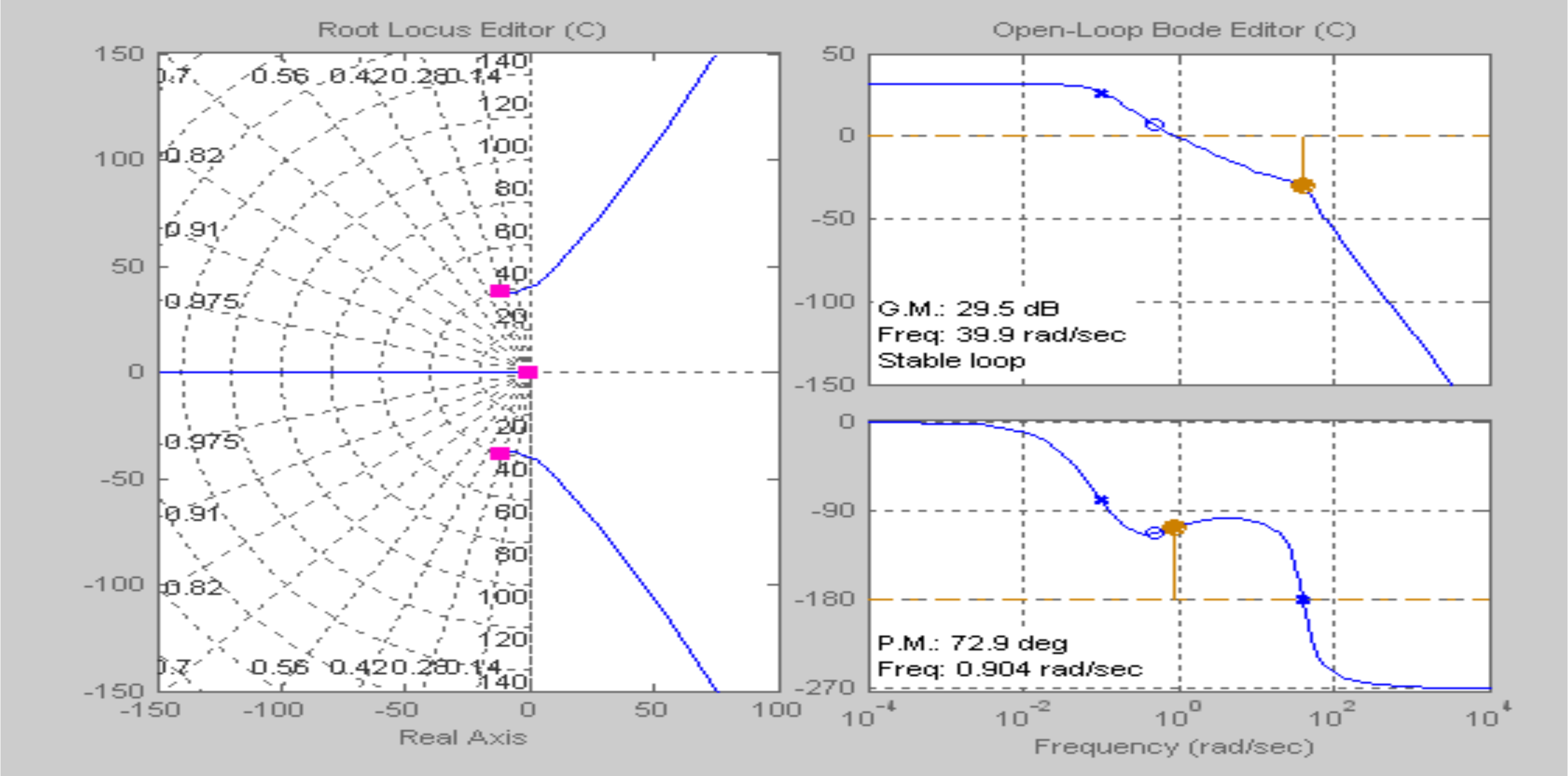
```
>> rlocus(tf([1 8], conv(conv([1 0],[1 2]),[1 8 32])))
```



Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Control System Toolbox

Design Tool: sisotool



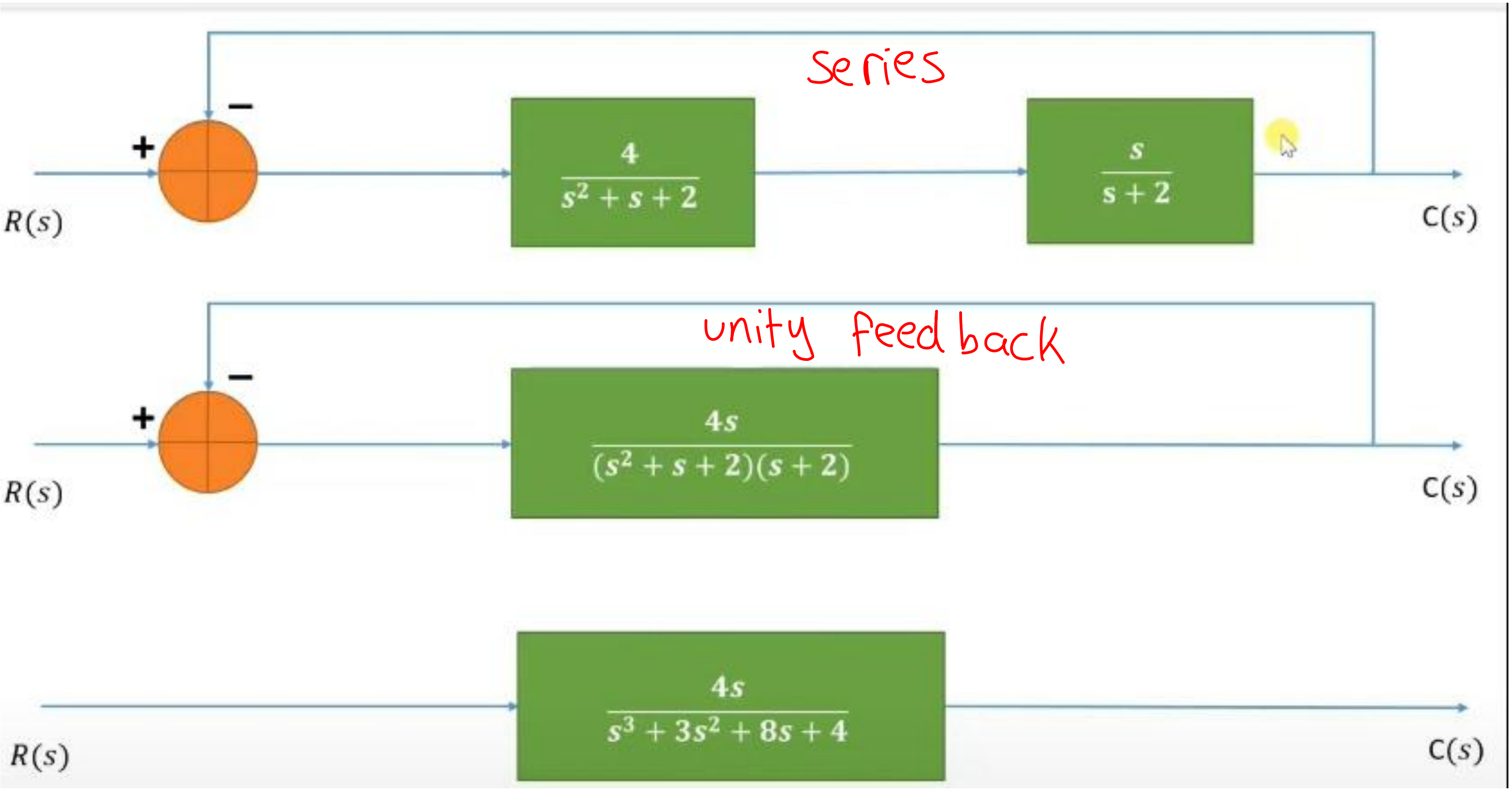
G.M.: 29.5 dB
Freq: 39.9 rad/sec
Stable loop

P.M.: 72.9 deg
Freq: 0.904 rad/sec

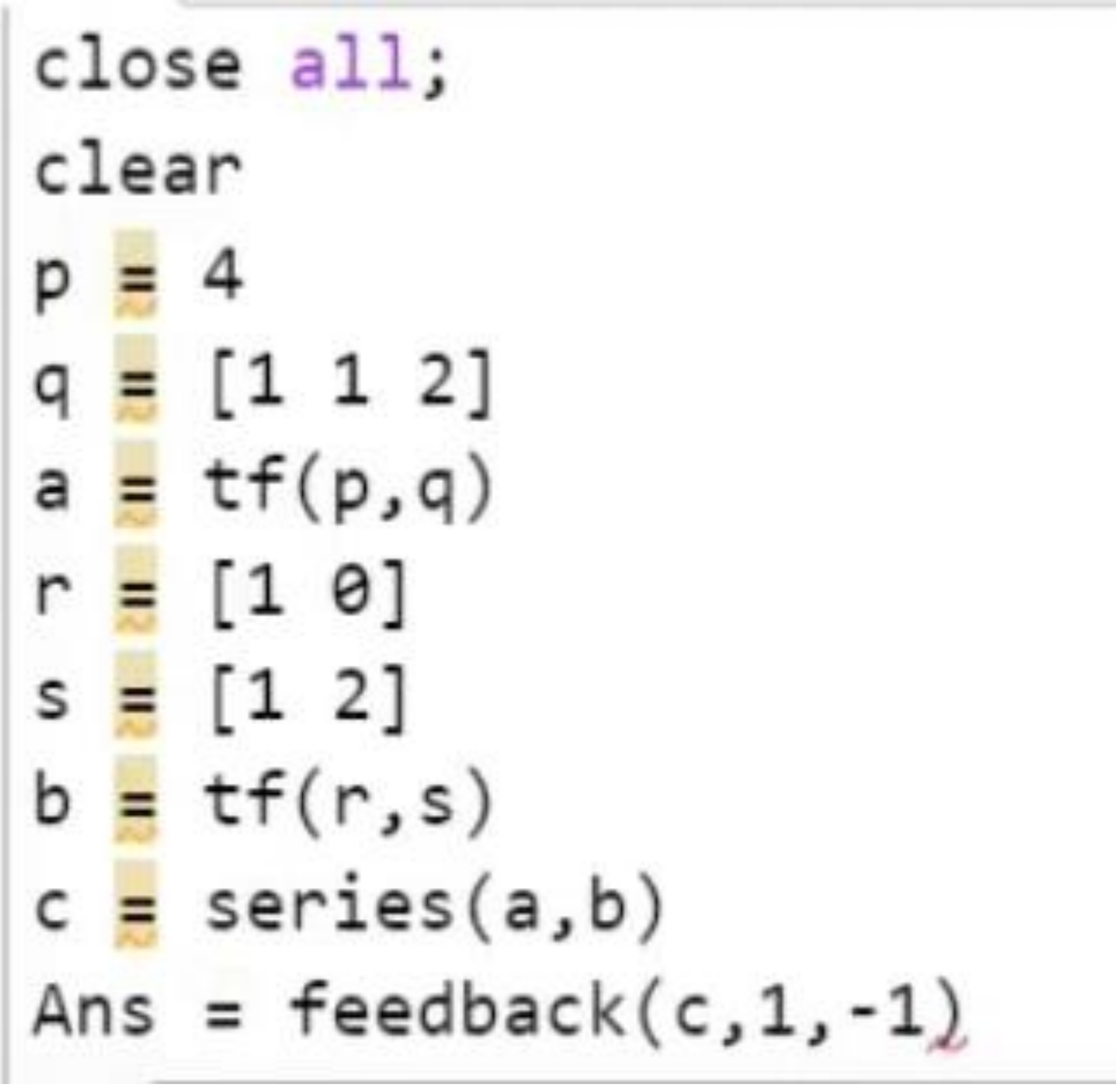
Design with root locus, Bode, and Nichols plots of the open-loop system.

Cannot handle continuous models with time delay.

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved
PEARSON



Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved PEARSON



```
close all;
clear
p = 4
q = [1 1 2]
a = tf(p,q)
r = [1 0]
s = [1 2]
b = tf(r,s)
c = series(a,b)
Ans = feedback(c,1,-1)
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Solving differential Equation using MATLAB

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Solving Differential Equations in MATLAB

MATLAB have lots of built-in functionality for solving differential equations. MATLAB includes functions that solve ordinary differential equations (ODE) of the form:

$$\boxed{\frac{dy}{dt} = f(t, y),} \qquad \boxed{y(t_0) = y_0}$$

MATLAB can solve these equations numerically.

Higher order differential equations must be reformulated into a system of first order differential equations.

* مهما كانت درجة ال diffrential الأولى
لازم نحول لمعادلات مافيها الا diffrentiation بالدرجة الأولى

Note! Different notation is used:

$$\boxed{\frac{dy}{dt} = y' = \dot{y}}$$

Not all differential equations can be solved by the same technique, so MATLAB offers lots of different ODE solvers for solving differential equations, such as **ode45**, **ode23**, **ode113**, etc.

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Bacteria Population

In this task we will simulate a simple model of a bacteria population in a jar.

The model is as follows:

$$\text{birth rate} = bx$$

$$\text{death rate} = px^2$$

Then the total rate of change of bacteria population is:

$$\dot{x} = bx - px^2$$

b p

Set ***b=1***/hour and ***p=0.5*** bacteria-hour

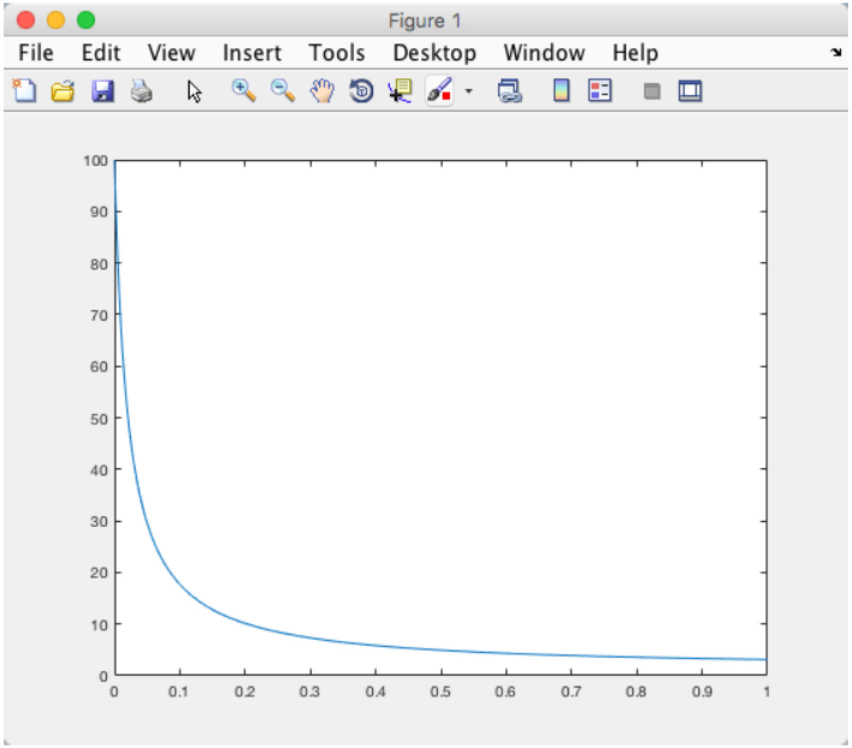
→ Simulate the number of bacteria in the jar after **1 hour**, assuming that initially there are **100 bacteria** present.

x_0

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved
PEARSON

```
function dx = bacteriadiff(t,x)
% My Simple Differential Equation

b=1;
p=0.5;

dx = b*x - p*x^2;
```

```
clear
clc
       صفر ساعة   ساعة واحدة
tspan=[0 1];
x0=100;

[t,y]=ode45(@bacteriadiff, tspan,x0);
plot(t,y)

[t,y]
```



Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Passing Parameters to the model

Given the following system (1.order differential equation):

$$\boxed{\dot{x} = ax + b}$$

where $\boxed{a = -\frac{1}{T}}$,where T is the time constant

In this case we want to pass a and b as parameters, to make it easy to be able to change values for these parameters

We set $\boxed{b = 1}$

We set initial condition $\boxed{x(0) = 1}$ and $\boxed{T = 5}$.

Solve the Equation and Plot the results with MATLAB

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved PEARSON

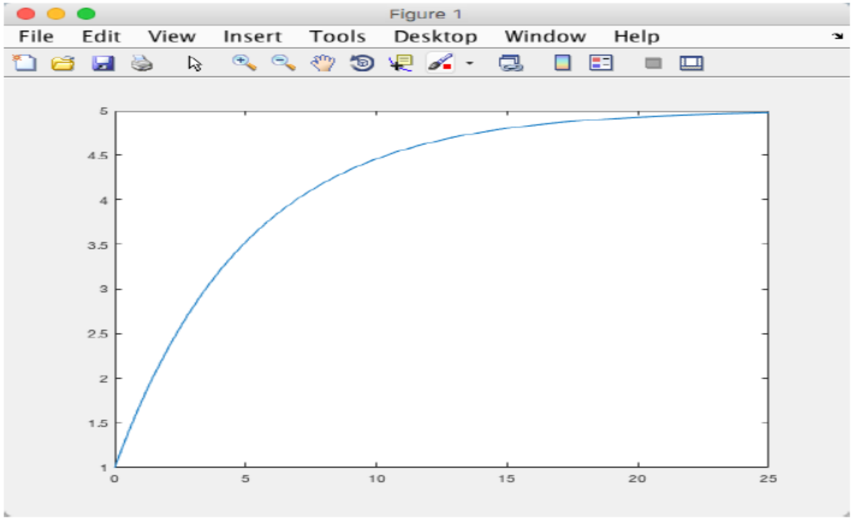
```
function dx = mysimplediff(t,x,param)
% My Simple Differential Equation

a=param(1);
b=param(2);

dx=a*x+b;
```

By doing this, it is very easy to changes values for the parameters a and b.

Note! We need to use the 5. argument in the ODE solver function for this. The 4. argument is for special options and is normally set to "[]", i.e., no options.

?

```
tspan=[0 25];
x0=1;
a=-1/5;
b=1;
param=[a b];

[t,y]=ode45(@mysimplediff, tspan,
x0,[], param);
plot(t,y)
```



Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Differential Equation

Use the ode23 function to solve and plot the results of the following differential equation in the interval $\boxed{[t_0, t_f]}$:

$$\boxed{w' + (1.2 + sin10t)w = 0}$$

Where:

$$t_0 = 0$$
$$t_f = 5$$
$$w(t_0) = 1$$

tspan[0 5]

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved
PEARSON

Differential Equation

We start by rewriting the differential equation:

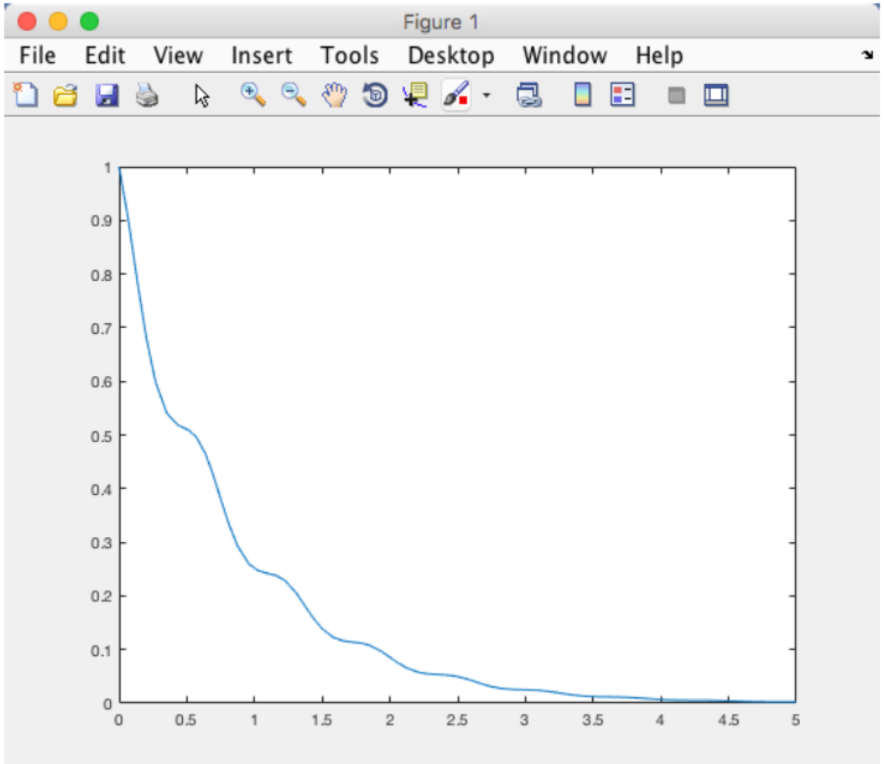
$$\longrightarrow \boxed{w' = -(1.2 + sin10t)w}$$

Then we can implement it in MATLAB

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

```
function dw = diff_task3(t,w)

dw = -(1.2 + sin(10*t))*w;
```

```
tspan=[0 5];
w0=1;

[t,w]=ode23(@diff_task3, tspan, w0);
plot(t,w)
```



Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

2.order differential equation

Use the ode23/ode45 function to solve and plot the results of the following differential equation in the interval $[t_0, t_f]$:

عشان اتخلص من ال $\ddot{w}$

$w = x_1$
$\dot{w} = x_2$
بعرف two variables

$$(1 + t^2)\ddot{w} + 2t\dot{w} + 3w = 2$$

هنا المثال
من الدرجة الثانية

* لازم احوله للدرجة الأولى

Where; , $t_0 = 0, t_f = 5, w(t_0) = 0, \dot{w}(t_0) = 1$

Note! Higher order differential equations must be reformulated into a system of first order differential equations.

Tip 1: Reformulate the differential equation so $\ddot{w}$ is alone on the left side.

Tip 2: Set:

$$\boxed{w = x_1}$$
$$\boxed{\dot{w} = x_2}$$

$$\ddot{w} = \frac{2 - 2t\dot{w} - 3w}{1 + t^2}$$

$$\ddot{w} = \dot{x}_2 = \frac{2 - 2t x_2 - 3x_1}{1 + t^2}$$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved
PEARSON

من ناحية الـ function

اذا كان سطر dx بكون سطر
اذا كان عمود dx بكون عمود

```
function dx = diff_secondorder(t,x)

[m,n] = size(x);
dx = zeros(m,n)

dx(1) = x(2);
dx(2) = (2-2*t*x(2)-3*x(1))/(1+t^2);
```

عادي نضع اي اسم لكن بشرط نستخدمه نفسه

اسطر اعمدة

$\longrightarrow \dot{x}_1 = x_2$

from previous slide

$\longrightarrow \dot{x}_2 = \frac{2-2tx_2-3x_1}{(1+t^2)}$

من ناحية الـ Matlab
Command window

الفترة الزمنية

```
tspan=[0 5];
x0=[0; 1];

[t,x]=ode23(@diff_secondorder, tspan, x0);
plot(t,x)
legend('x1','x2')
```

semicolumn
يعني سطر جديد

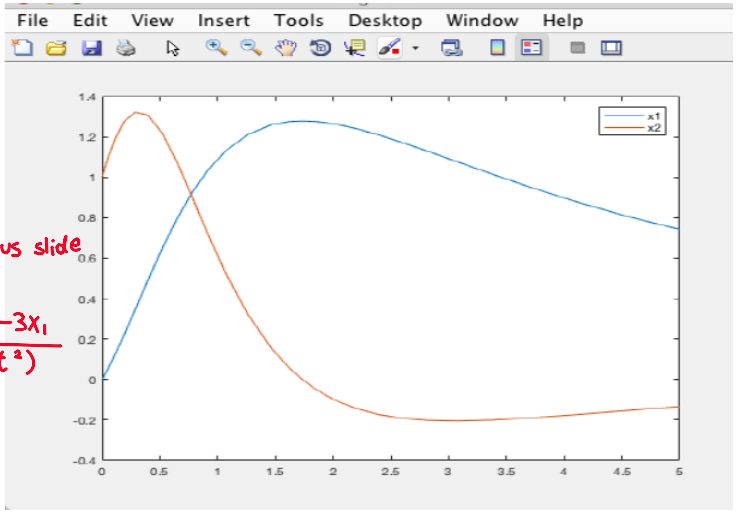
يوجد functions مختلفة
كل واحد بيعمل integration خاص فيه

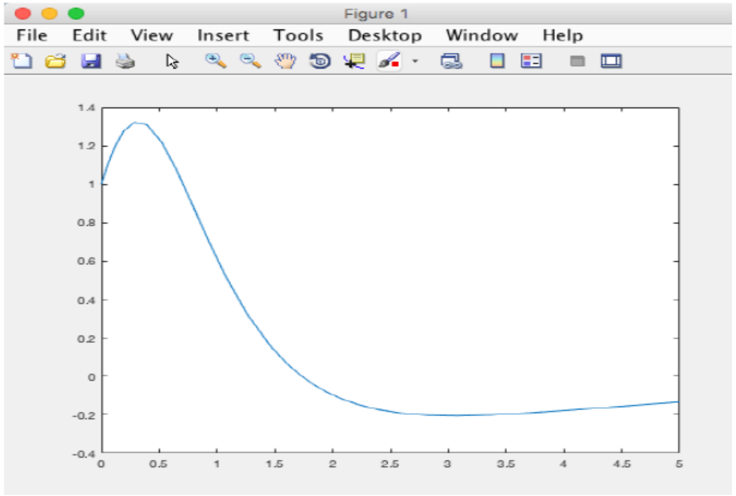
ode23
ode24
ode33
⋮

الفترة الزمنية

شرط استخدام نفس الاسم

```
tspan=[0 5];
x0=[0; 1];

[t,x]=ode23(@diff_secondorder, tspan, x0);
plot(t, x(:,2))
```





Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved PEARSON

On Matlab :-

نكتبه فوق في ال function →

اي اسم مابفرق

```
function  dx = dfe(t,x)
   [m,n]=size(x);
     dx = zeros(m,n);
     dx(1) = x(2);
     dx(2) = (2-2*t*x(2)-3*x(1))/(1+t^2);
```

→ عندما نضغط على save راح يكون اسمه dfe نفس الاسم تبعي

الان نكتب على ال command window (تحت)

```
>>tspan=[0 5]
       0   5
>>x0 =[0 1]
   x0 =  0  1
```

```
>>x0 =[0;1]
```

x_{node}

```
   x0 =
       0
       1
```

انتباه يوجد فرق بينهم

; ← يعني سطر جديد

```
>>[t, x] = ode45(@dfe,tspan,x0)
```

```
>>Plot(t,x)
```

اذا بدي اياه يرسملي ال x فقط وما بدنا يرسم $\dot{x}$ →

```
>plot(t,x(:,1))
```

← يعني كل الاسطر في العمود الأول

2.order differential equation

$w = x_1$

اشتقاق ↓

$\dot{w} = \dot{x}_1 = x_2$

اشتقاق ↓

$\ddot{w} = \ddot{x}_1 = \dot{x}_2$

Tip1: First we rewrite like this:

$$\ddot{w} = \frac{2 - 2t\dot{w} - 3w}{(1 + t^2)}$$

Tip2: In order to solve it using the ode functions in MATLAB it has to be a set with 1.order ode's. So we set:

$$\boxed{w = x_1}$$

$$\boxed{\dot{w} = x_2}$$

← معناه مشتقة

ال x_1 هي x_2

→ $\dot{x}_1 = x_2$

This gives 2 first order differential equations:

$$\boxed{\dot{x}_1 = x_2}$$

$$\ddot{w} = \dot{x}_2 = \frac{2 - 2tx_2 - 3x_1}{(1 + t^2)}$$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Symbolic Differentiation

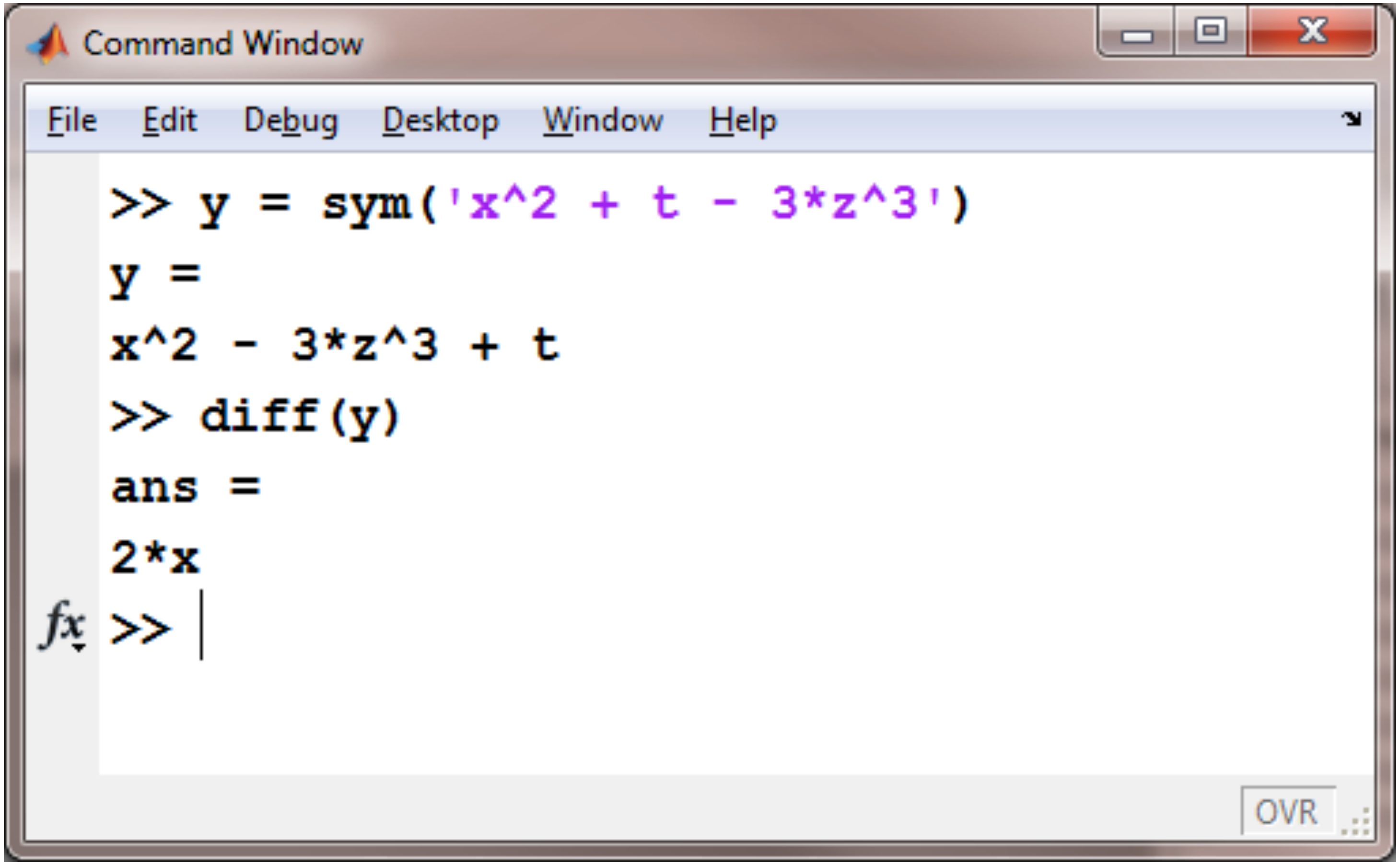
`diff(f)`	Returns the derivative of the expression **`f`** with respect to the default independent variable	**y=sym('x^3+z^2')** **diff(y)** **ans =** **3*x^2**
`diff(f,'t')`	Returns the derivative of the expression **`f`** with respect to the variable **`t`**.	**y=sym('x^3+z^2')** **diff(y,'z')** **ans =** **2*z**
`diff(f,n)`	Returns the **n**th derivative of the expression **`f`** with respect to the default independent variable	**y=sym('x^3+z^2')** **diff(y,2)** **ans =** **6*x**
`diff(f,'t',n)`	Returns the **n**th derivative of the expression **`f`** with respect to the variable **`t`**.	**y=sym('x^3+z^2')** **diff(y,'z',2)** **ans =** **2**

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Partial Derivatives

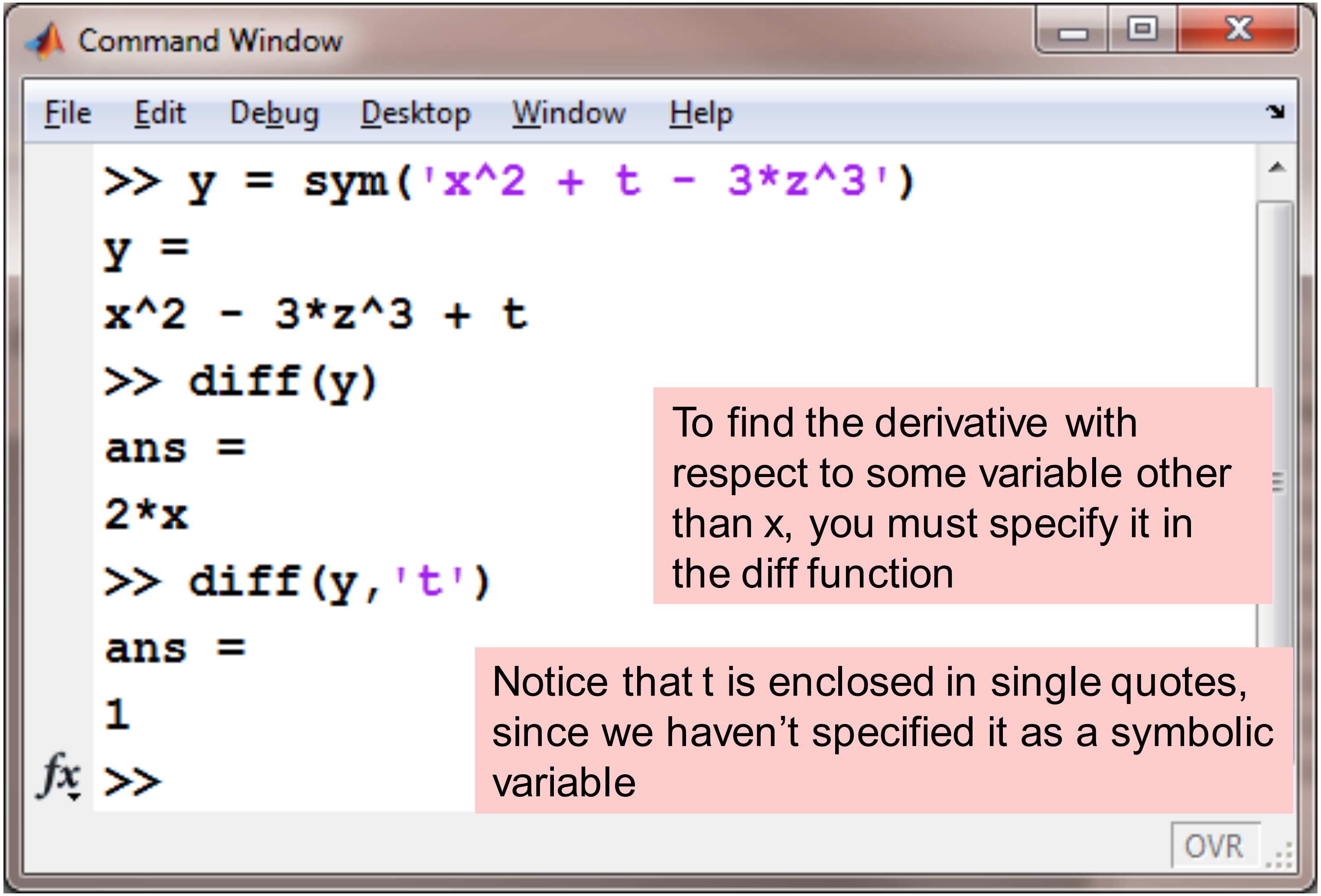
- If you have multiple variables, MATLAB takes the derivative with respect to x – unless you specify otherwise
- All the other variables are kept constant

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved



```
>> y = sym('x^2 + t - 3*z^3')
y =
x^2 - 3*z^3 + t
>> diff(y)
ans =
2*x
>>
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved



```
>> y = sym('x^2 + t - 3*z^3')
y =
x^2 - 3*z^3 + t
>> diff(y)
ans =
2*x
>> diff(y,'t')
ans =
1
>>
```

To find the derivative with respect to some variable other than x, you must specify it in the diff function

Notice that t is enclosed in single quotes, since we haven't specified it as a symbolic variable

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Integration

- Usually introduced in Calculus II
- Often visualized as the area under a curve
- MATLAB has built in symbolic integration capability.

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Consider a piston cylinder device

- Work done by a piston cylinder device as it moves up or down, can be calculated by taking the integral of P with respect to V

$$W = \int_1^2 P dV$$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved PEARSON

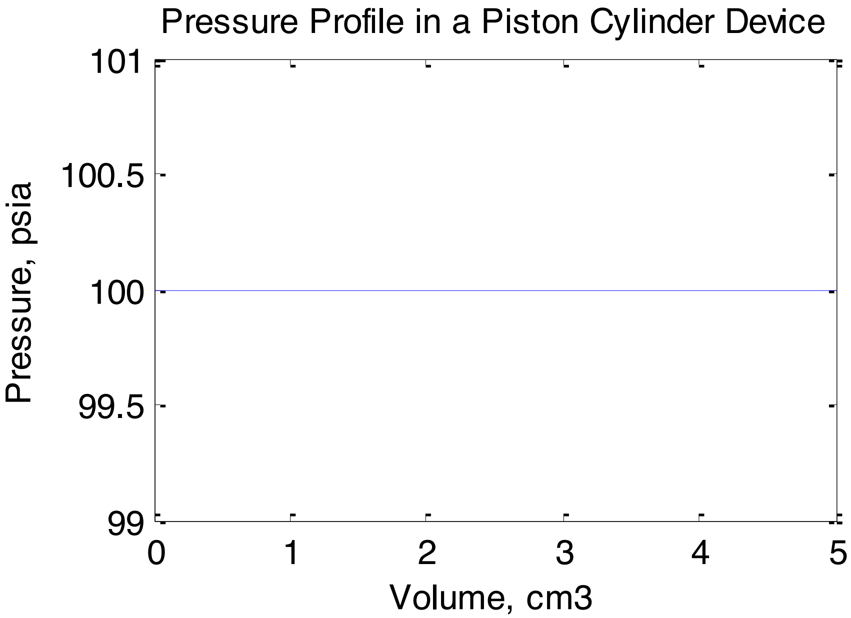
To perform the integration we need to know how P changes with V

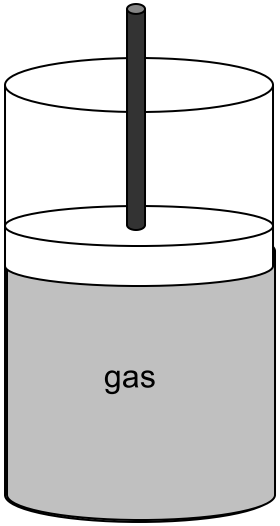
- If P is constant the problem becomes

$$W = P\int_1^2 dV$$

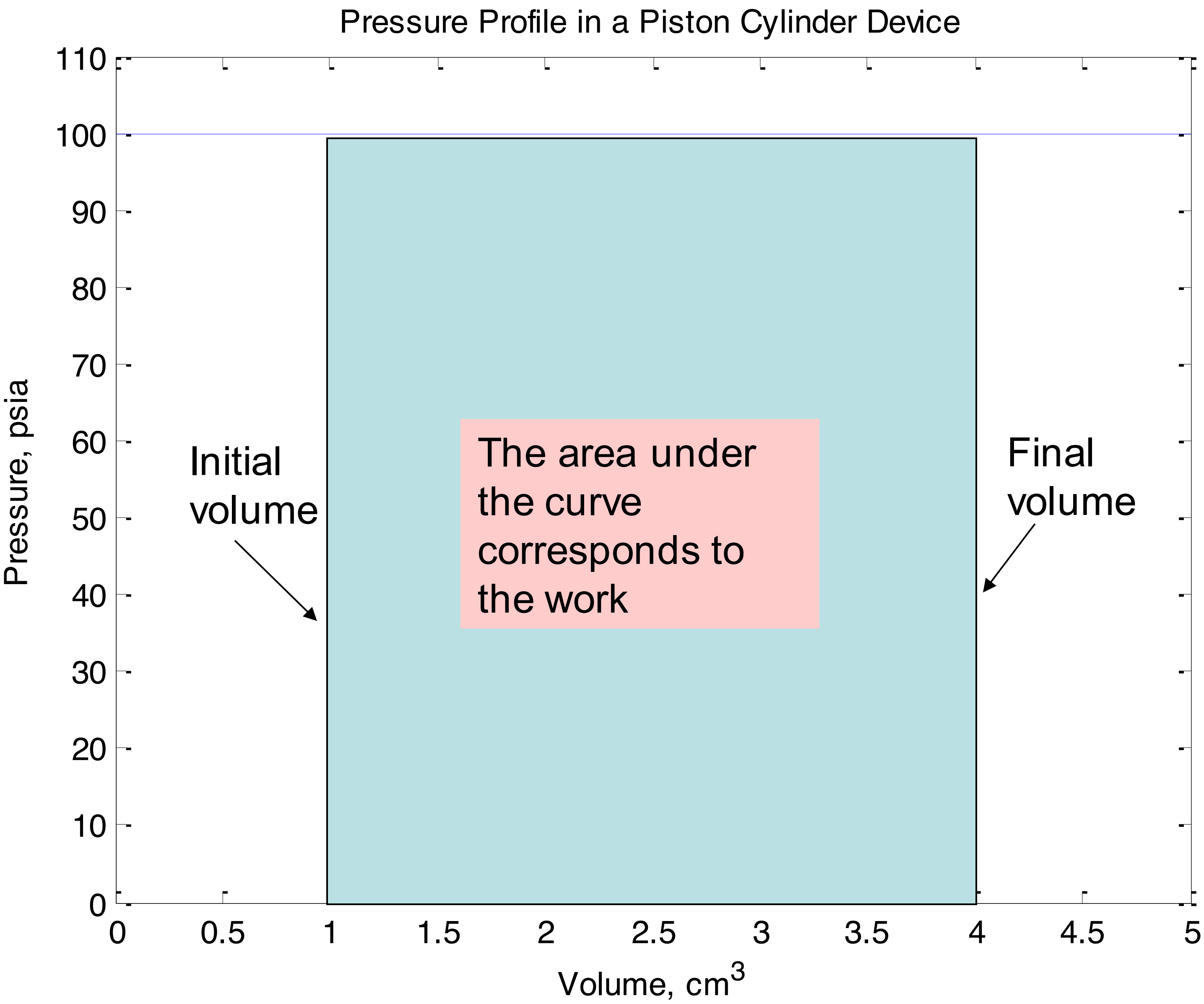
Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Model of the behavior of a piston cylinder device





Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved
PEARSON



The area under the curve corresponds to the work

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Hand Calculation

$$W = \int_1^4 PdV = P\int_1^4 dV = PV\Big|_1^4 = PV_4 - PV_1 = P\Delta V$$

$$if \quad P = 100\,psia$$

$$W = 3cm^3 * 100\,psia$$

Read this as: Work is equal to the integral of P with respect to V, from V=1 to V=4

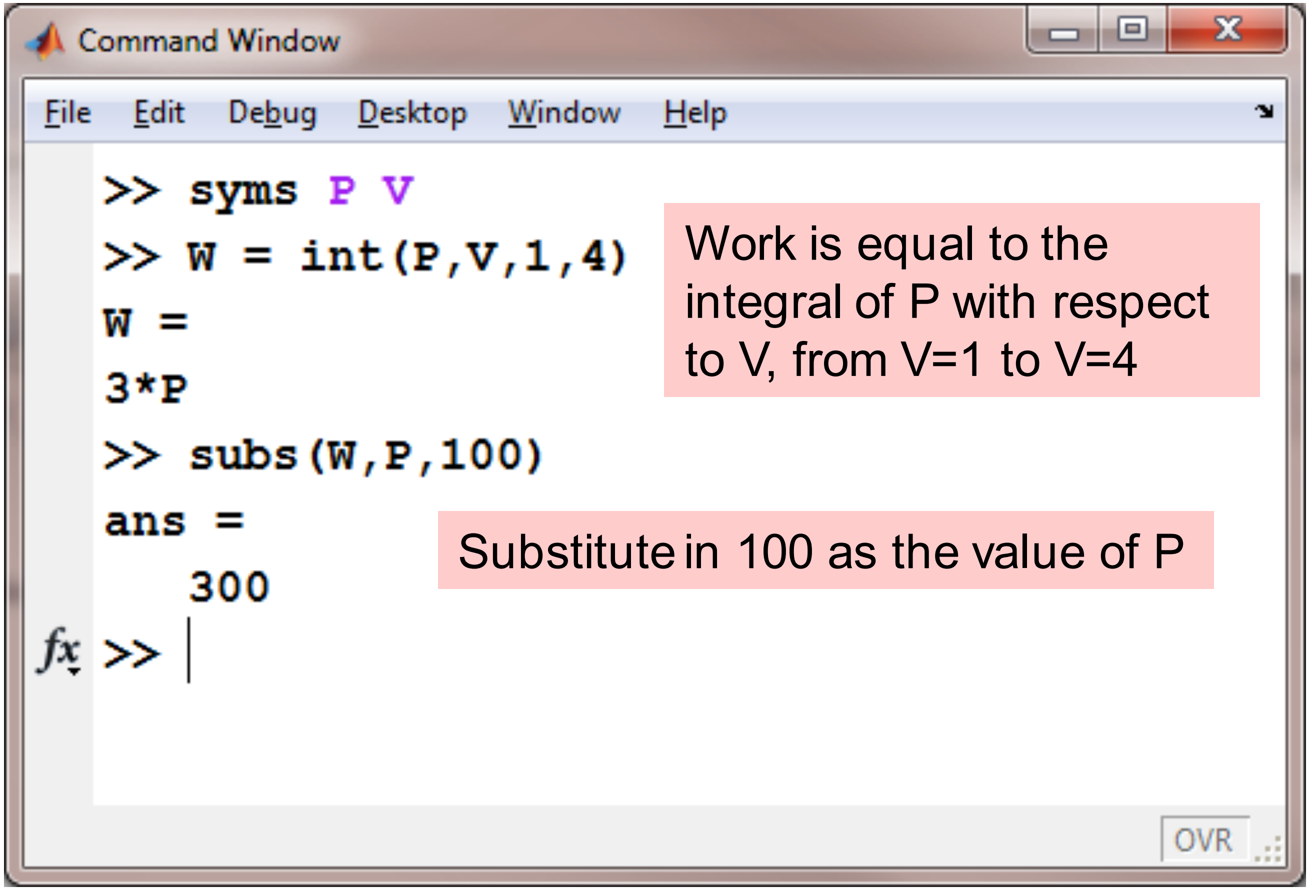
Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

* Note :-

Numirical solution there is always a solution

Symbolic there is not always a solution

MATLAB Solution



```
>> syms P V
>> W = int(P,V,1,4)
W =
3*P
>> subs(W,P,100)
ans =
   300
>> 
```

Work is equal to the integral of P with respect to V, from V=1 to V=4

Substitute in 100 as the value of P

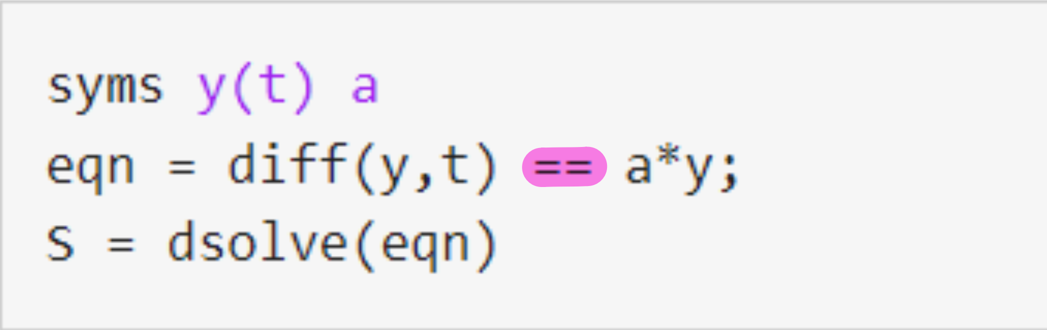
Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Symbolic Integration

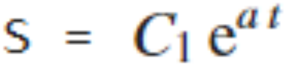
`int(f)`	Returns the integral of the expression **`f`** with respect to the default independent variable	**y=sym('x^3+z^2') int(y) ans = 1/4*x^4+z^2*x**
`int(f,'t')`	Returns the integral of the expression **`f`** with respect to the variable **`t`**.	**y=sym('x^3+z^2') int(y,'z') ans = x^3*z+1/3*z^3**
`int(f,a,b)`	Returns the integral with respect to the default variable, of the expression **`f`** between the numeric bounds, a and b.	**y=sym('x^3+z^2') int(y,2,3) ans = 65/4+z^2**
`int(f,'t',a,b)`	Returns the integral with respect to the variable **t**, of the expression **`f`** between the numeric bounds, a and b.	**y=sym('x^3+z^2') int(y,'z',2,3) ans = x^3+19/3**
`int(f,'t',a,b)`	Returns the integral with respect to the variable **t**, of the expression **`f`** between the symbolic bounds, a and b.	**y=sym('x^3+z^2') int(y,'z','a','b') ans = x^3*(b-a)+1/3*b^3-1/3*a^3**

Symbolic solution of differential equation

اهم من التكامل



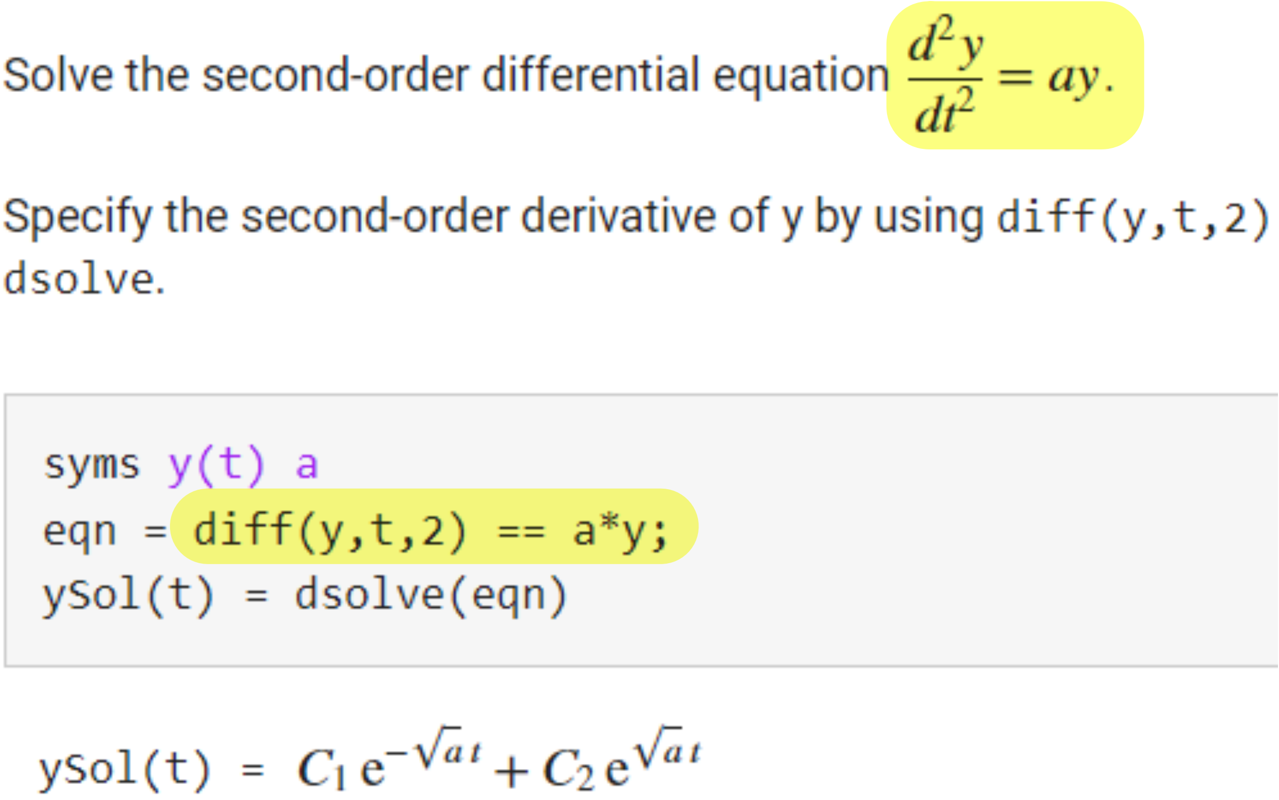
```
syms y(t) a
eqn = diff(y,t) == a*y;
S = dsolve(eqn)
```



$S = C_1 e^{a t}$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Second Order



Solve the second-order differential equation $\frac{d^2y}{dt^2} = ay$.

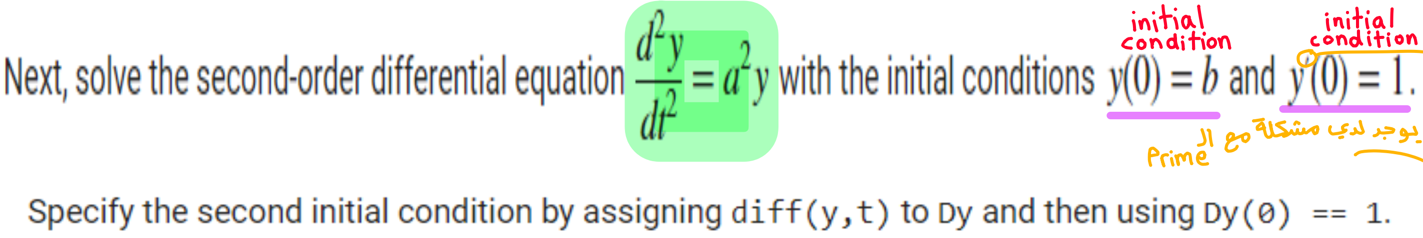
Specify the second-order derivative of y by using `diff(y,t,2)`
`dsolve`.

```
syms y(t) a
eqn = diff(y,t,2) == a*y;
ySol(t) = dsolve(eqn)
```

`ySol(t) =` $C_1\,e^{-\sqrt{a}\,t} + C_2\,e^{\sqrt{a}\,t}$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

With initial conditions



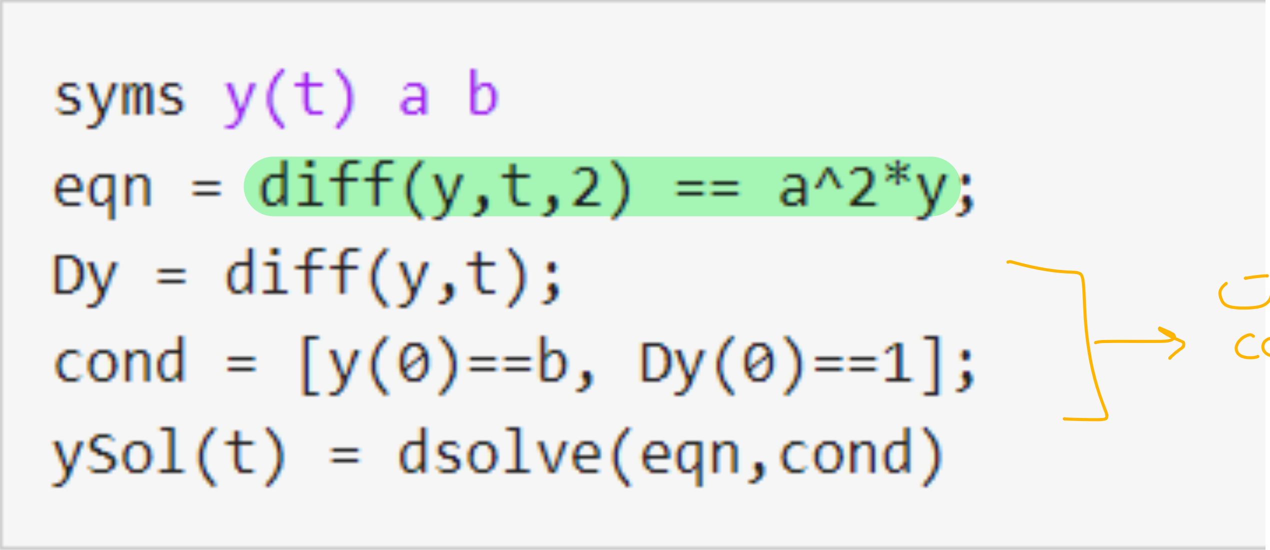
Next, solve the second-order differential equation $\frac{d^2y}{dt^2} = a^2y$ with the initial conditions $y(0) = b$ and $y'(0) = 1$.

initial condition

initial condition

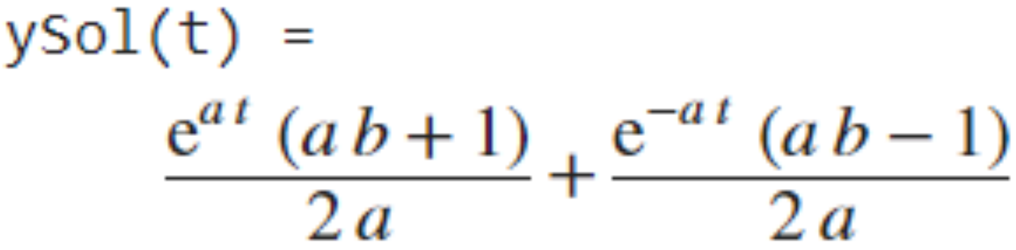
يوجد لدي مشكلة مع ال Prime

Specify the second initial condition by assigning `diff(y,t)` to `Dy` and then using `Dy(0) == 1`.



```
syms y(t) a b
eqn = diff(y,t,2) == a^2*y;
Dy = diff(y,t);
cond = [y(0)==b, Dy(0)==1];
ySol(t) = dsolve(eqn,cond)
```

عملت condition



ySol(t) =

$$\frac{e^{a\,t}\,(a\,b+1)}{2\,a} + \frac{e^{-a\,t}\,(a\,b-1)}{2\,a}$$

y Pearson Education, Ltd.
All Rights Reserved

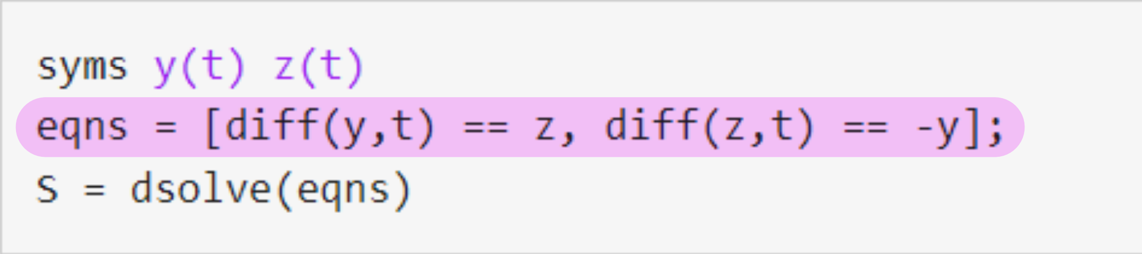
Square
Bracets
[ ]

لما يصير System بستعمل

System of differential equations

$$\frac{dy}{dt} = z$$

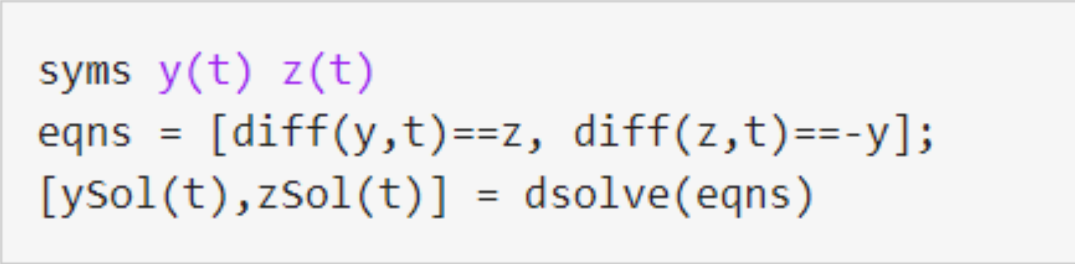
$$\frac{dz}{dt} = -y.$$



```
syms y(t) z(t)
eqns = [diff(y,t) == z, diff(z,t) == -y];
S = dsolve(eqns)
```

```
S = struct with fields:
    z: C2*cos(t) - C1*sin(t)
    y: C1*cos(t) + C2*sin(t)
```

Without assignment



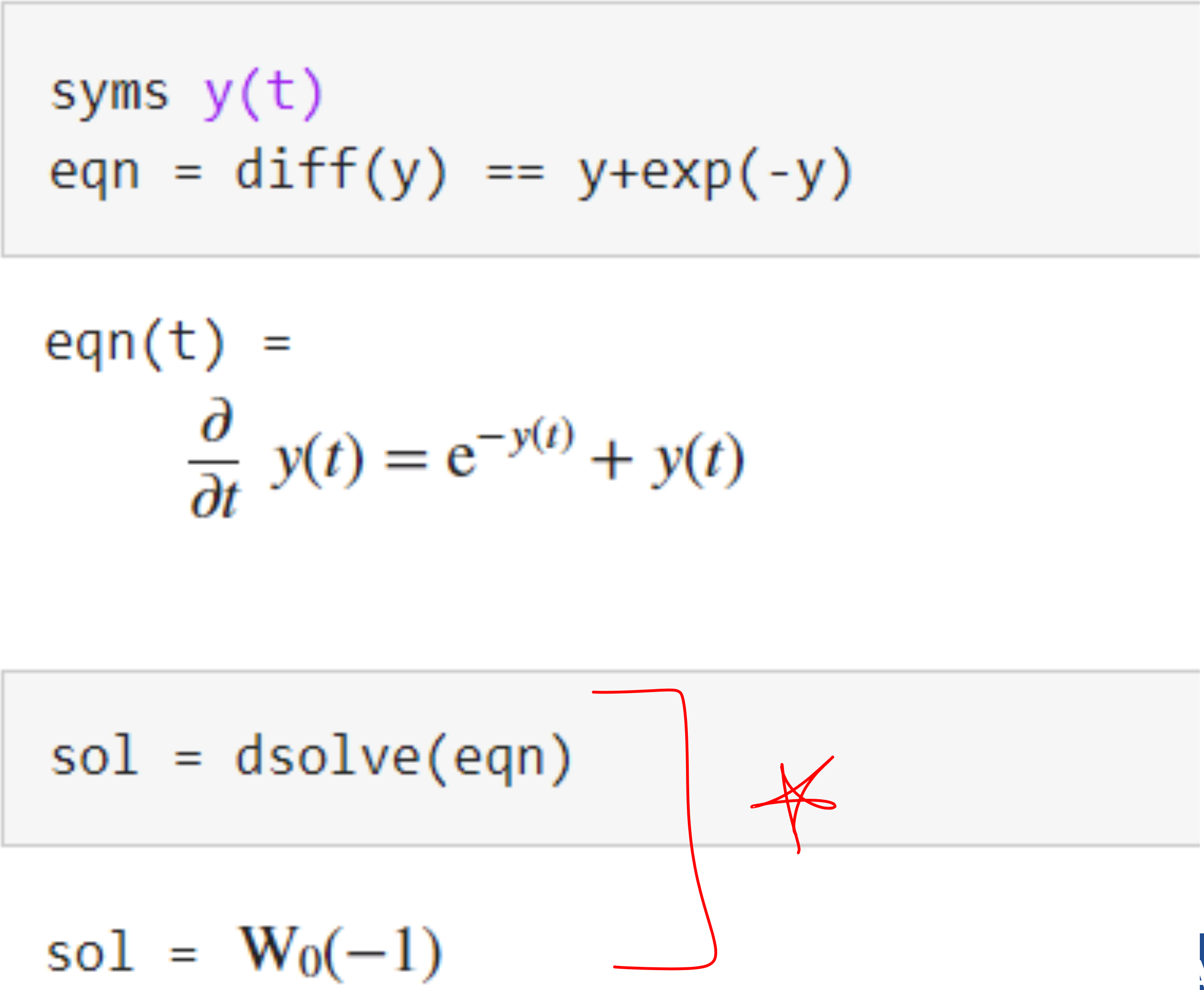
```
syms y(t) z(t)
eqns = [diff(y,t)==z, diff(z,t)==-y];
[ySol(t),zSol(t)] = dsolve(eqns)
```

`ySol(t)` $= C_1\cos(t) + C_2\sin(t)$

`zSol(t)` $= C_2\cos(t) - C_1\sin(t)$

With Assignment

Solving the differential equations



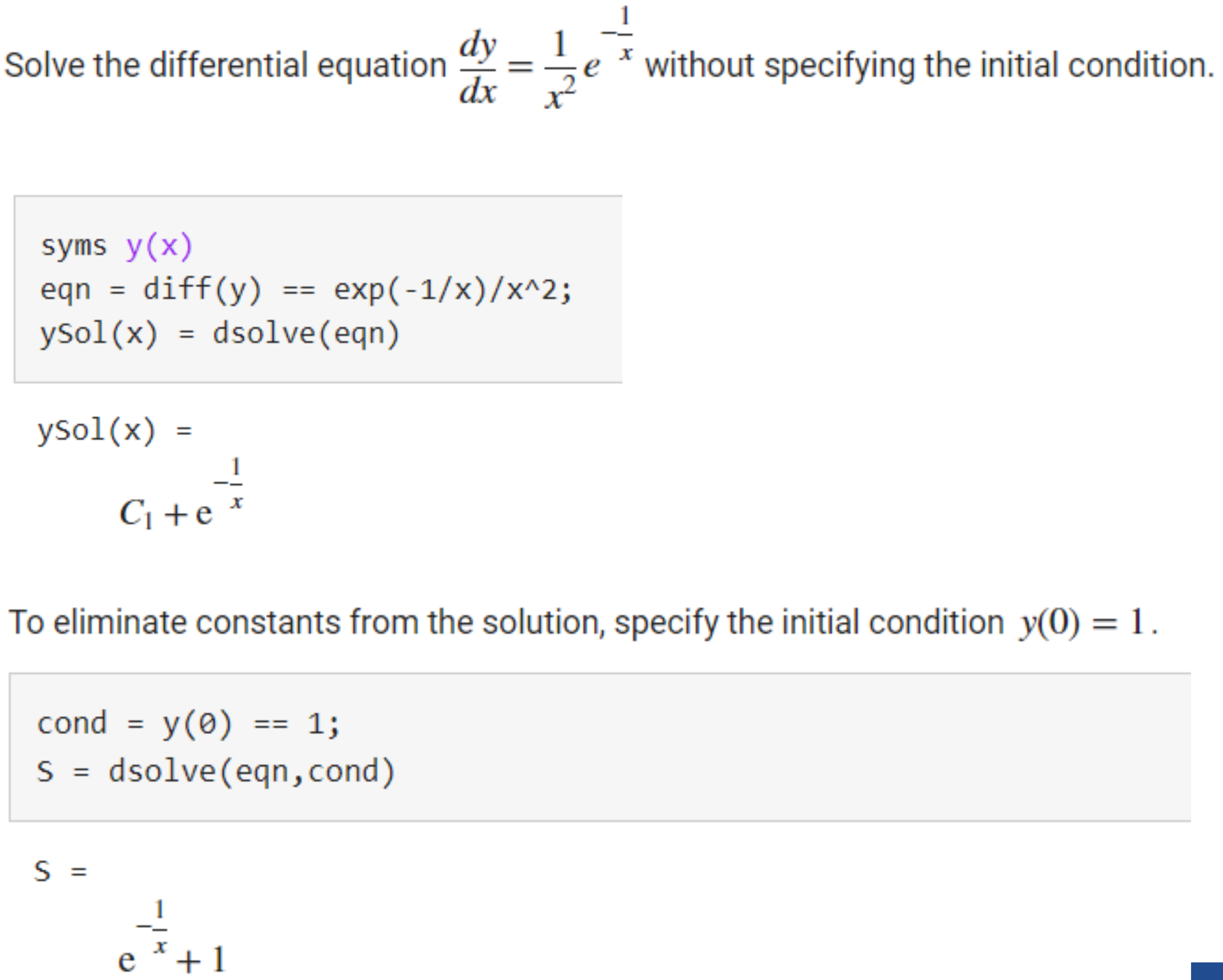
```
syms y(t)
eqn = diff(y) == y+exp(-y)
```

eqn(t) =

$$\frac{\partial}{\partial t}\, y(t) = e^{-y(t)} + y(t)$$

```
sol = dsolve(eqn)
```

sol = $W_0(-1)$



Solve the differential equation $\frac{dy}{dx} = \frac{1}{x^2} e^{-\frac{1}{x}}$ without specifying the initial condition.

```
syms y(x)
eqn = diff(y) == exp(-1/x)/x^2;
ySol(x) = dsolve(eqn)
```

```
ySol(x) =
```

$$C_1 + \mathrm{e}^{-\frac{1}{x}}$$

To eliminate constants from the solution, specify the initial condition $y(0) = 1$.

```
cond = y(0) == 1;
S = dsolve(eqn,cond)
```

```
S =
```

$$\mathrm{e}^{-\frac{1}{x}} + 1$$

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

in Command window in Matlab :-

```
>> syms y(t) a
>> eqn = diff (y,t) == a* y
eqn (t) =
  diff(y(t),t) ==a*y(t)
 >> dsolve (eqn)
ans =
 C1 *exp (a *t)
```

```
>> syms y(t) a
 >> eqn =diff(y,t,2) = = a*y ;
                 المشتقة الثانية
  >> ysol(t)= dsolve (eqn)
ysol(t) =
  c1*exp(-a^(1/2)*t) +C2*exp(a^(1/2)*t)
```

```
>> syms y(t) a b
>> eqn =diff (y,t,2)== a^2 *y ;
>> Dy =diff (y,t)
Dy(t) =
 diff(y(t),t)
>> co = [y(0)==b , Dy(0)==1] ;
>> ysol (t) =dsolve (eqn,co)
ysol(t) =
   (exp (a*t)*(a*b+1))/(2*a) +(exp(-a*t)*(a*b-1))/(2*a)
```

```
>> syms y(t) z(t)
 >> eqn =[diff (y,t)== z , diff (z,t) ==-y]
  >> [ ysol (t) , zsol (t)] = dsolve(eqn)
   ysol (t) =
       c1*cos(t) +C2*sin(t)
   zsol(t) =
       c2*cos(t) - C1* sin(t)
```

```
>> syms y(t)
 >> eqn = diff(y , t) == y + exp (-y)
  >> ysol (t) =d solve (eqn)
ysol (t) =
   lambertw(0,-1)
```

```
>> syms  y(x)
>> eqn = diff(y,x)= =exp (-(1/x))/x^2
eqn (x)=
   diff(y(x), x)= =exp(-1/x)/x^2
```

أو y(x)
نفس الاشي

```
>> sol (x) =d solve (eqn)
sol (x) =
  c1+exp(-1/x)
>> c = y(0)= = 1
c =
  y(0) = =1
>> sol(x) =dsolve(eqn,c)
sol(x) =
  exp(-1/x) +1
```

في الامتحان ممكن الدكتور
يطلب الحل → numirical
→ symbolic

12.5
Differential Equations

- Differential equations contain both
  - the derivative of the dependent variable with respect to the independent variable
  - the dependent variable

$$\frac{dy}{dt} = y$$

is a differential equation

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

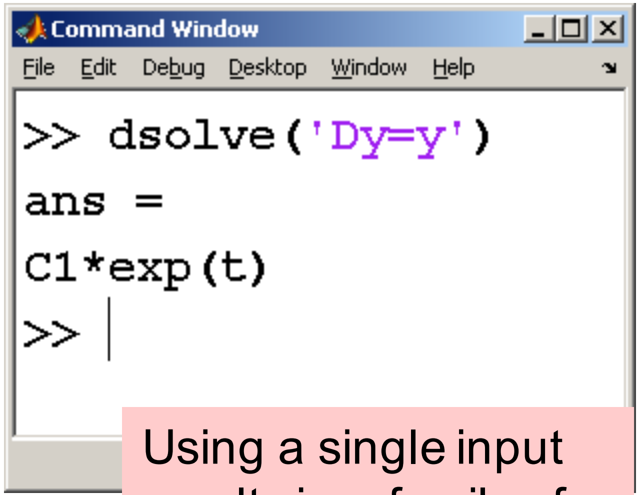
Default variable

- Although any symbol can be used for either the independent or the dependent variable, the default independent variable is t in MATLAB (and is the usual choice for most ordinary differential equation formulations.)

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

dsolve

- When we solve a differential equation, we are looking for an expression for y in terms of t
- dsolve requires the differential equation as input
  - use the symbol D to specify derivatives with respect to the independent variable

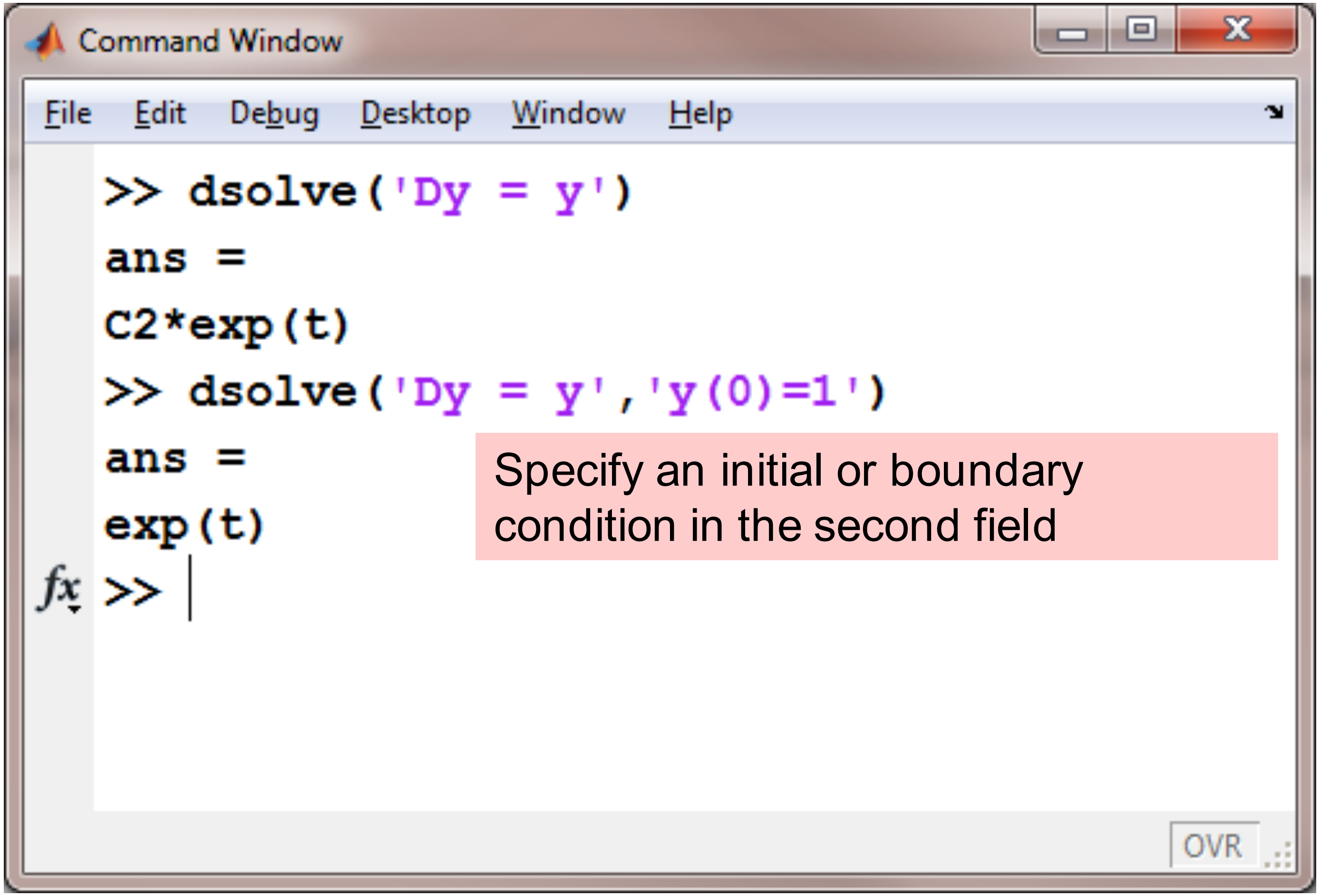


```
>> dsolve('Dy=y')
ans =
C1*exp(t)
>>
```

Using a single input results in a family of results

dsolve is a “function function”

Copyright © 2017 by Pearson Education, Ltd. All Rights Reserved PEARSON

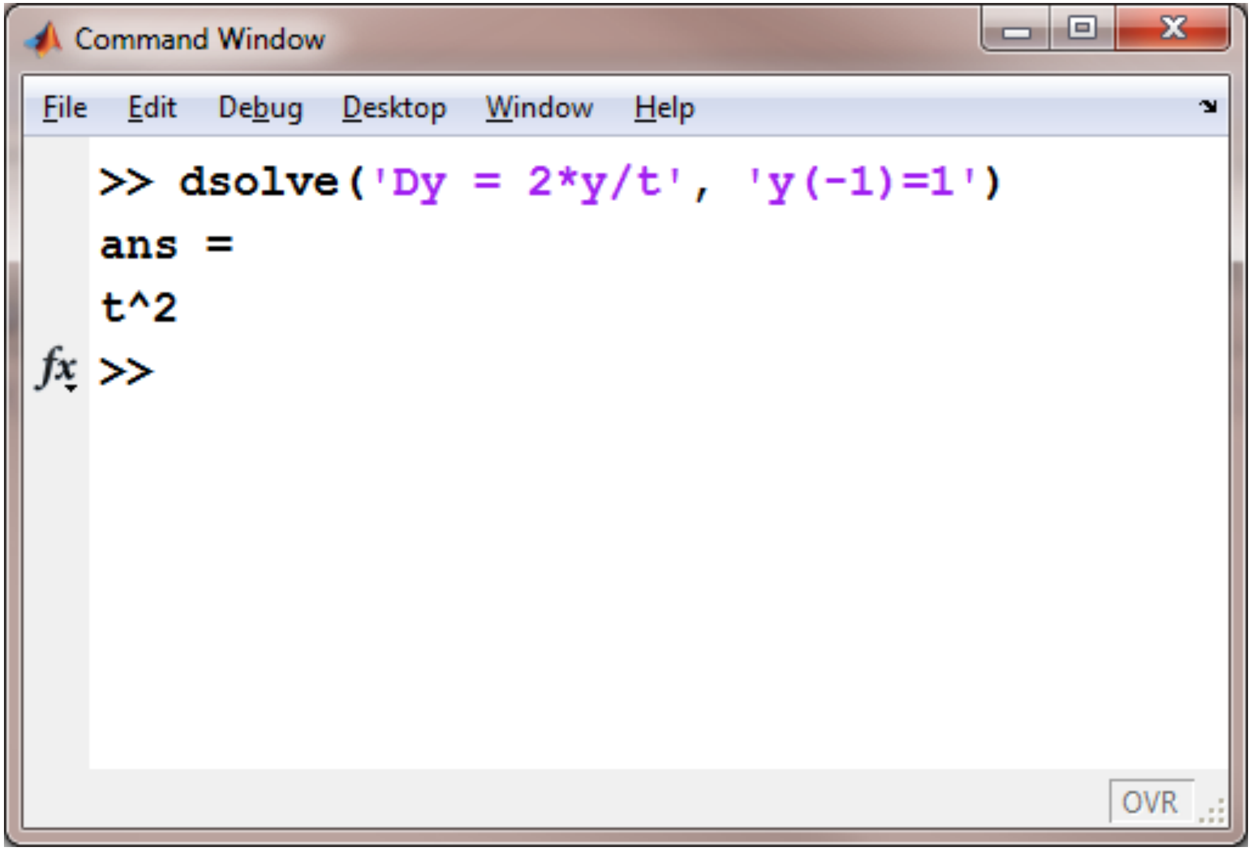


```
>> dsolve('Dy = y')
ans =
C2*exp(t)
>> dsolve('Dy = y','y(0)=1')
ans =
exp(t)
>>
```

Specify an initial or boundary condition in the second field

Copyright © 2017 by Pearson Education, Ltd. All Rights Reserved

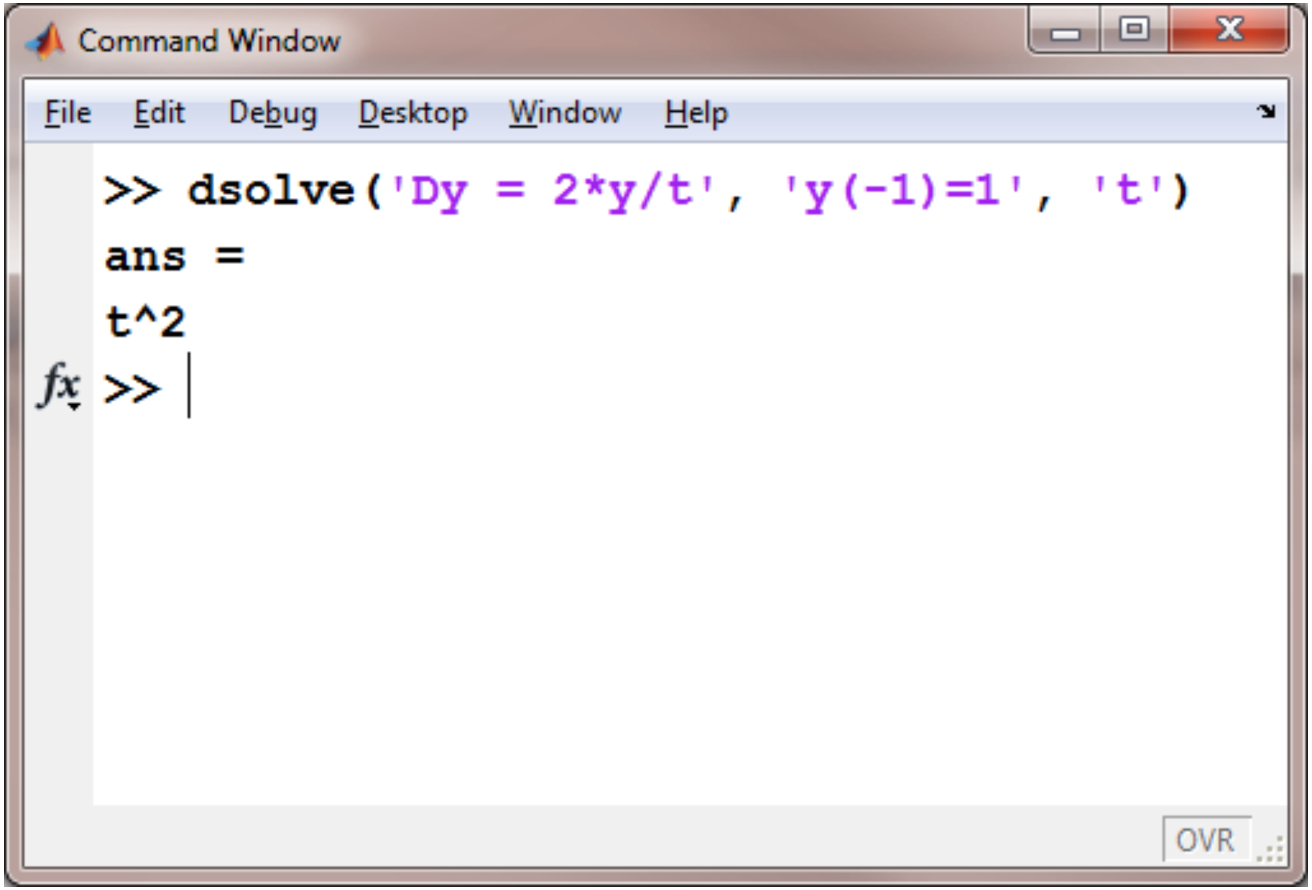
Here's a more complicated example



```
>> dsolve('Dy = 2*y/t', 'y(-1)=1')
ans =
t^2
>>
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

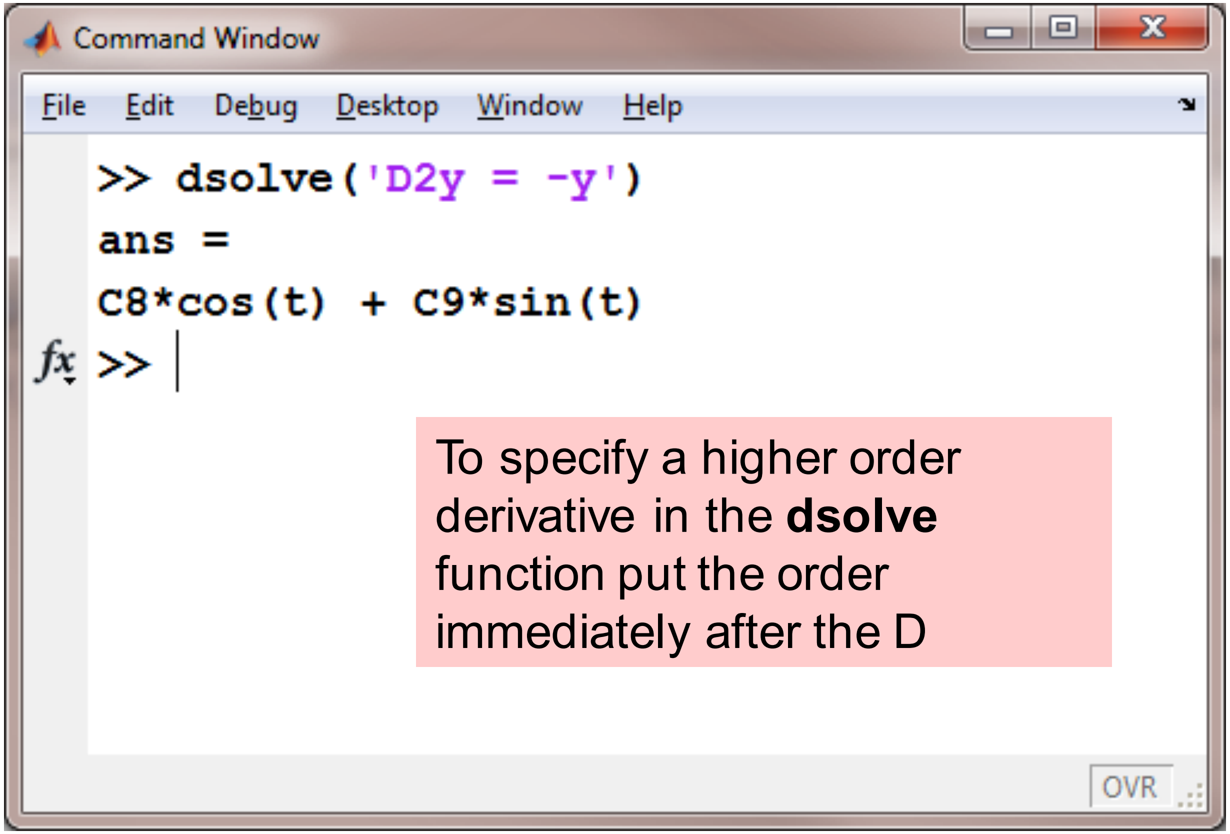
You can specify the independent variable in the third field



```
>> dsolve('Dy = 2*y/t', 'y(-1)=1', 't')
ans =
t^2
>>
```

Copyright © 2017 by Pearson Education, Ltd. All Rights Reserved

Higher Order Derivatives



```
>> dsolve('D2y = -y')
ans =
C8*cos(t) + C9*sin(t)
>>
```

To specify a higher order derivative in the **dsolve** function put the order immediately after the D

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

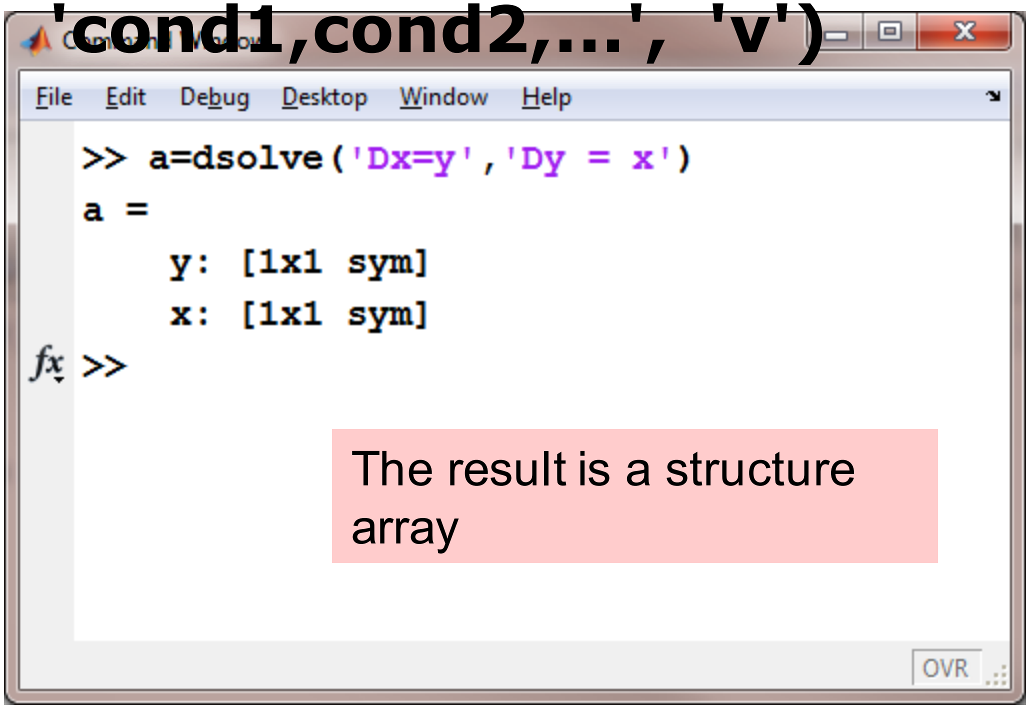
Hint

- Don’t use the letter D in your variable names in differential equations.
- It will confuse the function into thinking you are trying to specify a derivative

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Use the dsolve function to solve systems of equations

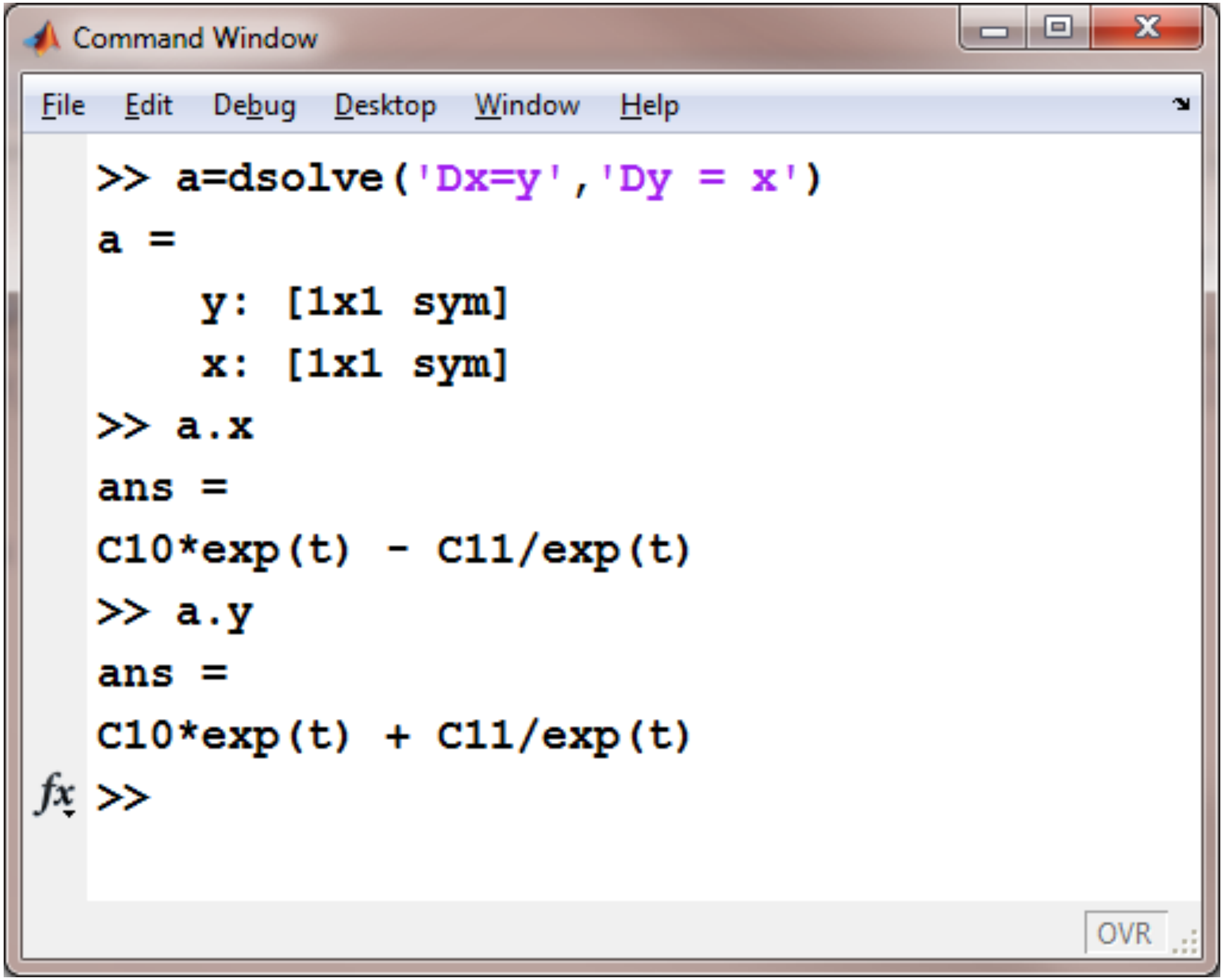
- **dsolve('eq1,eq2,...', 'cond1,cond2,...', 'v')**



```
>> a=dsolve('Dx=y','Dy = x')
a =
    y: [1x1 sym]
    x: [1x1 sym]
>>
```

The result is a structure array

Copyright © 2017 by Pearson Education, Ltd. All Rights Reserved PEARSON



```
>> a=dsolve('Dx=y','Dy = x')
a =
    y: [1x1 sym]
    x: [1x1 sym]
>> a.x
ans =
C10*exp(t) - C11/exp(t)
>> a.y
ans =
C10*exp(t) + C11/exp(t)
>>
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

MATLAB can not solve every differential equation symbolically.

- For complicated (or ill behaved) systems of equations you may find it easier to use MuPad
    - Remember that MATLAB’s symbolic capability is based on the MuPad engine
- There are many differential equations that can’t be solved analytically at all
    - The numerical techniques described in Chapter 13 can be used to solve many of these equations.

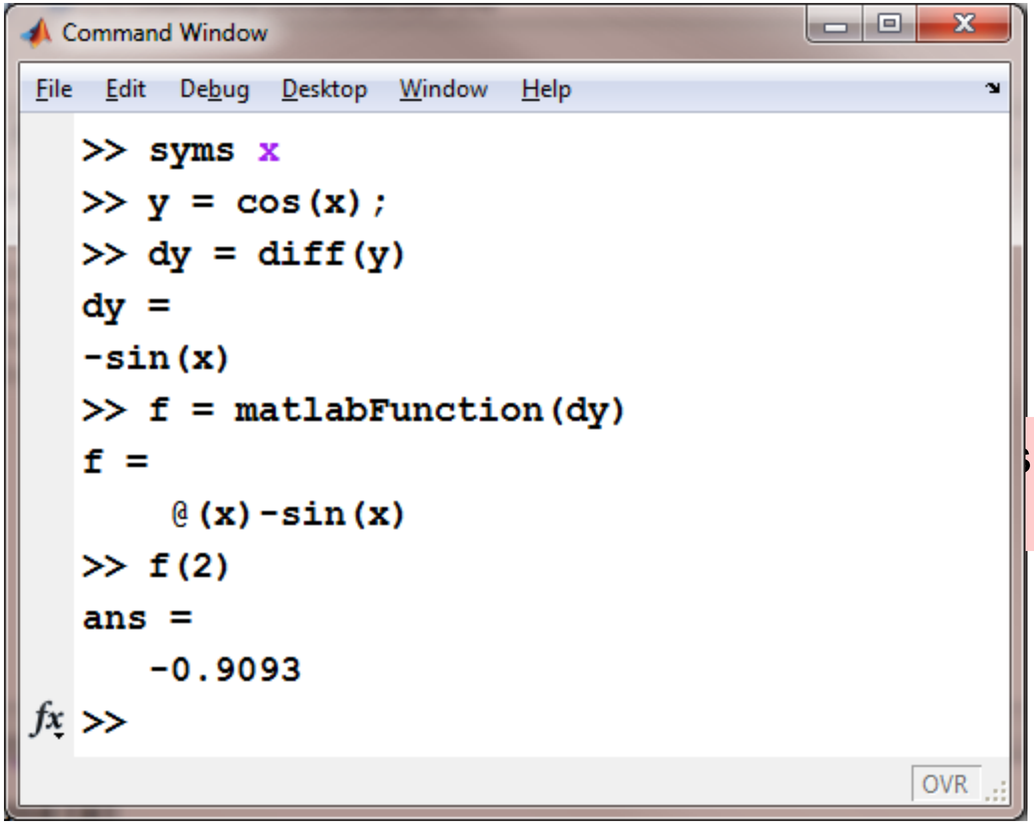
Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

12.6 Converting Symbolic Expressions to MATLAB functions

- It is often useful to manipulate expressions symbolically … but then to perform numeric calculations using more traditional MATLAB functions
- matlabFunction converts a symbolic expression to an anonymous function

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

matlabFunction



```
>> syms x
>> y = cos(x);
>> dy = diff(y)
dy =
-sin(x)
>> f = matlabFunction(dy)
f =
    @(x)-sin(x)
>> f(2)
ans =
   -0.9093
>>
```

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved PEARSON

Summary

- MATLAB uses MuPad as its symbolic engine
- The symbolic toolbox is an optional component of the professional version
- A subset is included with the student version

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Summary – Variable Definition

- Use either
  - sym
  - syms
- The sym command can be used to create symbolic expressions or equations
- The syms command can create multiple symbolic variables in one step

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Summary – Composition of expressions

- Once symbolic variables have been created they can be used to create more complicated expression

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Summary
Equations vs Expressions

- Equations are set equal to something
- Expressions are not
- If you set one expression equal to another, you've created an equation

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Summary – Symbolic functions

- numden
- expand
- factor
- collect
- simplify
- simple

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Summary – Solve

- If the input to solve is an expression MATLAB sets it equal to 0 and solves
- If the input is an equation, MATLAB solves the equation for either the default variable, or a user defined variable
- solve can also solve systems of equations

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Summary - dsolve

- Used to solve differential equations
- D signifies a derivative
- Can be used to solve systems of equations
- Not all differential equations can be solved analytically

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved

Summary - Calculus

- diff - finds the derivative
- int - takes the integral

Copyright © 2017 by Pearson Education, Ltd.
All Rights Reserved