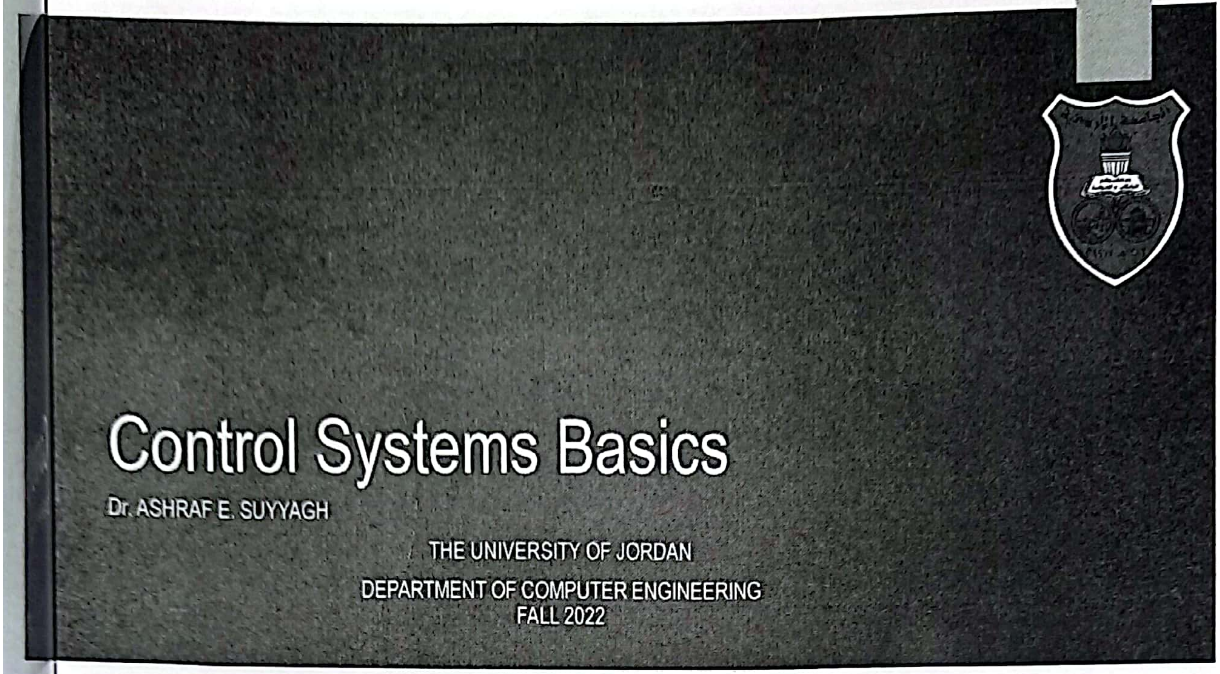
CONTROL

REEM MUIN

POWERUNIT

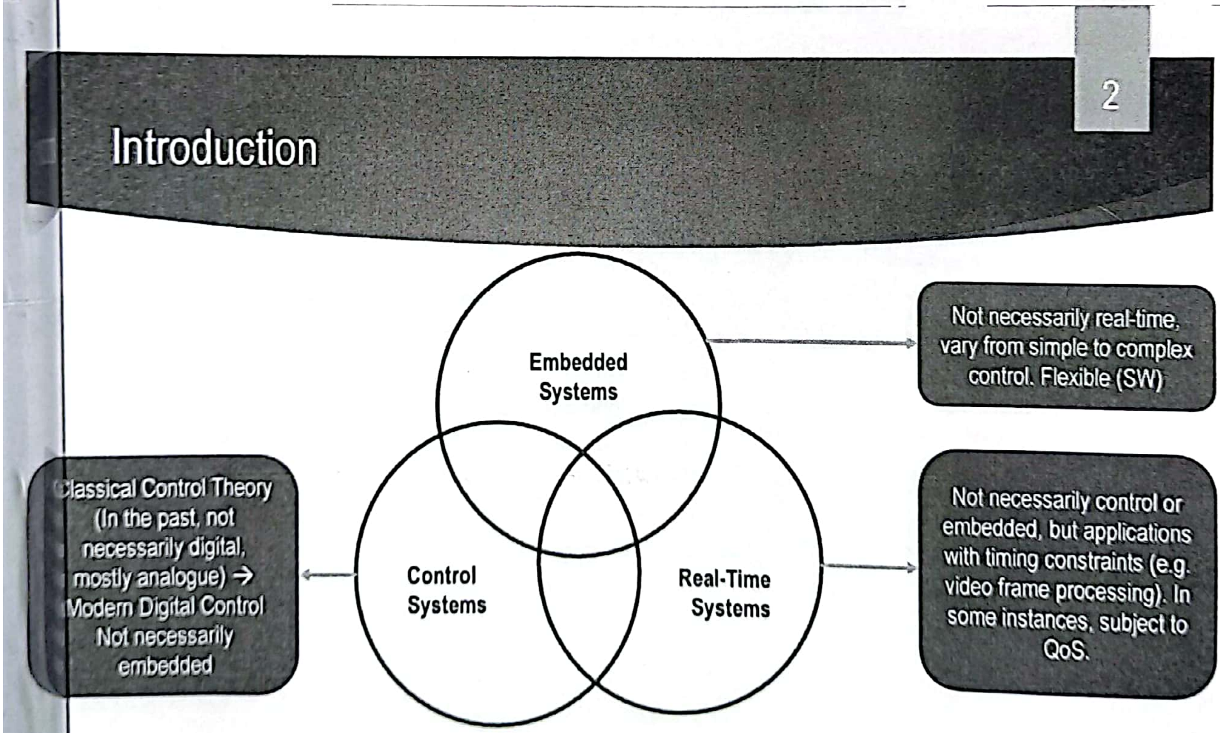


Control Systems Basics

Dr. ASHRAF E. SUYYAGH

THE UNIVERSITY OF JORDAN

DEPARTMENT OF COMPUTER ENGINEERING
FALL 2022



Introduction

Not necessarily real-time, vary from simple to complex control. Flexible (SW)

Classical Control Theory (In the past, not necessarily digital, mostly analogue) → Modern Digital Control Not necessarily embedded

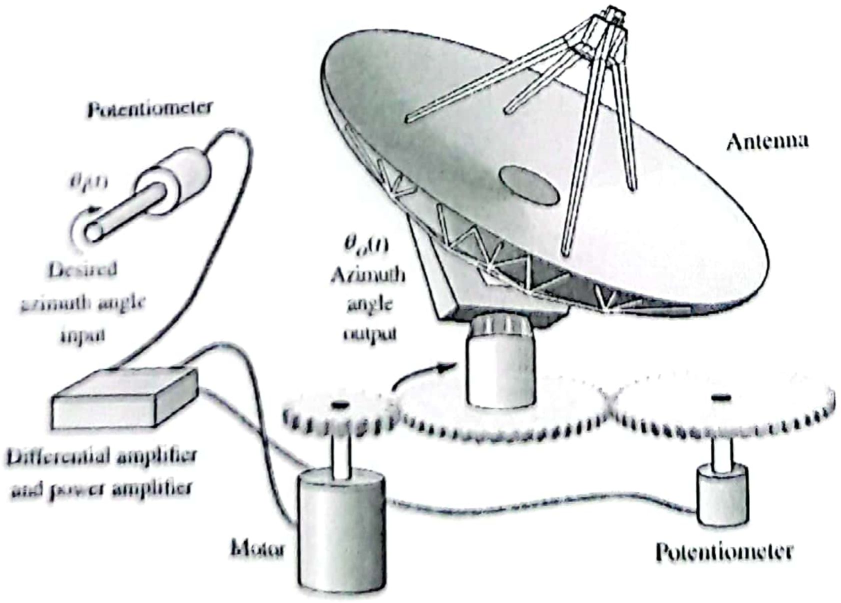
Not necessarily control or embedded, but applications with timing constraints (e.g. video frame processing). In some instances, subject to QoS.

Classical Control Development

- Control Systems is not a human engineered discipline → Best controllers are found in nature!
- Biochemical Controllers → Human body
  (numerous control systems (central and distributed), *e.g., pancreas, immune system, eyes following an object, eye-hand coordination, fight or flight response, etc.*)
- Species declining or increasing population can be modeled as a self-regulating control system (wolves/rabbits populatio
- First feedback control systems were developed in Greece in 300 B.C. (water clocks)
- Steam pressure and temperature control with safety valves as far back as 1681, windmill blade adjustment to wind spee 1745
- Foundation of Control Systems as we know today in 1868 (*Maxwell, Routh-Hurwitz, Chebyshev*) → system description differential equations, criteria for stability
  → stability of steering ships and applying power through hydraulic systems, gun platforms for military ships (achieved b 1922 by Minorsky) (Adaptive Control field)
- In 1930's and 40's: focus on developing of math to analyze modern systems (Bode, Nyquist) → Bode Plots and Root L

Simple Control System Example (All Analogue Components)

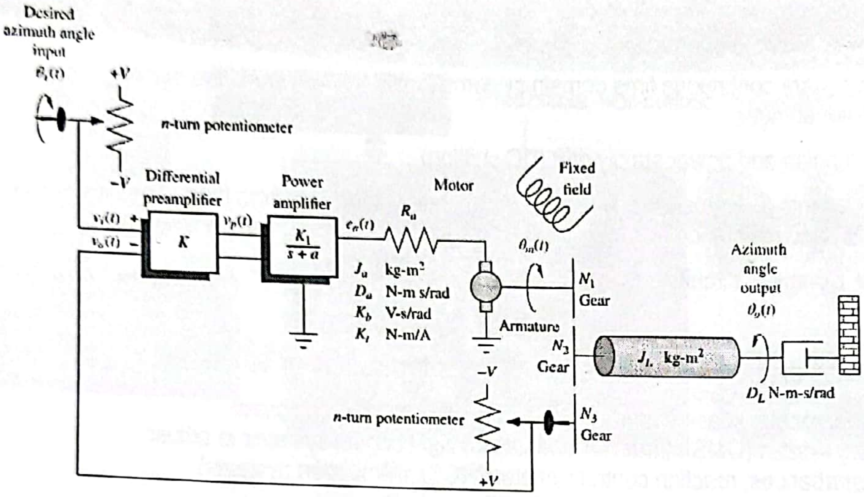
Layout


Potentiometer
$\theta_i(t)$
Desired azimuth angle input
Antenna
$\theta_o(t)$ Azimuth angle output
Differential amplifier and power amplifier
Motor
Potentiometer

Antenna Azimuth Position Control System

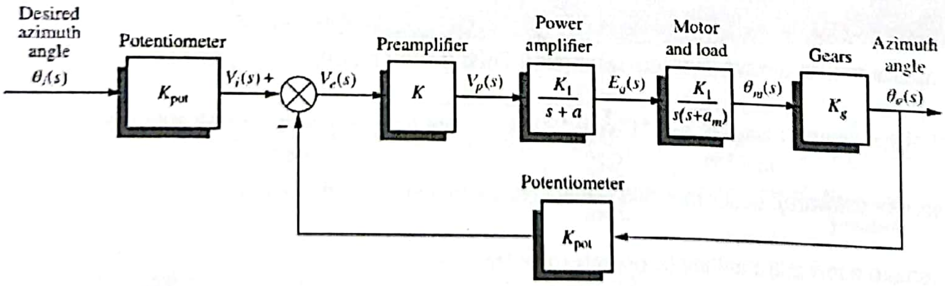
Simple Control System Example (All Analogue Components)

Schematic



Simple Control System Example (All Analogue Components)

Block Diagram



You can notice that there are no **computers** or **digital controllers** in the layout/schematic/block diagrams → analogue control

Digital (Computer) Control

Classical (Analogue) Control systems are continuous time domain systems (no sampling), the entire signal (voltage, curr processed). They have several disadvantages:

- highly susceptible (sensitive) to noise and power supply drift (DC shifting)
- Since they use analogue components (*e.g.*, resistors and capacitors), their values vary from their nominal values due tolerances (*e.g.*, 1000 ± 100Ω) , and resistance changes for example by circuit/environment temperature.
- Difficult to modify or update → Complete circuit redesign

Last colour band: Tolera

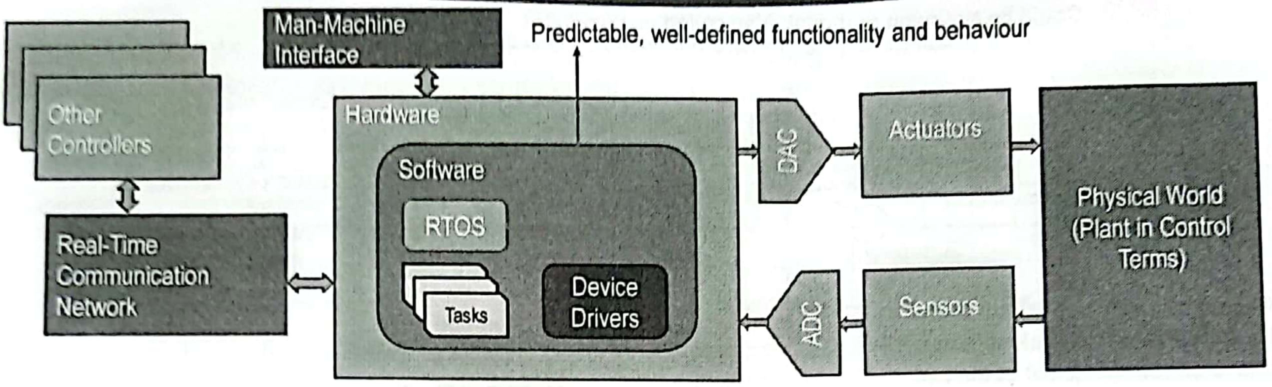
Most modern control systems are digital → They use main computer frames or embedded controllers

- Could be complex: Industrial robots, space craft/rover, chemical/nuclear process control (orbital maneuvering system (OMS), thrust for orientation, flight control systems to adjust for atmosphere disturbances, reaction control systems (RCS), life support systems)
- Could be simple: home automation, DVD player, HDD controller

Digital (Computer) Control Advantages

- Discrete time signals and approximation always involved **(Sampling Theory : Shannon's Theory of Sampling / Nyquist Rate)**
- Accurate representation of digital signals using "0" and "1" with 12 bits or more for a single number → Negligible errors (de on ADC resolution)
- Digital controllers (in firmware or software): easily modified without complete replacement of the original controller → **no ci redesign or re-wiring**
- Complex digital controller require a few extra arithmetic operations or libraries
- Faster hardware allows short sampling periods (high sampling rate).
- With short sampling periods, digital controllers monitor controlled variables almost continuously
- Advances in VLSI technology provide better, faster and more reliable integrated circuits at lower prices.
- Many hardware circuits can be replaced by software code **(*e.g.* Filters can be coded instead of being built as analogue circuits)**
- **Can be real-time control**
- **Computer can schedule multiple control systems at the same time**

Overview of a Real-Time Control System

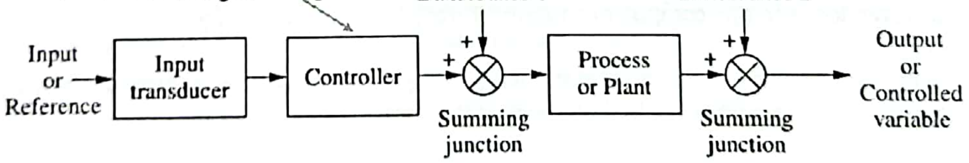


Predictable, well-defined functionality and behaviour

Tasks share resources, communicate/synchronize with each other

Open-Loop Control Systems

Could be analogue or digital

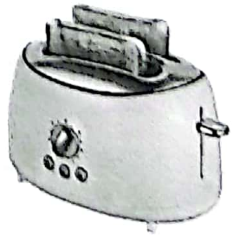


Main characteristic of open-loop systems is that they

- Are only commanded by the input and therefore sensitive to disturbance
- Cannot compensate/correct for any disturbances (noise) that add to the controller's driving signal (Disturbance 1), or disturbances at the output (Disturbance 2, for example a physical object in the way)

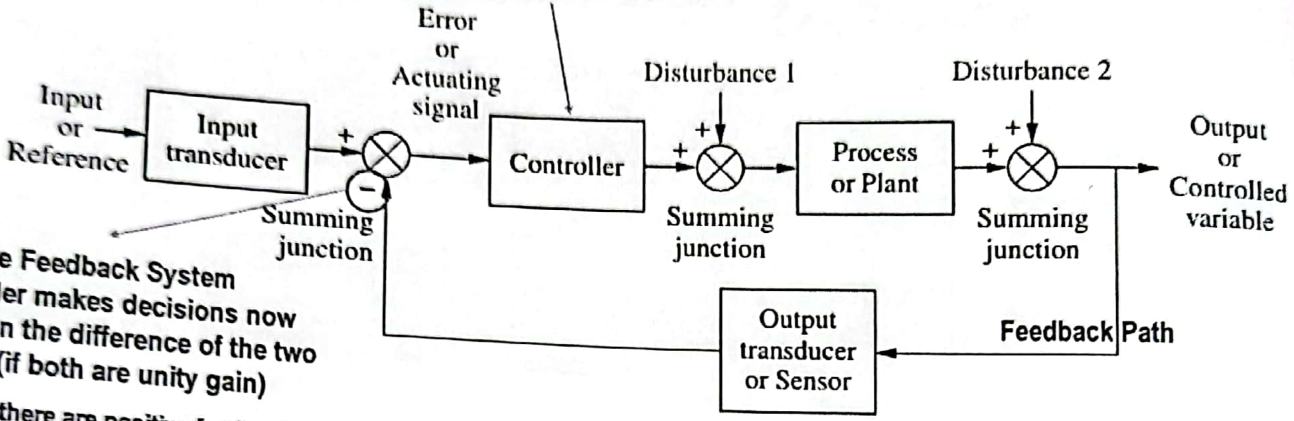
- A kitchen microwave or toaster is a basic example

- Very simple to design, but often times unstable for complex systems



Closed-Loop Control Systems I

Could be analogue or digital. Also called compensator



Negative Feedback System
Controller makes decisions now based on the difference of the two signals (if both are unity gain)

Note that there are positive feedback systems as well. A simple NAND latch that forces that output to 0s and 1s is a positive feedback sys with desirable output. However, a microphone and speaker loop in a conference hall is a positive feedback system with undesirable outp

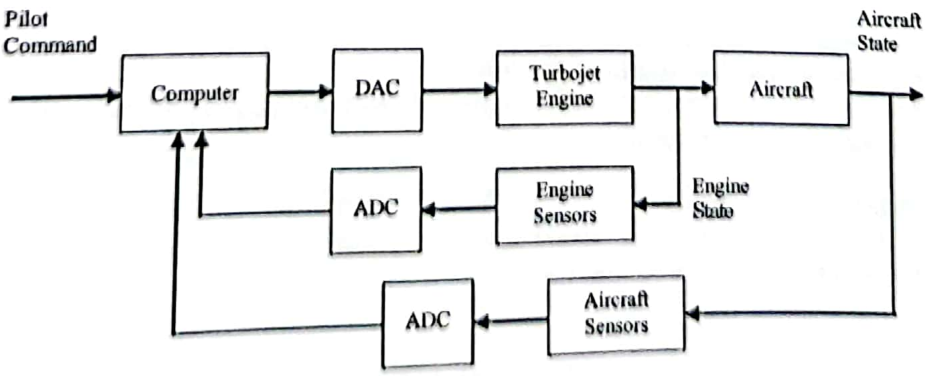
Closed-Loop Control Systems II

uld be multiple closed loops to one computer controller: Turbojet Engine Control System

e feedback loops could be:

Multiple sensors connected to the same physical object in the plant (engine temperature, engine vibration, engine fuel flow)

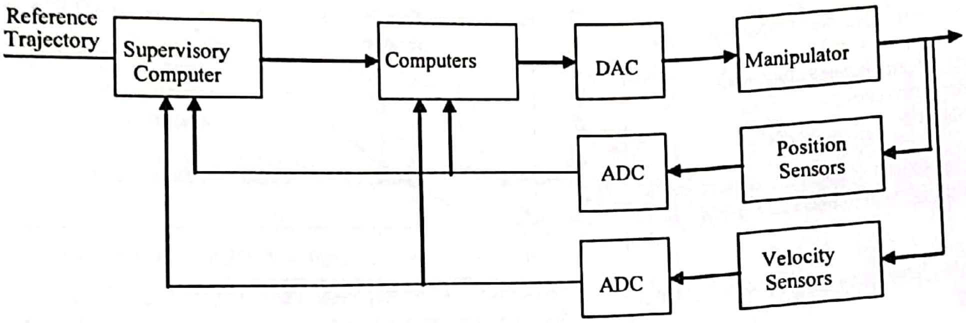
Multiple sensors connected to the different physical objects in the plant



Closed-Loop Control Systems III

Could be multiple closed loops, to multiple computer controllers and a supervisory computer

Controllers could be digital or analogue in the same system!



Closed-Loop Control Systems IV

- Have the obvious advantage of greater accuracy than open-loop systems
- Less sensitive to noise, disturbances, and changes in the environment
- Transient response and steady-state error can be controlled more conveniently and with greater flexibility in closed-loop systems (more in following slides)
- Closed-loop systems are more complex and expensive than open-loop systems (extra sensors, wiring, ADC if digital, and processing time)

- **Question**: What do you need to do to redesign the simple toaster as a closed-loop feedback system? What will you be sensing to get a perfect toasted piece of bread?
- **Home Exercise**: Think of another existing open-loop system at your home, and think of what you need to convert it to a closed-loop system? Is it worth it?

Analysis and Design of Control Systems (I)

Temporal Characteristics of the Plant

Input: Press the 4th floor button representing our desired output (reach 4th floor)

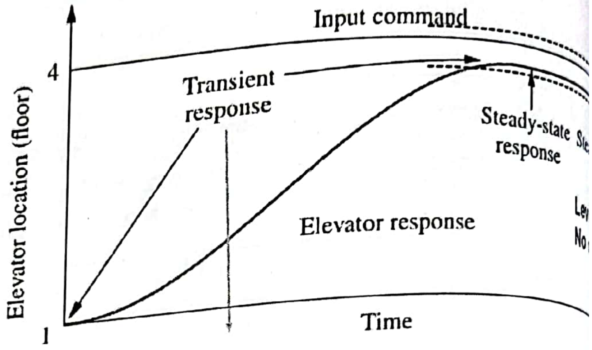
A button ideally represents a unit step function u(t)

Two major measures of performance are apparent:
(1) the transient response and
(2) the steady-state error.

Analysis Stage: find values for transient response and steady state error that are within specifications and acceptable

Design Stage: apply control system parameters that satisfy the values analyzed

Elevator Example



Too slow transient response: user patience is sacrificed
Too fast transient response : user comfort is sacrificed, could cau
damage/death

Analysis and Design of Control Systems (II)

Temporal Characteristics of the Plant

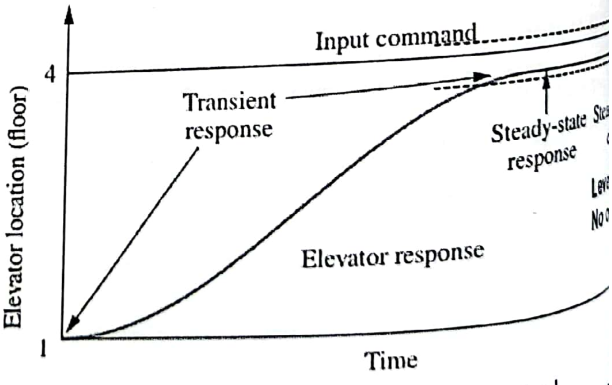
The transient response should be as quick as possible (ideal case), but in many systems, this is not feasible either due to its catastrophic effects or simply it is physically impossible

e.g., an elevator going from G-level to 4th or 100th floor incautiously will kill you; also, mechanical systems are slow and physically infeasible to have instantons response.

The steady-state means we should reach our desired output without error. However; error is always there.
Error band specifies the maximum acceptable margins for errors (0.01%, 1%, 5%)

Elevator could stop at the 4th floor with its level 1cm above the floor ground (acceptable), 5cm → You could trip and fall, 50 cm (something is visibly wrong, and it is dangerous)

Elevator Example



Too slow transient response: user patience is sacrificed
Too fast transient response : user comfort is sacrificed, could caus
damage/death

Analysis and Design of Control Systems (III)

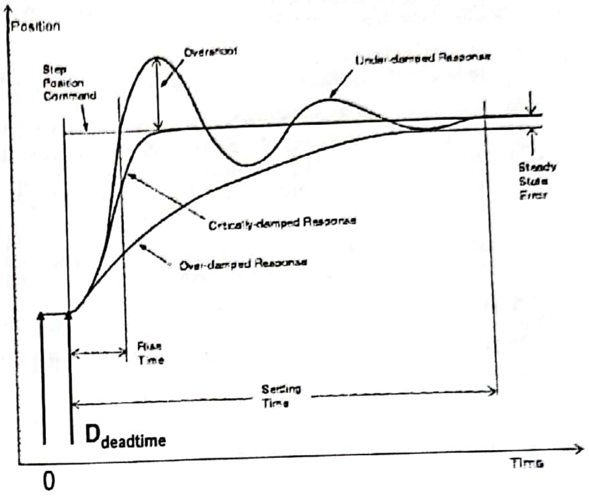
When responding to controllers' commands, changing the old state and reaching the new stead-state is not necessarily a smooth ride. It depends on the plant components, physical characteristics and equations

Given the system characteristics, we could have different type of responses:

1. **Overdamped** response
2. **Critically damped** response
3. **Underdamped** response → Overshoots

Overshooting means that the output oscillates until it settles on the stead-state value.

Imagine using an elevator to go to the 4th floor, but it first goes to the 5th floor, then 3rd floor, then goes to between 4th and 5th, then to between 3rd and 4th before eventually settling on the 4th floor.



Analysis and Design of Control Systems (IV)

Oscillations could be not only inconvenient but also dangerous, if they exceed the plant physical limitations.

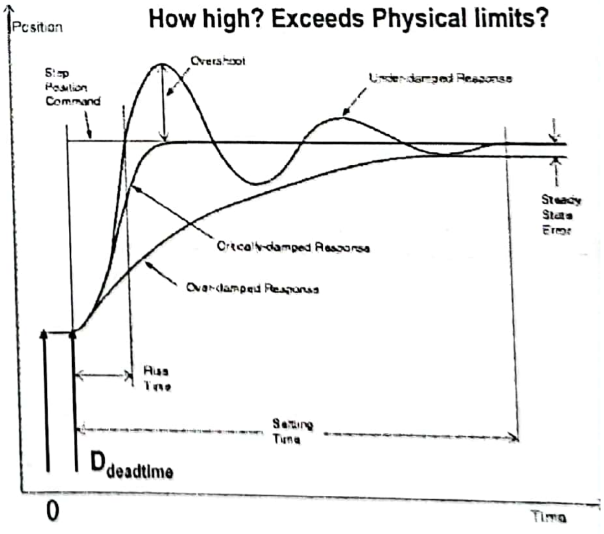
Imagine if the 4th floor is the last floor, if they elevator overshoots, then means it will hit the roof causing damage, injury, or even death.

Temporal characteristics of the physical plant:

Settling time: the time elapsed from the application of an ideal instantaneous step input to the time at which the output has entered and <u>remained</u> within a specified error band.

Rise time: the time required for the response to rise from 0% to 100% of its final value **(Ideal Definition), In practice to 90%, or 95%**

Dead time: Physical systems do not always respond immediately (*e.g.*, motor). When you send a current to a motor, it takes time to generate a field to overcome the inertia and initial torque to start moving


How high? Exceeds Physical limits?

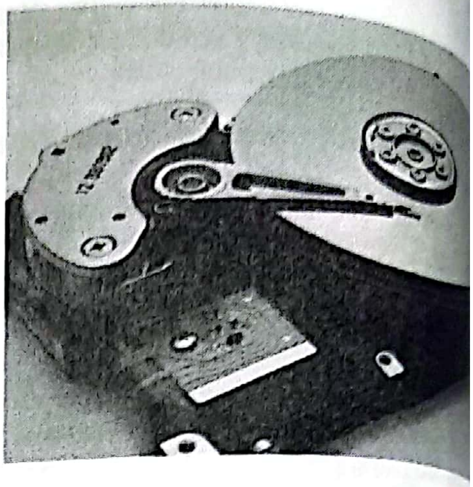
Analysis and Design of Control Systems (V)

Another Example is the HDD

- In a computer HDD, transient response contributes to the time required to read from or write to the computer's disk storage
- Since reading and writing cannot take place until the head stops, the speed of the read/write head's movement from one track on the disk to another influences the overall speed/performance of the computer. The faster the cylinder rotates; the quicker we move the information to the head to read.
- Steady state response: head of a disk drive finally stopped at the correct track → otherwise errors

Think: Why do we rarely see HDDs with speeds more than 7200 RPMs?

Think: What do you think are the error margins for the steady state?



Control System Stability

- Transient Response and Steady-State error is irrelevant if the system is not stable.
- To understand stability, one must understand the system total response.

$$\textbf{Total response} = \textbf{Natural response} + \textbf{Forced response}$$

Natural response:

Any physical system either mechanical or electrical has certain properties governed by physics equations, when it turns on, it responds initial conditions and no other inputs (no forced inputs)

Examples: brick initially wired onto a spring, charged RC circuit

Forced response is the system response when there is a certain input.

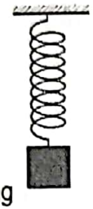
Examples: Adding additional weight to the brick/spring system or opening a switch in the charged RC circuit.

In any stable system, the natural response must decay, if not, and it **must not grow Indefinetly** otherwise the system will become unstable

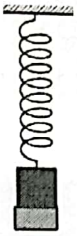
A Simple Stability Example

Initially, suppose a certain spring can hold a maximum weight of 7KGs, beyond which, the spring loses its elasticity and cannot return back to its original shape, and is therefore damaged.

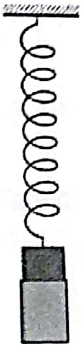
Consider that our system design initially requires attaching a weight of 3KGs. The spring stretches and reaches an equilibrium state. Now the spring system (spring + weight) is said to have a natural response.



During system operation, say we exert an external weight of 2KGs and the spring stretches. Since 2 + 3 < 7, when we remove the 2KG weight, the system will bounce and return to its former state. The 2KG weight is a forced input which gave a forced response



During system operation, say we exert an external weight of 5KGs and the spring stretches beyond its physical limitations. Since 5 + 3 > 7, when we remove the 5KG weight, the system will NOT bounce back and return to its former state. The 5KG weight is a forced input which gave a forced response that caused Instability and irrecoverable system damage



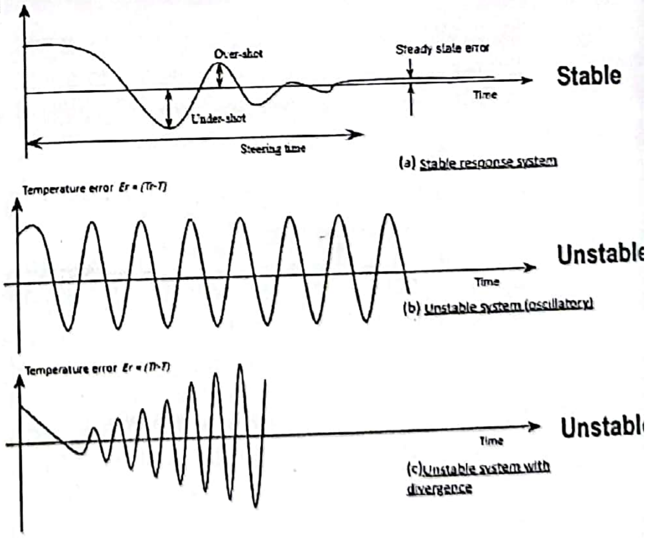
Instability

Instability, could lead to self-destruction of the physical device if limit stops (protection) are not part of the design.

Examples:

- an elevator would crash through the floor or exit through the ceiling;
- an aircraft would go into an uncontrollable roll;
- an antenna commanded to point to a target would rotate, line up with the target, but then begin to oscillate about the target with *growing* oscillations and *increasing* velocity until the motor or amplifiers reached their output limits or until the antenna was damaged structurally.

A time plot of an unstable system would show a transient response that grows without bound and without any evidence of a steady-state response.



(a) Stable response system

(b) Unstable system (oscillatory)

(c)Unstable system with divergence

Control

11 / 10 / 2022

* Control System basics :

1- Embedded System : * Complex * lots of processing * flexible (لإنها بتبرمج)

2- Real time System : * hard real system : * every task has deadline
إذا لم تنتهي قبل الـ deadline ستحصل كارثة → (لازم تخلص المهمة قبل الـ deadline)
→ Examples: Cars or heart .

* Soft real system : * لو تجاوز الـ deadline مابتحصل مشكلة كبيرة ولكن
بأثر عالـ performance مثلًا لو ماوصلت كل Frame الفيديو للشاشة "وتأخر جزء منها"
، مابصير كارثة بس بتصير الصورة مغبشة أكثر لـ 24 or 29

* الـ Real-time System مو شرط يكون دائمًا إله علاقة بالـ embedded system
ولو كان إله بصير اسمه : " Real-time embedded System "

* في عندك chips للهواتف المحمولة : 1- الـ gsm والشبكة (للكهرباء) 2- الـ cpu 3- MBU
للبطارية (embedded)

3- Control System : * أقدم من الـ embedded sys * Clasical Control (Signals
and Mathematics) and Digital Control → (بس مو شرط دائمًا تكون digital)

* الـ Sensors مثال على الـ Real-time sys مثل عداد الخطوات .

* Clasical Control (Slide 3) : مثل وجود الحيوانات في الغابات

* أكبر مثال على الـ Clasical هو الإنسان، الطبيعة أيضًا مثال على الـ clasical .

(slide 4) : clasical control System because it is Mechanical not digital . ↴

* laplace transform → ما فيه computer ← بحول أي معادلة إلى معادلة جبرية بسيطة .

* Z transform → المكافئ لـ laplace ولكن الـ digital System

* الـ clasical controls كانت بتأثر وبتركز على الـ capacitors والـ transistors
فقط بعدين بالـ digital صار في عنا hardwares أكثر مثل الـ Filters وغيرها .

* Slide 7 ← المركبات الفضائية فيها digital control sys للتكيف خارج الغلاف الجوي .

للتخلص من معادلات التفاضل و التكامل

* Impulse function : transient (حدث معين صار وحدث وبعدين خلص اختفى)
تغيرات مفاجئة → ($\delta(t)$)

* Step function : Switch (لحظة تشغيل وإطفاء الـ System) . ($u(t)$)

16/10/2022

* Control Systems : 1- Open-Loop
2- Closed-Loop

(Slide 10):

open-Loop : - بكون بالsystems البسيطة غير المعقدة - بسيط و سهل التعامل

- عندك interface بدخله كل الinput وبطلعلك الoutput بدون ما نعرف شو بصير جوا بالزبط مثل المايكرويف والغسالة وحماصة الخبز وهكذا. (ما فيها feedback)

* الـ transducer Input بترجم المدخلات للصيغة اللي بقدر يتقبلها الSystem و تدخل للController.

* الJunction بكون ++ أو -+ في الControl Systems.

* plant : الاشي اللي بدي اتحكم فيه (هيك معناها بالControl).

* لو اجانا عائق داخلي أو خارجي ما بقدر أعرفه أو أعدله بالopen loop.

(Slide 11):

closed-Loop : - لازم أقيس فيها إذا الoutput طلع صح أو لأ.

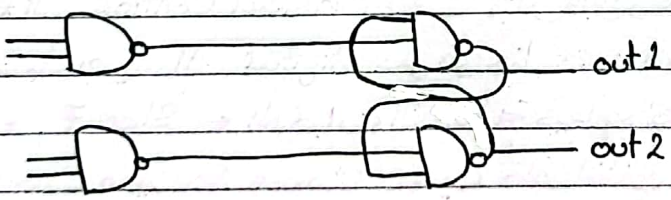
- فيها feedback، مثال المكيفات اللي فيها Sensors بتقيس حرارة الغرفة وبتضبط المكيف عالمناسب.

* الoutput transducer بخلي الoutput يتوافق مع الinput لحتى أعرف شو المشاكل.

* لو كان فيه علاقة - بين الoutput transducer والinput transducer معناها في عنا negative feedback. (الأكثر شيوعاً هو الnegative مش الPositive.)

* مثال سيئ للPositive feedback sys : في اجتماع على الZoom والدكتور بيحكي فالصوت بفوت وبطلع فهذا ممكن يسببلنا احنا إزعاج لانه بطلع صوت تشويش أو تضخيم.

* مثال جيد على الPositive feedback : الSR-Latch



* مش بالضرورة الfeedback يكون جاي من جهة وحدة ممكن يكون من أكثر من جهة. * الclosed أفضل ولكن more complex + hardware

* سؤال امتحان : لو بدي أحول التوستر لـ closed-loop شو بعمل ؟ بنضيف Sensor light بقيس درجة التحميص مثلاً و Sensor لنوع الخبز بس هنا حيزيد التكلفة لانه لازم ما نحرق الSensors لازم بالنهاية جوا حماصة هم.

(Slide 15) :

* مثال المصعد: أنا بالطابق الأول وبدي أطلع للطابق الرابع:

- ممكن يطلع المصعد شوي شوي أو بسرعة أو بسرعة متوسطة.

- الـ Switch بالـ Signals هو unit step function.

- لو كان سريع كثير ماشي منيح بس حيطلعك بسرعة وهذا مش آمن ولو كان بطيء كثير انت رح تزهق فلازم ناخذ كل الأمور بعين الاعتبار لنوازن كل شيء بالـ System.

- Steady-state response: النتيجة النهائية، مش ضروري دايما تكون النتيجة دقيقة مية بالمية بس لازم تكون أقرب ما يكون للدقة.

(Slide 16) :

* settling time: قديش بده وقت من ما بلش يطلع وينزل لحد ما ثبت →

* في حالات سيئة للاستجابة أسوأ هذه الحالات هي: underdamped ← بضل المصعد يطلع وينزل عن النتيجة النهائية تاعتك (مثل الطابق الرابع لما تطلعه بالمصعد بضل يتنقل بين الثالث والخامس وهيك) لازم نحاول نقلل هذا الخطأ قدر الإمكان حتى لو صار ما يأثر كثير. الحالة الثانية: overdamped ← بكون بطيء كثير. والحالة الثالثة هي الـ Critically ← وهي أفضلهم والأقرب للصواب (95 %) والـ rise time إلها منيح.

* الـ dead time: الوقت الميت الغير مستعمل قبل ما الـ System يبلش يشتغل منيح.

* الـ Rise time: الوقت من ما أعطيت أمر للـ control لحد الوصول للـ (95%) Steady-state

(Slide 19) : * مثال آخر هو الـ Computer HDD. الـ underdamped سريع والـ overdamped بطيء

- يعني هون الـ Rise time يكون أفضل ما يمكن بس لو علّيت كثير الـ response time ممكن يتفتت القرص ويخرب حسب قواعد الفيزياء (circular motion)

(Slide 20): الـ Steady state هون لحتى أوصل للـ File اللي بدي اياه من القرص (مافيو مزح)

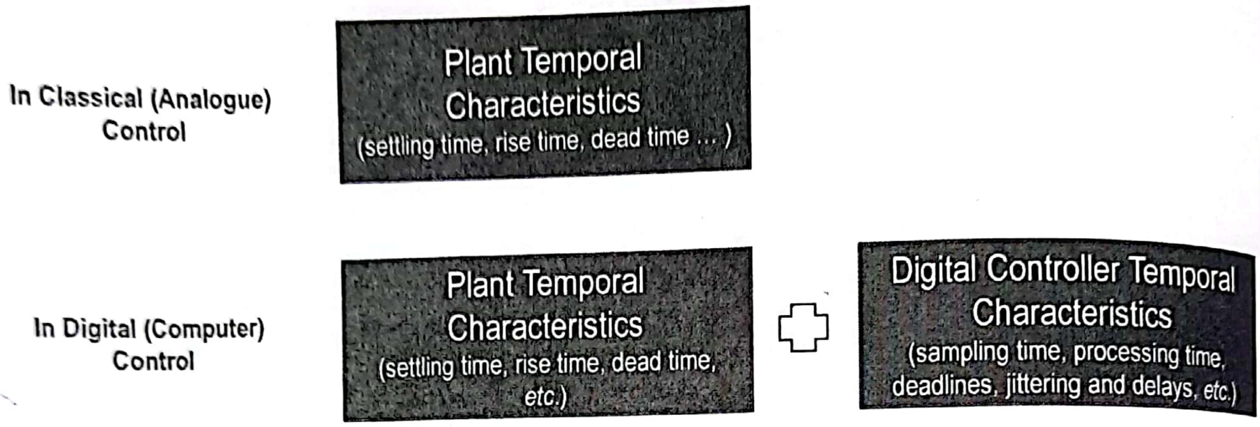
Stability: معناها هل الـ System حيضل تمام طول الوقت ولا ممكن يصير اشي لقدام.

A- Natural Response :

اللي بتحوي فقط الـ initial states اللي محطوطة هالـ inputs.

B- Forced Response : My own input → قوة أنا بعطيها بإيدي هالـ system ممكن يخرب شوي ويرجع وممكن يخرب الـ System بالكامل حسب القوة اللي دخلت عليه: مثل لو كان زمبرك حامل كيلو بس وأنا شديته وصار يحمل أكثر فممكن يتحمل لدرجة معينة بعدين ينقطع أو يخرب بالكامل.

Temporal Requirements in Computer (Digital) Control (I)



In Classical (Analogue) Control

Plant Temporal Characteristics (settling time, rise time, dead time …)

In Digital (Computer) Control

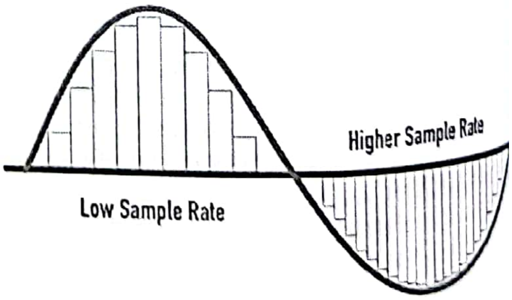
Plant Temporal Characteristics (settling time, rise time, dead time, *etc.*)

+

Digital Controller Temporal Characteristics (sampling time, processing time, deadlines, jittering and delays, *etc.*)

Temporal Requirements in Computer (Digital) Control (II)

- In a classical control system without digital controllers, the temporal requirements are restricted to analyzing the timing of the physical system (settling time, deadtime, risetime, transient time, *etc*).
- In a digital system, more timing requirements are introduced → *e.g.*, ADC sampling rate ($T_{sampling}$)
- Some sensors are digital (they have a built-in ADC and give us digital, (mostly serial) output. The programmer selects the output data rate (ODR, *i.e.* sampled rate) in software.
- Many other sensors are analogue which must be interfaced to an ADC, correct sampling rate must be chosen.



Temporal Requirements in Computer (Digital) Control (III)

- Choosing the sampling rate is not as easy as one thinks!
- Must comply with Shannon theory of sampling and the lower bound is Nyquist rate (*min* 2πf). Sampling at higher frequencies is allowed (though not often necessary) and called oversampling.
- Sampling below this Nyquist rate is undersampling, and results in aliasing; that is, the correct original signal cannot be reconstructed
- Between each sample and the next, the controller might need to do online calibration, signal filtering, processing and conrtol. We call this time ($T_{processing}$).
- $T_{processing} \leq T_{sampling}$
- In this case, the deadline to finish processing is before the next sample arrives

* Sampling Rate ← ال ADC بكون Built in جوا ال digital sensor.

*كل فترة زمنية معينة بيآخذ منه ال ADC قطعة (discrete time signal)

Temporal Requirements in Computer (Digital) Control (IV)

" حلول لو خلص بعد ال deadline " التزم بال Minimum Sample rate الي هو ال Nyquist لو ما بقدر أنزل بشروط.

ما خلص بالوقت المناسب

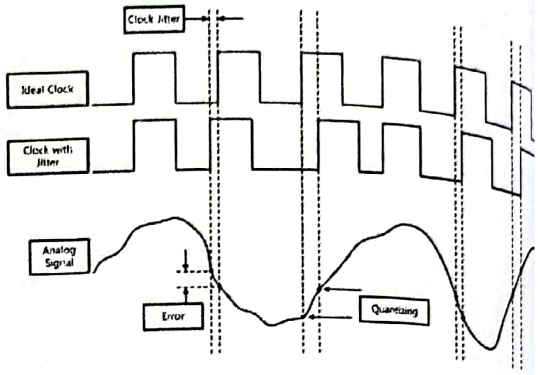
- Can the controller finish the required tasks ($T_{processing}$) related to one sample before the new sample comes in? If not, what to do?
  1. Reduce the sampling rate to the lower bound *if the specification allows you to* → ($T_{sampling}$) increases giving us more time.
     أنزل بس بشروط معينة
  2. Use compiler setting to optimize your code for speed (be careful as the compiler can remove variables necessary for code operation: Ports)
     * نكتب Assembly code لنحل المشكلة.
  3. Optimize your control code by using efficient algorithms or libraries, or in extreme cases, revert to assembly coding. Remember some sorting algorithms are O(n^2), others are O(log(n)). → نغير ال Algorithm.
     → كل الأرقام حتى مع الفواصل ← Integers subset of real Num.
  4. If your control code uses real numbers, use fixed-point arithmetic instead of floating-point arithmatic. Fixed-point arithmetic uses integer hardware and is therefore faster. Many DSP and Control libraries are fixed-point.
- Suppose sampling rate is 10KHz, processor is running at 50Mhz, the task to process one sample consists of 25000 assembly instructions, and on average, each instruction takes 1.5 clock cycles to execute, could it work?

$T_{sampling}$ = Deadline = 1 / 10,000 = 0.0001 sec

$T_{processing}$ = 1 / 50,000,000 (cycle time) x 25,000 (Instructions) x 1.5 (CPI) = 0.0005 → More time than the deadline

Temporal Requirements (Jittering Issues)

- Jittering as a word means to suffer from slight irregular movement, variation, or unsteadiness. We have two types:
  1. Jittering Δt due to uneven clock cycles that affect sampling (see adjacent figure)
  2. Jittering $\Delta t_{computer}$ due to uneven response time to a certain event
- If Δt << $T_{sampling}$ considered almost constant, system can handle it!
- If Δt is proportionally significant, that is the clock shifts between cycles is large compared to the cycle itself →big problem
- Another source of jittering is $\Delta t_{computer}$ *which is jittering in response time to an incoming event*. If event A happens, it can interrupt the processing and call the ISR say in 10μs, but if the current task being processed is of a higher priority than event A, then the ISR of A cannot start until the current task finishes, this could be tens or hundreds of microseconds later.
- *Some digital controllers can guarantee a maximum response time, useful to design for the worst case!*



Temporal Requirements in Real-Time Control Systems

Some control systems might have real-time requirements. Real-time means:

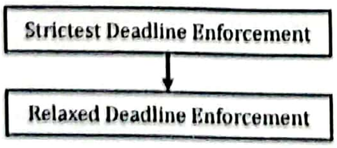
1. Tasks must finish within a certain deadline
2. Tasks must not start before a certain time
3. System must respond to external events quickly

Timing Requirements (*e.g.* deadlines) are categorized into:

1. Absolute: response must occur at defined deadlines (an action must happen every 10 ms).
2. Relative: response must occur within a specified period of time following an event (when event A happens, we have 10 at most 10 ms finish associated task).

Real-Time Systems Classification by Deadline Type:

1. **Hard Real-Time Systems**
2. **Firm Real-Time Systems**
3. **Soft Real-Time Systems**


Strictest Deadline Enforcement

Relaxed Deadline Enforcement

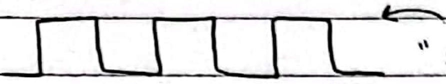
18/10/2022

Slide 23 : Temporal Requirements in computer Control:

هون حنبلش نضيف خواص ال Computer قبل كنا نحكي بس علاقات لحظية:

* بال ADC بنركز على ال frequency لل Signal.

* أي Signal لها: amplitude و frequency , phase

* Max frequency component : *  "Squarewave"

"Sin wave" ↓ بقسمها لـ Sin waves وبشوف وين أكبر وحدة فيهم وهي بتكون ال Max

واللي بنطلع ال frequencies هي ال (T) fourier Series وبعدين بغير

أبرمج ال Sensor حسب الناتج وال Max Frequency ← Transform

* كل Signal هي عبارة عن مجموعة مركبة من ال Sin waves أو cos waves.

* كل Signal بتيجيني لازم تخلص Processing خلال الوقت المحدد فلو كثرت ال Signals ممكن ال codes ما تقدر تعمل ال Processing بشكل صحيح ويصير ياخذ ال data بشكل سريع وهذا غلط (high sample Rate).

* Dsampling ≥ Dprocessing.

*سؤال امتحان: (اخر اشي) Slide 26 : ما بقدر انزله من 2πF (لما أزيد ال Dsampling) دا

time ← D Sampling = الفرق بين كل Signals والثانية = $\frac{1}{10\text{kHz}}$. $\frac{1}{10,000}$

Number of instruction

$$D_{Processing} = \frac{1}{50,000,000} \times 25,000 \times 1.5$$ → each inst. takes 1.5 cycle

*اذا كان ال Dprocessing أعلى بازيد ال Dsampling بتقليل ال rate (لحد 2πF أقل اشي)

أو بقلل عدد ال instruction (clock cycle, optimization) وهذا يعني تقليل Dprocessing.

* Nyquist's : لو عندي Signal ال frequency الأكبر فيها 200 Hz فلازم ال Sampling يكون على الأقل 400 Hz.

Slid 27 :

* ال clock عندي مش ideal فما بنقدر ناخذها بشكل دقيق فبصير عندي إزاحات يمين وشمال (clock with Jitter) (ΔT)

* ال Source الأول لل Jitter كهربائي ما دخلنا فيه، أما ال Source الثاني حاسوبي من الكمبيوتر نفسه

idle ← off 50% , on 50% بس ممكن بالواقع تختلف

* انا بعمل Sampling بشكل كبير بكاميرة وال Jitter باخذ وقت كثير هون بكون في مشكلة:

* ال deadline هون إنه تيجي Sample ثانية *ممكن ال System يبلش قبل بسبب ال Jitter.

* لما يجيني interrupt وبدي استجيب إله كاميرة بستجيب بوقت معين بس بنحصل نفس النتيجة.

* الجيد إنه عند ال controllers بتحسبلنا ال worst case (max response time) فبنشتغل عليها.

Real-Time Systems Types (I)

1. Hard Real-Time (HRT) Systems

→ Under all circumstances, ALL 'hard (critical)' tasks MUST meet ALL their deadlines.

→ If not, system failure causes catastrophe or death.

→ Imperative that responses occur and associated tasks finish within the required deadline.

→ Response after deadline has no value!

- Guaranteed services required → functional correctness and timing correctness (response is worthless if you finish before deadline with the wrong result)
- Hard Real-Time Systems must be PREDICTABLE and DETERMINISTIC. Systems must always behave in the same way.
- Analysis of estimated worst-case time → Scheduling algorithm and system must pass schedulability test
- In practice, the time bounds for HRT ranges from microseconds to milliseconds.
- Deadline does not necessarily imply "imminency" → Hard real-time task does not need to be completed within the shortest time possible → Only within the bound and before the deadline

Real-Time Systems Types (II)

2. Firm Real-Time Systems

→ Tasks missing their deadline **will not** result in a system failure, and no catastrophe or death

→ But, Infrequent misses lead to *performance degradation* **(loss of QoS, quality of service)**

→ Response following a deadline has no value (*e.g.*, Video frame processing)

3. Soft Real-Time Systems

→ Deadlines desired to be enforced, but they are not strict. (**Best-effort service** → deals with average response times)

→ Frequent deadline misses do not cause errors, but the result of the task **might** no longer be as useful.

→ Response following a deadline is not wasted, but degrades as more time passes.

→ Usually specified by some probability? What is the probability that task A misses its deadlines 10% of the time?

→ Probabilistic analysis → complexity at design time!

→ Time bounds between fraction of a second to few seconds (example is Home IoT, if room temperature reading misses its deadline, no big deal)

- Complex real-time control systems could consist of subsystems of any of the three types (airplane control systems)

* Real time Means:

① تخلص الـ task قبل الـ deadline. الي لازم يكون:

- absolute: ثابت، كل وقت معين بصير اشي (زي كل عشر ثانية بصير شي) ← كل عشر ثانية بده يصير ① اشي.
- relative: كل ما يصير اشي لازم يخلص خلال وقت (هون من لما يبلش بال input مثلاً عشر ثانية من وقتها) ② ← من يوم ما يجي event معك عشر ثانية إنك تعمل اشي.

② ما لازم الـ Task الثانية تبلش قبل ما نخلص الي قبلها.

③ استجابتها للـ Interrupts تكون سريعة.

* Example:

```
int x = 1;
int y,z = 1, m;
y = z+x;
cout << y;
```

* الـ compiler هون رح يكون حجز مكان للـ x,y,z فقط لأنّا الـ m لأ لأنو ما استخدمها بس بالايمبيديد هنا ممكن يسبب مشكلة لأنه مرات التغيير بكون بالهاردوير فال compiler مو فاهم إنه مثلاً كان لازم يحجز لـ m مكان.

* Imminency (السرعة) :

* لازم ما نخلط بين السرعة وبين الـ deadline. (يعني ممكن نخلص الـ task قبل الـ deadline).

* Soft Real time → best effort Service.

* Firm Real time → مثال عليها الصورة، لازم يخلص قبل ما يحس المشاهد بفرق.

* Functional Requirements : " شو بدنا الـ System يعمل "

A- Data Collection.

B- Signal Conditioning.

C- Plant / process Control.

D- (optional) Alarm monitoring / Data logging / Man Machine Interaction.

* طالما الـ Noise مش كبير كثير وما بقدر يقلب الحالة من 0 ← 1 أو العكس فحتى وجود الـ Noise عادي عندي في الـ digital أما بالـ Analog ما بقدر أشيل الـ noise فما بقدر أعرف القيمة الأصلية للـ Analog.

* الـ noise ممكن تكون إشعاعات أو Temprecture أو غيرها.

* الـ Filtering للـ noise ممكن تخففها بس ما بتشيلها نهائياً.

* اتجهنا للـ digital لأنه حتى مع وجود الـ noise بنقدر نعرف القيمة الأصلية لأنه ما في إلا 0 و 1، كمان بنقدر نخفف الـ noise بالـ error detection and correction.

* لما نصنف الـ control systems غير إنه بنتبه للـ Temporal requirements
والـ timing والـ Functional req. بنتبه عأشياء أخرى مثل Dependability
plant or ADC
والـ Performance والـ Robustness.

Control Systems Requirements I

A. Data Collection / Acquisition

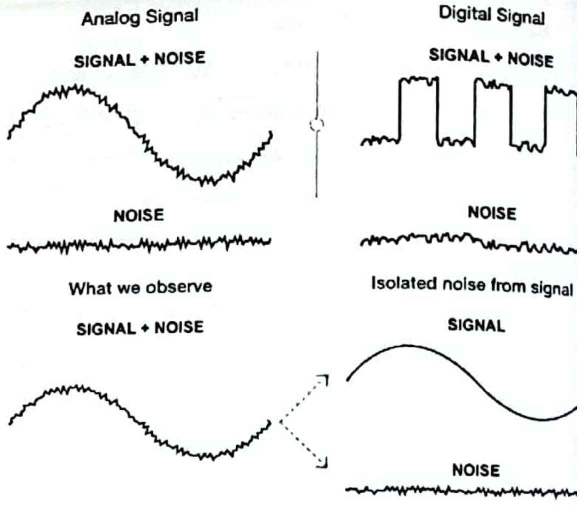
B. Signal Conditioning

Signal conditioning is used to refer to all the processing steps that are necessary
to obtain meaningful *measured data* of plant from the raw sensor data.

- Sensors produce raw data! (*e.g.*, voltages, currents).
- Scaling to required values (*e.g.*, sensor voltage to input voltage of port pin)
- Inherent Measurement errors (*e.g.* A/D Quantization).
- Sensors values also need calibration.
- Noise effects → Must be filtered out (Anti-Aliasing filters, DC Filters, Digital Filters, *etc.*)

C. Plant / Process Control

D. (Optional) Alarm Monitoring / Data Logging / Man Machine Interaction



Control Systems Requirements II

1. Dependability Requirements: ← بعد مايمر وقت معين هل حيظل الـ System شغال ولا لأ.

A- **Reliability (survivability):** probability that a control system will provide the plant control task until time t, given that the system was operational at $t = t_0$

e.g. $R(t) = e^{-\lambda(t-t_0)}$, where λ is constant failure rate in failures/hour ← كلما كانت أصغر كلما كان

Ultra-high reliability when $\lambda < 10^{-9}$ (automotive-avionics) ← عادة بنلاقيه بالطيارات والسيارات لإنها لازم تكون reliable.

↓ أسوأ واحد ←

Mean Time To Failure (MTTF) = $1/\lambda$ ← reliability factor

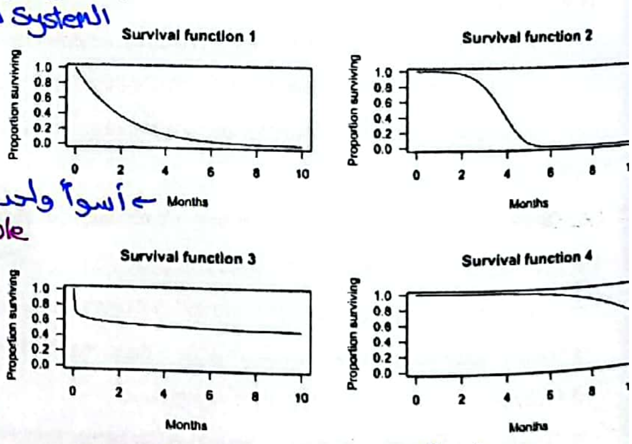
reliable أكثر.

If a car company produces 1 million cars in one year, and on average each cars drives two hour a day each day of the year, and only one car fails during this whole year:

total ساعات التشغيل ← $1{,}000{,}000 \times 2 \times 365 = 730{,}000{,}000$ hours → قسمت 1 ÷ واخذت المقلوب

$1/730{,}000{,}000 = 1.36 \times 10^{-9}$, which means that the reliability λ is close to the order of 10^{-9}

↓ قريب جداً من الـ Standard للسيارات الي بساوي 1 وهذا معناه انه Reliable.



← أعلى Survival لأنه الـ System ظل شغال منيح قرابة الـ 6 أشهر بعدين بلش ينزل شوي.

* لو مثلاً فلك خربت 1000 سيارة فبنقسم 1000 على 730,000,000 وكمل نفس الحل وهكذا. → بنشوف بعدها لو reliable ولا لأ.

* لو عندي أخطاء معينة يكتشفها
(detection + correction) (parity bits, ecc codes)
* لو عندي critical failure يعرف يتصرف الـSystem

Control Systems Requirements III

1. Dependability Requirements (continued): → تحددت حسب Standards معينة مو احنا بنحددها.

Safety requires certification

B- ▸ **Safety**: defined as 1. responses to protect the system from harm (e.g., error detection)
2. reliability against critical failure modes (e.g., plane crash, self-driving car accident)
For example, *safeRTOS* is a certified safe version of *freeRTOS* ← ماتور الطيارة
components must successfully pass certain tests like the FCC, CE, EMC
Must comply with certified industry safety standards (*e.g.*, aviation or automotive safety standards)

C- ▸ **Fault-tolerance:** protection from design and operational faults? How? → odd Number of sensors دائمًا الأفضل نستخدم
Hardware redundancy → *e.g.*, Two lock-step processors in tandem (*e.g.*, ARM Cortex-R processors) / multiple sensors
In software, roll back/recovery and checkpoints (similar to computer games ☺) however; in hard real-control systems:

1. Difficult to guarantee a deadline when error occurs → roll-back and recovery can take unpredictable time.
2. The error could have caused irrevocable action (remember we are connected to other hardware which affects the plant controlled)
3. Temporal accuracy of the checkpoint data is invalidated by passage of time

مثال
الطيارة الي وقعت لإنه كان فيها
2 sensors وخربانين

يعملوا نفس العمليات
ككل القطع ولازم يتطابقوا ليكونوا
سليمين 100 %

D- ▸ **Security**: protect system from intentional harm or access
↓
ما يصير
هكر.

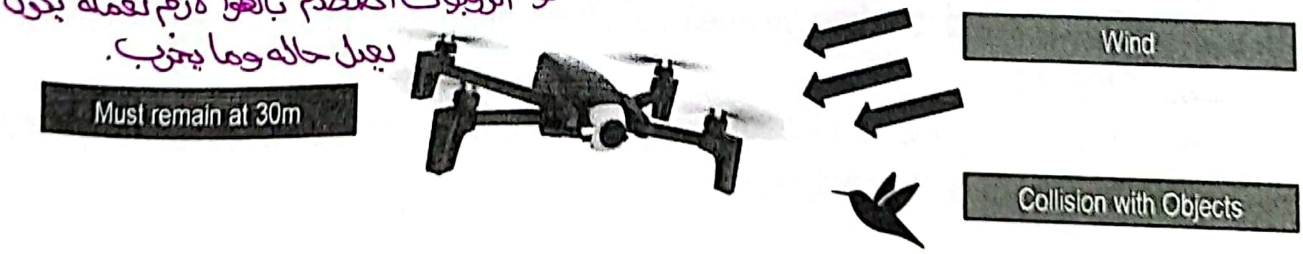
إحدى الحلول إنه يرجعني لأقرب مكان قبل ما يخرب
زي الألعاب بس حل صعب وأساسًا هيك هيك الغلط صار وحدث وأثّر وحتى لو رجعت الـ data قيمة ما

Control Systems Requirements IV

2. Performance: timing of responses or throughput necessary? → يستجيب ويعطيك الشغل بوقت سريع

3. Robustness: protection from external interference and perturbations

(مثلاً لو الروبوت اصطدم بالهوا لازم نعمله يكون قابل
يعدل حاله وما يخرب.



4. Scalability: Perform reasonably in an environment with added load. We have a swarm of three drones doing a task, and we add more drones to help to complete a task that the three cannot do on their own. ↓

مثلاً لو روبوت اجا يدفع قطعة لقاها ثقيلة
بنادي روبوت ثاني (تفاعل الروبوتات مع بعض)

Fail-safe and Fail-Operational Control Systems

يجب أن يعود ال System لل Stage معينة يا بوقف كلها يا بيلش منها.

- Some control systems can have safe states (**fail-safe**) → When system fails, go to safe state

  عندي system بـ control معين إذا صار عندي fail كيف بدي اتعامل معه ؟

  1-fail-safe 2-fail-operational

  Examples:

  Electronic → Simple Electric Fuse

  In control and embedded software → Watchdog timers

  مثل التلفزيون لما أفصل الكهرباء لو علق ال System

  ما بطفي بعمل restart كل مدة معينة عشان ما يصير في overflow. (أصعب من ال Fuse)

  In Nuclear reactors → Magnetically held lead rods

  الرصاص بمتص كل النيوترونات

  * Requires high error-detection coverage → the probability that an error is detected, provided it has occurred, must be close to one

  لو صار خلل بالمفاعل بدخل الرصاص بمتص كل النيوترونات فبتعالج المشكلة.

- In certain applications, you cannot identify a safe state!

  Example: Flight control system of airplane or space craft!

  * Must provide minimum level of service to avoid catastrophe even if failures occur, or sound alarms.

  * If main power shuts down, switch to auxiliary power in hospitals.

  * These systems are called ***fail-operational*** → لازم لما يكون عندي fail يكون عندي Mechanism معينة يا تعطيني Alarm تنبه سائق الطيارة مثلاً لو بيقدر يحل المشكلة أو هيك.

يعني بيطفش مثل الطيارة تعملها Restart بنص الشغل فواي

الفرق الأساسي بينهم إنه هون لازم ال system يضل شغال. → النوعية من المشاكل تسمى fail-operational

Control System Block Diagram Exercise

The control of the recording head of a dual actuator hard disk drive (HDD) requires two types of actuators to achieve the required high real density:

- The first is a coarse voice coil motor with a large stroke but slow dynamics
- The second is a fine piezoelectric transducer (PZT) with a small stroke and fast dynamics.

A sensor measures the head position, and the position error is fed to a separate controller for each actuator.

Draw a block diagram for a dual actuator digital control system for the HDD.

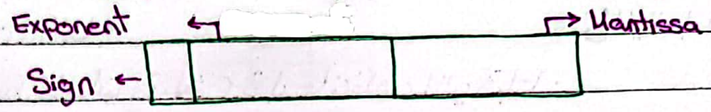
مواصفات لل hard disk ارسملها ال block diagram مبين النوع لو open أو closed ، ترسم ال sensors واتجاهاتها بأكملها.

23/10/2022

* ADC جزء أساسي من ال Control system

* ال feedback بجيني من ال Sensors. ← أسعارها تتراوح بين few Cents لإكا مئات الدولارات.
لـ بسولها Analog / digital وبضيفوا ADC عالجوين. * Slide 26.8

* ال floating point داخل ال Computer : ← بتستهلك وقت كثير لتعريفها بهاي الطريقة.



* بال Python والماتلاب بتكون معرّفة direct, انها "double"

* لو بدك تعرف float / double بال control system بياخذ وقت طويل ومساحة و Power.

* Fixed-point : زي ما أخذنا بال logic

... 2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	...
	1	0	0	0	1	1		

Ex: → = 8.75 ←

لـ صاروا يعتمدوا بتعريف ال float / double عشان ما ياخذ وقت طويل.

* FPU : Floating point unit (ال hardware اللي بيعمل ال float/doubles) مش متوفر بكل ال Controllers

* ال Fixed-point بختلف عن تعريف ال float/double لكنه أصبح يستخدم لذلك لأنه أسرع، ويمكننا تحويل ال float/double إلى صيغة ال Fixed-point باستخدام المكتبات الخارجية في ال C وال C++. (هو أسرع لأنه بيستخدم ال integer hardware).

* CMSIS DSP ← مكتبة خارجية ← - f32 ← ال floating العادية (دقيقة) (طويلة)
"كلهم نفس الوظيفة بس هي كمثال والفرق بينهم" → - q31 ← ال Fixed-point (32 bits)
لو استخدمتها بقدر أعمل Parallel → - q15 ← ال Fixed-point (16 bits).
Processing بس بال q31 أ، ال q31 أدق بس أبطئ وبتاخذ Storage أكبر.

* ال floating بوصل لـ 64 bits بس ال Fixed-point أقصى سعة إله 32 bits.

* SNR ← Signal to noise Ratio (Self learning Material 2).

* Improving Resolution ← لو بدي أحصل شيء منيح بس بسعر مناسب.

* لما تجمع 2 frequencies نفس بعض بكون ال Amplitude عالي أما لو كان ال 2 frequencies مختلفين بكون ال Amplitude أقل. → Convolution

* PID Controller ← Proportional-Integral-derivative controller ← سنأخذه بالمادة.

Computer Controlled Systems (Digital Control)

University of Jordan

School of Engineering

Department of Computer Engineering

Material prepared by Dr. Ashraf Suyyagh

Review of Signals, Fourier Transform, and ADCs

Table of Contents

In embedded, cyber-physical, and control systems, the controller interfaces with the real physical world to either sense or actuate the plant. In feedback control systems, sensors are attached to the plant to provide the controller with measurements about the state of control to enable the controller to tune its actuating/control signal to achieve the desired output. The physical plant could be interfaced to either analogue or digital sensors. Analogue sensors must be interfaced to an intermediate ADC before their value is used by the digital controller/computer. Digital sensors have their ADCs built-in, and through software they are configured to provide data at a given output data rate (ODR); that is provide data at a certain sample rate.

*إذا نزلنا عن Nyquist بصير under Sampling وهذا مشكلة.

* ال Nyquist رائع بيعطينيال freq المناسبة للـ Sampling.

The Sampling Rate

Per Shannon's Theorem of sampling, we must sample the analogue signal at a minimum signal sampling frequency. This frequency should be at least twice the maximum frequency component that exists in the finite bandwidth signal. This sufficient sampling rate is called the Nyquist rate. Sampling below this rate is called undersampling and this means we did not capture the entire information (*e.g.*, shape) of the signal, and thus when we recreate the original signal, it will be distorted. We call this problem aliasing. Sampling at above the Nyquist rate is called oversampling. A signal is said to be

← آخذ قراءات أكثر s resolution قليلة مثل bits ما تكافئ أخذ قراءات قليلة s resolutions عالي (عشان أخفف تكلفة)

* لو ال Signal = 100Hz ← ال Nyquist = 200Hz ← هون الشرح بوضحلك
كيف لو اشتغلت s rates أعلى وليه مرات بشتغل عليها.

oversampled by a factor of ***N*** if it is sampled at ***N*** times the Nyquist rate. Oversampling can improve resolution, signal-to-noise ratio (SNR), and simplify building anti-aliasing filters. We shall discuss the first two benefits of oversampling:

Reducing SNR →

ال Signal لازم تكون أقوى من ال Noise.

When we oversample by a factor of **N**, the *SNR* improves by a factor of $\sqrt{N}$. Suppose we have a signal with a maximum frequency of 100 Hz; this means the minimum sampling frequency should be 200Hz. If we sample the signal at 800 Hz, this means that we oversampled by a factor of 4 and improved the SNR by a factor of $\sqrt{4}=2$.

* استخدامي لـ Samples أكثر بعطيني قراءات أكثر فبزيد ال SNR.

Improving Resolution

ADCs output is a digital set of binary bits. ADCs can have either a fixed-width output (You buy them with only one fixed output width), or configurable width output (they can provide different output widths that you configure in software). ADCs nominally have 10, 12, 14, 16, 20. 24, 32 bits among other configurations. High-resolution ADCs are more expensive, slower in converting an analogue sample to digital representation, yet more accurate and precise. ADCs also vary at their maximum sampling speeds that range from kSPS (kilo samples per second) to tens or hundreds of MSPS (mega samples per second). ADCs with GSPS (giga samples per second) also exist but are often used in high-speed communication systems. The higher the digital resolution and the higher the sampling rate, the more expensive is the ADC. ADCs could cost from few cents to hundreds of dollars.

Oversampling is a technique that allows the use of lower-resolution ADCs to give a comparable performance of higher-resolution ADCs. That is, we can sample our data at many times the Nyquist rate using a 10-bit ADC to give similar performance of a 12-bit ADC.

Formally, the extra number of samples we need to take to increase the resolution by one bit *n* is:

$$\text{No. Samples} = 2^{2n} \qquad (1)$$

So, if we have a 16-bit ADC, the number of samples we need to have a resolution similar to that of a 18-bit ADC is $2^{2x2}=2^4=16$ times the samples. So, if our signal has a maximum frequency of 1000 Hz, the Nyquist Rate is 2000 Hz. We can either use an 18-bit ADC at 2000 Hz, or a 16-bit ADC at 32,000Hz.

As we learnt in class, choosing a sampling rate is not an easy task. It should at least meet the Nyquist rate for that sampled signal, and the time between the sample and the next should be sufficient to finish sample processing (*e.g.,* calibrating, filtering, control, *etc*). Yet, we have just learnt that to reduce the cost of using a more expensive high-resolution ADC and increase the SNR, we can use a cheaper lower-resolution ADC but with oversampling. But oversampling means taking more samples with less time in between making our deadline constraints stricter.

سؤال امتحان ←

Exercise 1

In a certain control application, suppose that the plant is connected to an analogue sensor whose signal is fed back to a digital controller though an ADC. The sensed signal has a maximum frequency component of 15 kHz.

Given the following ADC specifications, answer the questions that follow:

	Cost	Resolution	Max. Samples
ADC 1	5 $	12-bit	24 kSPS (kHz)
ADC 2	6 $	12-bit	50 kSPS (kHz)
ADC 3	8 $	12-bit	1 MSPS (MHz)
ADC 4	10 $	14-bit	50 kSPS (kHz)
ADC 5	14 $	16-bit	100 kSPS (kHz)

- بدون قيود ← Suppose that it is sufficient for the control application purposes to use a resolution of 12-bits, which one of the ADCs would you use?
- مع قيود ← Suppose that our control application requires processing data samples that have a resolution no less than 14-bits or lower resolution with *equivalent* performance. Which one of ADC3, ADC4, or ADC5 would you choose assuming that the control task time will always finish by the deadline?
- مع قيود ← Suppose that the control task takes 0.00001 seconds on the digital controller used in our system, and that our control application requires processing data samples that has a resolution no less than 14-bits or lower resolution with *equivalent* performance. Which one of ADC3, ADC4, or ADC5 would you choose?

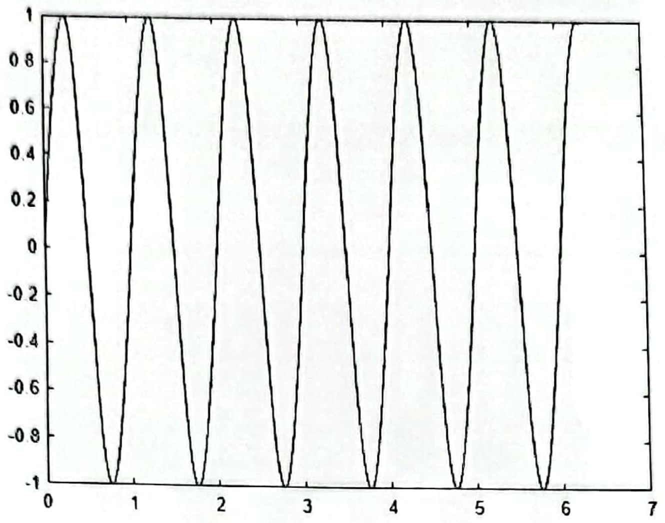
Review of Periodic Signals and Basic Analysis

As we have seen in the previous section, all our decisions fundamentally require us to have some knowledge of the properties of the signal that we want to sense, more critically we need to know its bandwidth (BW); that is **the maximum frequency component** that exists in the signal.

← هي الأساس لأعرف ال Nyquist rate وال Sampling وكل اشي.

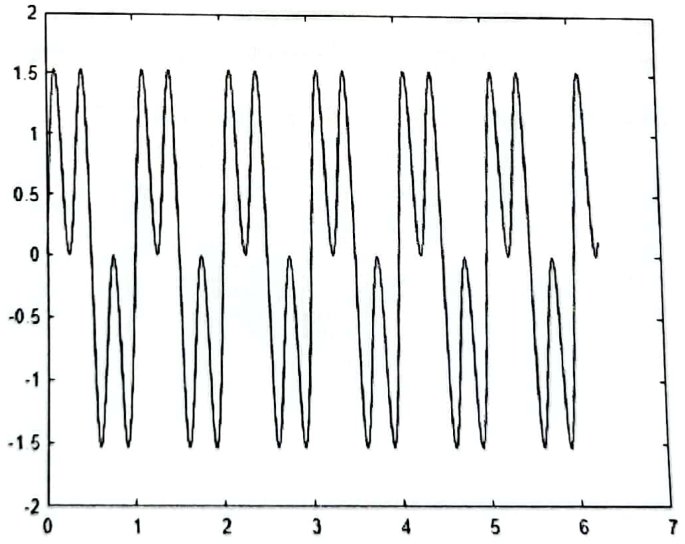
As you have studied in the *Signals and Systems* course, periodic signals can be constructed from the summation of sinusoidal waves. For example, a ***periodic square wave*** can be constructed by adding the odd harmonics of a sine wave, the more harmonics we add, the closer we get to a square wave. Let us draw a sine wave over the range of 0 to 2π with steps of 0.01 and use MATLAB's plot command to view the signal:

```
Ts = 0.01;
x = 0:Ts:2*pi;
y =sin (2*pi*x);
plot (x,y)
```



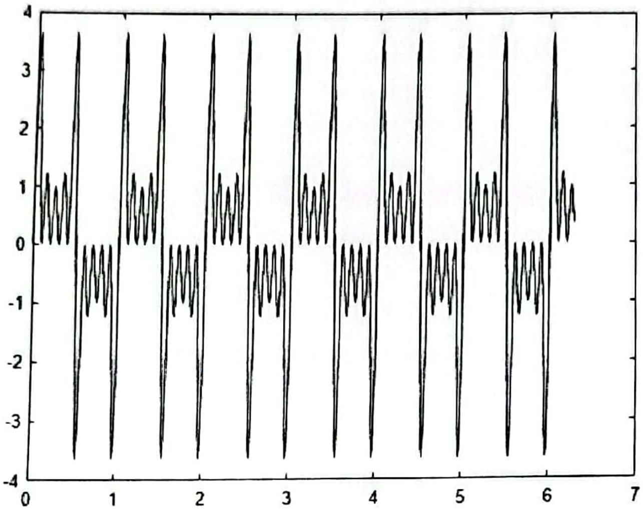
Now, let us add the third harmonic, that is a sine wave with three times the original frequency and plot the signal again:

```
Ts = 0.01;
x = 0:Ts:2*pi;
y =sin (2*pi*x) + sin(2*pi*3*x);
plot (x,y)
```



Now let us add the fifth, seventh, and ninth harmonics, and plot the signal again:

```
Ts = 0.01;
x = 0:Ts:2*pi;
y =sin (2*pi*x) + sin(3*2*pi*x)+ sin(5*2*pi*x)+ sin(7*2*pi*x)+ sin(9*2*pi*x);
plot (x,y)
```

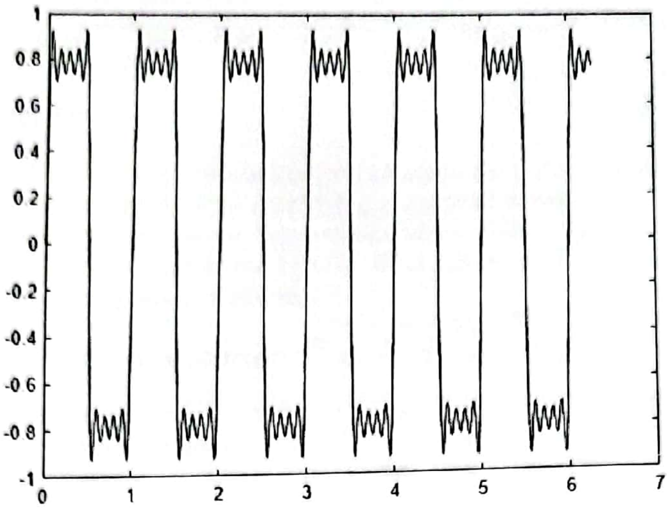


"لحد هون فشلنا لإنه كنا نغير بس عال frequency".

As you can see, the square wave that we got does not look like as the square wave we expect it to be; this is because we changed the frequencies (harmonics) but not the amplitude of the additional sine waves. Normally, we can construct different waveforms from sinusoidal waveforms through changing ***frequency***, ***amplitude*** and the ***phase*** of the underlying sinusoidal waveforms.

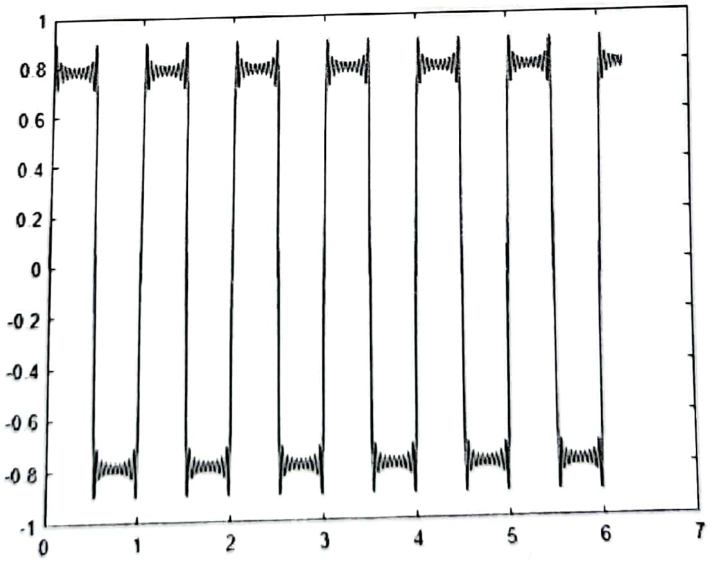
Now let us add the fifth, seventh, and ninth harmonics with varying amplitudes, and plot the signal again:

```
 Ts = 0.01;
 x = 0:Ts:2*pi;
 y =sin (2*pi*x) + (1/3)* sin(3*2*pi*x) + (1/5)*sin(5*2*pi*x)+
(1/7)*sin(7*2*pi*x)+ (1/9)*sin(9*2*pi*x);
 plot (x,y)
```



The more harmonics we add, the closer we get to a square wave:

```
 Ts = 0.01;
 x = 0:Ts:2*pi;
 y =sin (2*pi*x) + (1/3)* sin(3*2*pi*x)+ (1/5)*sin(5*2*pi*x)+
(1/7)*sin(7*2*pi*x)+ (1/9)*sin(9*2*pi*x) + (1/11)*sin (11*2*pi*x)...
     + (1/13)* sin(13*2*pi*x)+ (1/15)*sin(15*2*pi*x)+ (1/17)*sin(17*2*pi*x)+
(1/19)*sin(19*2*pi*x);
 plot (x,y)
```



Do not concern yourself at this stage how we got the correct amplitudes and frequencies of the square wave. We simply got them from an equation from a book. You will always be given any equations if necessary.

The Fourier Transform

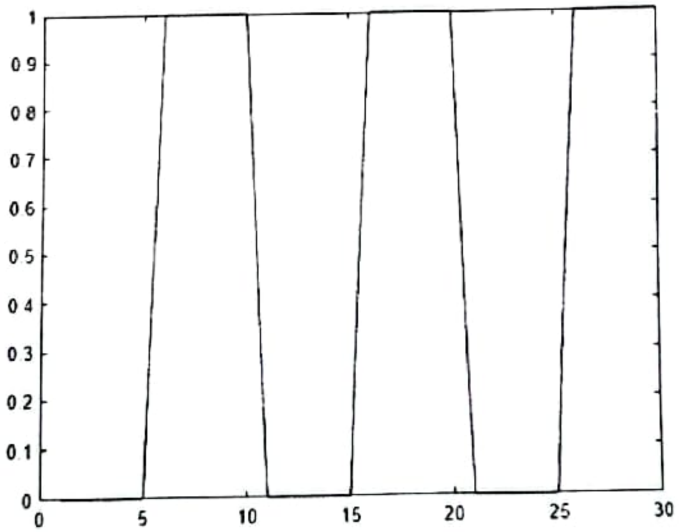
In the previous section, we have seen the underlying characteristic that periodic waveforms can be constructed from the summation of multiple underlying sinusoidal waveforms. This is fundamental to understand the basics of Fourier transform. But initially, let us review the basic of convolution. You learnt that in convolution, we flipped one of the signals, then started sliding it onto another signal while multiplying and summing (or integrating) them together.

Quick Visual Review of Convolution

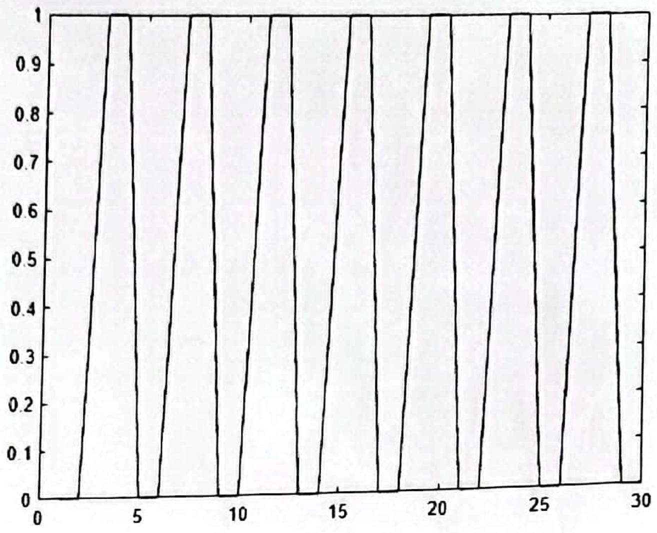
→ أهم العمليات بالهندسة : 1 Signal ووحدة ثانية
بقلب وحدة منهم وبصير أمشيهم فوق بعض واضربهم
أو أجمعهم أو هكذا!.

Suppose we have these two waveforms which differ in their frequency. To plot each of the plots in a separate window, use the figure command

```
 x = 1:30;
 y1 = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1];
 y2 = [0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0,
0, 1, 1, 0, 0];
 figure
 plot (x, y1)
```



```
figure
plot (x, y2)
```

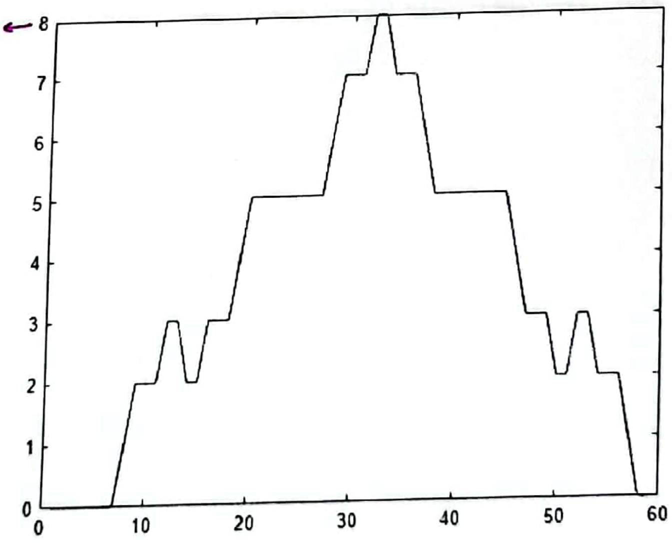


Now, let us do a convolution between the two different signals and plot the result:

```
z = conv (y1, y2);
plot (1:length(z), z);
```

Max Amplitude ←

(البطيئة مع السريعة)



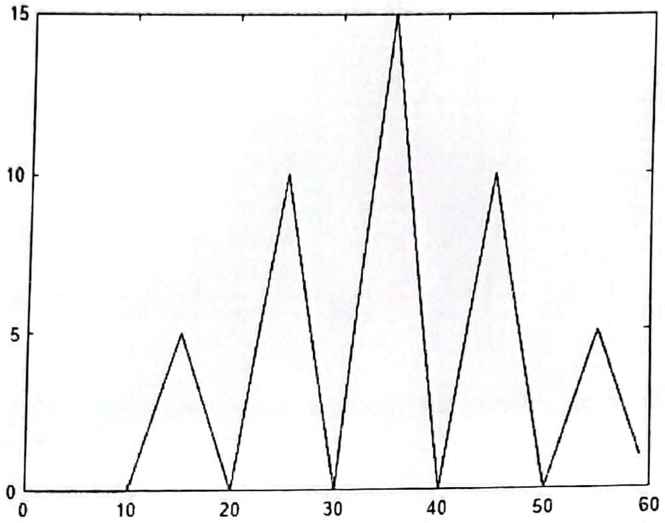
Notice that the frequency of *y1* is different from *y2*, and the maximum amplitude in the convolved signal is 8.

Let us do a convolution between the first signal *y1* and itself and plot it on a new figure:

```
figure
z = conv (y1, y1);
plot (1:length(z), z);
```

Max Amplitude ←

(البطيئة مع حالها)



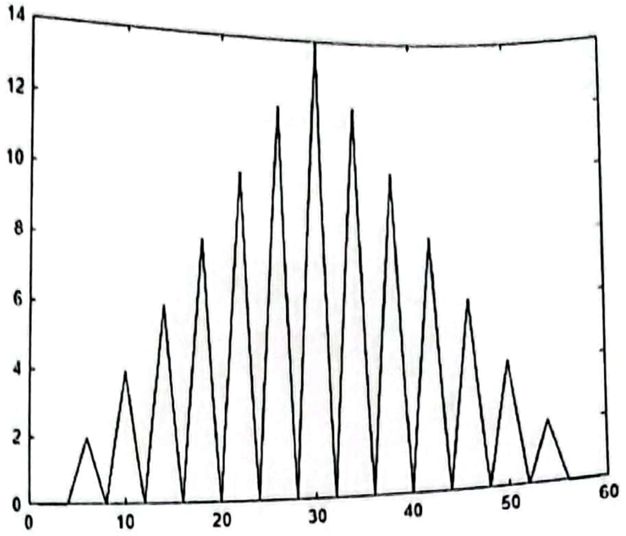
Notice that the frequency of *y1* and the convolved signal *y1* (obviously) , and the maximum amplitude in the convolved signal is 15.

Let us do a convolution between the second signal *y2* and itself and plot it on a new figure:

```
figure
z = conv (y2, y2);
plot (1:length(z), z);
```

Max Amplitude ←

(السريعة مع حالها)



Notice that the frequency of *y2* and the convolved signal *y2* (obviously) , and the maximum amplitude in the convolved signal is 14.

What we want to emphasize from the previous discussion is that we obtain the maximum amplitude when we convolve the signal with itself, that is we are convolving a signal at some frequency with itself at the same frequency, and the resulting signal has a higher amplitude than if we convolve a signal with another signal at a different frequency. This forms the basis of the Fourier transform.

The Fourier Transform

The Fourier transform is a mathematical formula that transforms a signal sampled in time or space to the same signal sampled in temporal or spatial frequency. In signal processing, the Fourier transform can reveal important characteristics of a signal, namely, its frequency components. Take a look at the definition of the Fourier transform in the continuous-time domain:

كأنك Sin/Cos بخطر عبالك بعمله Conv مع الـSignal ←

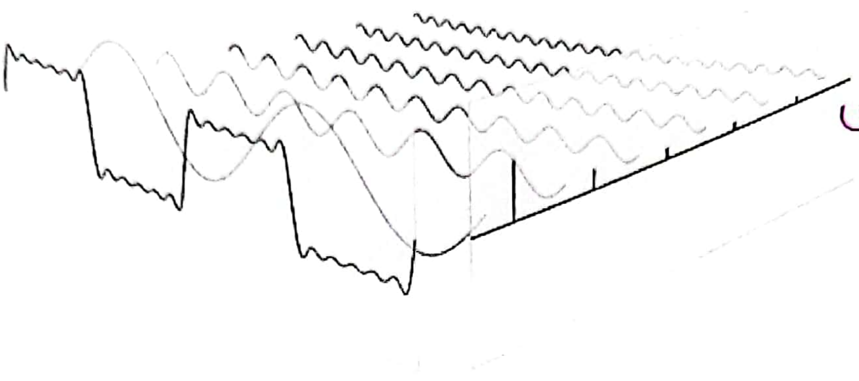
(مبنية من Sin waves) Periodic ← X(t) تعني

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)\underbrace{e^{-j\omega t}}_{\hookrightarrow \text{Sin} + j\,\text{cos}}dt$$

What it basically means, is that we are doing something similar to a convolution. We are taking a signal *x(t)* and convolving it with sinusoidals $e^{-j\omega t}$ at virtually all frequencies from $-\infty$ to ∞. Remember that *x(t)* is a periodic signal, and all periodic signals can be constructed from the summation of sinusoids at different frequencies, phases, and amplitudes. So, what happens with the Fourier transform, is that since we are integrating from $-\infty$ to ∞, it means we are scanning for the underlying sinusoids at all different frequencies.

Suppose our signal *x(t)* is constructed from the summation of sinusoids at 30, 40, and 80 Hz. The equation of the Fourier transform means that we will convolve this signal with sine waves at all frequencies. So, if the frequency is 1Hz, and we convolve it with the signal, it will give a certain amplitude, same thing for a signal of 2Hz, and 3Hz, and 10 Hz. But when we integrate x(t) with a sine wave of frequency of 30Hz, since the signal x(t) is constructed from a sine wave of 30Hz, here the convolution will give a high amplitude, or a spike. The same thing happens with the 40Hz and 80Hz cases. This is how the Fourier transform is able to transform a signal from the time domain to the frequency domain, by finding all fundamental frequencies that the signal is constructed from.

هم الحالتين الوحيدات اللي ال Amplitude عالي (spike)



You can watch this video for a more visual explanation.

Fourier Transform in MATLAB

The `fft` function in MATLAB® uses a fast Fourier transform algorithm to compute the Fourier transform of data. Consider the square wave that we constructed earlier from the summation of multiple sinusoidal waveforms (first three terms)

```
Ts = 0.01;
x = 0:Ts:2*pi;
y =sin (2*pi*x) + (1/3)* sin(3*2*pi*x)+ (1/5)*sin(5*2*pi*x);
```

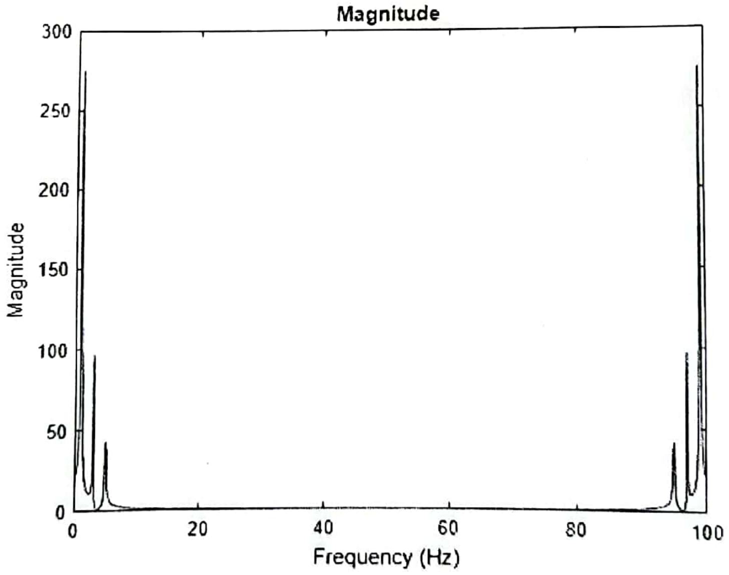
We can compute the Fourier transform of the signal and create the vector `f` that corresponds to the signal's sampling in frequency space.

```
z = fft(y);
fs = 1/Ts;
f = (0:length(z)-1)*fs/length(z);
```

When you plot the magnitude of the signal as a function of frequency, the spikes in magnitude correspond to the signal's frequency components of 1 Hz, 3 Hz and 5 Hz. Yet, since the output of the `fft` function is not zero-centered, you will see these frequencies shifted.

Frequencies

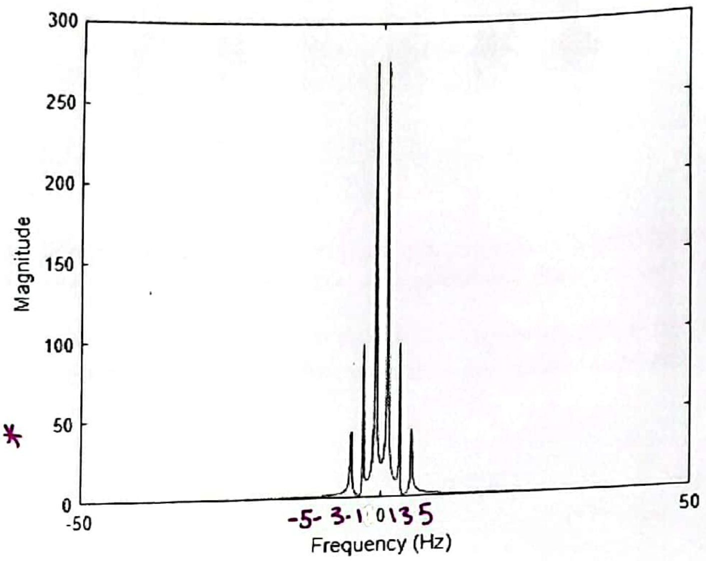
```
figure
plot(f,abs(z))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('Magnitude')
```



← بطلع غلط لأنه ال
fft بالماتلاب
مابكون Centered
عال Zero فلازم نتيجة
ال fft اعملها Shift
ليطلعوال Frequencies
الصح

The transform also produces a mirror copy of the spikes, which correspond to the signal's negative frequencies. To better visualize this periodicity, you can use the **fftshift** function, which performs a zero-centered, circular shift on the transform.

```
n = length(y);
fshift = (-n/2:n/2-1)*(fs/n);
yshift = fftshift(z);
plot(fshift,abs(yshift))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
```



* هيك بعرف الـ Max Freq = 5 فال Nyquist = 10.

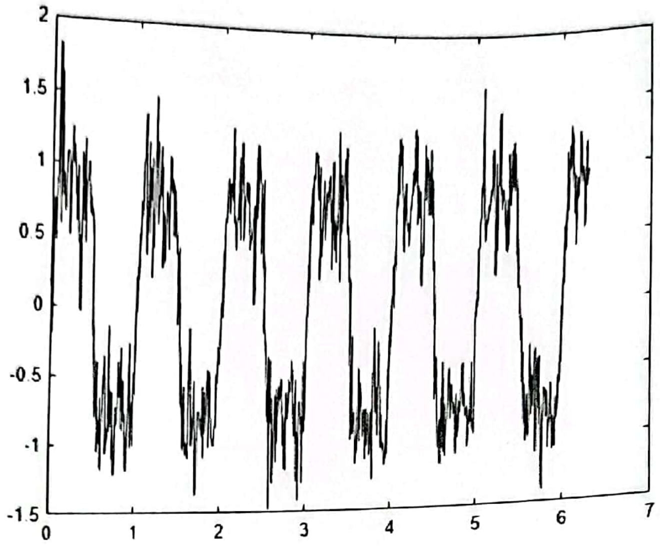
Fourier Transform and Noisy Signals

→ هون ببينلك إنه ما بتأثر

In scientific applications, signals are often corrupted with random noise, disguising their frequency components. The Fourier transform can process out random noise and reveal the frequencies.

For example, create a new signal, xnoise, by injecting Gaussian noise into the original signal, x.

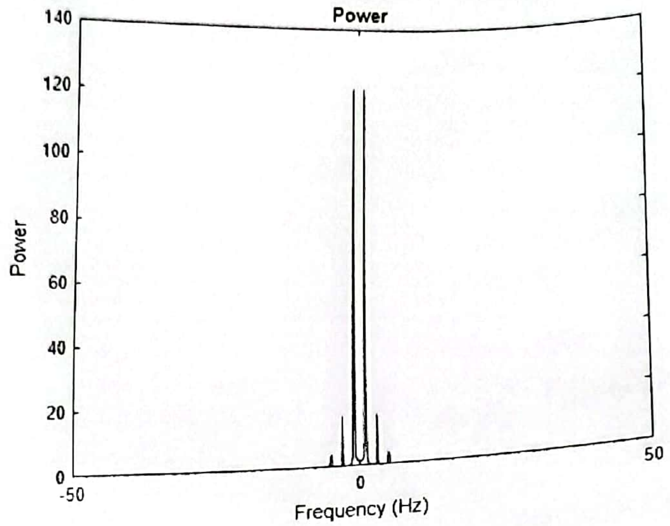
```
rng('default')
Ts = 0.01;
x = 0:Ts:2*pi;
y =sin (2*pi*x) + (1/3)* sin(3*2*pi*x)+ (1/5)*sin(5*2*pi*x);
ynoise = y + 0.25*randn(size(y));  % We are adding random Gaussian noise to the
original signal
plot(x, ynoise)
```



Signal power as a function of frequency is a common metric used in signal processing. Power is the squared magnitude of a signal's Fourier transform, normalized by the number of frequency samples.

Compute and plot the power spectrum of the noisy signal centered at the zero frequency. Despite noise, you can still make out the signal's frequencies due to the spikes in power.

```
n = length(x);
fshift = (-n/2:n/2-1)*(fs/n);
ynoise = fft(ynoise);
ynoiseshift = fftshift(ynoise);
power = abs(ynoiseshift).^2/n;
plot(fshift,power)
title('Power')
xlabel('Frequency (Hz)')
ylabel('Power')
```



Determining the Nyquist Rate

Now that we have used the Fourier transform analyse the signal in the frequency (spatial) domain, we can look for the maximum frequency. This in turns determines the Nyquist rate and all subsequent analysis.

Exercise 2

The file *signal.mat* contains a signal synthesized in MATLAB. The signal also has white Gaussian Noise. You can load the file *signal.mat* by double clicking it, or drag and dropping it into MATLAB, or through the Home Import Wizard. The variable ynoisy_signal will appear in the workspace which contains the signal.

- Draw the signal in the time domain from [0, 1] and assume the spacing is 0.0001.
- You can zoom in the figure to see the signal closely.
- You are required to find the maximum frequency in this signal and determine the Nyquist rate.
- You can zoom in the figure to see the signal closely

Answer: The signal has two frequency components, one at 400 Hz, another at 950Hz. Therefore, Nyquist's rate should not be less than 1.9 kHz.

25/10/2022

Chapter 2 : Introduction to System Modelling :

*حناخذ فيو الـ concept والتطبيق الرياضي حيكون عالماتلاب.

*هون الفكرة نعرف متى الـ System تبعي يكون Stable وكيف يكون الـ rising time وما الى ذلك.

*لبناء وتحليل الـ System لازم أعرف المعادلات لكل جزء بحاله وبعدين بكون عندي طريقتين:

1- Transfer function in frequency domain. (حناخذه).

زي الـ fourier transform وهكذا.

2- State space equations in the Time Domain. (ماحناخذه).

بِحُط عالمعادلات المعقدة وبتعامل معها لذلك هو صعب.

*Kirchhoff laws : KCL + KVL → مجموع كل الـ Voltages بيساوي صفر.

↲ التيار الداخل على load معين بيساوي التيار اللي خارج منها.

*ohm's law :(V=IR) فرق الجهد الكهربائي بين طرفي ناقل يتناسب طرديا مع شدة التيار الكهربائي المار فيه)

*وجود التكامل والاشتقاق بالمعادلات هي المشكلة اللي خلتنا نلجأ لل transfer لتبسيط المعادلات.

* linear System ← الـ output بتغير بطريقة خطية بناء عالـ input.

* LTI ← linear time Invariant System (بنحبها لأنها بسيطة بس الحياة كلها non-linear فاحنا بنلجأ للتقريب لـ linear بأقل نسبة خطأ ممكنة لتبسيط المشاكل.)

* Slide 3 : فوق عجلات السيارات مثلاً عندي System بزنبرك معين عشان يحمي السيارة لو طلعت عمطب عالي أو هيك. (معادلته معقدة وكلها مشتقات).

* Slide 4 : "RLC" ← التيار حطه r(t) مش i لانه وقيمته بتنوزع عالـ Circuit (برضه فيها تفاضل وتكامل المعادلة).

"شالهم الدكتور" خلاهم بـ Slide 11+12

← الرسمة اللي تحت Amplifier ← عبارة عن Circuit فيها transistors وظيفتها تعمل تكامل وتفاضل وجمع وطرح و Amplification بس Analog قبل اختراع الـ computers ، أكبر تطبيق لها في الصوتيات بالتلفونات وغيرها (برضه فيها تفاضل وتكامل).

* Slide 5 : linear Systems : الـ transistor بشتغل بـ 2 modes of operation

↳ Digital : بنحب تكون الـ transistor بالـ cutoff (0) أو بالـ Saturation وما كنت بدي يقضي فترة طويلة بالمرحلة اللي بالنص بينهم.

↳ Analog (Amplification) : ما بنهتم بالـ Saturation والـ cutoff بنهتم بالـ plant مثلاً لما بدي أزيد الصوت بزيد الـ Volt ليزيد التيار.

* الـ curve كله non-linear بس أنا باخذ الجزء الـ linear كتقريب وبشتغل عليه للتسهيل.

* الـSystem non-linear اللي فيها liquids أو farmans معقدة ما بناخدها هون لأنه مستحيل نتقرب لـ linear.

* Slide 6+7 : كيف أعرف إنه الـSystem تاعي linear :

① Principle of Superposition : مجموع الـ 2 inputs الأولانيين لازم يعطيني نفس الجواب لكل واحد على حدى مجموعين.

Example: $Y=5X$	Example: $Y=5e^{x}$ (non linear مشكلتها)
* $X_1=3 \rightarrow Y_1=15$	* $X_1=3 \rightarrow Y_1=100.4277$
* $X_2=4 \rightarrow Y_2=20$	* $X_2=4 \rightarrow Y_2=272.9908$
* $X_3=7 \rightarrow Y_2=35$	* $X_3=7 \rightarrow Y_3=5483.165792$
if $15+20=35$ and is therefore linear	$100.4277+272.9908 \neq 5483.165792$ and is therefore non-linear

② Homogeneity : Scaling.

← بتحقق الـSuperposition بس بتخالف الـhomogeneity

Example: $Y=5X$	Example: $Y=5x+4$ (Scaling with shift)
* $X_1=3 \rightarrow Y_1=15$	* $X_1=3 \rightarrow Y_1=19$
* $X_2=4 \rightarrow Y_2=20$	* $X_2=4 \rightarrow Y_2=24$
* $X_3=7 \rightarrow Y_3=35$	* $X_3=7 \rightarrow Y_3=39$
$15+20=35$ and is therefore linear	$19+24 \neq 39$ and is therefore non-homogenous → non-linear.

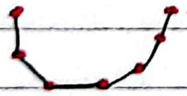
- معناه رياضيًا الاثنين عبارة عن معادلة خطية لكن من ناحية System منه ما بحقق الـScaling

* Slide 8 : اللي مهم نعرفه بالمعادلات اللي فيها اشتقاقها هي 1st order و 2nd order.

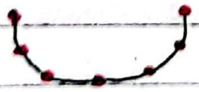
← درجة المشتقة هي اللي بتحدد الـ order تبعه.

← كيف أعرف إنه linear ؟ "if $r(x)=0$ → linear"

* Matlab → linear Interpolation : كنا نعطيه نقاط ويرسمها كخطوط linear.



→ Spline : كنا نعطيه نقاط ويربطها non-linear.



Slide 9 : Time Variant and Invariant System :

① Time Invariant System : لا يتأثر ولا يتغير بالوقت (مثل قانون اوم ، مثلاً
لو التيار 5 والمقاومة 5 فالفولتية دائماً 25 شو ماكان الوقت) . (مش معقد)

② Time Variant System : بكون ال t بالمعادلة برة (خارجية Variables بحد
ذاتها مش جوا ال function معناه ال input بتغير معي بتغير الوقت) . (معقد)

* احنا هيك بنبسط المعادلات قدر الإمكان عشان نقدر نحلها (لإنه مثلاً بقانون أوم لو حطيت
المقاومة وحطيت فيها تيار فبتسخن فبتتغير مقاومتها) بس احنا اعتبرنا هاي التغيرات طفيفة
تكاد لا تؤثر.

* Slide 10+11+12 ← معادلات

* Slide 13 ← كيف بدنا نحل المعادلات المعقدة هاي ؟ هلأ ال computers صارت تحلها.

* Slide 14 ← fourier Series هي حالة خاصة من laplace.

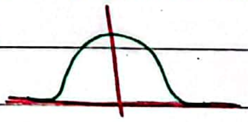
* ال fourier ل بقدر يعرفلي ال Signals لما تكون periodic.
ل كان يكامل ال function (f(t)) ويضربه بـ $e^{-j\omega t}$ ، الاختلاف انه بـ laplace بدل jω
حاط s ← s مو رقم بل Complex : $e^{-(\alpha+j\omega t)}$ ، لما تكون =0 بكون (fourier).

* لو كانت ال Signal مش Periodic فال laplace أفضل وأعم. حنستخدم ال laplace
لنحول المعادلات التفاضلية والتكاملية الى معادلات جبرية بسيطة (تربيع ، تكعيب ، خطي ، ....)

* laplace ← Continuous time domain ← المكافئ الها Z transform بعملها ع Samples.

* Slide 15 ← المهم 5+6+7) حنشتغل بإنه الدكتور بعطيك المعادلة بتطلع ع ال table
وبتحول المعادلة للصيغة المكافئة الها من هذا ال table) . (fo ← initial ← لو ماحطها الدكتور بتكون
(o =

* if My signal is periodic → like :



* Slide 16 ← بقسم ال System لـ sub systems وبتعامل معهم كـ R and C وبمسك كل وحدة
بحولها لـ laplace عشان نخلص من التكامل والاشتقاق.

* دائماً بال system تبعي بدي أعطي input مشان يعطيني output اتحكم فيه ، حيصير يهمنا اشي
اسمه ال transfer function وهو حاصل ال ratio لـ output على ال input.

* هيك بال System أحط V(t) مجموعة و r(t) مجموعة واقسمهم ببعض وهون صعب لوجود التكامل
فبلجأ لـ laplace

* Slide 17 ← أصبحت معادلة جبرية فسهل أربطها وهيك بقدر أعرف كل اشي بدنا.

* ال Partial fractions : $\frac{1}{2}+\frac{1}{3}$ ← نوحد مقامات ونجمع ، ال Partial هي العملية العكسية
لهذا الموضوع . * التفاضل بضربه بـ s والتكامل بقسمه ع s.

Modelling of Control Systems

Dr. ASHRAF E. SUYYAGH

THE UNIVERSITY OF JORDAN

DEPARTMENT OF COMPUTER ENGINEERING

FALL 2022

* حنفترض لكل ال Systems
اللي حنتعامل معها حتكون
linear
→ superposition
→ homogenety

١- اكتب المعادلات ميكانيكيا ورياضيا وهذا
صعب

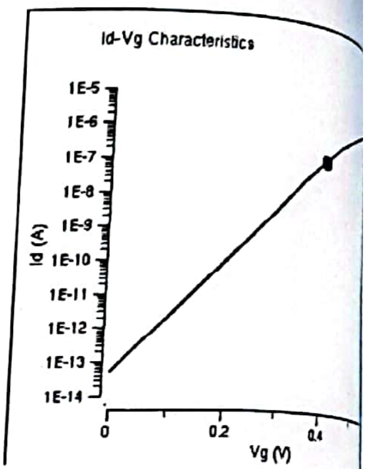
2- ننقل للـ Frequency domain

Introduction to System Modelling

- In Chapter 1 we discussed the analysis of temporal and functional requirements among others than are required in control systems design. We have also seen basic schematics and block diagram of a sample Azimuth angle system.
- Next step is to develop the mathematical model in order to analyze the systems:
  1. Transfer Function in Frequency Domain (This course)
  2. State Space Equations in the Time Domain (Another Advance Course)
- Modelling requires applying the fundamental physical laws of science and engineering: Ohm's law, Kirchhoff's laws, Newton's laws, *etc.*
- Many of these governing equations have derivatives, integrals, and many are described in differential equations
- We concern ourselves in this course with linear systems, many systems can be treated as linear if their inputs remain within a certain range.
- Yet, in general most of nature is not always linear. We deal with **approximations** to simplify the design. Non-linear systems are handled in advanced control systems.

Quick Review of Linear Systems (I)

- A great majority of physical , systems are linear within some range of the variables. For example, the operation of a FET transistor:
  - Linear operation (amplification) if Vg between 0 – 0.4 Volts
  - Non-Linear beyond Vg = 0.4 Volts → Goes towards saturation
- In general, many systems ultimately become nonlinear as the variables are increased without limit (*e.g.* spring response)
- The linearity of many mechanical and electrical elements can be assumed over a reasonably large range of the variables. This is not usually the case for thermal and fluid elements, which are more frequently nonlinear in character



A linear system satisfies the properties of superposition and homogeneity.

Quick Review of Linear Systems (II)

This Principle of Superposition

In general, a **necessary condition** for a linear system can be determined in terms of an excitation *x(t)* and a response *y(t)*.

- When the system at rest is subjected to an excitation $x_1(t)$, it provides a response $y_1(t)$.
- Furthermore, when the system is subjected to an excitation $x_2(t)$, it provides a corresponding response $y_2(t)$.
- For a linear system, it is necessary that the excitation $x_1(t)+ x_2(t)$ result in a response $y_1(t)+ y_2(t)$.

Example: $y = 5x$

- $X_1 = 3$ → $Y_1 = 15$
- $X_2 = 4$ → $Y_2 = 20$
- $X_3 = 7$ → $Y_3 = 35$

15 + 20 = 35 and is therefore linear

Example: $y = 5e^X$

- $X_1 = 3$ → $Y_1 = 100.4277$
- $X_2 = 4$ → $Y_2 = 272.9908$
- $X_3 = 7$ → $Y_3 = 5483.165792$

100.4277 + 272.9908 ≠ 5483.165792 and is therefore non-linear

Quick Review of Linear Systems (III)

This **Property of Homogeneity**

The magnitude scale factor must be preserved in a **linear system**.

Example: y = 5x

- ❖$X_1 = 3 \rightarrow Y_1 = 15$
- ❖$X_2 = 4 \rightarrow Y_2 = 20$
- ❖$X_3 = 7 \rightarrow Y_3 = 35$

15 + 20 = 35 and is therefore homogenous

Example: y = 5x + 4

- ❖$X_1 = 3 \rightarrow Y_1 = 19$
- ❖$X_2 = 4 \rightarrow Y_2 = 24$
- ❖$X_3 = 7 \rightarrow Y_3 = 39$

19 + 24 ≠ 39 and is therefore non-homogenous → non-linear System

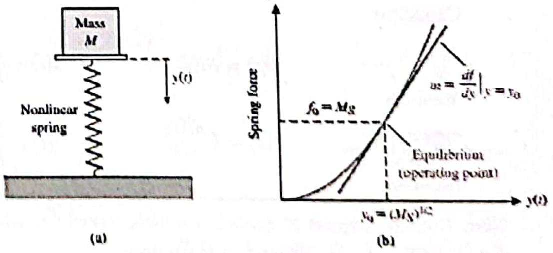
Quick Review of Linear Systems (IV)

y(t) = x²(t) (non-linear, does not satisfy the superposition property)
y(t) = mx(t) + b is not linear, because it does not satisfy the homogeneity property

y' + p(x)y = r(x) is a **first order LINEAR** differential equation
if r(x) = 0, for all values of x, then homogenous, else non-homogeneous
y" + p(x)y' + q(x)y = r(x) is a **second order LINEAR** differential equation
if r(x) = 0, for all values of x, then homogenous, else non-homogeneous

What if the system is non-linear? Then we must apply techniques for linear approximation before analyzing the system Is linear.

Approximations are as accurate as the underlying assumptions → Model errors



* ال RLC أبسط سيركت شفنا مثال الها وطبقنا كانون كيرشوف عليها .

Quick Review of Time Variant and Invariant Systems

Any system has inputs and outputs.

A time invariant system is a system in which the output does not depend on the time at which the input arrives. That is; regardless of when the input comes in, now or delayed; the output to this input remains always the same result.

We can spot atime invariant system if the equation has no dependance on time as A SEPARATE variable.

V = IR (Time-invariant)
V(t) = I(t)R (Also time invariant)

بتغير جواب ال input مع اختلاف الوقت .

A time variant system is a system in which the output changes <u>if the same input arrives at different times</u>. That is; if the input x = 5 arrives at t = 0, it will produce different result if x = 5 arrives delayed by 10 seconds.

We can spot an time variant system if the equation has dependance on time as A SEPARATE variable.

system دخلت عليه ال input بأي وقت حيعطيني نفس ال output .

Y = tX (Time-variant)
y(t) = tx(t) (Time-variant)
V(t) = I(t) R(t) where R(t) changes due to component temperature over time (Time-variant)

* In the Signals and Systems Course, and in this Control course, we assume we are working with only Linear Time-Invariant Systems (LTI)

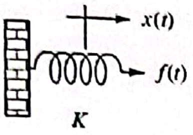
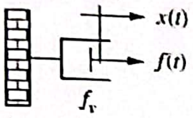
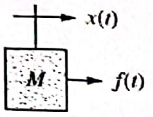
* مش مطالبين بتحليل السيركت الدكتور يعطيك ياها وقلك هي ال equation Time Domain تبعها.

Simple Dynamics of Electrical Components

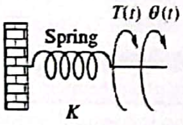
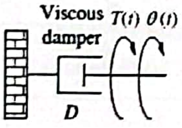
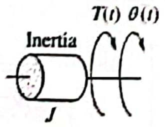
Component	Voltage-current	Current-voltage	Voltage-charge	Impedance $Z(s) = V(s)/I(s)$	Admittance $Y(s) = I(s)/V(s)$
Capacitor	$v(t) = \frac{1}{C}\int_0^t i(\tau)d\tau$	$i(t) = C\frac{dv(t)}{dt}$	$v(t) = \frac{1}{C}q(t)$	$\frac{1}{Cs}$	Cs
Resistor	$v(t) = Ri(t)$	$i(t) = \frac{1}{R}v(t)$	$v(t) = R\frac{dq(t)}{dt}$	R	$\frac{1}{R} = G$
Inductor	$v(t) = L\frac{di(t)}{dt}$	$i(t) = \frac{1}{L}\int_0^t v(\tau)d\tau$	$v(t) = L\frac{d^2q(t)}{dt^2}$	Ls	$\frac{1}{Ls}$

Note: The following set of symbols and units is used throughout this book: $v(t)$ – V (volts), $i(t)$ – A (amps), $q(t)$ – Q (coulombs), C – F (farads), R – Ω (ohms), G – Ω (mhos), L – H (henries).

Simple Dynamics of Mechanical Components (I)

Component	Force-velocity	Force-displacement	Impedence $Z_M(s) = F(s)/X(s)$
Spring 	$f(t) = K\int_0^t v(\tau)d\tau$	$f(t) = Kx(t)$	K
Viscous damper 	$f(t) = f_v v(t)$	$f(t) = f_v\frac{dx(t)}{dt}$	$f_v s$
Mass 	$f(t) = M\frac{dv(t)}{dt}$	$f(t) = M\frac{d^2x(t)}{dt^2}$	Ms^2

Simple Dynamics of Mechanical Components (II)

Component	Torque-angular velocity	Torque-angular displacement	Impedence $Z_M(s) = T(s)/\theta(s)$
	$T(t) = K\int_0^t\omega(\tau)d\tau$	$T(t) = K\theta(t)$	K
	$T(t) = D\omega(t)$	$T(t) = D\frac{d\theta(t)}{dt}$	Ds
	$T(t) = J\frac{d\omega(t)}{dt}$	$T(t) = J\frac{d^2\theta(t)}{dt^2}$	Js^2

Note: The following set of symbols and units is used throughout this book: $T(t)$ – N-m (newton-meters), $\theta(t)$ – rad (radians), $\omega(t)$ – rad/s (radians/second), K – N-m/rad (newton-meters/radian), D – N-m-s/rad (newton-meters-seconds/radian). J – kg-m^2(kilograms-meters^2 – newton-meters-seconds^2/radian).

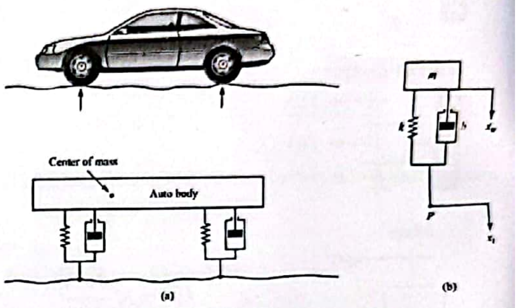
Mechanical Example of System Modelling

Automobile Suspension System

A very simplified version of the suspension system is shown. Assuming that the motion (displacement) x_i at point P is the input to the system and the vertical motion x_o of the body is the output, then, the equation of motion for the system is:

$$m\ddot{x}_o + b(\dot{x}_o - \dot{x}_i) + k(x_o - x_i) = 0$$

$$m\ddot{x}_o + b\dot{x}_o + kx_o = b\dot{x}_i + kx_i$$

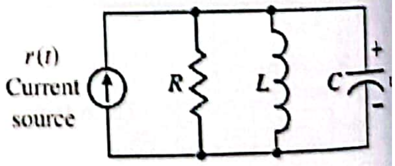


Electrical Example of System Modelling

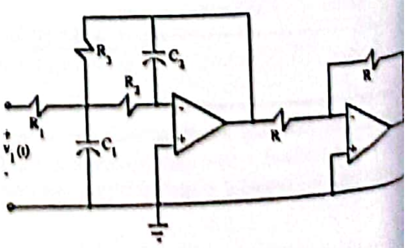
Simple RLC Circuit

One may describe the electrical *RLC* circuit shown in the figure by utilizing Kirchhoff's current law. We will obtain the following integrodifferential equation:

$$\frac{v(t)}{R} + C\frac{dv(t)}{dt} + \frac{1}{L}\int_0^t v(t)\,dt = r(t)$$

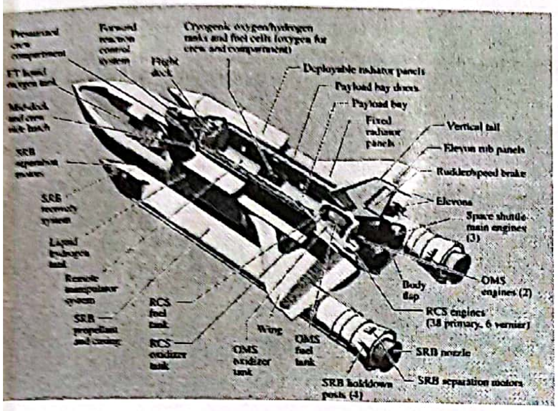


$$v_2''(t)+(\frac{1}{R_1C_1}+\frac{1}{R_2C_1}+\frac{1}{R_3C_1})v_2'(t)+\frac{1}{R_2R_3C_1C_2}v_2(t)=\frac{1}{R_1R_2C_1C_2}v_1(t)$$

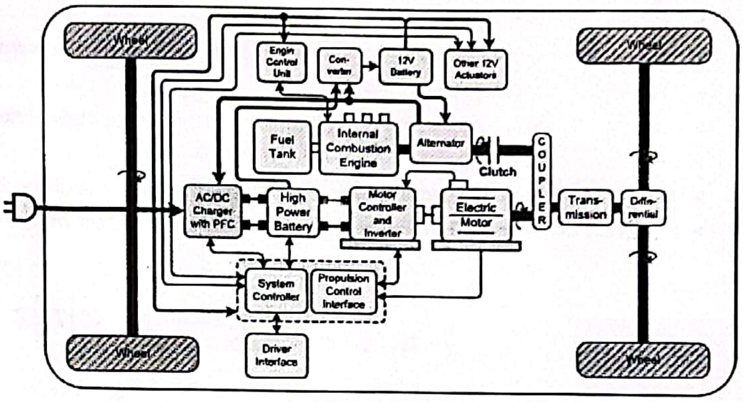


Modelling Complexity

Real systems have many electrical, mechanical, hydraulic and pneumatic components. Putting them all together and writing their equations in the time domain then trying to solve the system (e.g., finding the relationship between the output and the input) in the time domain can be difficult.



Ex.1: Shuttle Control

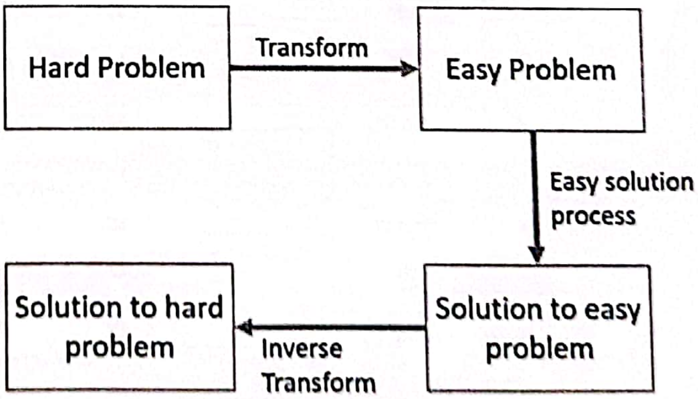


Ex.2: Hybrid Car Control

Complex Dynamics of Control Systems

- Most systems even when constructed from simple components will quickly turn to be more complex to analyze in the time domain, as many of the basic components are governed by integrodifferential equations.
- Need a much-simplified solution to transfer the hard problem into simpler one
- Laplace transform to the rescue

بتشبه شبيهة
ال Fourier
transform.



Laplace Transform

The Laplace transform mathematically resembles Fourier transform:

$$\mathcal{L}_t\,[f\,(t)]\,(s) \equiv \int_0^\infty f\,(t)\,e^{-s\,t}\,d\,t,$$

- But here, *s* is a complex number, so s is *α* + *jw*
- Fourier Transform is a special case of Laplace, when *α* = *0*, that is there are no underlying exponentials in the signal.
- Laplace is a more generic transform
- Laplace's Transform beauty lies in its ability to transform functions, even integral, differential into simple ALGEBRAIC notation.
- Algebra is much easier to solve
- Notice that Laplace is a continuous time-domain transformation into the spatial (frequency domain) (Analogue Systems)
- In discrete time systems, where we have samples fed into digital controllers, we have the equivalent z-transform

$$X(z) = \sum_{n=0}^{\infty} x[n]z^{-n}$$

- MATLAB has functions for Fourier, Lapalce, and z-transforms and their inverse transformations.

Laplace Transform Table

S.no	$f(t)$	$\mathcal{L}\{f(t)\}$	S.no	$f(t)$	$\mathcal{L}\{f(t)\}$
1	1	$\frac{1}{s}$	11	$e^{at}\sinh bt$	$\frac{b}{(s-a)^2-b^2}$
2	e^{at}	$\frac{1}{s-a}$	12	$e^{at}\cosh bt$	$\frac{s-a}{(s-a)^2-b^2}$
3	t^n	$\frac{n!}{s^{n+1}}$	13	$t\cos at$	$\frac{s^2-a^2}{(s^2+a^2)^2}$
4	$\sin at$	$\frac{a}{s^2+a^2}$	14	$t\sin at$	$\frac{2as}{(s^2+a^2)^2}$
5	$\cos at$	$\frac{s}{s^2+a^2}$	15	$f'(t)$	$sF(s)-f(0)$
6	$\sinh at$	$\frac{a}{s^2-a^2}$	16	$f''(t)$	$s^2F(s)-sf(0)-f'(0)$
7	$\cosh at$	$\frac{s}{s^2-a^2}$	17	$\int_0^t f(u)du$	$\frac{1}{s}F(s)$
8	$e^{at}t^n$	$\frac{n!}{(s-a)^{n+1}}$	18	$t^n f(t)$ Where $n = 1,2,3,..$	$(-1)^n\frac{d^n}{ds^n}\{F(s)\}$
9	$e^{at}\cos bt$	$\frac{s-a}{(s-a)^2+b^2}$	19	$\frac{1}{t}\{f(t)\}$	$\int_s^\infty F(s)ds$
10	$e^{at}\sin bt$	$\frac{b}{(s-a)^2+b^2}$	20	$e^{at}f(t)$	$F(s-a)$

$\delta(t)$ 1

Laplace Transform Table Example

17

*PROBLEM: Find the Laplace transform of $f(t) = te^{-5t}$.

From the Laplace Table, we can find this relationship:

$e^{at}t^n$	$\frac{n!}{(s-a)^{n+1}}$

$n! \rightarrow = 1 \rightarrow$ لأنها ال Power
تبعت t

So, in our case, a = -5 and n = 1, substituting in the Laplace transform, we get:

ANSWER: $F(s) = 1/(s+5)^2$

*PROBLEM: For the following differential equation, find the LaPlace trans

$$\frac{d^2x}{dt^2} + 4\frac{dx}{dt} + 5x = 1$$

If the initial conditions are x(0) = 1, and x'(0) = -1

القيمة عندما تكون ال X=0 ↳
لو ما أعطانا اياها خلص بنعتبرها 0.

ANSWER: From the Laplace Table, we can find this relationship:

$f'(t)$	$sF(s) - \overset{=1}{f(0)}$
$f''(t)$	$s^2F(s) - s\underbrace{f(0)}_{=1} - \underbrace{f'(0)}_{=-1}$

$X' = t^n$	$\frac{n!}{s^{n+1}}$

So, for the 2nd derivative we get:

$$s^2F(s) - s(1) - (-1) = s^2F(s) - s + 1$$

for the 1st derivative we get:

$$4(sF(s) - 1)$$

For the 3rd term, L(5x), here n = 1, so

$$\frac{5}{s^2}$$

And the right-hand term is 1, so L(1) = $\frac{1}{s}$

$$s^2F(s) - s + 1 + 4sF(s) - 4 - \frac{5}{s^2} = \frac{1}{s}$$

Then simplify

↱ "مهمة جداً" → يعني لكل block وللـ system كامل.

The Transfer Function

العلاقة التي تربط الـ input بالـ output
شو ما كانت العلاقة تكون.

18

A control system has an input signal r(t) which is going to provide an output (a control signal) c(t) as you can see in the simple open-loop control system (Figure a).

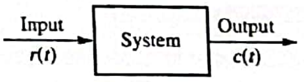
When a system is complex, and it has many electrical or mechanical components, we divide it into subsystems. But still, we have an input signal r(t) which is going to provide an output (a control signal) c(t) as you can see in the simple open-loop control system (Figure b).

For example, in the adjacent RLC circuit, the current *I* (*i.e.*, r(t)) controls the output voltage v(t) (*i.e.*, c(t))

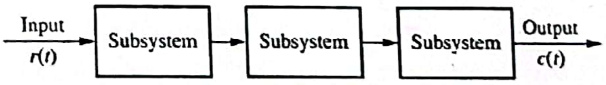
The relationship between the output and the input is governed by the equations of the system.

We are interested to know the relationship of the output to the input; that is: G(t) = $\frac{c(t)}{r(t)}$ which we call the transfer function.

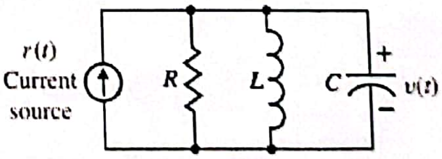
But in the time domain (complex) $\frac{v(t)}{R} + C\frac{dv(t)}{dt} + \frac{1}{L}\int_0^t v(t)\,dt = r(t)$



(a)



(b)

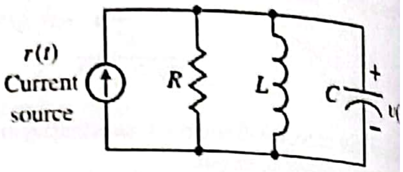


The Transfer Function

This is why we use the Laplace transform and express the Transfer Function as:

$$G(s) = \frac{c(s)}{r(s)}$$

1. Use Algebra instead of integrodifferential equations
2. Spatial domain provides means to analyze the system in ways unavailable to us in the time-domain. Exposes and allows using tools to determine transient response, steady-state, stability.



$$\frac{v(t)}{R} + C\frac{dv(t)}{dt} + \frac{1}{L}\int_0^t v(t)\,dt = r$$

Assuming no initial conditions, and using Laplace Tables, we can rewrite the system equations as:

$$\frac{V(s)}{R} + sCV(s) + \frac{V(s)}{sL} = R(s)$$

$$V(s)\left(\frac{1}{R} + sC + \frac{1}{sL}\right) = R(s)$$

$$\mathrm{G}(s) = \frac{V(s)}{R(s)} = \frac{1}{\left(\frac{1}{R} + sC + \frac{1}{sL}\right)}$$

$$\mathrm{G}(s) = \frac{V(s)}{R(s)} = \frac{sRL}{s^2RLC + sL + R}$$

Transfer Functions are valid when the syste LINEAR; that is, the governing equations a linear.
If not, we need to apply linearization to no linear components first; then retrieve the tra function.

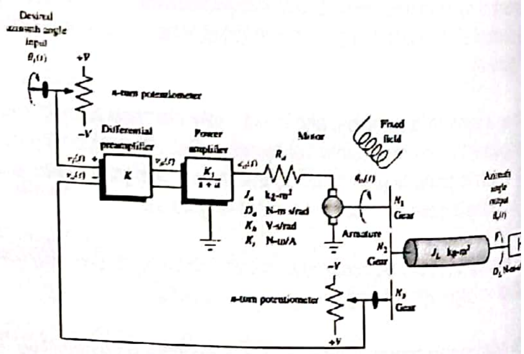
* اغلب ال Systems الي حنشوفها closed-loop System (closed feedback)

The Transfer Function of Complex Systems

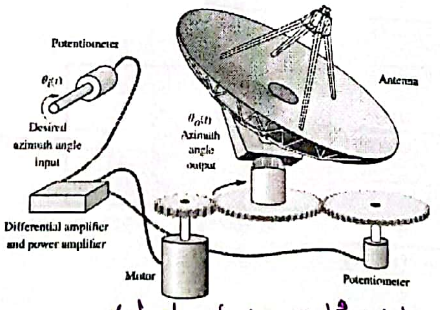
Most control systems are far more complex.

1. Divide them into subsystems
2. Retrieve the transfer function for each subsystems individually
3. Simplify using block diagram reduction techniques or by maths

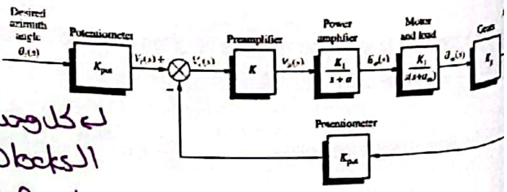
Schematic



Layout


Potentiometer
θi(t)
Desired azimuth angle input
θo(t)
Azimuth angle output
Antenna
Differential amplifier and power amplifier
Motor
Potentiometer

Block Diagram



لـ كل وحدة من هاي ال blocks عبارة عن transfer function (الـ معادلة طُبق عليها laplace)

معادلة مكتوبة بال S domain التي تعبر عن معادلة مكتوبة بال Time Domain (ميكانيكية وكهربائية)، كُتبت بال S domain للتسهيل.

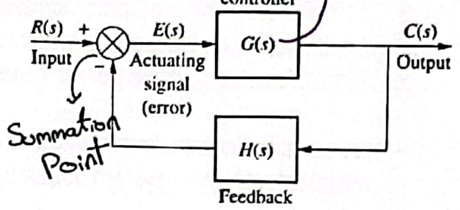
← نسبة ال output على ال input .

* موقعي أوجد العلاقة النهائية بين ال input وال output (معادلة السيستم).

Transfer Function of Systems with Negative Feedback

- The input to this system is the signal x(s)
- Since this is a negative feedback system, the actual controlling signal is the difference (error) between the input signal and the feedback signal (say previous reading) which is C(s) H(s)
- Therefore, the difference (error) signal is R(s) – C(s)H(s)
- The new output signal C(s) is the result of applying the control (system) G(s) to the difference (error) signal G(s)(R(s) – C(s)H(s))

not classical ← لأنه لسا ما شفني Digital Computers ، لو كانت G(Z) بتكون Digital.



So:

$$C(s) = G(s)(R(s) - C(s)H(s))$$

$$C(s) = G(s)R(s) - G(s)C(s)H(s)$$

$$C(s) + G(s)C(s)H(s) = G(s)R(s)$$

$$C(s)(1 + G(s)H(s)) = G(s)R(s)$$

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

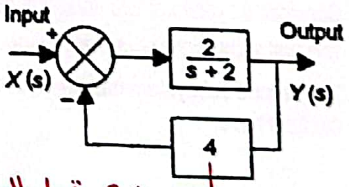
→ صرنا نقدر نفهم كل ال blocks ونحط block وحدة فيها هاي المعادلة .

A positive feedback system will have the same transfer function, except that the sign will be negative in the denominator

* Negative feedback system بشتغل عالفرق بين ال desired وال actual
output كل ما شاف في فرق بضل يتحكم فيه ليوصل لل accuracy الي بدي ياها.

Examples

Example 1) Determine the overall transfer function for the adjacent control system which has a negative feedback loop with a transfer function of a gain equal 4 and a forward path transfer function of 2/(s + 2).



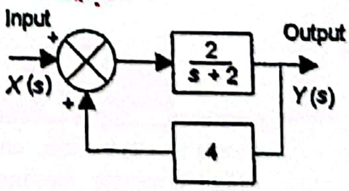
The overall transfer function of the system is:

$$G_{overall}(s) = \frac{\frac{2}{s+2}}{1+4\times\frac{2}{s+2}} = \frac{2}{s+10}$$

تعويض بالمعادلة الي فوق $\left(\frac{G}{1+HG}\right)$

بيكون رقم لحاله معناه gain بزيد من قوة ال Signal وممكن يكون Amplifier بسيط .

Example 2) Determine the overall transfer function for the adjacent control system which has a positive feedback loop with a transfer function of a gain equal 4 and a forward path transfer function of 2/(s + 2).



The overall transfer function of the system is:

$$G_{overall}(s) = \frac{\frac{2}{s+2}}{1-4\times\frac{2}{s+2}} = \frac{2}{s-6}$$

تعويض $\left(\frac{G}{1-HG}\right)$ ← لأنه positive feedback

1/11/2022

* معادلات الـ Time Domain هي اللي بتربط الـ input بالـ output بس جدًا معقدة فبنحولها لـ Frequency Domain

* الـ System مكون من Multiple components سواء كهرباء أو ميكانيك.

* Z-transform ← for Digital / laplace ← Clasical

* حاليًا مافي sys كله Clasical أو كله Digital دايمًا مدارفيه بالـ sys الواحد Digital و Clasical مع بعض (Mechanic + Computer Control)
↳ Digital ↲ clasical

Slide 21 :

$$Y(S)\ (\text{output}) = \underbrace{E(S)}_{\text{error or difference}}\ \underbrace{G(S)}_{\text{System}}$$

$\hookrightarrow X(S) - \underbrace{Y(S)(H(S))}_{\text{feedback}}$

حرف الـ S لنعبر إنه المعادلات مكتوبة بصيغة laplace (S domain) اللي أصلها كانت معادلات الـ Time domain لـ System معين، لقدام حنكتبهم بدون S للسرعة.

$\rightarrow Y(s) = (X(S) - Y(S)(H(S)))\,G(S)$

$* \quad Y = (X - YH)G \rightarrow Y = XG - YHG \rightarrow Y + YHG = XG$

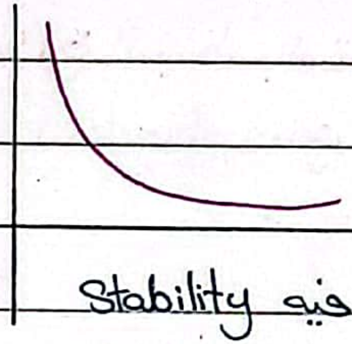
$\rightarrow Y(1 + HG) = XG \rightarrow \dfrac{Y}{X} = \dfrac{G}{1 \oplus HG}$.

* دائمًا بالـ sys الـ negative feedback الـ transfer function هو نسبة الشق العلوي (actual controller (G)) على 1 + الـ G مضروبة بالـ feedback.
↲ عكس إشارة الـ negative feedback

* نفس الـ Sys اللي بـ Slide 21 لو كان Positive feedback بتكون المعادلة : $\dfrac{G}{1 \ominus HG}$

* في فرق بين الـ Matlab والـ Simulink
↲ Command lines وبنعمل فيه شوية Plots.
↲ commands for control موجودين بمكتبة اسمها control system بالـ python.
↲ لما نكتب الـ commands تبعت الـ Python والماتلاب فعليًا منكتب كل sys لحال ولازم نعملها بإيدينا أما بالـ Simulink هو لحاله بعطيك الـ transfer funct. وبحلل.

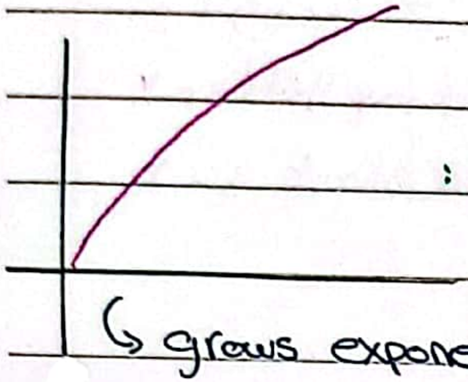
8 Slide 22 لمعادلة الـ negative feedback sys لما نرجعها للـ Time domain باستخدام الـ Laplace transform table حتكون رسمتها هيك :



↓ فيه Stability

وبوصل الـ steady state

بينما الـ positive feedback بكون هيك :



↳ grows exponential

ما بتوقف ولا بتوصل لقيمة

(unstable)

* بالـ S Domain العمليات كلها ضرب مش تعويض ، بالـ Time domain بنعوض (لقدام بناخذ بالتفصيل).

* لتبسيط الـ System يا إما بنحل المعادلة وبنخلي X بجهة و Y بجهة يا إما بنستخدم الـ block Simplification (block Manipulation).

Transfer Function Simplification using Block Manipulation

- Control systems may have many elements and sometimes more than one input. → systems أنواع ال
- A single input-single output system is often termed a SISO system.
- A multiple input-single output system is a MISO system.
- A multiple input-multiple output system is a MIMO system.

- We need to find relationships (transfer functions) between each input and each output. That is; how each input will affect any given system output based on the circuit/system design and controller(computer) decisions.

- We need to simplify the control system block diagram to find these transfer functions

- The following are some of the ways we can reorganise the blocks in a block diagram of a control system in order to produce simplification and still give the same overall transfer function for the system:

1. Merging Block in Series and Parallel Forms (Cascade Form)
2. Moving take-off points (distribution points)
3. Moving a summing point
4. Eliminating Feedback loops

* بال SISO ← ال transfer funct هو علاقة ال output بال Input.

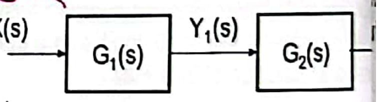
* بال MISO ← ال transfer funct هو علاقة كل Input لحاله بال output.

* بالMIMO ← ال transfer funct هو علاقة كل Input لحاله بكل output لحاله، مثلاً لو عندي 3 inputs و 3 outputs بطلع معاي 9 transfer functions.

← أسهل وحدة.

Block Manipulation: Transfer Function of Systems in Series *1*

← كلما نشوف 2 blocks in series أو اكثر بشيلهم وبخليهم block وحدة وبضربهم ببعض.



Consider a system of two subsystems in series.

The first subsystem $G_1(s)$ has an input of $X(s)$ and an output of $Y_1(s)$; thus, $G_1(s) = \frac{Y_1(s)}{X(s)}$.

The second subsystem has an input of $Y_1(s)$ and an output of $Y_2(s)$ thus, $G_2(s) = Y_2(s)/Y_1(s)$

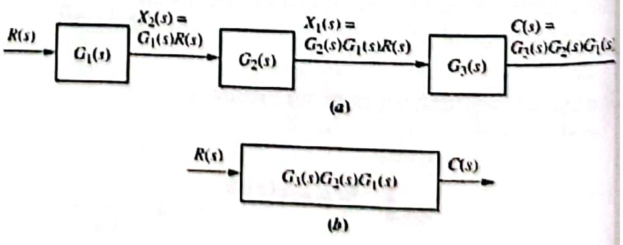
We thus have:

$$Y_2(s) = G_2(s)Y_1(s) = G_2(s)G_1(s)X(s)$$

The overall transfer function $G(s)$ of the system is $Y(s)/X(s)$ and so:

$$G_{overall}(s) = G_1(s)G_2(s)$$

The overall transfer function f system composed of eleme in series is the product of t transfer functions of the individual series elements

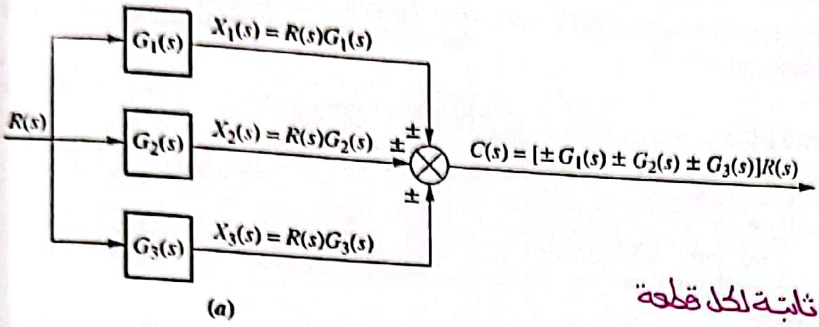


Determine the overall transfer function for a system which consists of two elements in series, one having a transfer function of 1/(s + 1) and the other 1/(s + 2).

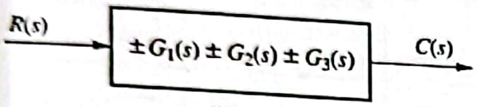
$$G_{overall}(s) = \frac{1}{s+1} \times \frac{1}{s+2} = \frac{1}{(s+1)(s+2)}$$

م نقاط التفرع ، كثير مهمة وكثير مستخدمة.

Block Manipulation: *2* Transfer Function of Systems in Parallel



(*a*)



(*b*)

$$G_e(s) = \pm G_1(s) \pm G_2(s) \pm G_3(s)$$

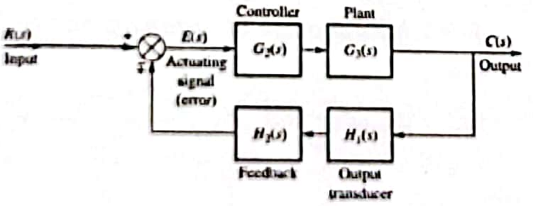
The overall transfer function for a system composed of elements in series is the sum of the transfer functions of the individual series elements.

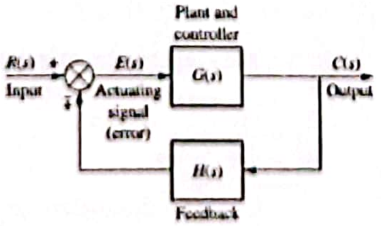
* فكرتي فيها بإنه يجب أن تبقى المعادلة ثابتة لكل قطعة
صغيرة ، فحتى لو جمعتهم كلهم بـ block وحدة
لما ترجعلي تطبقي بده يطلعلك نفس ما كان يطلعلك
قبل.

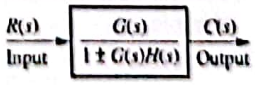
* احنا ما بنغير بال System كهربائية أو ميكانيكيًا احنا رياضيًا بنعبر عنه
بطريقة أسهل.

Block Manipulation: *3* Changing Feedback Loops

We follow the rule for the feedback loop transfer function, but we need to make sure that we do not have any distribution points in the middle of our forward or feedback paths.







* بال Physical
system بخلصنك زي ما هو احنا التبسيط هنا نظريا فقط .

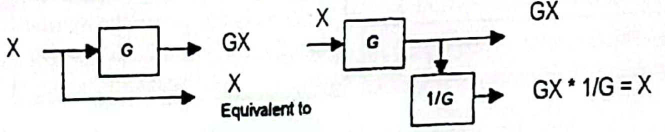
4

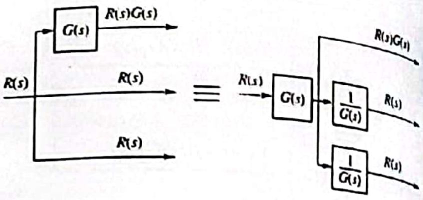
Block Manipulation: Moving Take-off Points

ل فيها نقاط تفرع أيضًا ، مهمة ومستخدمة

As a means of simplifying block diagrams, it is often necessary to move take-off points. The following figures demonstrate the basic rules for such movements:

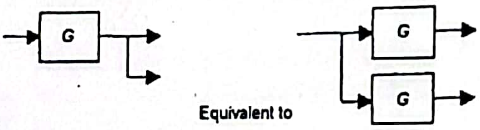
Rule 1: Moving a take-off point to beyond a block

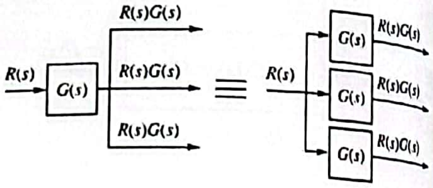




ل ضفنها عشان يطلعلي X زي ما كان يطلع قبل

Rule 2: Moving a take-off point to ahead of a block (distributive)





* بالعادة يا بضرب يا بقسم G s *

* هاي ال Slides حتكون معطاة إلنا
بالامتحان .

5

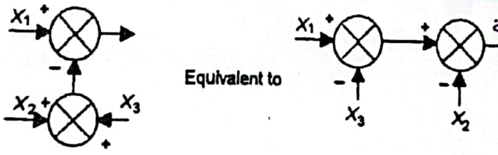
Block Manipulation: Moving a Summing Point

As a means of simplifying block diagrams, it is often necessary to move summation points. The following figures demonstrate the basic rules for such movements:

↰ خاصية التبادل بين الجمع والطرح (بشرط ما في خطوط متفرعة وطالعة من ال blocks) .

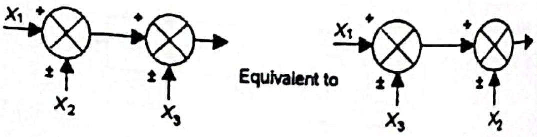
Rule 1: ***Rearrangement of summing points***

$X_1 - (X_2+X_3) = X_1 - X_2 - X_3$ → كأني عملت

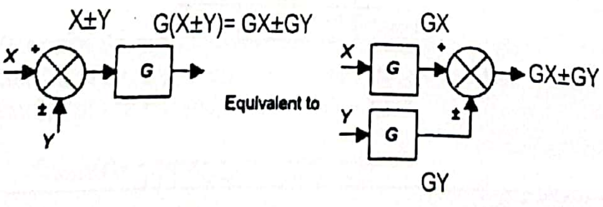


خاصية تفريع .

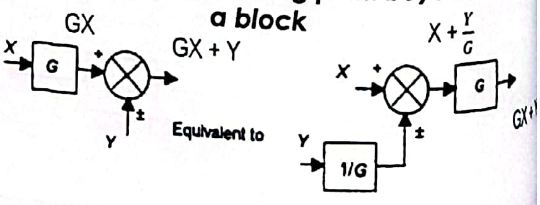
Rule 2: ***Interchange of summing points***



Rule 3: ***Moving a summing point ahead of a block***



Rule 4: ***Moving a summing point beyond a block***



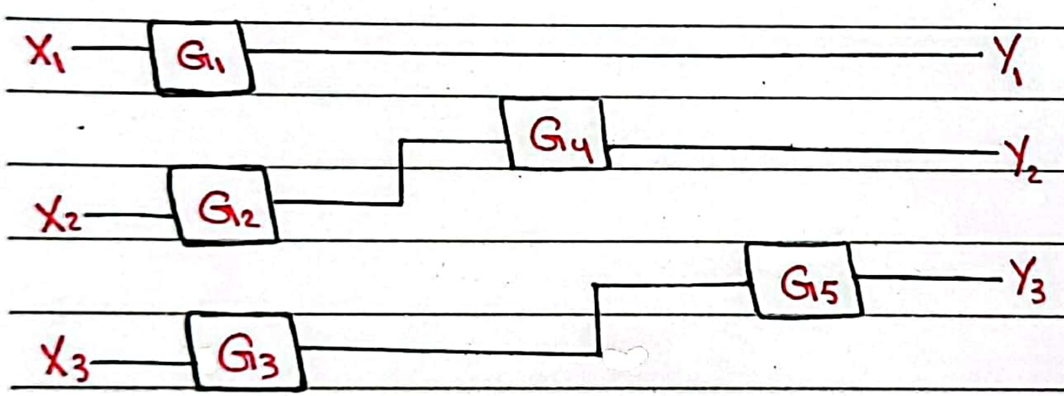
3/11/ 2022

* الـ Control Systems الحديثة المعقدة هي اللي بكون فيها أجهزة ميكانيكية و هيدرولوكية وكهربائية لها خواص فيزيائية معينة فبالتالي معادلات معينة.

* الـ output النهائي ليطلع بده يمر بكل الـ Subsystems ولازم نتأكد إنه هاي النتيجة اللي بدنا ياها.

* لازم أوصل دائمًا لإنه الـ input يكون كله بجهة والـ output كله بجهة أخرى لحتى أقسم y على x ليطلع معي الـ transfer function.

Slide 23 : MIMO → if we have 3 inputs + 3 outputs = 9 transfer functions:



$\frac{Y_1}{X_1}, \frac{Y_2}{X_1}, \frac{Y_3}{X_1}$ /

→ $\frac{Y_1}{X_2}, \frac{Y_2}{X_2}, \frac{Y_3}{X_2}$ /

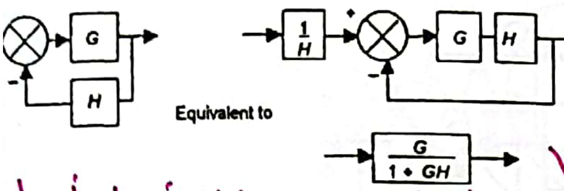
$\frac{Y_1}{X_3}, \frac{Y_2}{X_3}, \frac{Y_3}{X_3}$

* الـ input ممكن تكون Sensors, كبسات Switches وهكذا، الـ output مثلاً أوامر من الـ motor يتحرك أو يولع ضواو معينة أو يحرك ذراع آلية أو يضخ غازات معينة أو يحرك اشي ثقيل وهكذا.

Block Manipulation: *6* Changing feedback and forward paths

لو عندك 2 forward paths وبدك تشيل واحد منهم لازم تضيفه للي فوق وتربطه عشان يطلع نفس الناتج.

Rule 1: Removing a block from a feedback path

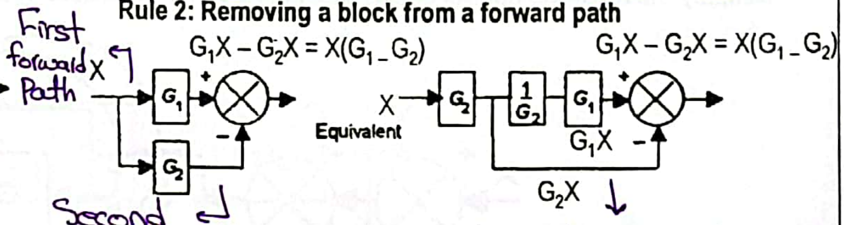


حاولنا نخليهم in series لإنه أسهل وأبسط.

Rule 2: Removing a block from a forward path

$G_1X - G_2X = X(G_1 - G_2)$

$G_1X - G_2X = X(G_1 - G_2)$



ال Path اللي ماعليها أي block سواء كانت forward أو feedback تسمى unity gain Path

Rule 3: Eliminating feedback loop →

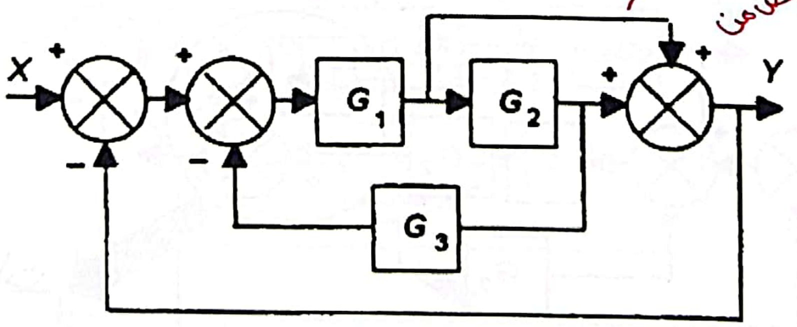
ال block العلوي مقسوم على 1 معكوس الإشارة بال block العلوية بال block السفلية.

like Slide 21+22+26

← فعليا ماعندي system ال Signal مارقة زي ماهي، زي كأنه ال system عبارة عن block فيها رقم 1 .

Example 1 on Control System Block Simplification

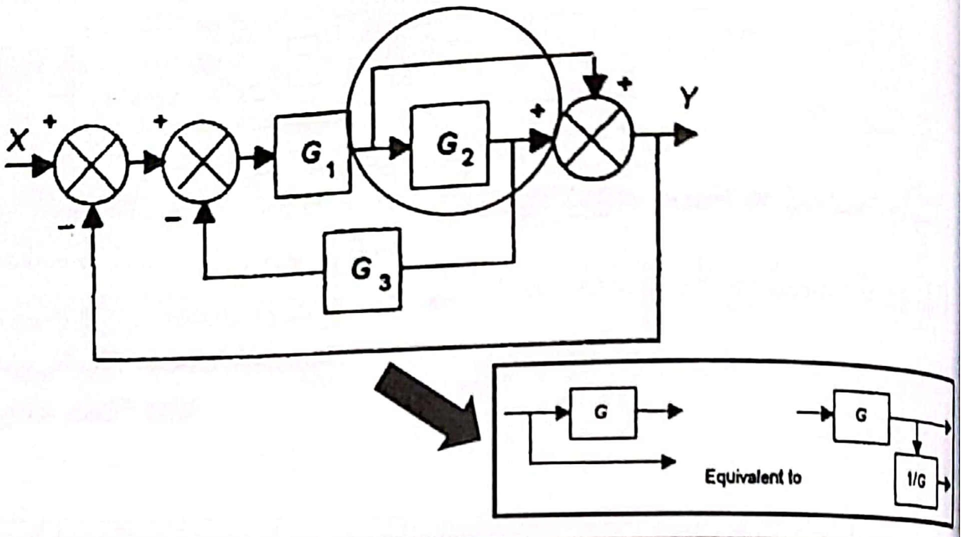
Simplify the following control circuit using the block manipulation rules:



هاي نقطة البداية (بدي اتخلص من نقطة التشعب هاي) بدي أحاول أخلي G1, G2 يكونوا Series

Example 1 on Control System Block Simplification

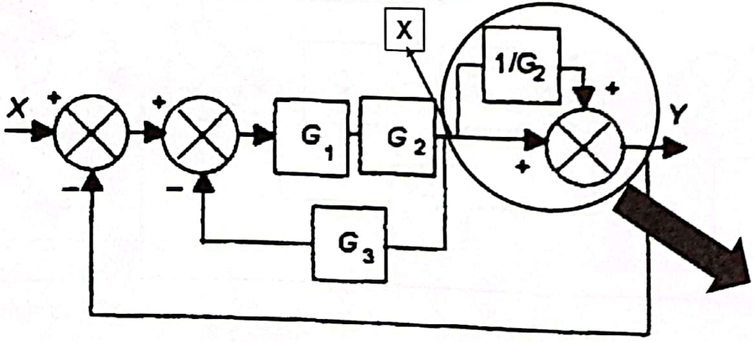
Simplify the following control circuit using the block manipulation rules:



Example 1 on Control System Block Simplification

Simplify the following control circuit using the block manipulation rules:

افترضناه
لنحل المعادلة



حليناها بالرياضيات

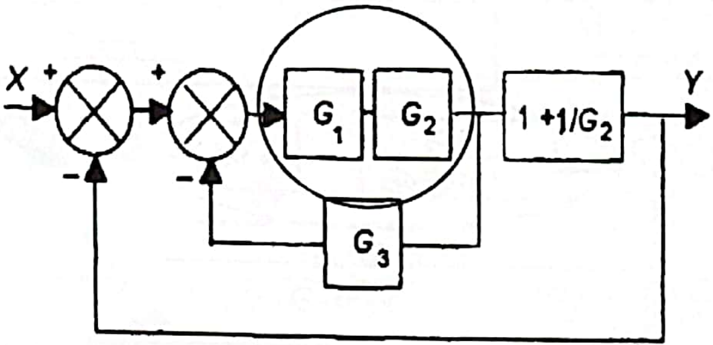
$$Y = X + \frac{X}{G_2}$$

$$Y = X(1 + \frac{1}{G_2})$$

$$\frac{Y}{X} = 1 + \frac{1}{G_2}$$

Example 1 on Control System Block Simplification

Simplify the following control circuit using the block manipulation rules:

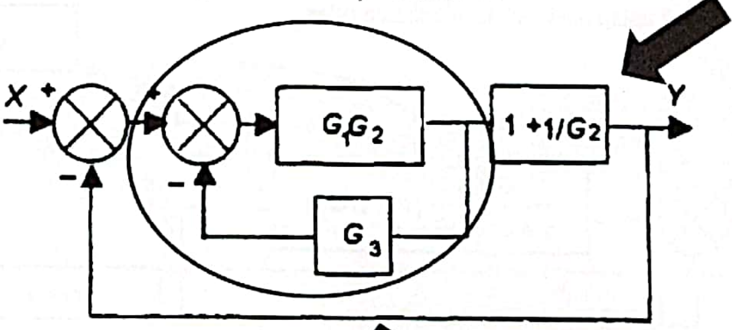


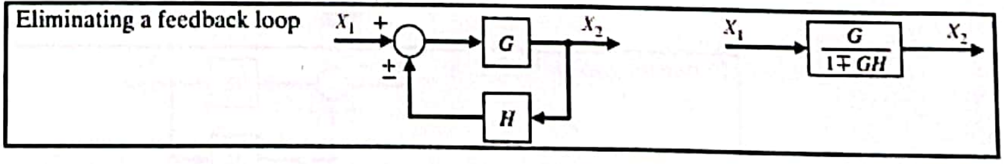
$G_{overall} = G_1G_2$

Example 1 on Control System Block Simplification

Simplify the following control circuit using the block manipulation rules:

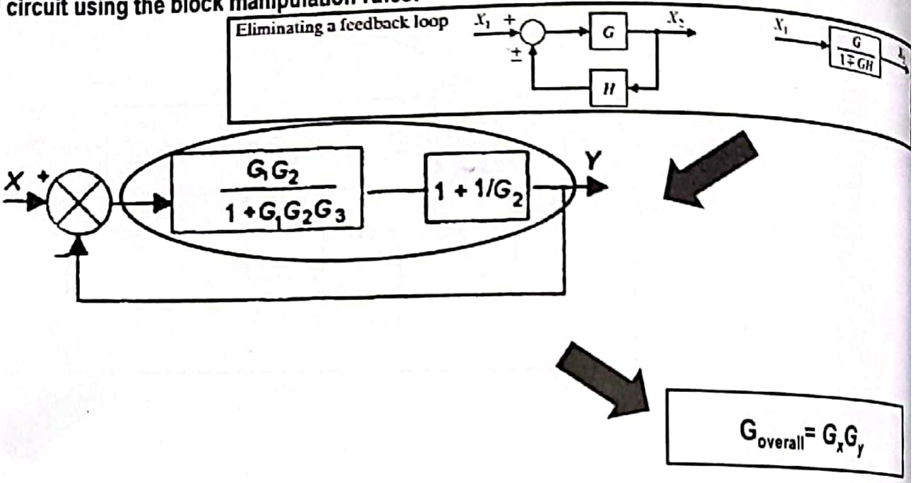
$G_{overall}(s) = G_1G_2$




Eliminating a feedback loop

Example 1 on Control System Block Simplification

Simplify the following control circuit using the block manipulation rules:



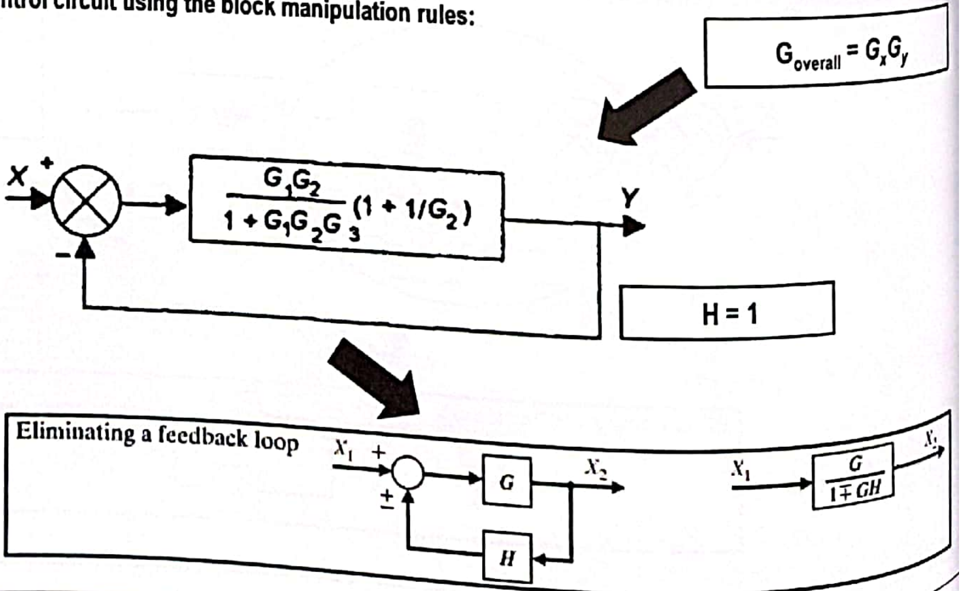
Eliminating a feedback loop

$G_{overall} = G_xG_y$

Example 1 on Control System Block Simplification

Simplify the following control circuit using the block manipulation rules:

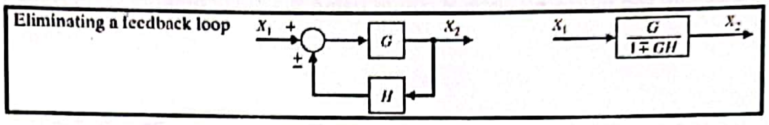
$G_{overall} = G_xG_y$



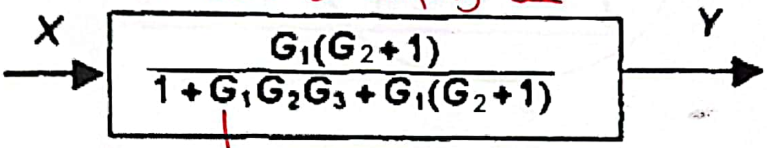
Eliminating a feedback loop

Example 1 on Control Block Simplification

Simplify the following control circuit using the block manipulation rules:



عمل Simplify لوصل لهون:



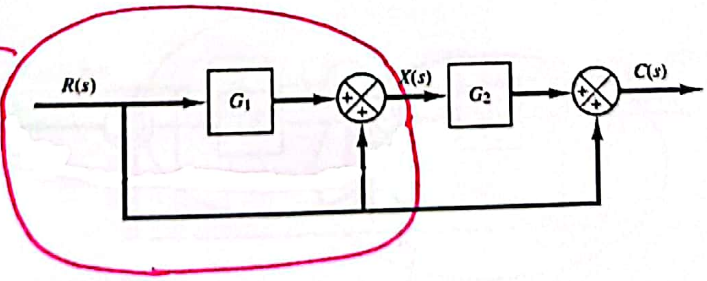
$$\frac{G_1(G_2+1)}{1+G_1G_2G_3+G_1(G_2+1)}$$

G1, G2, G3 عبارة عن functions
كانت من الأساس معادلات بال Time Domain.

Example 2 on Control System Block Simplification

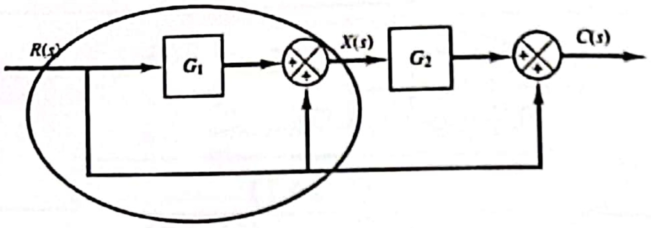
Simplify the following control circuit using the block manipulation rules:

بلش
منهنا
الجزء.



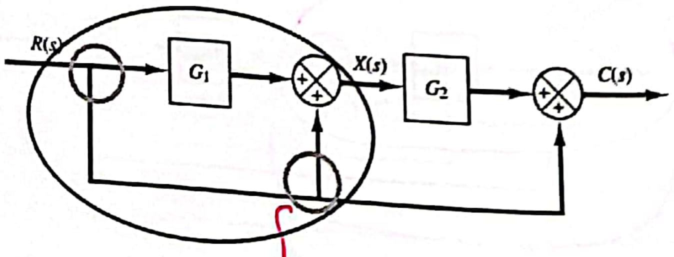
Example 2 on Control Block Simplification

Simplify the following control circuit using the block manipulation rules:



Example 2 on Control Block Simplification

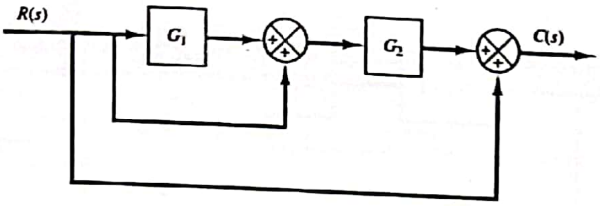
Simplify the following control circuit using the block manipulation rules:



لما إنهم الاتنين زي بعض
رتبتهم بطريقة أفضل.

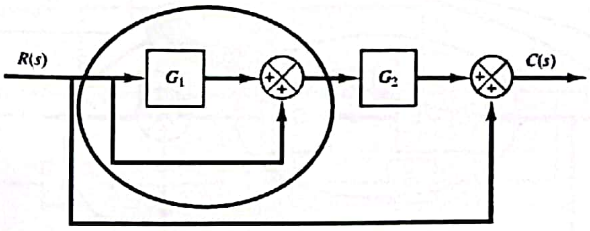
Example 2 on Control Block Simplification

Simplify the following control circuit using the block manipulation rules:



Example 2 on Control Block Simplification

Simplify the following control circuit using the block manipulation rules:



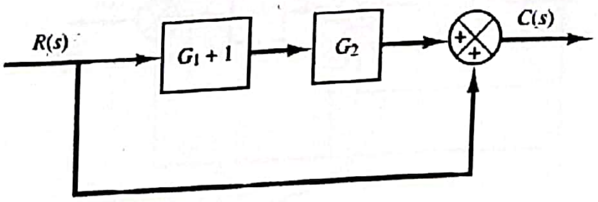
$* \; Y = XG_1 + X$

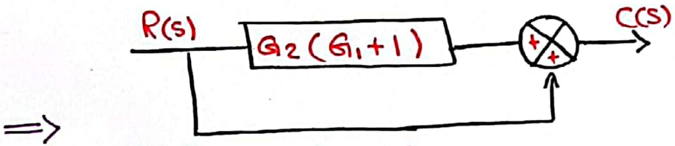
$\rightarrow Y = X(G_1 + 1)$

$\rightarrow \dfrac{Y}{X} = (G_1 + 1)$

Example 2 on Control Block Simplification

Simplify the following control circuit using the block manipulation rules:

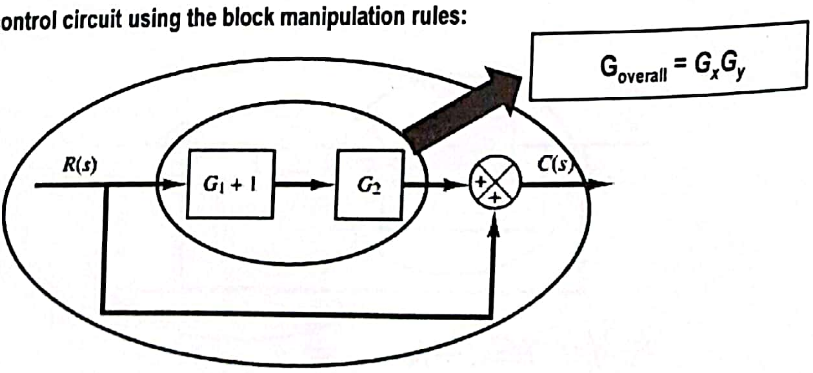


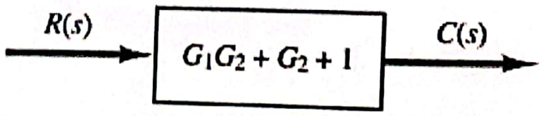


لـ حليها رياضيات أيضًا
(معادلة الfeedback)

Example 2 on Control Block Simplification

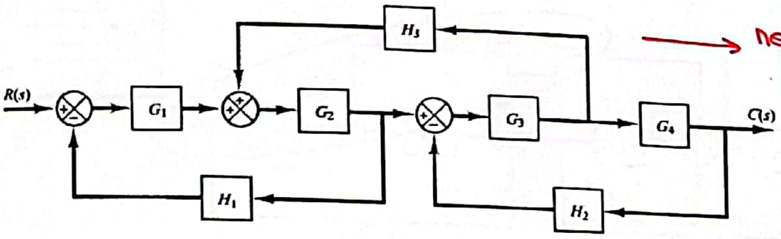
Simplify the following control circuit using the block manipulation rules:





Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).

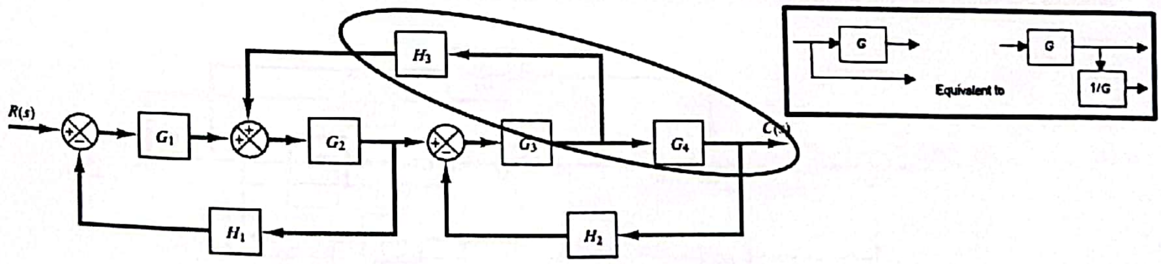


لازم نبلش بالـ negative
feedback الي فوق

* دائماً الحلول الأفضل انك تودي المشاكل للأطراف ماتخليها بالنص.

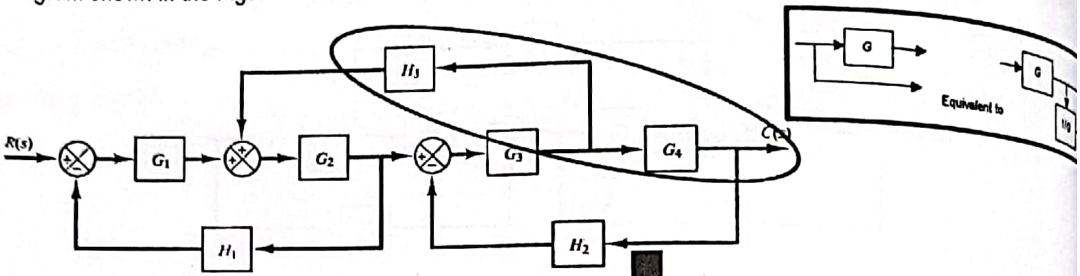
Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).

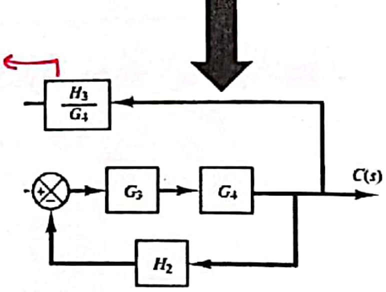


Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).

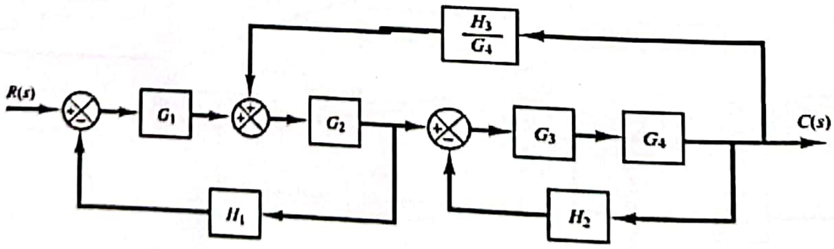


← لإنهم كانوا in series ف direct رسمالك
ال block بعد ما ضربهم ، وهاي خلص
انحلت بنحلها بال Negative feedback.



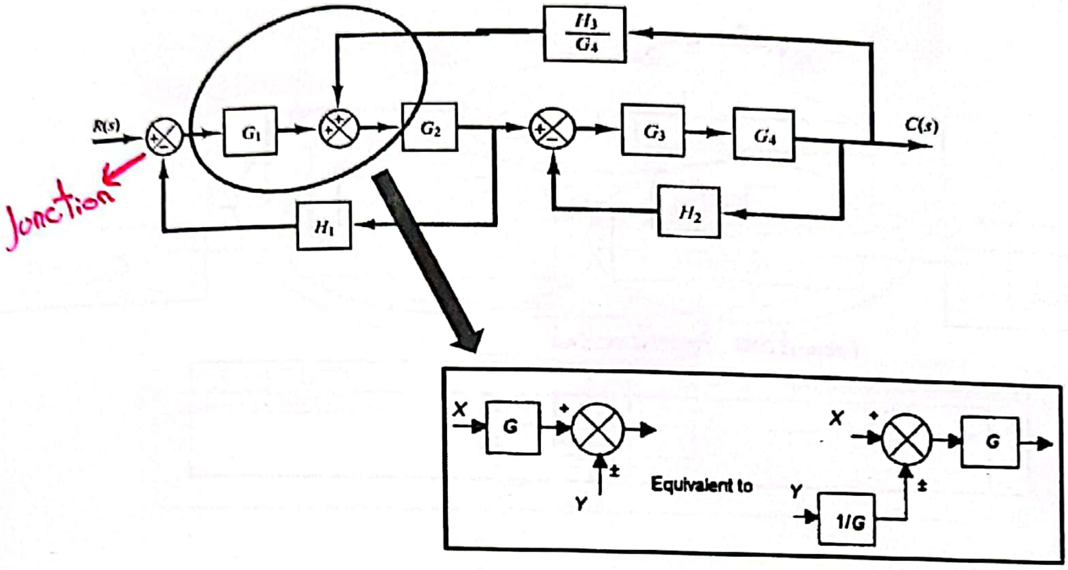
Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).



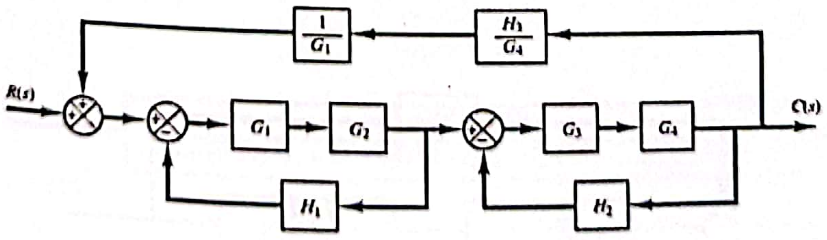
Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).



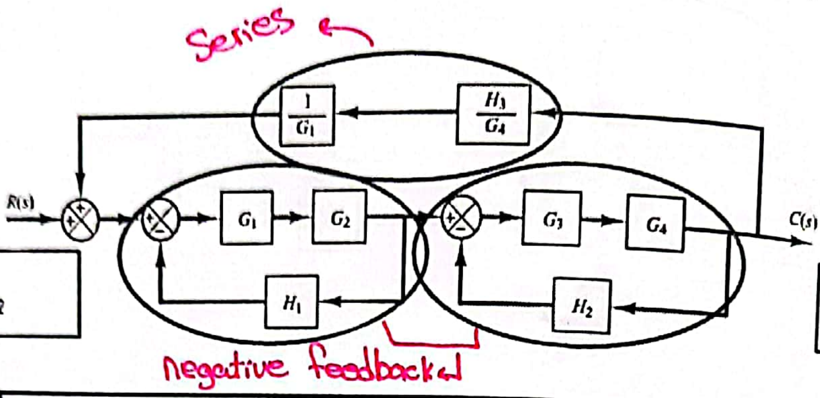
Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).



Example 3 on Control Block Simplification

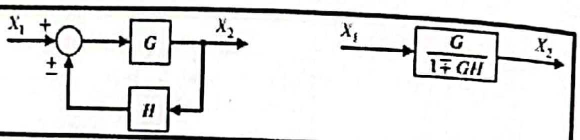
Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).



$G_{overall}$= G_1G_2

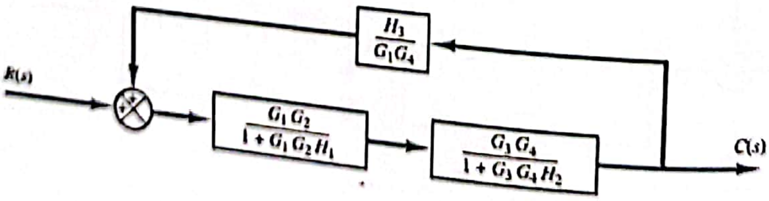
$G_{overall}$= G_3G

Eliminating a feedback loop



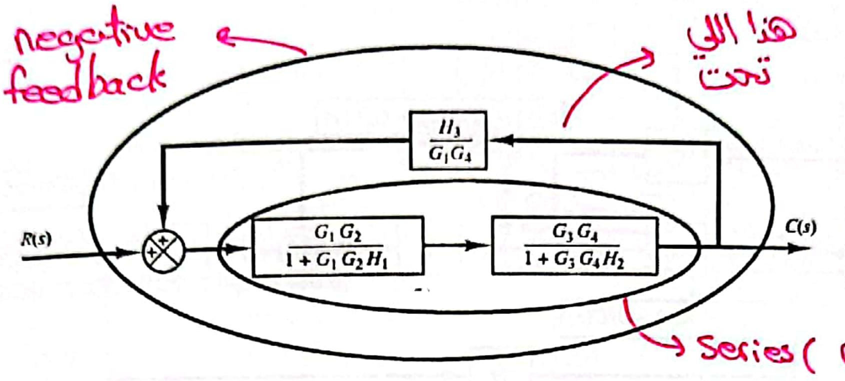
Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).



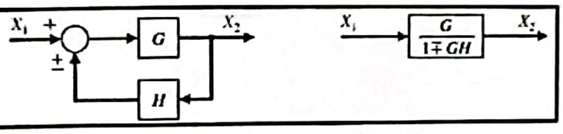
Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).



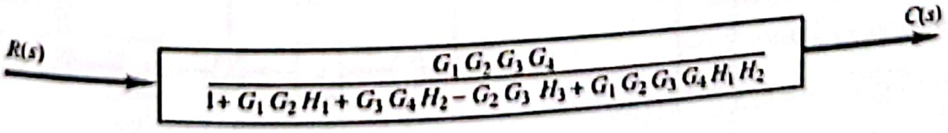
Rule: $G_{overall} = G_x G_y$

Eliminating a feedback loop



Example 3 on Control Block Simplification

Simplify the block diagram shown in the Figure. Then obtain the closed-loop transfer function C(s)/R(s).


R(s)
$\frac{G_1 G_2 G_3 G_4}{1 + G_1 G_2 H_1 + G_3 G_4 H_2 - G_2 G_3 H_3 + G_1 G_2 G_3 G_4 H_1 H_2}$
C(s)

* مرات بنحتاج نجمع بين الحل الرياضي والـsimplificate blocks لو كان الـSystem أو الرسمة كثير Complex .

* معادلة الـloop negative feedback = البسط على 1 معكوس الإشارة البسط بالمقام .

* أي block مش موجودة بعتبرها ك block جواها رقم 1 (unity gain block).

Example 4 on Control Block Simplification MISO

Using Equations

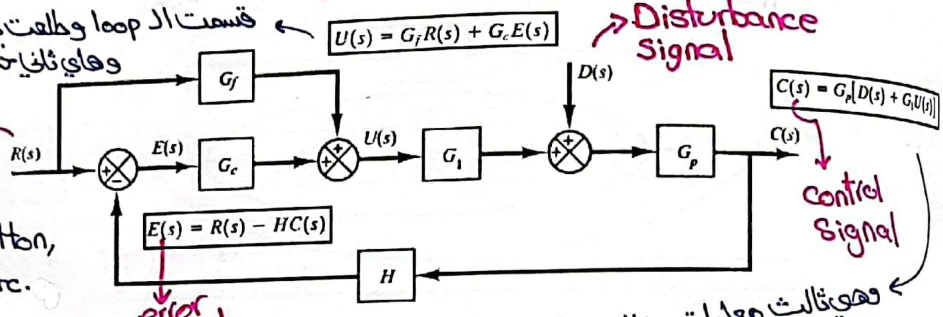
Mathematically obtain transfer functions ① C(s)/R(s) and ② C(s)/D(s) of the system shown in the Figure:

output

inputs

"حنحله باستخدام المعادلات"

لأنو الـ System يعني مش idle قد يكون فيها Noise.



$U(s) = G_f R(s) + G_c E(s)$

قسمت الـ loop وطلعت هاي العلاقة وهاي ثاني خطوة.

Disturbance Signal

$C(s) = G_p[D(s) + G_1 U(s)]$

control Signal

وهي ثالث معادلة وثالث خطوة

reference Signal

like Push-button, Commands, etc.

$E(s) = R(s) - HC(s)$

error Signal (negative feedback)

بالعادة دايما بالـ Systems اللي حناخذها حتكون هاي العلاقة موجودة فأول معادلة بنطلعها هي هاي.

* دايما حاول تطلع أول علاقة بين الـ input والـ feedback
بعدين أي loop بالنص خذ بدايته ونهايته بمعادلة وبعدين أحطهم بالعلاقات.

Example 4 on Control Block Simplification

Using Equations

Mathematically obtain transfer functions C(s)/R(s) and C(s)/D(s) of the system shown in the Figure:

Substitute U_s into C_s:

$$C(s) = G_p D(s) + G_1 G_p[G_f R(s) + G_c E(s)]$$

Substitute E_s into the above equation:

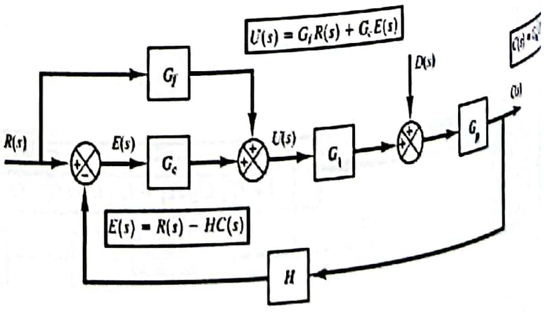
$$C(s) = G_p D(s) + G_1 G_p\{G_f R(s) + G_c[R(s) - HC(s)]\}$$

Solve for C_s:

$$C(s) + G_1 G_p G_c HC(s) = G_p D(s) + G_1 G_p(G_f + G_c)R(s)$$

Rearrange:

$$C(s) = \frac{G_p D(s) + G_1 G_p(G_f + G_c)R(s)}{1 + G_1 G_p G_c H}$$



$U(s) = G_f R(s) + G_c E(s)$

$E(s) = R(s) - HC(s)$

* دايمًا بنحاول نخلي الـ output على جهة والـ input على جهة ثانية.

Building this block or others in → Matlab → commands (language)(بحتية)
→ Simulink → GUI (مستخدم بالصناعة)

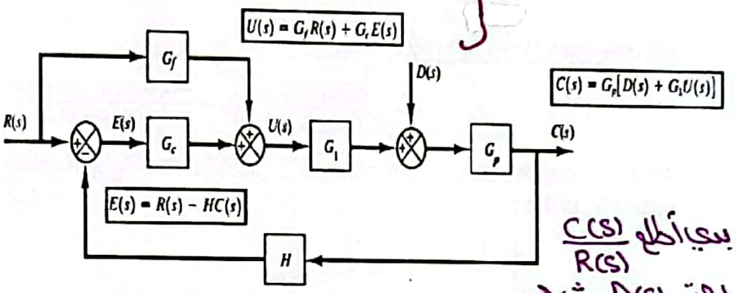
Example 4 on Control Block Simplification

لو بدي أحلها باستخدام ال blocks مرة بصفر D(s) ومرة بصفر R(s)

Using Equations

Mathematically obtain the transfer functions C(s)/R(s) and C(s)/D(s) of the system shown in the Figure:

$$C(s) = \frac{G_p D(s) + G_1 G_p (G_f + G_c) R(s)}{1 + G_1 G_p G_c H}$$



Notice that this is a multiple input system, one for the actual Input R(s), and another for some disturbance D(s)

To obtain the transfer functions C(s)/R(s) and C(s)/D(s), we must only let one input present, and all other inputs are 0, therefore:

C(s)/R(s) = $\frac{C(s)}{R(s)} = \frac{G_1 G_p (G_f + G_c)}{1 + G_1 G_p G_c H}$

C(s)/D(s) = $\frac{C(s)}{D(s)} = \frac{G_p}{1 + G_1 G_p G_c H}$

بدي أطلع $\frac{C(S)}{R(S)}$ بعتبر D(S) مش موجود (بصفره)

بدي أطلع $\frac{C(S)}{D(S)}$ بعتبر R(S) مش موجود (بصفره)

* عشان أطلع $\frac{C(S)}{R(S)}$ أو $\frac{C(S)}{D(S)}$ اللي ما بدي أحسبه من ال input بعتبره مش موجود وبكمل حلي.

* بالعادة الأفضل تحل ال Multiple inputs باستخدام ال equations، وال Single input باستخدام ال blocks.

Example 5 on Control Block Simplification MIMO

Using Equations → الطريقة الأولى :

Figure 2-24 shows a system with two inputs and two outputs. Derive ① C₁(s)/R₁(s), ② C₁(s)/R₂(s), ③ C₂(s)/R₁(s), and ④ C₂(s)/R₂(s)

2 inputs
2 outputs

Solution:

$$C_1 = G_1(R_1 - G_3 C_2)$$ → أخذ نقطة معينة وحلوة بقدر أعوض C2 فيها.

$$C_2 = G_4(R_2 - G_2 C_1)$$ → أخذ نقطة معينة وحلوة بقدر أعوض C1 فيها.

→ كسرت العلاقة اللي بتربط ال output الأول بالثاني

Substitute C₂ into C₁

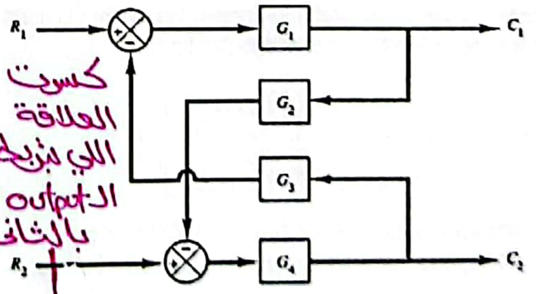
$$C_1 = G_1[R_1 - G_3 G_4 (R_2 - G_2 C_1)]$$

Substitute C₁ into C₂

$$C_2 = G_4[R_2 - G_2 G_1 (R_1 - G_3 C_2)]$$

Solve the equation for C₁

$$C_1 = \frac{G_1 R_1 - G_1 G_3 G_4 R_2}{1 - G_1 G_2 G_3 G_4}$$



→ صارت كل معادلة عبارة عن output واحد و عدد من ال inputs وال blocks.

Solve the equation for C₂

$$C_2 = \frac{-G_1 G_2 G_4 R_1 + G_4 R_2}{1 - G_1 G_2 G_3 G_4}$$

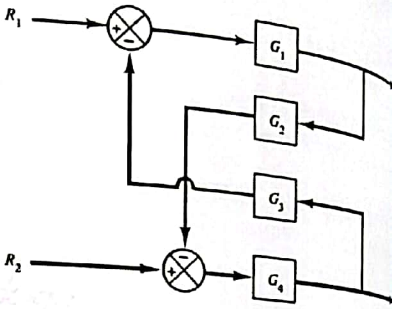
Example 5 on Control Block Simplification

Using Equations

Figure 2–24 shows a system with two inputs and two outputs. Derive $C_1(s)/R_1(s)$, $C_1(s)/R_2(s)$, $C_2(s)/R_1(s)$, and $C_2(s)/R_2(s)$

We now have these two relationships:

$$C_1 = \frac{G_1R_1 - G_1G_3G_4R_2}{1 - G_1G_2G_3G_4} \qquad C_2 = \frac{-G_1G_2G_4R_1 + G_4R_2}{1 - G_1G_2G_3G_4}$$



We can derive the transfer functions of $C_1(s)/R_1(s)$ and $C_2(s)/R_1(s)$ by setting R_2 to 0, so

$$\frac{C_1(s)}{R_1(s)} = \frac{G_1}{1 - G_1G_2G_3G_4},$$

$$\frac{C_2(s)}{R_1(s)} = -\frac{G_1G_2G_4}{1 - G_1G_2G_3G_4},$$

Similarly, we can derive the transfer functions of $C_1(s)/R_2(s)$ and $C_2(s)/R_2(s)$ by setting R_1 to 0, so

$$\frac{C_1(s)}{R_2(s)} = -\frac{G_1G_3G_4}{1 - G_1G_2G_3G_4}$$

$$\frac{C_2(s)}{R_2(s)} = \frac{G_4}{1 - G_1G_2G_3G_4}$$

بالكتب بحبوا يكتبوهم ك (Matrix form).

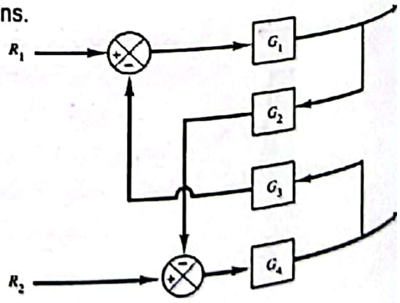
Example 5 on Control Block Simplification

Using Equations

لـ مش مطالبين فيها.

Figure 2–24 shows a system with two inputs and two outputs. Derive $C_1(s)/R_1(s)$, $C_1(s)/R_2(s)$, $C_2(s)/R_1(s)$, and $C_2(s)/R_2(s)$

In control systems, we prefer to present the system in matrix/vector notations. The equations can be combined in this form:



$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} \dfrac{G_1}{1 - G_1G_2G_3G_4} & -\dfrac{G_1G_3G_4}{1 - G_1G_2G_3G_4} \\ -\dfrac{G_1G_2G_4}{1 - G_1G_2G_3G_4} & \dfrac{G_4}{1 - G_1G_2G_3G_4} \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$$

Then the transfer functions $C1(s)/R1(s)$, $C1(s)/R2(s)$, $C2(s)/R1(s)$ and $C2(s)/R2(s)$ can be obtained as follows:

$$\frac{C_1(s)}{R_1(s)} = \frac{G_1}{1 - G_1G_2G_3G_4}, \qquad \frac{C_1(s)}{R_2(s)} = -\frac{G_1G_3G_4}{1 - G_1G_2G_3G_4}$$

$$\frac{C_2(s)}{R_1(s)} = -\frac{G_1G_2G_4}{1 - G_1G_2G_3G_4}, \qquad \frac{C_2(s)}{R_2(s)} = \frac{G_4}{1 - G_1G_2G_3G_4}$$

Example 5 on Control Block Simplification

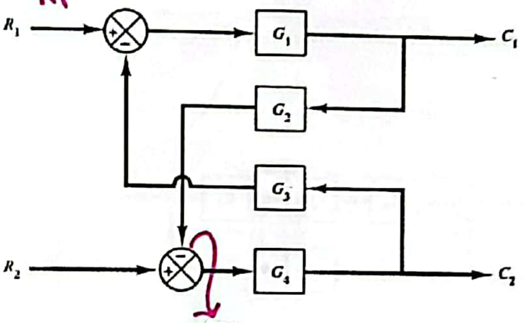
Using Equations – Alternative Solution

→ طريقة ثانية للحل :

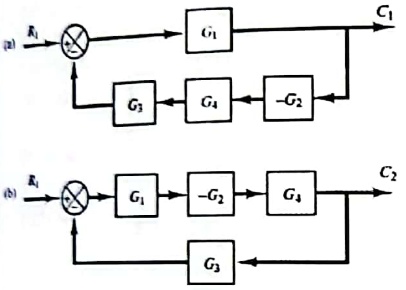
When setting $R_2 = 0$, the block diagram can actually be simplified to:

R_1 مرة صفرنا R_2 وطلعنا معادلة → $\frac{C_1}{R_1}$, $\frac{C_2}{R_1}$

R_2 ومرة صفرنا R_1 وطلعنا معادلة → $\frac{C_1}{R_2}$, $\frac{C_2}{R_2}$



إشارة G_2 بقيت موجودة بعد موت ال jonction.



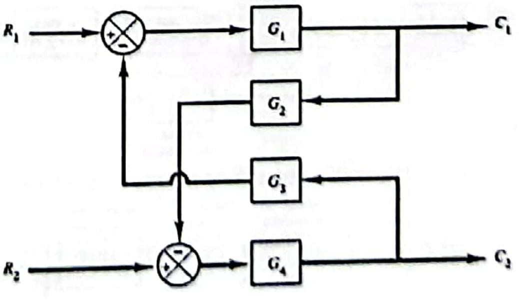
$$\frac{C_1}{R_1} = \frac{G_1}{1 - G_1 G_2 G_3 G_4}$$

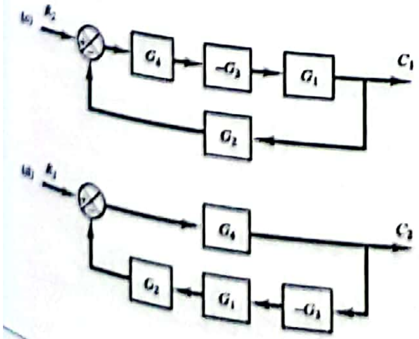
$$\frac{C_2}{R_1} = \frac{-G_1 G_2 G_4}{1 - G_1 G_2 G_3 G_4}$$

Example 5 on Control Block Simplification

Using Equations – Alternative Solution

When setting $R_1 = 0$, the block diagram actually can be simplified to:





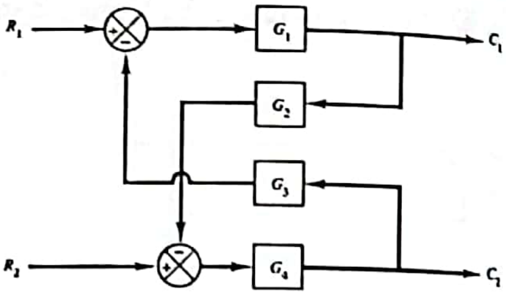
$$\frac{C_1}{R_2} = \frac{-G_1 G_3 G_4}{1 - G_1 G_2 G_3 G_4}$$

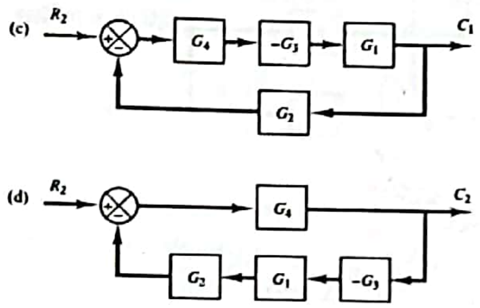
$$\frac{C_2}{R_2} = \frac{G_4}{1 - G_1 G_2 G_3 G_4}$$

Example 5 on Control Block Simplification

Using Equations – Alternative Solution

When setting $R_1 = 0$, the block diagram actually can be simplified to:





$$\frac{C_1}{R_2} = \frac{-G_1G_3G_4}{1 - G_1G_2G_3G_4}$$

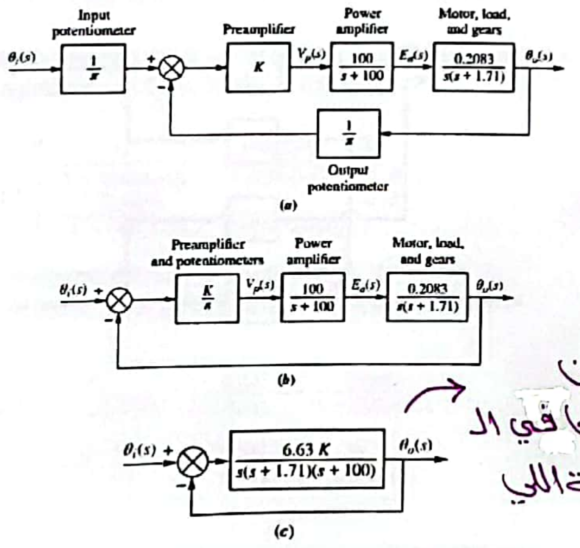
$$\frac{C_2}{R_2} = \frac{G_4}{1 - G_1G_2G_3G_4}$$

Potentiometer مربوط مع Amplifier بتحكم بـ Motor اللي بحرك الـ gears
لتبعت الـ Antenna ومن الـ Antenna برجع بني feedback تاني مشان
يشوف أنا وصلت صح ولا ما وصلت صح.

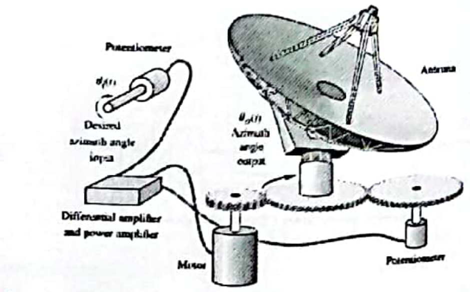
كل Component هي subsystem ولها معادلات كهربائية وميكانيكية
وهكذا ، وحولنا كل وحدة فيها للـ S-Domain.

Example 6 on Control Block Simplification

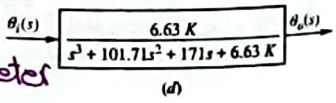
Antenna Azimuth Control System



Layout



وصلنا لعلاقة بين
الزاوية اللي بحركها في الـ
Potentiometer والزاوية اللي
بتحركها الستلايت



لـ الهدف بس نخلص إني أكتب كل واحد بالـ S-Domain مشان الصيغة النهائية
اللي حوصلها حتعبر عن الـ System كامل بالـ S-Domain.

Partial Fractions Form

بعطيك النتيجة النهائية وانت فكلي ياها ورجعها لأصولها.

Now that we have found transfer functions that relate the control system outputs to its inputs, this transfer function can be analyzed to find the control system parameters that we have discussed in Chapter 1: settling time, overshoot, stability, etc.

But first, we need to mathematically transform the fin $\frac{1}{s}-\frac{2}{(s+4)}+\frac{1}{(s+8)}$ into the partial fractions form, from which it is even e: inverse Laplace transform to get back to the time-doı mathematically easy to do so, we shall do it in MATLAB in self-learning material.

For example: thv function Y(s) below can be rewritten as:

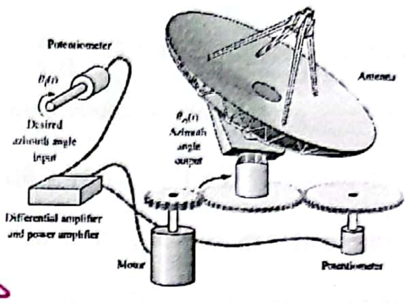
$$Y(s)=\frac{32}{s(s+4)(s+8)}=\frac{1}{s}-\frac{2}{(s+4)}+\frac{1}{(s+8)}$$

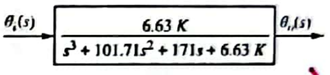
معادلة تكعيبية

هيك أفضل عشان مع العودة إلى جداول laplace حنلاقي صيغة لجدول المعادلات مش المعادلة النهائية (التكعيبية)

Partial Fractions Form is عكس توحيد المقام والعودة إلى الكسور الأصلية

Layout





(d)

حتكون شغلنا بال ch الجاي عشان نحلل ال system.

Math Review: Partial Fractions Example 1 (Optional)

Determine the partial fractions of: $\frac{s+4}{(s+1)(s+2)}$

The partial fractions are of the form: $\frac{A}{s+1}+\frac{B}{s+2}$

Then, for the partial fraction expression to equal the original fraction, we must have: $\frac{s+4}{(s+1)(s+2)}=\frac{A(s+2)+B(s+1)}{(s+1)(s+2)}$

and consequently: $s+4=A(s+2)+B(s+1)$

Pick values of s that will enable some of the terms involving constants to become zero and so enable other constants to be determined:

Let s = -2 then $(-2)+4=A(-2+2)+B(-2+1)$
so B = -2

Let s = -1 then $(-1)+4=A(-1+2)+B(-1+1)$
so A = 3

Therefore $\frac{s+4}{(s+1)(s+2)}=\frac{3}{s+1}-\frac{2}{s+2}$

Math Review: Partial Fractions Example 2 (Optional)

Determine the partial fractions of: $\frac{3s+1}{(s+2)^3}$

The partial fractions are of the form: $\frac{A}{(s+2)}+\frac{B}{(s+2)^2}+\frac{C}{(s+2)^3}$

Then, for the partial fraction expression to equal the original fraction, we must have: $\frac{3s+1}{(s+2)^3}=\frac{A}{(s+2)}+\frac{B}{(s+2)^2}+\frac{C}{(s+2)^3}$

and consequently: $3s+1=A(s+2)^2+B(s+2)+C \quad =A(s^2+2s+1)+B(s+2)+C$

We start equating the same power terms on each side to determine A, B, and C

Equating s^2 terms gives 0 = A. Equating s terms gives 3 = 2A + B and so B = 3. Equating the numeric terms gives 1 = A + 2B + C and so C =-5. Thus:

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

Math Review: Partial Fractions Example 3 (Optional)

Determine the partial fractions of: $\frac{2s+1}{(s^2+s+1)(s+2)}$

The partial fractions are of the form: $\frac{As+B}{s^2+s+1}+\frac{C}{s+2}$

Then, for the partial fraction expression to equal the original fraction, we must have: $\frac{2s+1}{(s^2+s+1)(s+2)}=\frac{As+B}{s^2+s+1}+\frac{C}{s+2}$

and consequently: $2s+1=(As+B)(s+2)+C(s^2+s+1)$

We start equating the same power terms on each side to determine A, B, and C

With S = -2 then -3 = 3C and so C = -1. Equating s^2 terms gives 0 = A +C and so A= 1. Equating s terms gives 2 = 2A + B+ C and so B = 1. Thus

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

Math Review: Partial Fractions Example 4 (Optional)

Determine the partial fractions of: $\frac{2s^2+2}{(s+4)(s-2)}$

Notice the power of numerator and denominator is the same

We must first use division:

$$\begin{array}{r} 2 \\ s^2+2s-8\,\overline{)\,2s^2+2} \\ \underline{2s^2+4s-16} \\ -4s+18 \end{array}$$

And the expression becomes: $2+\frac{-4s+18}{(s+4)(s-2)}$

Then, for the partial fraction expression to equal the original fraction, we must have: $\frac{-4s+18}{(s+4)(s-2)}=\frac{A}{s+4}+\frac{B}{s-2}$

and consequently: $-4s+18=A(s-2)+B(s+4)$

We start equating the same power terms on each side to determine A, B, and C

With s = 2, then B = 5/3. With s =-4, then A = -17/3. Thus, the expression can be written as:

$$2-\frac{17}{3(s+4)}+\frac{5}{3(s-2)}$$

References and Textbook Material

The material in these slides are based on:

Control Systems Engineering, Norman S. Nise, 7th Edition (2014), John Wiley And Sons

- **Chapter 2 – System Modelling in the Frequency Domain**

  Sections *2.1, 2.2, 2.3, 2.4, 2.5, 2.6* (Students kindly note that these sections involve lots of math, and we only described the ideas as we will use MATLAB instead)
- **Chapter 5 – Reduction of Multiple Subsystems**

  Sections *5.1, 5.2*

Instrumentation and Control Systems, 1st Edition , 2004, Elsevier (Newer versions available 3rd edition 2021, but I used the first one)

- **Chapter 9 – Transfer Function**

  Sections *9.1, 9.2, 9.3, 9.4, 9.5*

6/11/2022

(Self-learning Material 3+4)

Page5 ← Partial Fractions ← Partfrac(F(S)) بفككلك المعادلة
وبعطيك يلها بالشكل البسيط.

* بأول Part بعلمك كيف تعرف الـ Functions باستخدام الـ Syms وكيف بعمل
تفاضل وتكامل وبطلع النتائج بصيغة Symbolic، وكيف ممكن تعمل Multiple
Variables بـ Sub Index مختلف، وهكذا.

* Part 2

↲ laplace ← بتكون معرف المعادلة بالـ Time Domain وبعطيك ياها
بالـ S-Domain

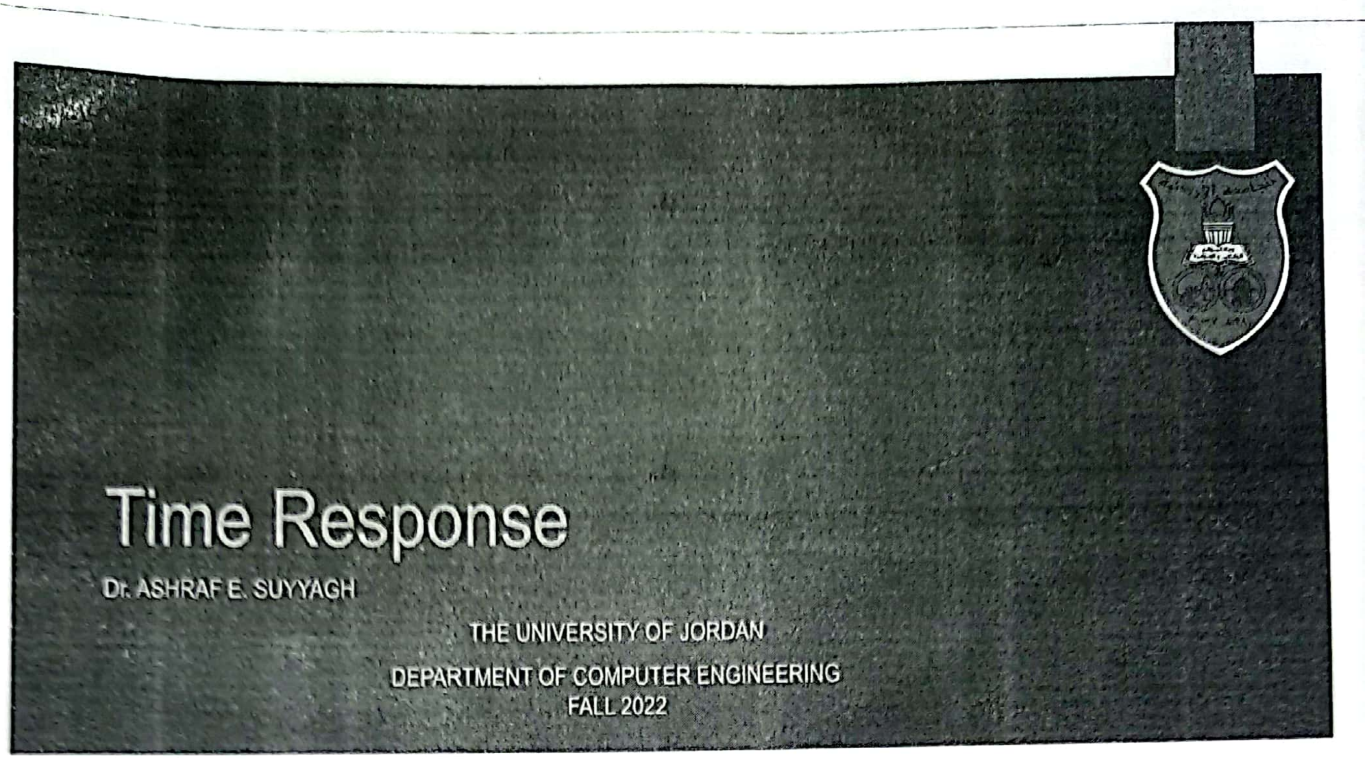
↲ مهمة جدًا ة إنا أعطيت الـ Function لـ laplace ورجعلك الجواب نفس
الـ Input أعرف إنه ما إله laplace أو عامل غلط انت معين.

↲ ilaplace ← تعطيه المعادلة بالـ S-Domain وهو يرجعلك ياها بالـ
Time Domain

* آخر Part مراجعة لـ Polynomials.

* كمان بعلمك إنا مابدك تشتغل Symbolic بدك Numeric كيف تعمل الي عملناه (التفاعل
مع البسط لحال ومع المقام لحال)، في Command بتحول بين الـ Symbolic والـ
Numeric، numden Command بفصل البسط عن المقام وبحفظ كل واحد بمتغير.

* residue ← بعمل Partial fraction لـ Numeric s مش Symbolic.

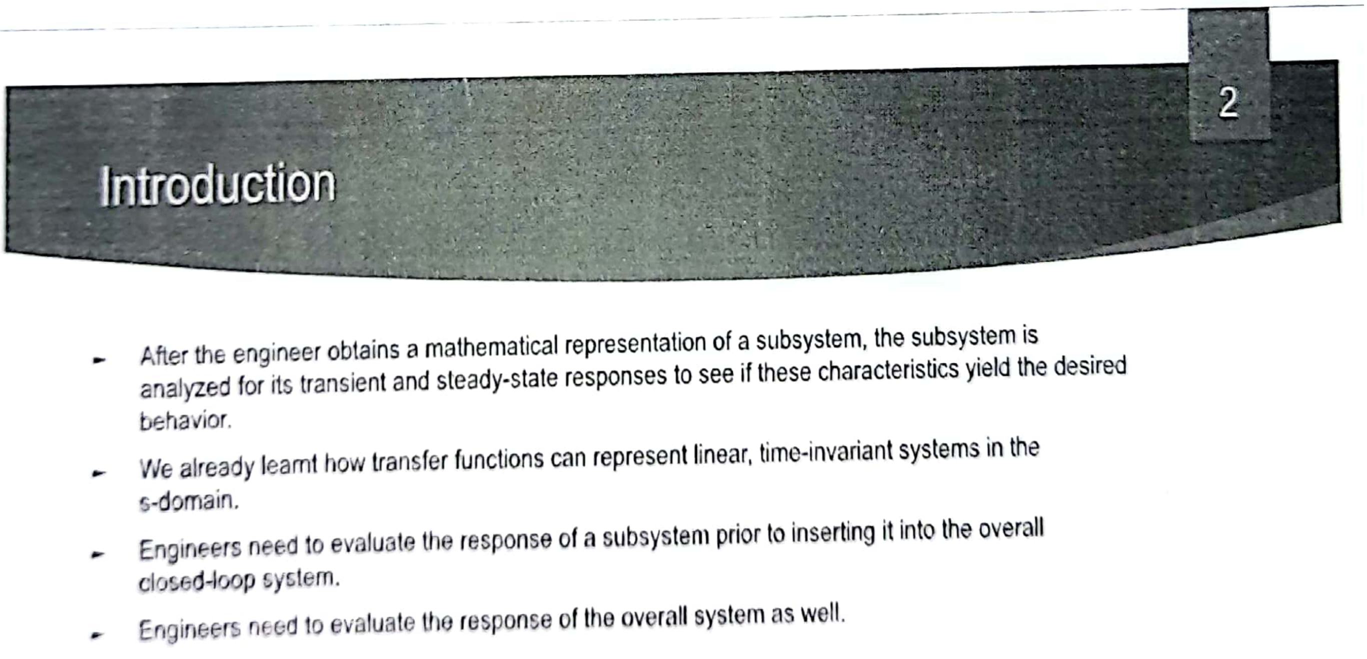


Time Response

Dr. ASHRAF E. SUYYAGH

THE UNIVERSITY OF JORDAN

DEPARTMENT OF COMPUTER ENGINEERING
FALL 2022



Introduction

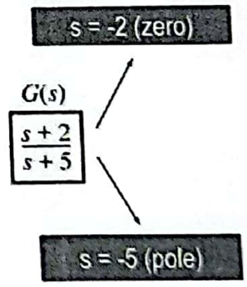
- After the engineer obtains a mathematical representation of a subsystem, the subsystem is analyzed for its transient and steady-state responses to see if these characteristics yield the desired behavior.
- We already learnt how transfer functions can represent linear, time-invariant systems in the s-domain.
- Engineers need to evaluate the response of a subsystem prior to inserting it into the overall closed-loop system.
- Engineers need to evaluate the response of the overall system as well.

Poles, Zeros, and System Response

- The output response of a system is the sum of two responses:
  - Forced response (also called stead-state response, or particular solution)
  - Natural response (also called homogenous solution)
- Solving in the time-domain or inverse Laplace transform is time-consuming and laborious.
- We use the transfer function, and the new concepts of poles and zero to reach qualitative and quantitative solutions to learn quickly all about our system

Poles and Zeros of a Transfer Function

- The **poles** of a transfer function are
  1. The values of the Laplace transform variable, **s**, that cause the transfer function to become infinite or
  2. Any roots of the denominator of the transfer function that are common to roots of the numerator.
- The zeros of a transfer function are
  1. The values of the Laplace transform variable, **s**, that cause the transfer function to become zero, or
  2. Any roots of the numerator of the transfer function that are common to roots of the denominator

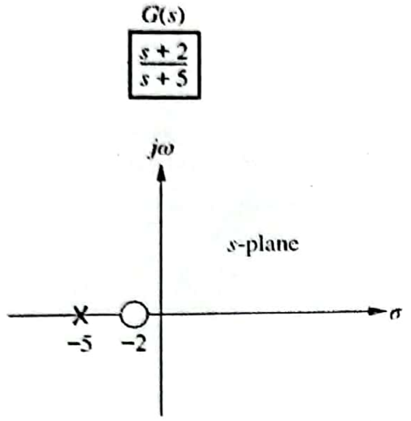


$$H(s) = \frac{(s+2)(s+4)(s-1)}{(s-1)(s-2)(s+5)(s+1)}$$

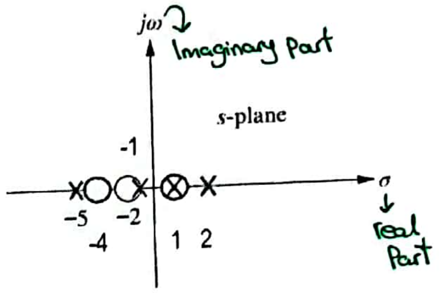
Zeros of H(s) → -4, -2, 3

Poles of H(s) → -5, -1, 1, 2

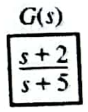
Poles and Zeros - S-Plane



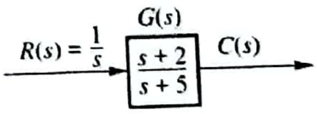
$$H(s) = \frac{(s+2)(s+4)(s-1)}{(s-1)(s-2)(s+5)(s+1)}$$



Poles and Zeros of a First-Order System I



- The system G(s) is a first –order system, given that the denominator has a maximum power for s as 1

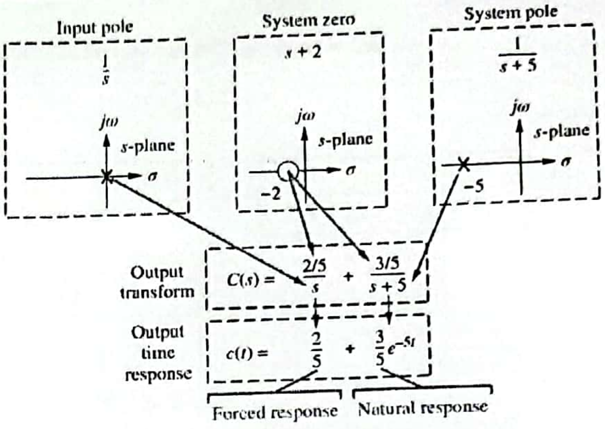


- Suppose that the input of this system is a simple input switch, which in the time domain is represented as a unit step function, which in the s-domain is represented as $\frac{1}{s}$
- The total system response is therefore:

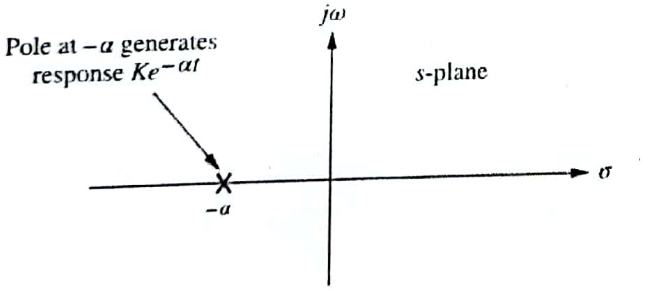
$$C(s) = \frac{(s+2)}{s(s+5)} = \frac{2/5}{s} + \frac{3/5}{s+5} \qquad c(t) = \frac{2}{5} + \frac{3}{5}e^{-5t}$$

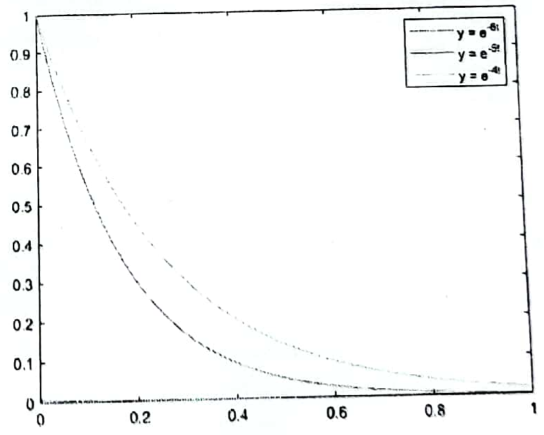
Poles and Zeros of a First-Order System II

- Analysis:
  1. A pole of the input function generates the form of the forced response (that is, the pole at the origin generated a step function at the output).
  2. A pole of the transfer function generates the form of the natural response (that is, the pole at -5 generated e^{-5t}).
  3. A pole on the real axis generates an exponential response of the form $e^{-\alpha t}$, where $-\alpha$ is the pole location on the real axis.
  4. The zeros and poles generate the amplitudes for both the forced and natural responses



Effect of a real-axis pole upon transient response



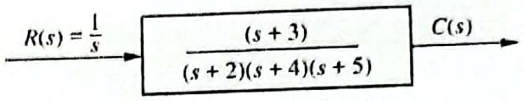


Thus, the farther to the left a pole is on the negative real axis, the faster the exponential transient response will decay to zero Remember the natural response of the system must decay at some point to only leave the system with the forced response, otherwise, it will be an unstable system

Evaluating Response Using Poles – Example I

- Given the system in the adjacent figure, write the output, c(*t*), in **general terms by Inspection.** Specify the forced and natural parts of the solution.

$$C(s) \equiv \underbrace{\frac{K_1}{s}}_{\text{Forced response}} + \underbrace{\frac{K_2}{s+2} + \frac{K_3}{s+4} + \frac{K_4}{s+5}}_{\text{Natural response}}$$



$$c(t) \equiv \underbrace{K_1}_{\text{Forced response}} \underbrace{+K_2 e^{-2t} + K_3 e^{-4t} + K_4 e^{-5t}}_{\text{Natural response}}$$

- Notice that the transfer function quickly made us understand how the output would be like.
- To know the amplitudes of each term (The Ks), write C(s) in MATLAB and use the **partfrac (symbolic)** command or **residue (numeric)** command

Evaluating Response Using Poles – Example II

- Given the system in the adjacent figure, write the output, c(*t*), in **general terms by inspection** if the input is a unit step. Specify the forced and natural parts of the solution.
- Given that the input is $\frac{1}{s}$, $C(s) = \frac{1}{s}G(s)$

$$G(s) = \frac{10(s+4)(s+6)}{(s+1)(s+7)(s+8)(s+10)}$$

$$c(t) \equiv A + Be^{-t} + Ce^{-7t} + De^{-8t} + Ee^{-10t}$$

Analysis of First Order Systems without Zeros

- If the input is a unit step, where R(s) = 1 / s, the Laplace transform of the step response C(s) = R(s)G(s) =

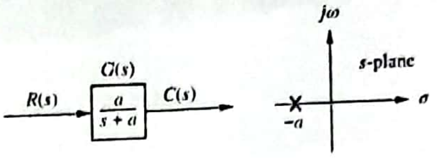
$$C(s) = R(s)G(s) = \frac{a}{s(s+a)}$$

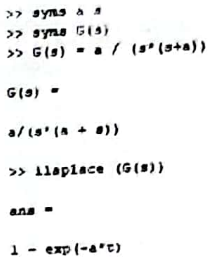
- The inverse Laplace Transform is

$$c(t) = c_f(t) + c_n(t) = 1 - e^{-at}$$

Where the input pole at the origin generated the forced response $C_f(t) = 1$, and the system pole at –a generated the natural response $C_n(t) = -e^{-at}$

when t = $\frac{1}{a}$ the exponential $e^{-at} = e^{-a/a} = e^{-1}$ = 0.37 And c(t) = 1 – 0.37 = 0.63

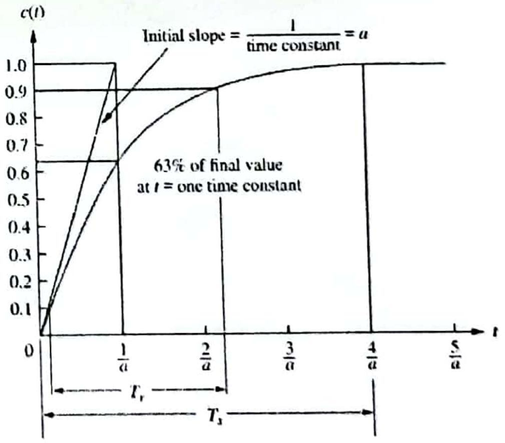




```
>> syms a s
>> syms G(s)
>> G(s) = a / (s*(s+a))

G(s) =

a/(s*(a + s))

>> ilaplace (G(s))

ans =

1 - exp(-a*t)
```

Time Constant of a First Order System

- We call $\frac{1}{a}$ the time constant of the response.
- The time constant can be described as the time for e^{-at} to decay to 37% of its initial value, and that c(t) reaches 63% of its final value.
- The reciprocal of the time constant has the units (1/seconds), or frequency. Thus, we can call the parameter **'a'** the exponential frequency.
- Thus, the time constant can be considered a transient response specification for a first-order system, since it is related to the speed at which the system responds to a step input.
- The time constant can also be evaluated from the pole plot, Since the pole of the transfer function is at a, we can say the pole is located at the *reciprocal* of the time constant, and the farther the pole from the imaginary axis, the faster the transient response.



First-order system response to a unit step

Rise Time and Settling Time of First Order Systems

- *Rise time (T_r)* is defined as the time for the waveform to go from 10% to 90% of its final value.
- To derive the equation, we know that the time-domain equation of a first-order system is $C(t) = 1 - e^{-at}$, and that at t = 0, c(t) equals 0, and as time goes by, c(t) will near to be 1, so the range is 0 to 1
  - 90% of the final value is $0.9 = 1 - e^{-at_{90}} \rightarrow e^{-at_{90}} = 0.1$ then solve for t by talking the natural logarithm for both sides
  - 10% of the final value is $0.1 = 1 - e^{-at_{10}} \rightarrow e^{-at_{10}} = 0.9$ then solve for t by talking the natural logarithm for both sides
  - $t_{90} = \frac{2.31}{a}$
  - $t_{10} = \frac{0.11}{a}$
  - *Rise time (T_r)* $= t_{90} - t_{10} = \frac{2.2}{a}$
- *Settling time (T_s)* is defined as the time for the response to reach, and stay within, 2% of its final value.
  - $T_s = \frac{4}{a}$

Rise Time and Settling Time Example

PROBLEM: A system has a transfer function, $G(s) = \frac{50}{s+50}$. Find the time constant, T_c, settling time, T_s, and rise time, T_r.

ANSWER: $T_c = 0.02$ s, $T_s = 0.08$ s, and $T_r = 0.044$ s.

10/11/2022

(Time Division)

(Time Response)

* هون حنربط Ch1 باللي وصلناله ب ch2 .

* ممكن نطبق خصائص هنا ال chapter على كل block على حدى أو ال block النهائية.

* كل ال Systems اللي بنشتغل عليها لازم تكون linear و Time-Invarient :

* Slide3 : (Poles and Zeros) ← من خلالهم بنقدر نحلل ال System بالنظر ونعرف كيف بشتغل وهيك .

↲ جاي من input بعطيه لل System (خارجي)

* لازم نتذكر إنه ال Natural response + Forced response = total response

↳ مواصفات ال System بحد ذاته (initial State)

* بيهمني إنه ال System تبعي المخرج النهائي تبعه إنه يعتمد بس على ال forced response (My own input) مش شغلاته ومعادلاته الداخلية وأي ال response اله يموت مع الزمن لنقدر نقول إنه ال System stable .

* إذا أطلع ال Forced response وال Natural response بال Time-Domain صعب جدًا حتى لو استخدمنا بال laplace .

* qualititative ← النظر الأفضل لل response تاع ال System .

* quantitative ← أطلع ال response رقميًا (أحسبه) .

* Slide 4 : (Poles and Zeros examples) :

* $S+2 \rightarrow S=-2$ (Zeros)

$S+5 \rightarrow S=-5$ (Poles)

* $H(S)=\frac{(S+2)(S+4)(S-1)}{(S-1)(S-2)(S+5)(S+1)}$

↳ Zeros of $H(S) \rightarrow -4, -2, 1$

↳ Poles of $H(S) \rightarrow 1, 2, -5, -1$

← ال (S-1) مشترك بين البسط والمقام فأجمعوا إنه نحسبها بال Zeros وبال Poles أيضًا

* أصفار البسط هي ال Zeros وأصفار المقام هي ال Poles .

* Slide 5 : Poles and Zeros on S-plane (Complex plane)

* تعريف ال S ب laplace = $\sigma + j\omega$ ولما نصفر ال σ بيرجع laplace ل fourier

فبكون ال real part (σ) (x-axis) وال imaginary part ($j\omega$) .

* بالعادة برسموا ال Zeros وال Poles على ال x-axis . ← Zeros = O

↲ Poles = X

* S-plane ← بفيدني بالنظر لأعرف سرعة ال System بالوصول لل Steady-State .

↲ كأنه شتان صار ده فنشطب ال System

* Slide 6 : كيف بنعرف إنو ال System كان ← First order / Second order / Third order وهكذا ؟ بنعرفها من شكل ال H(s) ، بنطلع عالمقام أنا أكبر درجة بالمقام s فهنا First order ولو s^2 فهنا Second order ولو s^3 فهنا Third order. ولو s^4 فهنا fourth order وهكذا ، كلما يزيد ال order لل system بكون complex أكثر. (احنا بالعادة حنغطي 1st and 2nd order).

* ال Signal اللي بنستخدمها لنفحص ال System : ← بالعادة هدول المستخدمين.
  - ↳ u(t) (unit step function) → Switch (on/of)
  - ↳ δ(t) (Pulse function) → حدث سريع اجا وراح شو ناتجه

← مثلاً حدث رعدي بالجو أدّى إلى إيقاف جهاز Computer فشو التأثير اللي سببه هذا الحدث المفاجئ السريع.

$$* \quad R(s)=\frac{1}{s}\longrightarrow \overset{G(s)}{\boxed{\frac{s+2}{s+5}}}\xrightarrow{C(s)}$$

↳ transfer function of u(t) → بحلل ال System كيف حيشتغل لما أكبس فيه كبسة.

→ $C(s)=\frac{(s+2)}{s(s+5)}=\frac{2/5}{s}+\frac{3/5}{s+5}$ (Partial fraction (make it in Matlab) كشان نرجع ال function لأصلها قبل توحيد المقام لأقدر أرجع لل laplace table).

→ from table : $c(t)=\frac{2}{5}+\frac{3}{5}e^{-5t}$ → فيه جزئين : 1: $\frac{2}{5}$ 2: $\frac{3}{5}e^{-5t}$

↳ In Matlab : الطريق لأوصل من ال C(s) لل c(t).

```
Syms  s  G(s)  C(s)
G(s) = (s+2) / (s+5)
C(s) = 1/s * G(s)
ilaplace ( partfrac ( c(s)) )
```

* الجزء اللي حيموت من معادلة ال c(t) مع الزمن هو e^{-5t} لأنه في بداية التشغيل لما t=0 بكون قيمته 1 ف $c(t)=\frac{2}{5}+\frac{3}{5}=1$ بس كلما يمر وقت بال sys ال e بتضل تقل بقل معها الجزء الثاني لحد فترة زمنية طويلة بصفر الجزء الثاني فبضل في بس $\frac{2}{5}$ ← وبثبت على 0.4 فبصير بال Stead-state وكل شي between لحظة التشغيل ولحظة الثبات على ال Steady-state كان ال transiant response.

* Slide 7 : ربطت ال sys اللي ب Slide 6 بال Zeros and poles .

* Zeros = -2 / Poles = -5 / input = 1/S ما إله Zero بس إله Pole = 0

* بال sys اللي المقام فيها من الدرجة الأولى لو أدخل عليها u(t) تأثير ال u(t) مع ال Zeros هو اللي بجيبلي ال Steady-state . وال transiant بسبب ال Pole.

* دايما بهذا النوع من ال sys حيطلع 2 terms واحد Steady وواحد transiant.

* $c(t) = \underbrace{\frac{2}{5}} + \underbrace{\frac{3}{5} e^{-5t}}$

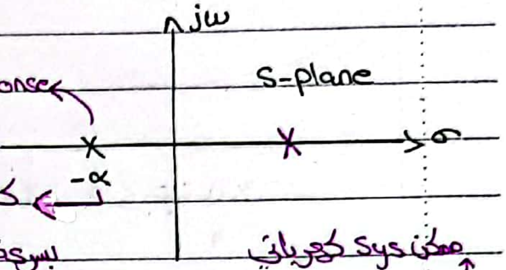
Forced response ↲ ↳ Natural response → هو الـ Transiant لأنه بموت

↳ Steady-State → هو اللي بضل.

* قديه بدي وقت لأوصل الـ Steady-State ؟ بناءً على الـ Natural response (القيمة اللي بالـ e^{-5t} بالمثال هون "اللي بتيجي من الـ Pole")

* Slide 8 :

Pole at $-\alpha$ generates response $ke^{-\alpha t}$



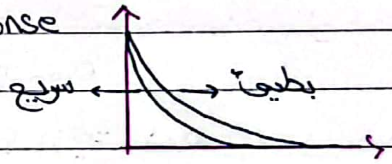
← كلما تبعد عن الـ 0 بتكون بتنزل بسرعة وكل ما تقرب للـ 0 بتكون بتنزل ببطء. فأنا من الرسمة بدون أي رياضيات عرفت إنه الـ sys بنزل بسرعة أو ببطء.

ممكن sys كهربائي

$s =$ → سريع جدًا ← $-10, -9$
↳ بطيء جدًا ← $-1, -0.5$

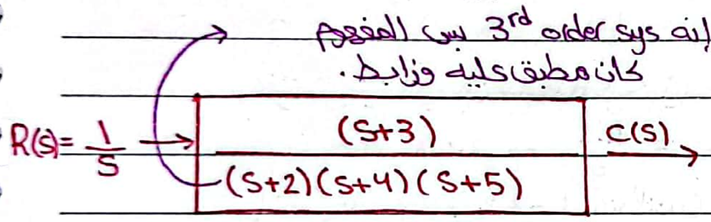
ممكن sys ميكانيكي أو هيدروليكي.

* لو الـ Pole موجودة مكان الـ X الوردية : معناها الـ sys ما رح ينزل نزول رح يطلع طلوع فما رح يموت الـ Natural response تبعه فاحتمال كبير ما يكون Stable (unstable).



رغم إنه 3rd order sys بس المفهوم كان مطبق عليه وزابط.

* Slide 9 : EX1 ←



$$R(s) = \frac{1}{s} \rightarrow \frac{(s+3)}{(s+2)(s+4)(s+5)} \xrightarrow{C(s)}$$

→ Poles : $-2, -4, -5$ ↴

كل وحدة منهم بتعطيني : $e^{-5t}, e^{-4t}, e^{-2t}$ ← أكبر وحدة فحتنزل بسرعة

↳ zeros : -3

$C(s) = \frac{k_1}{s} + \frac{k_2}{s+2} + \frac{k_3}{s+4} + \frac{k_4}{s+5}$

Forced Response ↲ ↳ Natural response

$\rightarrow c(t) = \underbrace{k_1} + \underbrace{k_2 e^{-2t} + k_3 e^{-4t} + k_4 e^{-5t}}$ → "من الـ Matlab بنطلعها"

Forced Response ↲ ↳ Natural Response

↲ قيم k بنطلع بس هلا ما بهمنا منهم إلا k_1 لأنه الباقي بموتوا.

* Slide 10 : EX2 ←

$G(s) = \frac{10(s+4)(s+6)}{(s+1)(s+7)(s+8)(s+10)}$

$\rightarrow c(t) = A + Be^{-t} + Ce^{-7t} + De^{-8t} + Ee^{-10t}$ → from Matlab

Steady-state ↲

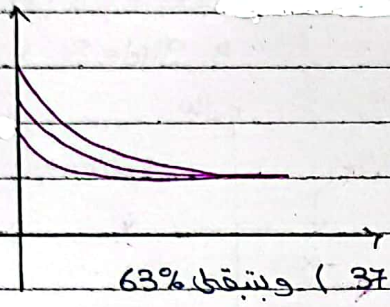
* Slide11: ← (First order System without Zeros)

بالبداية عادةً بخلو البسط ← $C(S) = R(S)\,G(S) = \frac{a}{S(S+a)}$ ← البسط constant ←

والمقام نفس ال constant ← from Matlab ← $c(t) = C_f(t) + C_n(t) = 1 - e^{-\alpha t}$

* ماعندي إلا Pole وحدة عند (-a).

* أكثر شغلة بتوصف هون إنه ممكن أوصل لل Steady-state بأكثر من طريقة ، اللي بحدد سرعة الوصول هي α.



* Slide12: $\frac{1}{\alpha}$ (Time constant of the sys).

← α (Frequency).

الفترة الزمنية اللي بوصل فيها ال sys ل 37% من قيمته النهائية (بموت فيها بنسبة 37%) وبتبقى 63%

* الرسمة خاصة بال First order وال unit step funct.

* لو عندي e^{-5t} فال Time const. بساوي $\frac{1}{5}$ يعني 0.2 يعني بعد $\frac{1}{5}$ ثانية ال Sys بوصل ل 63% من ال Steady-State ولو كانت α قيمتها 10 معناها بعد $\frac{1}{10}$ ثانية بوصل ال Sys لل Steady-State (كله جايبني من ال Pole ، أكبر قيمة إلها بتحدد سرعة الوصول لل Time response خاصة لو كانت First order ، كلما كبر الرقم كل ما وصلت أسرع).

* Slide13: (Rise Time and Settling Time)

* Rise Time → الفترة الزمنية اللي بستغرقها ال Sys ليوصل من 10% لغاية 90%.

$T_r = t_{90} - t_{10} = \frac{2.2}{\alpha}$ → 2.2 * Time constant ← مش مطلوب اشتقاقها.

* Settling Time → لما يكون ال sys (within 1-2%) من ال Final Value

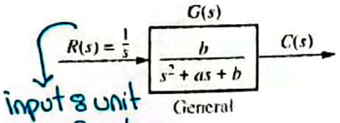
$T_s = \frac{4}{\alpha}$ → 4 * Time constant ← مش مطلوب اشتقاقها.

* بهيك ال Time const , Settling Time , Rise Time وهذول الي شرحناهم لل 1st order System.

مش بس بتأثر بال Speed زي ال First order
كمان ممكن تأثر بشكل الانتقال.

Second Order Systems: Introduction

- Varying a first-order system's parameter a simply changes the *speed* of the response.
- Changes in the parameters of a second-order system can change the *form* of the response
  - Can display characteristics much like a first-order system
  - Display damped or pure oscillations for its transient response.



input : unit step function
مسؤول عن forced response
اللي بوصلني للجواب النهائي (Steady state)

* بنفرض بأنه b اللي فوق نفس b اللي تحت.

* لما نشوف معادلات 2nd order (المقام فيه تربيعي) وتطلع الجذور تبعته حقيقية (real not complex)
فبنتوقع الصيغة النهائية يكون فيها 2 terms exponent بالإضافة إلى القيمة الثالثة (Steady-state)
شكلها بشبه كثير ال 1st order.

Second Order Systems: Overdamped

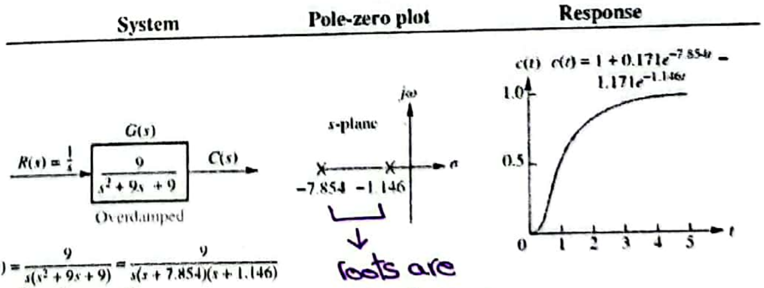
This function has a pole at the origin that comes from the unit step input and two real poles $-\sigma_1$, $-\sigma_2$ that come from the system.

The input pole at the origin generates the constant forced response;

Each of the two system poles on the real axis generates an exponential natural response whose exponential frequency is equal to the pole location.

The output can be estimated as

$$c(t) = K_1 e^{-\sigma_1 t} + K_2 e^{-\sigma_2 t}$$



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

↳ from $\frac{1}{s}$

roots are real num
↳ 2 poles not 1 like 1st order

$$c(t) = K_1 + \underbrace{K_2 e^{-7.854t} + K_3 e^{-1.146t}}$$

Steady-State
لـ لوين بدي أوصل

2 terms for exp. صار عندي
لـ شكل الوصول

(لما الجذور تكون complex) أهم شكل

Second Order Systems: Underdamped I

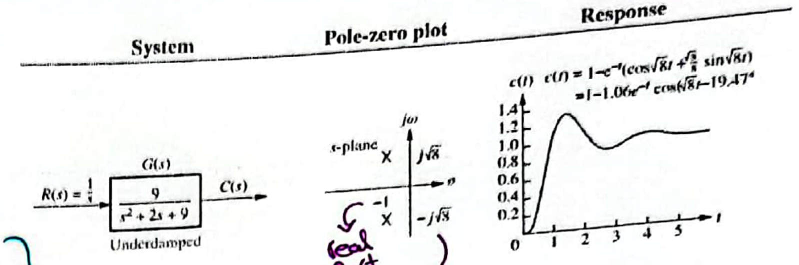
This function has a pole at the origin that comes from the unit step input and two **complex** poles $-\sigma_d \pm j\omega_d$ that come from the system.

The poles that generate the natural response are at $s = -1 \pm j\sqrt{8}$.

real ↵ ↳ Imagin

The real part of the pole matches the exponential decay frequency of the sinusoid's amplitude.

The imaginary part of the pole matches the frequency of the sinusoidal oscillation.



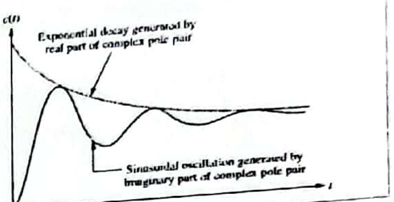
Sinusoid حتكون عبارة عن

Signal راكب عليها

exponential Signal

real Part

Imagin Part



*الـ sن كنا مربوطة دائمًا بال Signals

$*\ e^{-5}$ ← exponential / e^{-5j} ← Sinusoid

أول ما نشوف الـ Poles هيك

فيها إزاحة عن الـ real Part

بنفهم إنه هذا الـ System لما أوطيه

input مارح يوصل الـ output بسرعة

بده يعمل عمليات oscillation بعدين يوصل الـ output.

Second Order Systems: Underdamped II

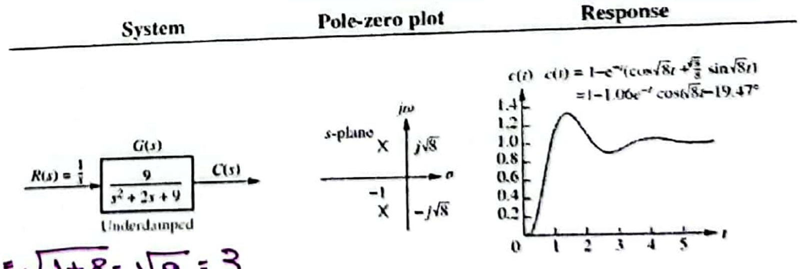
The time constant of the exponential decay is equal to the reciprocal of the real part of the system pole.

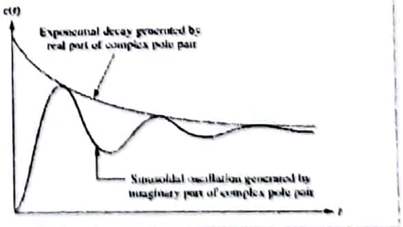
The value of the imaginary part is the actual frequency of the sinusoid. This sinusoidal frequency is given the name *damped frequency of oscillation*, ω_d (radian frequency)

$$\hookrightarrow \sqrt{(-1)^2+(\sqrt{8})^2} = \sqrt{1+8} = \sqrt{9} = 3$$

The output can be estimated as

$$c(t) = Ae^{-\sigma_d t}\cos(\omega_d t - \phi)$$





15/11/2022

"Revsion"

* unit step function يعبّر عنه بـ (1/S) وهو للتعريف بتشغيل الـ System أو كبسة التشغيل (هو اللي بنقلني من حالة لحالة) (بعبر عن الـ Forced response) وهو مسؤول عن الـ Steady state لكن شكل الانتقال من حالة إلى حالة مسؤول عنه الـ System المعبّر عنه بالـ Transfer funct (اللي هي الـ Poles تاعت الـ Function).

* 1st order بتغير فقط بالـ Speed بينما الـ 2nd order بتغير بالـ Speed والـ Form

* الـ Poles الموجودة بـ 1st order sys بتعبر عن exponential part.

* شكل الانتقال بالـ 1st order system هو شكل واحد بينما بالـ 2nd في أشكال مختلفة للانتقال.

* المبدأ اللي ماشين عليه حاليًا من slide16 وطالع إنه الرقم اللي بالبسط نفسه اللي تحت.

* Slide 16 : المعادلة بشكلها هيك صعب نحللها ونفهمها من شكلها فبنحتاج لتحليلها أو لاستخراج الجذور، لنطلع الجذور بالماتلاب : (هذا حل من الحلول).

eq = [1, 9, 9] → Vector بعد التأكد إنه المعادلة مرتبة،

roots (eq) → بعطيني الجذور وعددهم 2 لأنها معادلة تربيعية → real num مو شرط يكونوا ممكن يكونوا complex num لو كان تحت الجذر سالب.

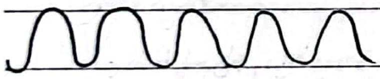
* Slide 17 :

* Note : الـ exponential لما نيجي نضربها بـ Sin waves شو بتسوي الـ exponential بالـ Sin wave ؟

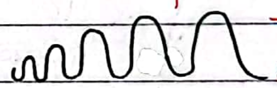
Sin wave

نتيجة ضربها بـ :

الناتج :



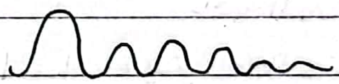
1- $e^{\alpha t}$



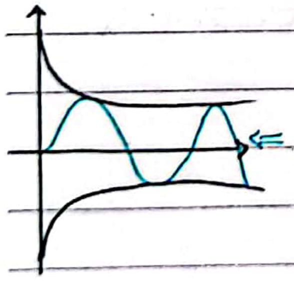
← الـ Sys بضل يتضخم لحد ما ينكسر لو ميكانيكي أو ينحرق لو كهربائي

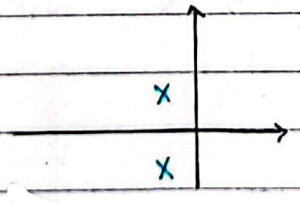
عادة هاي مش ← منيحة لأنه الـ Sys بكون عادة unstable لهيك ما كنا نحب نشوف Poles عالجهة اليمين

2- $e^{-\alpha t}$



الـ Sys بنزل لحد ما يوصل الـ Steady-State (هذا اللي بنفضله)





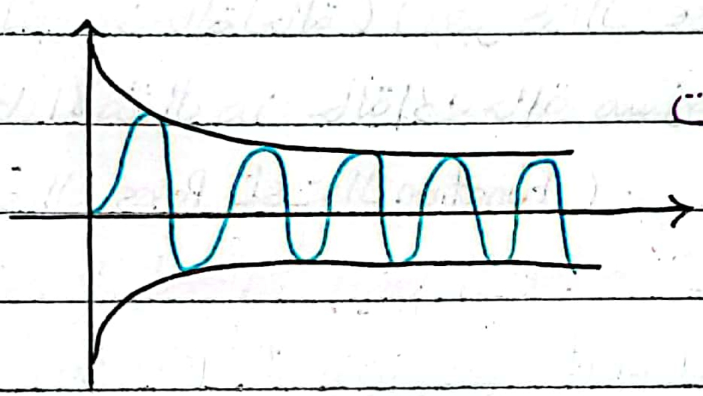
* افترضنا الـ Poles كانوا هيك :

→ الـ Frequency تبعتها بطيئة

(الـ Poles قريبة من الـ X- axis)

* بينما لو كانت الـ Poles بعيدة عن الـ X-axis :

* بعدنا عن الـ real-axis
بوضحلي الـ Frequency تاعت
الـ oscillation لأوصل
الـ Steady-state



الـ Frequency ← تبعتها
سريعة

* $\sqrt{8}$ هي ليست الـ Frequency نفسها ، هي تدل على سرعة أو بطء الـ Frequency لكن
لا تساويها . (Slide 17)

* الفرق الأساسي بين الـ 1st order والـ 2nd orders إنه الـ 1st order شكل الانتقال
للكل نفس الشكل ، بينما الـ 2nd order شفنا 4 أشكال مختلفة للانتقال (Undamped
(overdamped / critically damped / under-damped /

← معناها مداما في real part فما في exponential فكل ال Signal تبقى عبارة
Sinusoidal cs

لما تكون ال roots فيها بس Imagin ←
ما فيها real part .

Second Order Systems: Undamped

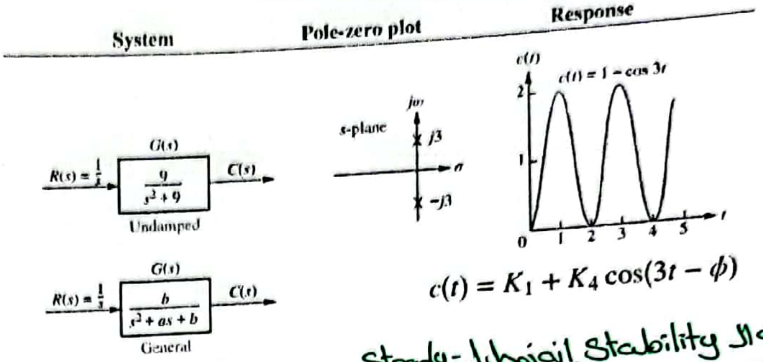
This function has a pole at the origin that comes from the unit step input and two imaginary poles $\pm j\omega_d$ that come from the system.

The input pole at the origin generates the constant forced response.

The two system poles on the imaginary axis at $j3$ generate a sinusoidal natural response whose frequency is equal to the location of the imaginary poles ↳ $\sqrt{9} = 3$ (سهل جدًا حسابه) → radian →

The output can be estimated as $c(t) = A\cos(\omega_1 t - \phi)$ $\omega_n = \sqrt{b}$

ما يحتاج أحولها بالماتلاب لأنها radian



$$c(t) = K_1 + K_4\cos(3t - \phi)$$

لأنه ال Stability انوصل ل Steady-State وهون مافي real-num يوصلنا .

هل هنا ال System يكون Stable ؟ يعتمد على ال System نتاعك شو بعمل ، قد يكون مطلوب ال out يكون Sinusoidal فهنا يطابق النتيجة فيكون Stable وقد يكون غير مطالب ال out يكون Sinusoidal فهيك يكون unstable .

الجذر (real) ما في complex بس الجذرين مساوين لبعض في القيمة .

Second Order Systems: Critically damped

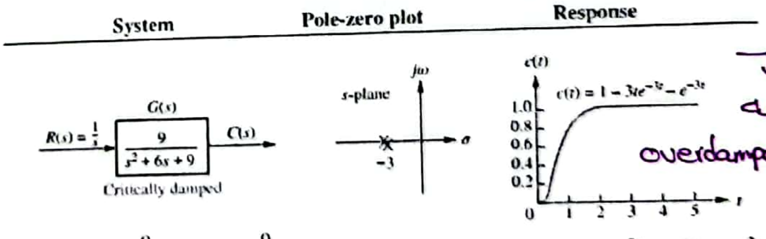
This function has a pole at the origin that comes from the unit step input and two real poles $-\sigma_1$ that come from the system.

The input pole at the origin generates the constant forced response

The two poles on the real axis at 3 generate a natural response consisting of an exponential and an exponential multiplied by time, where the exponential frequency is equal to the location of the real poles

The output can be estimated as $c(t) = K_1 e^{-\sigma_1 t} + K_2 t e^{-\sigma_1 t}$

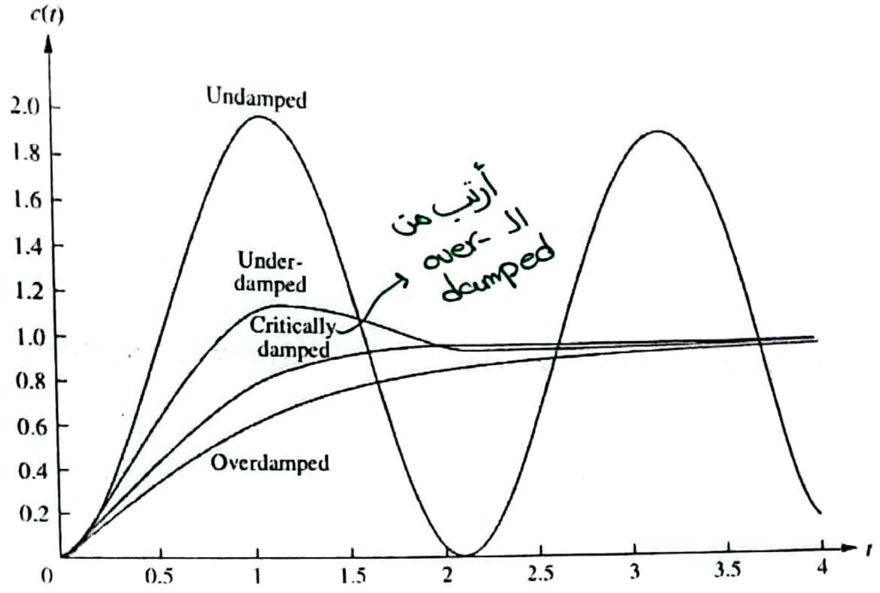
أحد ال exp. terms
بكون مضروب ب t
(time dependent)



$$C(s) = \frac{9}{s(s^2+6s+9)} = \frac{9}{s(s+3)^2}$$

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

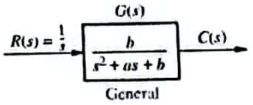
↓
بشبه
ال overdamped
بس
هنا
أفضل
وأرتب



أرتب من الـ over-damped

(مقارنتهم ببعض)

The General Second Order System



* من ζ لحالها ممكن نعرف شكل الـ System بدون الـ Transfer function .

Natural Frequency, ω_d of a second-order system is the frequency of oscillation of the system without damping

Damping Ratio, ζ compares the exponential decay frequency of the envelope to the natural frequency

$$\zeta = \frac{\text{Exponential decay frequency}}{\text{Natural frequency (rad/second)}} = \frac{1}{2\pi}\,\frac{\text{Natural period (seconds)}}{\text{Exponential time constant}}$$

نسبة الـ Poles = for exponential

for oscillation

للتحويل لـ radian وليقيم

تأثير الـ radian الموجود .

For the underdamped system, the complex poles have a real part σ, equal to −a / 2. → Frequency for exponential =Poles

← لإنه الـ Freq ما إلها معنى بالسالب .

$$\zeta = \frac{\text{Exponential decay frequency}}{\text{Natural frequency (rad/second)}} = \frac{|\sigma|}{\omega_n} = \frac{a/2}{\omega_n} \longrightarrow a = 2\zeta\omega_n \longrightarrow \boxed{G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}}$$

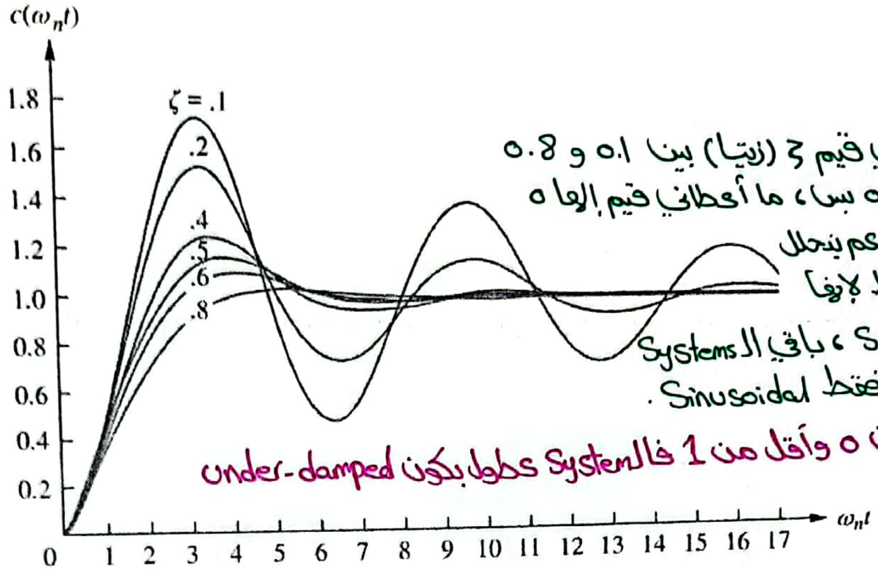
$\omega_n = \sqrt{b}$ → Frequency (Natural) → تقاس بالـ radian فلا حاجة لتحويلها إلى radian بالمثلاب لإنها أساسًا بالـ radian .

$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$ → roots

كتب المعادلة بدلالة ζ و ω_n و S

(عاد كتابة المعادلة من Algebra إلى Parameters Signals ترتبط بالـ Control systems)

Effect of ζ (dampening ratio) Visualized



* قاصد يعطيني قيم ζ (زيتا) بين 0.1 و 0.8
ممكن يضيف 0.9 بس، ما أعطاني قيم إلها 0
أو 1 وأكبر لإنه هو بيحلل
under-damped sys فقط لإنها
بتحوي exp و Sin wave ، باقي الـ Systems
يا إما فقط exp أو فقط Sinusoidal .

* طالما قيم زيتا أكبر من 0 وأقل من 1 فالـ System عطول بكون under-damped
↲ Important Note

* لما بندرس أي System بالدنيا عشان نشوف تأثير الـ Variable اللي قيد الدراسة على الـ System
لازم نثبت باقي الـ Variables الباقيين.

* هون : almost all Sin waves have Similar frequency ليسهل علينا عملية المقارنة (التغيير بـ ζ)
↲ كلما كانت قليلة بكون الـ exp. Frequency قليل فبكون الـ
dampening بطيء وكل ما زادت بزيد الـ dampening وبصير سريع.

Finding ζ and ω_n For a Second-Order System - Example

Given the transfer function of G(s), find ζ and ω_n.

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

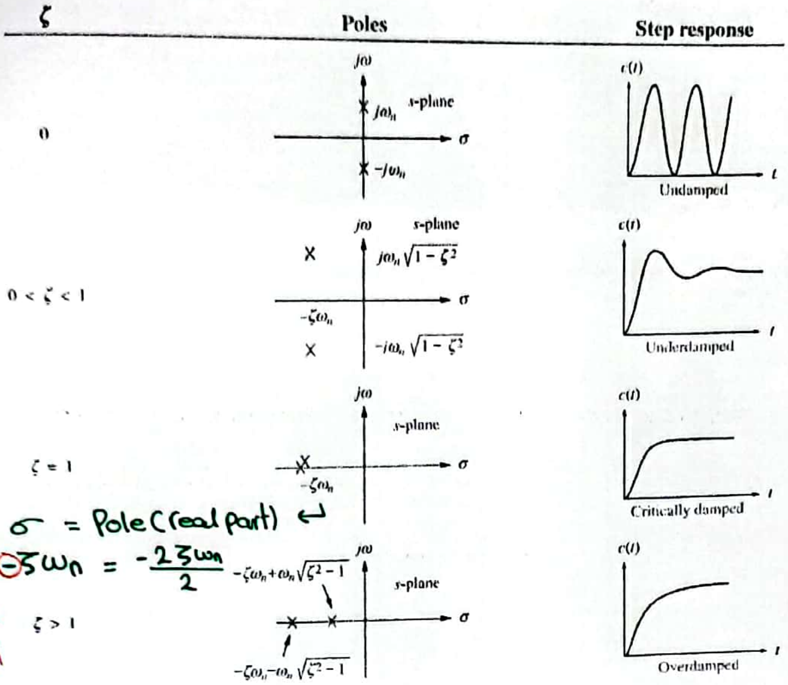
$$\boxed{G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}}$$

$\omega^2_n = 36$, therefore $\omega_n = 6$

Also, $2\zeta\omega_n = 4.2$, so after substituting the value of $\omega_n = 6$, w get $\zeta = 0.35$ (underdamped System)

* سؤال امتحان: بجيبلك 2 systems وانت
قارن بينهم واحكيله شو نوعهم (undamped
/ underdamped وهكذا!).

* سؤال امتحان : بعطيك رسمة ال Poles ويقلك أي وحدة من ال S-Domain Plots بطابق مواصفات رسمة ال Poles .



البسط بمعادلة ال ζ

→ we have complex Poles (we don't have real)

= 0 يعني مافي exp. Part بس في oscillation Part فإذا هو undamped

→ Poles and Zeros 8 Complex

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

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

→ الحالة الخاصة اللي بكون ζ فيها بساوي 1 .

Pole (real part) = σ = $\frac{-a}{2}$ =
↑
اللي مع S

$-\zeta\omega_n = -\frac{2\zeta\omega_n}{2}$

كشان تروح ال Poles للجهة الشمال.

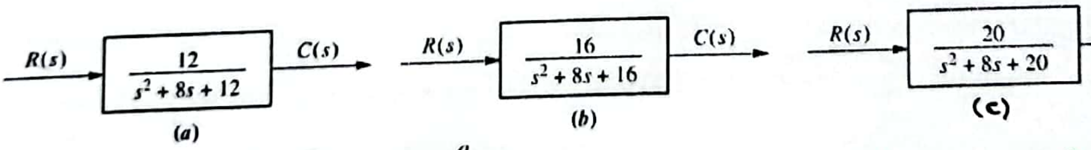
→ كبرت ζ فتقريبا انلغى تأثير الـ oscillation وصار ال Dominant هو ال exponential .

* لما تضل زيتا تكبر كثير بصير زي كإني بلغي كل تأثير الـ Oscillation فبترجعني للحالات اللي بغلب فيها الـ exponential .

Characterizing Response from the Value of ζ - Example

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

For each of the systems shown, find the value of ζ and report the kind of response expected.



→ Poles in −4
= −a/2
= −8/2
= −4

Since $a = 2\zeta\omega_n$ and $\omega_n = \sqrt{b}$, $\zeta = \frac{a}{2\sqrt{b}}$

* بالماتلاب ممكن ترسمهم بس بهاي الحالة هيك أسهل.

We find:

* ممكن يقلك حددلي أماكن ال Poles.

ζ = 1.155 for system (a), which is thus overdamped, since ζ > 1;
ζ = 1 for system (b), which is thus critically damped
ζ = 0.894 for system (c), which is thus underdamped, since ζ < 1.

17/11/2022

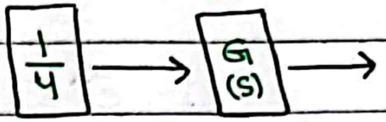
* Slide 22 :

* هون كنا مش زي ما فرضنا قبل ، هون البسط لايساوي المقام ($4 \neq 16$) فهل بنقدر نحلها زي باقي الـ Systems ؟

Suppose that we have : $\dfrac{4}{6S^2+2S+16}$ →

بنحاول نطلع عامل مشترك لنقدر نخلي القيمة اللي بالبسط تساوي القيمة اللي بالمقام :

$$\frac{1}{4}\ \frac{(4\times4)}{6S^2+2S+16} \rightarrow \frac{1}{4} * \frac{16}{6S^2+2S+16} \rightarrow$$



*كأنهم صاروا 2 Subsystems

← هنا إجراء رياضي فقط مش واقعي وحقيقي

← أصبحت كأنها gain من ناحية الـ Control System.

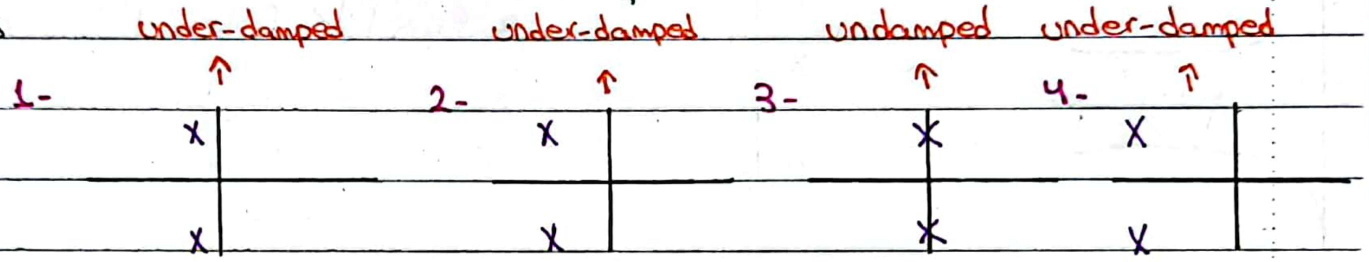
* هون σ قيمتها = -a/2 = -2/2 = |-1| ← يعني Pole عند -1

والـFrequency بتكون |-1| = +1 بمعادلة ζ

* $\omega_n = \sqrt{b} = \sqrt{16} = 4$ * $\zeta = \frac{1}{4}$ ← under-damped system

* Slide 26 :

→ Note :



← بجيبلك الـ System وبجيبلك ياهم وبقلك اختار الـ Poles اللي بتمثل الـ System المعطى.

Characterizing Response from the Value of ζ - Exercise

" Self Exercise "

For each of the systems shown, find the value of ζ and ω_n and report the kind of response expected. + Find the poles of each system.

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

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

a. $G(s) = \dfrac{400}{s^2 + 12s + 400}$

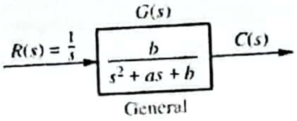
b. $G(s) = \dfrac{900}{s^2 + 90s + 900}$

c. $G(s) = \dfrac{225}{s^2 + 30s + 225}$

d. $G(s) = \dfrac{625}{s^2 + 625}$

بتهمني كتير لانها موجودة كتير بالطبيعة

Underdamped Second-Order Systems - Parameters



We have seen that in the underdamped case, it is assumed that $\zeta < 1$

Other parameters associated with the underdamped response are rise time, peak time, percent overshoot, and settling time.

مدام في oscillation فهو الزمن لحد ما يوصل أول Peak (منتصفها) (أول أعلى نقطة بال System).

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

1. *Rise time, Tr.* The time required for the waveform to go from 0.1 of the final value to 0.9 of the final value.
2. *Peak time, TP.* The time required to reach the first, or maximum, peak.
3. *Percent overshoot, %OS.* The amount that the waveform overshoots the steady-state, or final, value at the peak time, expressed as a percentage of the steady-state value. → قديه الSystem نط عندي فوق الSteady-state (ارتفاع الSystem)
4. *Settling time, Ts.* The time required for the transient's damped oscillations to reach and stay within 2% of the steady-state value. → أوصل 98% من الSteady-State

$$C(s) = \frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} = \frac{K_1}{s} + \frac{K_2 s + K_3}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Notice that the definitions for settling time and rise time are basically the same as the definitions for the first-order response. All definitions are also valid for systems of order higher than 2 → هذه التعريفات مش مقتصرة عالـ2nd order system خلاقيها بنفس المعنى للSystems الأخرى.

Underdamped Second-Order Systems - Parameters

كلهم بعطونا معلومات لسرعة وصولنا الـ Steady-State .

* Rise time, peak time, and settling time yield information about the **speed of the** *transient response*. →

This information can help a designer determine if the speed and the nature of the response do or do not degrade the performance of the system. → بساعدوا الـ designer يعرفوا هل الـ response هذا حيخرب الـ System ولا ماحيخربه →

For example, the speed of an entire computer system depends on the time it takes for a hard drive head to reach steady state and read data; passenger comfort depends in part on the suspension system of a car and the number of oscillations it goes through after hitting a bump

→ HDD : 1- overshoot : الـ motor يزيح الـ disk مايجي عالراس بالزبط به وقت ليستقر بالمكان الصح

2- Rise Time : الـ علاقة بسرعة الدوران قديه بجيبني قريب على الـ Steady-State

3- Settling : الوقت حتى استقر وأوصل 98% للـ Steady-state

وهكذا بيبنلك إنهم كيف ممكن يأثروا بسرعة الـ System للوصول إلى النتيجة النهائية.

Underdamped Second-Order Systems – Parameters Evaluation I

* كلهم مرتبطين ببعضهم البعض *

Peak time

$$\boxed{T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}}$$

→ بعتمد عالـ Frequency والـ damping

Percentage Overshoot

$$\boxed{\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100}$$

→ exp. ليتخلص من الـ دخل ln عالطرفين

$$\boxed{\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}}}$$

معادلة ثانية لـ ζ

Settling Time

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

* الـ Rise-Time ما قدروا يطلعولها معادلة (علاقة) بسبب كثرة المتغيرات فعملوا التالي (Slide 31).

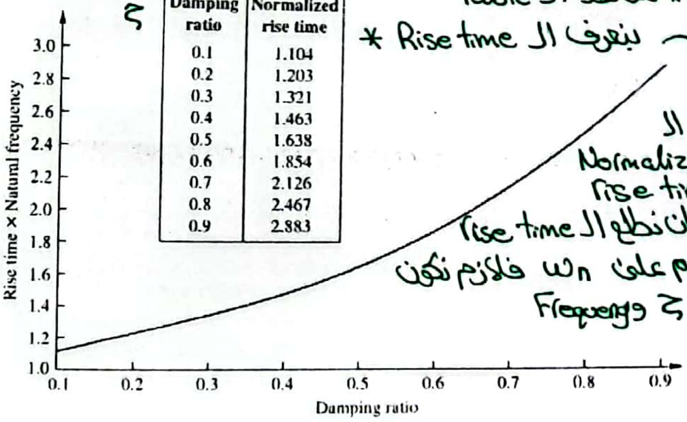
Underdamped Second-Order Systems – Parameters Evaluation II

An analytical relationship to derive T_r from the other system parameters is not easily available.

An experimental relationship exists.

- We set the value of c(t) to 0.9 and solved for $\omega_n t$.
- Then, we set the value of c(t) to 0.1, and solve again for $\omega_n t$.
- The rising time is the time required for the waveform to go from 0.1 of the final value to 0.9 of the final value, so T_r is the difference between the two values

Note; however, that we solved for $\omega_n t$ not t. This $\omega_n t$ is called the normalized rise time. We plot the resulting equation with respect to different value of ζ



ζ

Damping ratio	Normalized rise time
0.1	1.104
0.2	1.203
0.3	1.321
0.4	1.463
0.5	1.638
0.6	1.854
0.7	2.126
0.8	2.467
0.9	2.883

* من هنا الـ table

بنعرف الـ Rise time *

بطلع الـ Normalized rise time فعشان نطلع الـ rise time بنقسم على ω_n فلازم نكون نعرف ζ و Frequency

* الـ Rise Time : الوقت اللي بمضي ما بين تكون الـ Signal = 10% إلى 90% .

لـ انحلت مش بالنسبة لـ t بالنسبة لـ $\omega_n t$ تحت مسمى Normalized Rise time .

* مش مهم كثير التفاصيل الرياضية لأنه بالنهاية حنحله بالـ (Matlab) .

* سيئة الحل الـ Visual إنه مش دقيق (إجابات متقاربة لكن ليست دقيقة) .

* فبنحلها رياضيًا بالـ Matlab باستخدام الـ Interpolation

Underdamped Second-Order Systems – Parameters Evaluation III

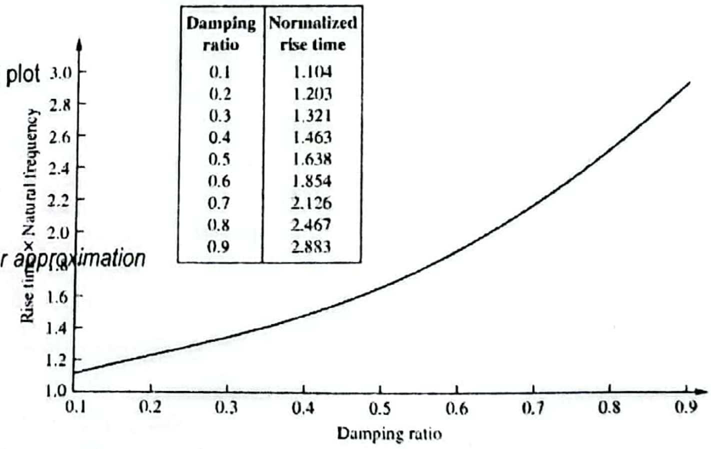
To find the rising time for an underdamped system,
First, we determine ζ, then we see to which value in the plot
It corresponds to $\omega_n t$
Finally , divide by ω_n

Visual is not accurate

Use MATLAB commands interp1 or spline to get a better approximation

```
X = 0.1 : 0.1 : 0.9
Y = [1.104, 1.203, 1.321, 1.463, 1.638 ...
     1.854, 2.216, 2.467, 2.883]
Interp1 (X, Y, your ζ)  // linear interpolation
Spline  (X, Y, your ζ)  // cubic  interpolation
// Then divide by ωn
```

non-linear Interpolation → أدق بكثير



Damping ratio	Normalized rise time
0.1	1.104
0.2	1.203
0.3	1.321
0.4	1.463
0.5	1.638
0.6	1.854
0.7	2.126
0.8	2.467
0.9	2.883

* مثلاً لو طلبنا الـ Rise time لـ System الـ damping
ratio (ζ) = 0.85 والـ Frequency = 5 ← بنحط قيم X وقيم
Y الثابتة من الجدول بعدين بنستخدم Spline أو Interp1 (أدق Spline)
و بتكون قيمة your ζ = 0.85 والناتج بنقسمه 5 ليعطيني الـ Rise time .

Underdamped Second-Order Systems – Parameters Evaluation Ex.

* $\sigma = \frac{a}{2}$

* $\zeta = \frac{\sigma}{\omega_n}$

Finding T_p, $\%OS$, T_s, and T_r from a Transfer Function

PROBLEM: Given the transfer function

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

find T_p, $\%OS$, T_s, and T_r.

$\to \sqrt{100}$ → underdamped ↱ σ / ω_n

* $\sigma = \frac{15}{2} = 7.5$

لـ ال Pole عالنقطة 7.5

SOLUTION: ω_n and ζ are calculated as 10 and 0.75, respectively. Now substitute ζ and ω_n into Eqs. find, respectively, that $T_p = 0.475$ second, $\%OS = 2.838$, and $T_s = 0.533$ second. Using the table , the normalized rise time is approximately 2.3 seconds. Dividing by ω_n yields $T_r = 0.23$ second. This problem demonstrates that we can find T_p, $\%OS$, T_s, and T_r without the tedious task of taking an inverse Laplace transform, plotting the output response, and taking measurements from the plot.

* اللي بغلب هو ال rising time لإنو ماإله معادلة .

* أول خطوة لازم نطلع ω_n و ζ لإنها المعادلات بتعتمد عليهم بعدين تطبيقات مباشرة .

* Important : ال Settling time وال overshoots وال Peak time لل oscillation فلازم ال System يكون فيه oscillation عشان تقدر تطبق عليه هاي المعادلات .

← بيجي كثير مقارنات زي هيك بالامتحان .

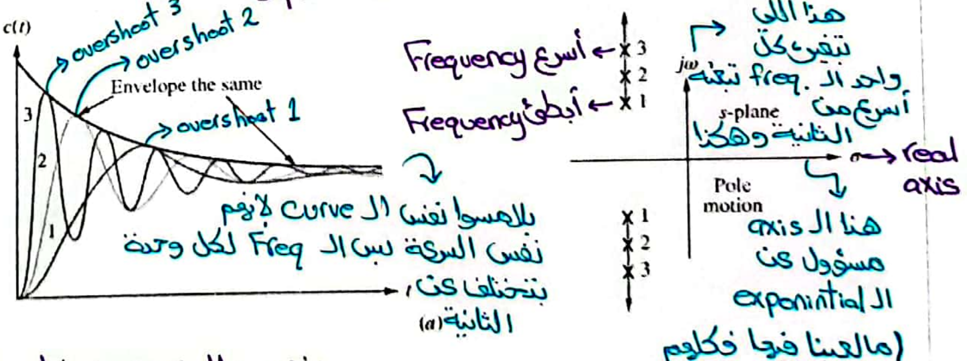
Pole Motion Effect on the Underdamped System Response I

* Systems with 2 complex Poles → under-damped ← طبقًا حسب شكل ال Poles عال S-plane

As the poles move in a vertical direction, the frequency increases, but the envelope remains the same since the real part of the pole is not changing.

The figure shows a constant exponential envelope, even though the sinusoidal response is changing frequency.

Since all curves fit under the same exponential decay curve, the settling time is virtually the same for all waveforms.



بالمسوا نفس ال curve لأنهم نفس السرعة بس ال Freq لكل وحدة بتختلف عن الثانية ،

↱ هذا اللي بتغير كل واحد ال freq. تبعته أسرع من الثانية وهكذا

هنا ال axis مسؤول عن ال exponential (مالعبنا فيها فكلهم نفس السرعة) (نفس ال Envelope)

Note that as overshoot increases, the rise time decreases.

① ② → بنحسب ال 90% من خط ال Steady-state (مو مرسوم هون بالرسمة)

③. Settling time ال تغيير مكان ال Poles ما بأثر عال * لـ كلهم وصلوا نفس المكان

* الأعلى Frequency ← أعلى overshoot ← أقل rise time .

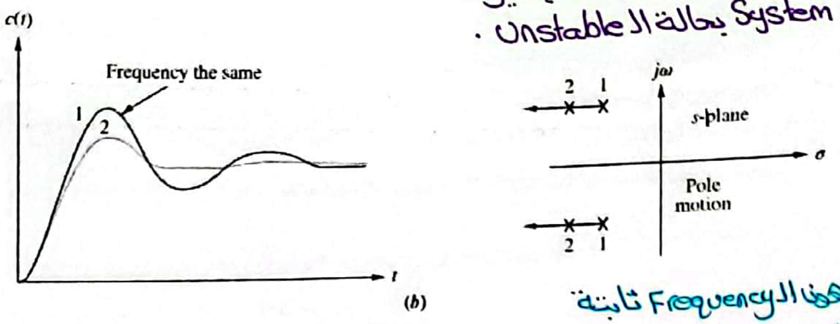
Pole Motion Effect on the Underdamped System Response I

* المقارنة هون بناء عال left side لانه لو رحت لليمين ممكن يصير ال System بحالة ال Unstable .

Let us move the poles to the right or left. Since the imaginary part is now constant, movement of the poles yields the responses shown in the figure.

Here the frequency is constant over the range of variation of the real part. As the poles move to the left, the response damps out more rapidly, while the frequency remains the same.

*
Notice that the peak time is the same for all waveforms because the imaginary part remains the same.



(b)

* هنا ال Frequency ثابتة
بس سرعة الانخساف بتتغير كلما بعدنا بكون أسرع .

* ال Peak Time : الوقت من أول ما بلشنا لغاية منتصف أول Peak ، مارح يختلف ال Peak time بتغير سرعة الانخساف.

Pole Motion Effect on the Underdamped System Response I

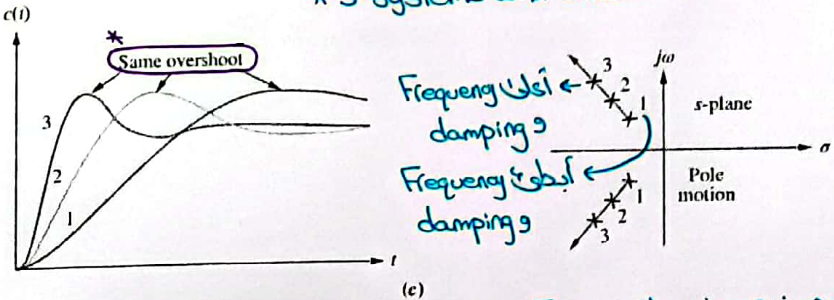
← ما بحرك بشكل عشوائي ← عمل شعاع diagonal ويمشي عليه ال System

Moving the poles along a constant radial line yields the responses shown in the figure

Here the percent overshoot remains the same.

Notice also that the responses look exactly alike, except for their speed. The farther the poles are from the origin, the more rapid the response

* 3 Systems with different Poles *



Frequency أعلى ← و damping

Frequency أبطئ ← و damping

(c)

* هون كم بغير بال Frequency و بال Damping.

* بنقارن Qualitative وليس رياضيًا (quantitative) .

* كيف بيجي عليها بالامتحان : ما بعطي الرسمة بشكل مباشر ، بعطيك الصيغة (المعادلة) ل 2 systems
وبقلك مثلًا هل الهم نفس ال overshoot أو هيك ، يا بتحل بالمعادلات (quantitative) أو بترسم ال roots
بال Matlab وبتقارنهم qualitative .

System Response with Additional Poles and Zeros

- * We analyzed systems with one or two poles. The formulas describing *percent overshoot, settling time,* and *peak time* were derived only for a system with two complex poles and no zeros.
- * If a system has more than two poles or has zeros, we cannot use the formulas to calculate the performance specifications.
- * However, under certain conditions, a system with more than two poles or with zeros can be approximated as a second-order system that has just two complex **dominant poles**. -> 2 dominant Poles يكون في 3 Poles الفكرة هل بقدر ألاقي من هذول الـ
- * Only then, the formulas for percent overshoot, settling time, and peak time can be applied to these *higher-order systems* by using the *location of the dominant poles.*
- * *We will restrict our analysis to a system of three poles (3rd degree), or a 2nd degree system with a zero*

الـ Pole اللي انهمله
Ignore ساعتها بكون
ما إله هالتأثير الكبير.

%OS و Ts و Tp

* الـ extra Pole أضافت new exponintial
term للـ System (equation)

$$C(s)=\frac{A}{s}+\frac{B(s+\zeta\omega_n)+C\omega_d}{(s+\zeta\omega_n)^2+\omega_d^2}+\frac{D}{s+\alpha_r}$$

Steady-State

$$c(t)=Au(t)+e^{-\zeta\omega_n t}(B\cos\omega_d t+C\sin\omega_d t)+De^{-\alpha_r t}$$

مع الزمن بختفي الـ exponintial terms
وبظل الـ Steady-State.

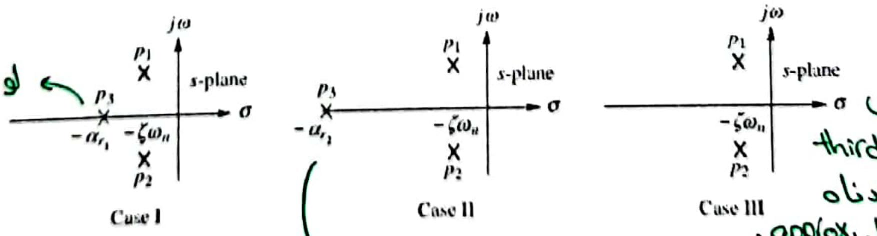
* هدفنا هون إنه نطلع
على هذا الـ System
اللي فيه extra poles and zeros
ونقدر لو بنفس بناء ع criteria
معينة؟ قول إنه بقدر أقرب
هذا الـ System وأتعامل معه كأنه 2nd order و Ts و %OS.
لـ الجواب بكون تقريبي.

* حندرس فقط حالات إنه يكون عندي 3 Poles أو 1 zero.
* بالـ Systems قبل ما كان في عنا zeros لإنه البسط كان Constant.

System Response with an Additional Real Pole I

- Suppose that in the denominator, we have a third real pole α_r, we will consider three cases:
  - Case I: $\alpha_r=\alpha_{r1}$ where it is not much larger than $\zeta\omega_n$
  - Case II: $\alpha_r=\alpha_{r2}$ where it is much larger than $\zeta\omega_n$
  - Case III: $\alpha_r=\infty$

* قيمة الـ Pole لازم يكون بعيد
لأقدر أهمله ignore ؟ حسب criteria معينة وعلاقة
رياضية معينة.



لو كان قريب كثير ما بقدر أهمله
ignore لإنه إله تأثير عال الـ Sys.

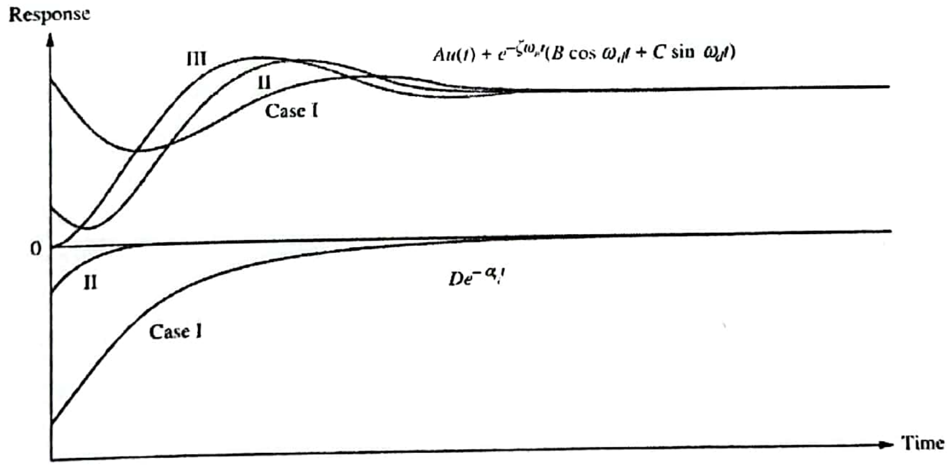
هذا الوضع الطبيعي
مناخذ وجود الـ third
Pole اللي كل ما بعدناه
كل ما كان أفضل للـ approx.

ممكن إنه نطلع 2 dominant
Poles ونعمل Ignore للـ Pole البعيدة واعتبره كـ 2nd order sys.

System Response with an Additional Real Pole II

- ✱ Case III → Pure Second Order System
- ✱ Case II → If $\alpha_r >> \zeta\omega_n$,the pure exponential will die out much more rapidly than the second-order underdamped step response. If the pure exponential term decays to an insignificant value *at the time of the first overshoot*, such parameters as percent overshoot, settling time, and peak time will be generated by the second-order underdamped step response component. Thus, the total response will approach that of a pure second-order system.
- ✱ If If $\alpha_r > \zeta\omega_n$ but **is not much greater** than $\zeta\omega_n$, the real pole's transient response will not decay to insignificance at the peak time or settling time generated by the second-order pair. In this case, the exponential decay is significant, and the system **cannot** be represented as a second-order system.
- ✱ ***How much farther from the dominant poles does the third pole have to be for its effect on the second-order response to be negligible?***
- ✱ Depends on the accuracy desired. The textbook book assumes that the exponential decay is negligible after five time constants. → ال Pole لازم يكون 5 مرات بعيد عن ال Dominant Pole لنقدر نهمله ignore .
- ✱ Thus, **if the real pole is five times farther to the left than the dominant poles**, we assume that the system is represented by its dominant second-order pair of poles
- ✱ Magnitude od the exponential term has less effect on the system as the pole moves further away to the left

System Response with an Additional Real Pole III



underdamped : أكبر من 0 وأقل من 1

Exercise

* $\omega_{n_1} = 4.954$, $\sigma_1 = \frac{9}{2} = 2$, $\zeta = \frac{\sigma_1}{\omega_{n_1}} = \frac{2}{4.954} = 0.4037$

- *▶ Find the step response of the following transfer functions and compare them. Which one of T_2 or T_3 can we apply the performance equations we learnt to?
- *▶ It is clear that T_1 is a second-order system while T_2 and T_3 are third order systems
- *▶ There is a frequency component ω_n = sqrt(24.542) = 4.954, therefore ζ = 4/(2* 4.954) = 0.4037, so T_1 is an underdamped system
- *▶ We notice that T_2 and T_3 have identical complex poles like T_1 but with extra real poles, one at -10 for T_2, and one at -3 for T_3, so expect the shapes of T_2 and T_3 to have similar shapes to T_1, but the effect of the dampening by the extra real pole will change them.
- *▶ We must know if the new shape will still resemble a second order system, so that we can approximate the third-order system as a second-order system and use the equations or not.

$$T_1(s) = \frac{24.542}{s^2 + 4s + 24.542} \rightarrow \text{Second order System}$$

$$T_2(s) = \frac{245.42}{(s+10)(s^2 + 4s + 24.542)}$$

$$T_3(s) = \frac{73.626}{(s+3)(s^2 + 4s + 24.542)}$$

3rd order

لـ أي واحد منهم بقدر إني أتعامل معه ك 2nd order زي T_1. كشان أقدر أتعامل معه ك Second order : لازم ال Pole الإضافي يكون 5 أضعاف بعيد عن ال Poles الأصليين عالأقل.

*سؤال امتحان : بطلبك مثلاً overshoot المعادلة ؛ قبل ما تبلش تحل تأكد إنها Second أو بنقدر نتعامل معها ك Second order غير هيك ما بصير تطبق المعادلات.

T_2 : x(-10) —— x(-3) x(-2) → $\frac{T_2\,(\text{Pole})}{T_1\,(\text{Pole})} = \frac{-10}{-2} = 5$ → بقدر أتعامل مع T_2 كأنها T_1 وكأنه ال Pole = -10 غير موجود.

T_3 : $\frac{T_3}{T_1} = \frac{-3}{-2} = 1.5$ → ما بعد 5 أضعاف فما بقدر أتعامل مع T_3 كأنه T_1

* نتائج المعادلات ع T_2 مش دقيقة 100 % زي T_1 ولكنها أقرب ما يكون الدقة :

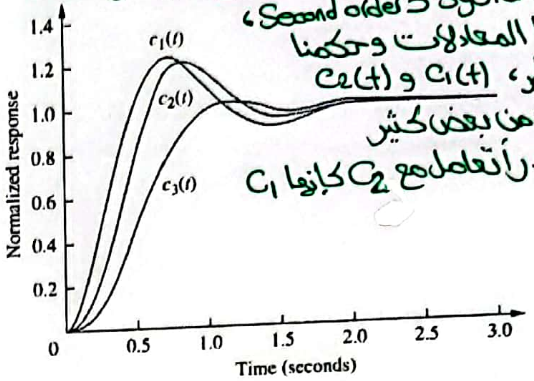
Solutions- Numerical

- * ▶ We should find the location of the complex poles , and see if the real pole is five times further away to the left
- * ▶ We know from the equations that the complex pole location will be at - $\zeta\,\omega_n$ = -σ/2 = - 0.4037 x 4.954 = -2
- * ▶ For T_2, α_r = -10 which five times larger than -2, so this system can be approximated as a 2nd order system. We can use the equations we learnt for settling time, overshoot, rise time, and peak time to ***APPROXIMATE*** the solutions.
- * ▶ For T_3, α_r = -3 which 1.5 times larger than -2, this system <u>**cannot**</u> be approximated as a 2nd order system. We cannot use the equations we learnt for settling time, overshoot, rise time, and peak time.

Solutions- Visual

- * Using plots, you can do an inverse Laplace transform on each of the transfer functions, and you get the time domain equation. Once you do that, plot the equations.
- * We can see how $c_2(t)$ is close to and resembles the shape of the second order function $c_1(t)$, while $c_3(t)$ differs in shape, and yields
more error if we use the performance equations on it.

*هون الحل بالنظر، بنحكم بالنظر إذا بنقدر نتعامل مع ال System كـ Second order ، رسمنا المعادلات وحكمنا بالنظر، c1(t) و c2(t) قراب من بعض كثير فبقدر أتعامل مع C2 كأنها C1

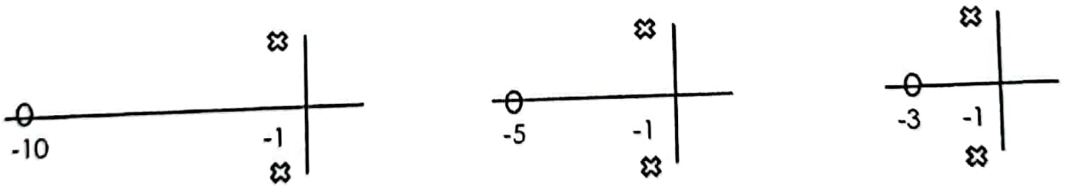


*ال Poles هم اللي بحددوا شكل ال response وال Speed تبعت الـ response وإذا ال response كان Stable و لا unstable ، فكل الشغلات الأساسية بالـ System موجودة يعني بال Poles فكنا نهمل ال Zeros .
*ال Zeros تأثيرها الوحيد عالقيمة النهائية للـ Steady-State (Forced response) (لوين ال sys بده يوصل) مش شكل وسرعة الوصول وال Stability .
↳ * لهيك كنا بنهتم بال Poles أكثر من ال Zeros

System Response with an Additional Real Zero I

البسط بدل ما يكون Constant حيصير فيه Zeros .

- * All the second order systems we have analyzed so far had no zeros, only poles. What if the transfer function had a zero as well?
- * We saw **(see slide 7)** that the zeros of a response affect the residue, or amplitude, of a response component but do not affect the nature of the response—exponential, damped sinusoid, and so on.
- * Suppose we have the system with the complex poles -1 ± 2.828j and we add each time a different zero at -3, -5, and -10



* نفس ال analysis اللي استخدمناها بال Poles حنستخدمها بال Zeros ، إنه لو كان ال Zero بعيد عن ال Pole عالأقل 5 أضعاف بنقدر نهمله ونتعامل مع ال System زي ما تعاملنا قبل (ما في Zeros بالبسط)

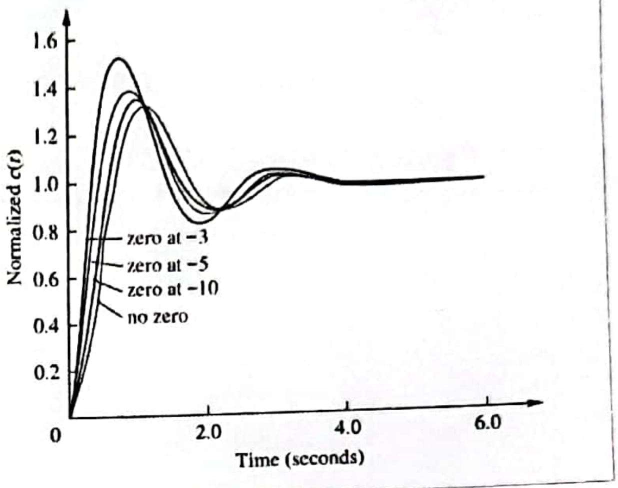
* Example : 1- $\frac{7(S-10)}{S^2+6S+7}$

↲ هذا اللي بنقدر نلغي فيه الـ zero ونتعامل معه زي قبل لأنه الـ zero هون كثير بعيد .

2- $\frac{7(S-0.5)}{S^2+6S+7}$ ⤷ هون الـ zero قريب صعب نهمله .

System Response with an Additional Real Zero II

* ▶ We can see that the closer the zero is to the dominant poles, the greater its effect on the transient response. As the zero moves away from the dominant poles, the response approaches that of the two-pole system.



*إضافة الـ zero إلى الـ Transfer function
معناه كأنه عندي .Transfer funct كمشتقة مضاف
لل function نفسه مضروب بـ gain معين

*ضرب أي .Transfer funct بـ S

*ضرب أي .Transfer funct بـ Constant (قيمة)
معناها gain (كأنه ال
Signal دخلتها s Amplifier) ،
وممكن ماتكون منو gain تكون
loss (تكون القيمة مثلاً 0.5
و 0.2 وهيك) reduction

$$* \frac{\overset{T(s)}{\boxed{90}}\,(s-2)}{s^2+2s+90} = \frac{90}{s^2+2s+90}(s) - 2\left(\frac{90}{s^2+2s+90}\right)$$

* Laplace { T`(s) } = ST(s)

Another way to think of systems with a zero

- Suppose we have a transfer function of a 2nd order function T(s) = $\frac{90}{s^2+2s+90}$, if T(s) were to have a zero, it would be for example in the shape of T(s) = $\frac{90\,(s+a)}{s^2+2s+90}$ where a is any number. Basically, we multiplied T(s) by (s+a)
- If we think of it , by using the laws of distributions, the version of a second order system with a zero is sT(s) + aT(s)
- sT(s) means the first derivative of T(s), while aT(s) is a scaled version of T(s), so in other words, adding a zero to a second order system basically has the effect of adding the derivative of the system to a scaled version of itself. For a unit step response, usually this derivative is positive at the beginning.
- When a is large, the scaled version overcomes the derivative part, and the derivative part become negligible, this is why we noticed that as the zero value increases, it approximates a second-order system
- When a is small, the derivative effects are more noticeable

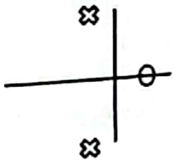
* شكل هذه المعادلة، إنه كونه المشتقة بالعادة بالبداية
بتكون موجبة ، إذا صدفت وكانت قيم α سالبة بتكون البداية سالبة (Input موجب والبداية سالبة)
هنا معناه بال Control ، أنا بسوق سيارة وعطيته order يروح عاليمين راحت زاحت السيارة عالشمال .

* فلو كان الـ Zero اللي انضاف بالبسط سالب بدل تدريبالك ، إنه هذا الـ System عارض الواقع بيعطي عكس اتجاه
الـ order المطلوب .

وهذا بالغالب مذموم.

*الـ control sys اللي بنعطيه order وبروح بعملك عكسه بالبداية حين مايرجع الـ sys يصير stable

Nonminimum-phase system



- What if the zero **a** is on the right-hand plane?
- We see that the derivative term, which is typically positive initially, will be of opposite sign from the scaled response term.
- Notice that the response begins to turn toward the negative direction even though the final value is positive. A system that exhibits this phenomenon is known as a **nonminimum-phase system**.
- If a motorcycle or airplane was a nonminimum-phase system, it would initially veer left when commanded to steer right.

كشان تصفى المعادلة constant بالبسط ومعادلة تربيعية بالمقام وتصير نتعامل معها زي قبل.

وحتى لو ما كانوا متطابقين 100% بس قراب هل بقدر أكنسلهم ؟

كنا قبل نعتبر لو في (S+2) مثلاً بالبسط والمقام فبكون Pole و zero الـ -2 ، السؤال هون هل بقدر أكنسل (S+2) من البسط والمقام ؟

Zero – Pole Cancellation I

- Suppose we have systems with many poles and zeros (high-order systems). Can we cancel zeros and poles so that we reach a system that approximates a 2nd order system that we already know how to analyze?
- Can we cancel the zero (s + 4) and the pole (s + 3.5) which are the closest to each other, so that we end up with a second order system?

*هل بقدر أشطب الـ (S+4) مع الـ (S+3.5) ؟

$$\underset{\text{unit Step funct.}}{C_1(s)} = \frac{\overset{\text{Constant}}{26.25}\overset{\text{zero}}{(s+4)}}{\underbrace{s(s+3.5)(s+5)(s+6)}_{\text{Poles}}}$$

أولاً لازم نحطها بالـ Partial fraction

Syms s C(s)

C(s)= 26.25*(S+4) / (S*(S+3.5)*(S+5)*(S+6))

Partfrac (C(s))

لازم نحطها بالـ Matlab

- We must do a partial fractions expansions and evaluate the residue, we notice that the residue of (s + 3.5) term is so close to the others, therefore, no cancellation.

$$C_1(s) = \underset{\text{unit step}}{\frac{1}{s}} - \frac{3.5}{s+5} + \frac{3.5}{s+6} - \frac{1}{\boxed{s+3.5}}$$

بالـ Matlab مثلاً بعطيك باها $\frac{7}{2(S+5)}$ فانت بس بسطها واقسم الـ 7/2 وزبط المعادلة.

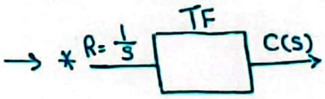
قريب عالـ 3.5 و 3.5 معناه الوزن تبع الـ 1 عالي فما بقدر أشطب الـ (S+3.5) مع الـ (S+4) اللي فوق فما حقدر أطبق المعادلات اللي تعلمناها عليه حيضله من الدرجة الثالثة وفيه zero بالبسط.

Zero – Pole Cancellation II

رياضيًا ما بصير نشطبهم احنا بنشطب تقريبًا

- Can we cancel the zero (s + 4) and the pole (s + 4.01) which are the closest to each other, so that we end up with a second order system?

$$C_2(s) = \frac{26.25(s+4)}{s(s+4.01)(s+5)(s+6)}$$



- We must do a partial fractions expansions and evaluate the residue, we notice that the residue of (s + 4.01) term is so far away from the others, therefore cancellation is possible

e^{-6}, e^{-5} → as time goes to ∞ (∞) → رح يروحوا

$$C_2(s) = \frac{0.87}{s} \overbrace{- \frac{5.3}{s+5} + \frac{4.4}{s+6} + \frac{0.033}{\boxed{s+4.01}}}^{\text{Natural response}}$$

بعيد عن ال 5.3
وأقرب لل 4.4، فبقسم لل 4.4
عال 0.033، إذا كانت حاصل
القسمة فوق ال 50 فبقدر ألغي (S+4) مع (S+4.01) فهيك بنرجع معادلة تربيعية بقدر أطبق عليها المعادلات.

Steady-state ، دائمًا البسط اللي مع ال $\frac{1}{s}$ هو اللي
بكون ال Steady-state لأنه الباقي أصلهم exp
فرح يروحوا مع الزمن وبظل بس ال 0.87.

$$C_2(s) \approx \frac{0.87}{s} - \frac{5.3}{s+5} + \frac{4.4}{s+6}$$

References

- The material in these slides are based on: Control Systems Engineering, Norman S. Nise, 7th Edition (2014), John Wiley And Sons
- **Chapter 4 – Time Response**
  - Sections *4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 and 4.8* (Students kindly note that some sections involve math derivations that we did not cover in class

* بعدنا ال chapter ما حنحل ال Stability زي المثال الي تحت، حنتعرف ع طرق
رياضية بتعرفني شو المواصفات الي أحطها عال Component الجديدة لحتى يبقى ال System Stable.

* الوظيفة الأساسية: بعدنا ال chapter لغايات ال design، إنه هل لو ضفت Component
جديدة عال System (like: Amplifier, etc...) هل سيبقى ال System stable ولا لأ؟

System Stability

Dr. ASHRAF E. SUYYAGH

THE UNIVERSITY OF JORDAN
DEPARTMENT OF COMPUTER ENGINEERING
FALL 2022

*Suppose we have this system: $\frac{s}{s^7+5s^2+3s+2}$, هل هو stable أو لأ؟

عدد ال roots = 7، حسب أكبر قوة للمقام. ← لـ ال stable بتحدد حسب المقام (Poles) واللي هي ال roots.

بنحول المقام ل Numirical form ←

→ In Matlab: * بنرتب الحدود * Vector: [1,0,0,0,0,5,3,2] * roots (vector)

roots كمثال لل ← * 6+0.000j → انتبه هنا هو Complex بس ال Matlab بطلعهم.

اللي بتطلع. Real ← 0+7.2j

Img ← 2+j
2-j → complex

* ال roots يا عال Real-axis يا عال Img-axi يا Complex فيها real part و Img- Part.

→ S-Domain: (left half Plane / Right half plane)

LHP Img RHP
x x x
x — Real
x x x

اللي عال RHP هم اللي unstable ↓ مجرد ما في عال Poles RHP فال Sys بكون unstable.

Definitions

2

① ← شكل الانتقال من نقطة إلى أخرى.
② أهم وحدة

- In Chapter 1, we saw that three requirements play a major role in the design of a control system: transient response, stability, and ③ steady-state errors. ← لما يوصل ال Steady-state قديه بكون بعيد عن القيمة النهائية.
- Stability is the most important system specification. If a system is unstable, transient response and steady-state errors are moot points. An unstable system cannot be designed for a specific transient response or steady-state error requirement.
- We can control the output of a system if the steady-state response consists of only the forced response. But the total response of a system is the sum of the forced and natural responses $c(t) = c_{forced}(t) + c_{natural}(t)$ input ← ; → خواص ال Sys لأول ال output. → مش موصلني لل steady-state بس هو مخرب ال System ↓ ما حقق المواصفات المطلوبة بس ما خرب ال System.
- Using these concepts, we present the following definitions of stability, instability, and marginal stability:
  - ✓ A linear, time-invariant system is stable if the natural response approaches zero as time approaches infinity.
  - ✓ A linear, time-invariant system is unstable if the natural response grows without bound as time approaches infinity.
  - ✓ A linear, time-invariant system is marginally stable if the natural response neither decays nor grows but remains constant or oscillates as time approaches infinity.
- Thus, the definition of stability implies that only the forced response remains as the natural response approaches zero

هاي التعاريف مربوطة بخواص ال System (natural response) بس
مش رابطها بال input، بلكي ال Sys نخله Input عالي جدًا فوق
مستوى فهوا مش مربوط بخواصه، لهيك أكاد صياغة هاي التعريفات.

"Bounded Input / Bounded output"

Definitions II : BIBO Stability

- The previous definition only takes into account the natural response and not the total response. What if the input (forced response) is from the beginning *unbounded*? BIBO Stability definition means:
  1. A system is stable if every bounded input yields a bounded output.
  2. 2. A system is unstable if any bounded input yields an unbounded output
- Under this definition, marginal stability, means that the system is stable for some bounded inputs and unstable for others.
- Physically, an unstable system whose natural response grows without bound can cause damage to the system, to adjacent property, or to human life. From the perspective of the time response plot of a physical system, instability is displayed by transients that grow without bound and, consequently, a total response that does not approach a steady-state value or other forced response.

*Poles in LHP → Stable.
* Poles in Img-axis → undamped → Marginally Stable.
* Poles in RHP → unstable.

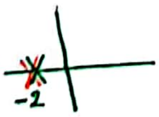
How do we determine if a system is stable?

- Stable systems have closed-loop transfer functions with poles only in the left half-plane (LHP).
- Unstable systems have closed-loop transfer functions with at least one pole in the right half-plane and/or poles of multiplicity greater than 1 on the imaginary axis.
  - Poles in the right half-plane yield either pure increasing exponentials or exponentially increasing sinusoidal natural responses. These natural responses approach infinity as time approaches infinity.
  - Poles of multiplicity greater than 1 on the imaginary axis lead to the sum of responses of the form $At^n \cos(\omega t + \varphi)$, where n = 1, 2, . . ., which also approaches infinity as time approaches infinity
- Marginally stable systems have closed-loop transfer functions with only imaginary axis poles of multiplicity 1 and poles in the left half-plane

* Multiplicity n :
n Poles (نفس بعض)
(راكبين فوق بعض)

2Poles منطبقين عبعض
(نفس بعض)

* $\frac{}{(S+2)(S+2)}$ →



→

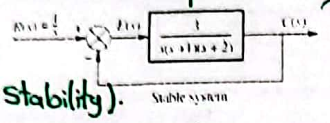
* $e^{-2t} + t e^{-2t}$ → هلقد ال System Stable or not →

بنحتاج $\lim_{t \to \infty}$ ، بعض الكتب نظريًا بتحكي بما في t فـ unstable وفي كتب لا بتحللوا بعدين بتحكم.

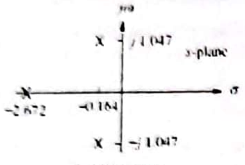
Examples

لأعرف إنه Stable أو لا لازم نطلع أول الـ Transfer fonct. لكل الـ System (overall system stability)

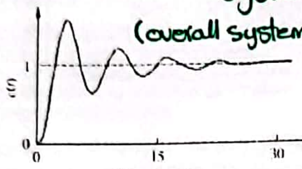
لو طلعت الـ roots بس لهاي (0,-1,-2) بتكون بتحسب (subsystem stability).



Stable system



Stable system's closed-loop poles (not to scale)

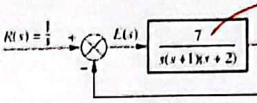


Time (seconds)

underdamped، قيمة ζ بين 0 و 1

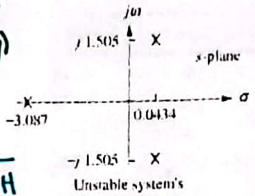
$$\frac{G}{1+GH}$$

$\hookrightarrow$ = 1

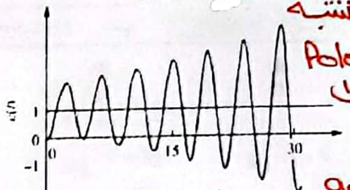


Unstable system

تغير بسيط بالـ gain خلاها للـ System unstable فإذا مو دايما بس ننتبه للـ Poles وننسى تأثير الـ gain.



Unstable system's closed-loop poles (not to scale)



Time (seconds)

$$\hookrightarrow \frac{G}{1+G} = \frac{\frac{3}{S(S+1)(S+2)}}{1+\frac{3}{S(S+1)(S+2)}} = \frac{3}{S(S+1)(S+2)+3}$$

هون بتطلع الـ roots وبتحدد إذا الـ System Stable or not.

* البسط لو كان بس Constant بكون كأنه gain (يعني زي Amplifier) :

$$-\boxed{3}-\boxed{\frac{1}{S(S+1)(S+2)}}-$$

قوتها بالـ design وضعفها بالـ analysis، بتطلع أماكن الـ Poles مش قيمهم. بتساعدني أعرف إيش القيمة المناسبة للـ gain (Amplifier) (أقصى قيمة للـ gain) اللي بقدر أحطها ويضل الـ System Stable.

→ بتطلع فقط أماكن الـ roots مش قيمهم

Routh-Hurwitz Criterion for Stability

3rd or 2nd لو انطلبنا نحل أمثلة رياضية عليها بتكون بسيطة و ← table الأسئلة الأكثر تحليل للـ

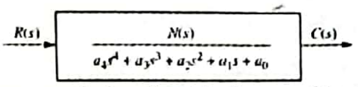
* ▸ Routh-Hurwitz Criterion is a method that yields stability information without the need to solve for the closed-loop system poles. Using this method, we can tell how many closed-loop system poles are in the left half-plane, in the right half-plane, and on the jω-axis. (Notice that we say *how many*, not *where*.) We can find the number of poles in each section of the s-plane, but we cannot find their coordinates.

* ▸ The method requires two steps: (1) Generate a data table called a Routh table and (2) interpret the Routh table to tell how many closed-loop system poles are in the left half-plane, the right half-plane, and on the jω –axis

* ▸ *If we can use computers to easily get the roots (poles), why would we need this method?* The power of Routh-Hurwitz criterion does not lie in the analysis stage but in the design stage, as it can let us know how can we change the range of a parameter and the system remains stable (check last slide).

Generating a Basic Routh Table

* ▸ Suppose we have the following equivalent closed-loop transfer function:



* ▸ Label the rows with powers of s from the highest power of the denominator of the closed-loop transfer function to s^0 رتب القوى أول اشي

* ▸ Next start with the coefficient of the highest power of s in the denominator and list, horizontally in the first row, every other coefficient. ← بنفصل الزوجي عن الفردي.

* ▸ In the second row, list horizontally, starting with the next highest power of s, every coefficient that was skipped in the first row.

* ▸ Each entry is a negative determinant of entries in the previous two rows divided by the entry in the first column directly above the calculated row.

* ▸ The left-hand column of the determinant is always the first column of the previous two rows, and the right-hand column is the elements of the column above and to the right.

* ▸ We can multiply/divide any row, if needed, by a positive constant without changing the analysis

TABLE 6.1 Initial layout for Routh table

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2			
s^1			
s^0			

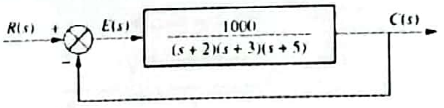
TABLE 6.2 Completed Routh table → الجدول مُعطى

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	$\frac{-\begin{vmatrix} a_4 & a_2 \\ a_3 & a_1 \end{vmatrix}}{a_3} = b_1$	$\frac{-\begin{vmatrix} a_4 & a_0 \\ a_3 & 0 \end{vmatrix}}{a_3} = b_2$	$\frac{-\begin{vmatrix} a_4 & 0 \\ a_3 & 0 \end{vmatrix}}{a_3} = 0$
s^1	$\frac{-\begin{vmatrix} a_3 & a_1 \\ b_1 & b_2 \end{vmatrix}}{b_1} = c_1$	$\frac{-\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$	$\frac{-\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$
s^0	$\frac{-\begin{vmatrix} b_1 & b_2 \\ c_1 & 0 \end{vmatrix}}{c_1} = d_1$	$\frac{-\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$	$\frac{-\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$

* المتحرك الوحيد هو الجزء اليمين من كل Metrix.

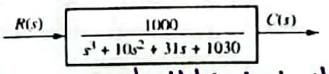
* مهم نعرف نحلل ال tables *

Example I



▸ Make the Routh table for the system shown in the right.

▸ First, we need to find the equivalent transfer function for the entire system, then fill in the table



قسم على رقم موجب " مابحيس سالب "
لحتى يبسط الأرقام ويسهل الحل.

s^3	1	31	0
s^2	~~10~~ 1	~~1030~~ 103	0
s^1	$\frac{-\begin{vmatrix} 1 & 31 \\ 1 & 103 \end{vmatrix}}{1} = -72$	$\frac{-\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}}{1} = 0$	$\frac{-\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}}{1} = 0$
s^0	$\frac{-\begin{vmatrix} 1 & 103 \\ -72 & 0 \end{vmatrix}}{-72} = 103$	$\frac{-\begin{vmatrix} 1 & 0 \\ -72 & 0 \end{vmatrix}}{-72} = 0$	$\frac{-\begin{vmatrix} 1 & 0 \\ -72 & 0 \end{vmatrix}}{-72} = 0$

$1*103 - 1*31 = 72*-1 = \frac{-72}{1} = -72$

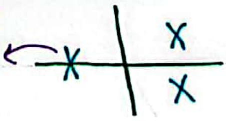
1	31	0
1	103	0
−72	0	0
103	0	0

من خلاله بنعرف إذا ال System stable أو لا.

مابقدر أعرف إذا هم Complex أو real

→ qaulititative ما بعرف القيم بعرف الأماكن

يعني أول عمود ، إذا كانت جميع القيم موجبة فمعناه جميع ال roots بال LHP يعني العا System stable ، لو في قيم سالبة فعدد مرات تغير الإشارة هو كدد الجذور الموجودة بالجزء اليمين (RHP) ، (ما بعرف القيم بس الأماكن) ← Sys unstable



← لأنه المعادلة تكعيبية وعرفنا 2 عاليمين فضل واحد عالشمال.

هاي الطريقة اخترعت عشان زمان ما كانوا بقدروا
يحلوها بأجهزة والآت حاسبة وهكذا.

Routh-Hurwitz Basic Interpretation

- The number of roots of the polynomial that are in the right half-plane is equal to the number of sign changes in the first column.
- If an entire row of zeros appear in the Routh Table, then we have jω poles
- In the previous example, we have a cubic equation, so we expect three roots.
  - We have two sign changes in the first column, this means two poles in the RHP → System Unstable
  - We have 3 – 2 = 1 root in the LHP
  - No row is entirely zero, no pure jw poles

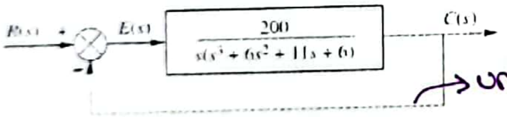
```
>> roots([1 10 31 1030])

ans =

 -13.4136 + 0.0000i
   1.7068 + 8.5950i
   1.7068 - 8.5950i
```

ال qualititative طلعت ←
مطابقة لنتائج ال quantitative .

لو بدنا نربطها بالواقع بنقدر نحكي إنه هذا عبارة عن إشارة بتجيني اتحكم فيها بال Motor
اللي اله معادلة Amplifier وبرجع ب feedback عبارة عن unity gain وبحسب ال error
بشكل مستمر.

Example 2

- Find the number of poles in the left half-plane, the right half-plane, and on the jω -axis for the system



→ unity gain

- First, we find the equivalent transfer function then complete the Routh Table

$$T(s) = \frac{200}{s^4 + 6s^3 + 11s^2 + 6s + 200}$$

* - At the s^1 row there is a negative coefficient; thus, there are two sign changes. The system is unstable, since it has two right–halfplane poles and two left–half-plane poles. The system cannot have jω poles since a row of zeros did not appear in the Routh table

System unstable ← 2 Poles in Right half Plane ← 2 changes

قسم ع أرقام موجبة
للتبسيط.

s^4		1	11	200
s^3	~~6~~	1	~~6~~ 1	
s^2	~~10~~	1	~~200~~ 20	
s^1		−19		
s^0		20		

الأماكن
الفاضية
= 0

```
>> roots([1 6 11 6 200])

ans =

  -4.2760 - 2.5409i
  -4.2760 - 2.5409i
   1.2760 - 2.5409i
   1.2760 - 2.5409i
```

المشكلة بالسطر الثالث طلع معنا 0 فاللي بعده حيكون بنقسم ع 0 ←
وهنا مابصير.

11

Routh-Hurwitz Criterion: Special Cases I

- Determine the stability of the closed-loop transfer function $T(s) = \frac{10}{s^5 + 2s^4 + 3s^3 + 6s^2 + 5s + 3}$

Second assumption ↑

.limit تنشبه مبدأ الـ ← First assumption ←

s^5	1	3	5
s^4	2	6	3
s^3	~~0~~ ϵ	$\frac{7}{2}$	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

Label	First column	$\epsilon = +$	$\epsilon = -$
s^5	1	+	+
s^4	2	+	+
s^3	~~0~~ ϵ	+	−
s^2	بطلع أقل من 7 ← $\frac{6\epsilon - 7}{\epsilon}$ $\frac{(-)(-)}{(+)(-)}$	−	+
s^1	$\frac{(-)\ 42\epsilon - 49 - 6\epsilon^2}{(-)\ (-)\ 12\epsilon - 14}$ (−) (+)	−	+
s^0	3	+	+

→ المفروض عدد الـ changes تكون نفسها بالعمودين لإنه نفس الـ System. (عندي تغيرين بالإشارة فإذاً في 2 poles بالـ RHP فالـ System بكون unstable) ← الـ 3 Poles الباقين بالـ LHP

* إذا طلعنا أي 0 بنبدله بـ ε (Epsilon) تعبر عن رقم صغير جدًا قريب من الصفر بس مش صفر، حطيناه عشان ما يخرب الـ System أو اضطر أقسم ع Zero.

* ε : أصغر رقم ممكن يمثله الـ Computer. الـ Computers بتعامل معها على انها Constant.

* الـ Computers غير قادرة على تمثيل الأرقام بشكل دقيق.

* قد تكون ε قيمة موجبة أو سالبة فلجنا حنفترض الاحتمالين ونعرف الإشارات للعمود الأول.

* هون ما حنتعامل معها بأرقام بنستخدمها عشان نعرف الإشارات فقط.

12

Routh-Hurwitz Criterion: Special Cases II

- Determine the number of right-half-plane poles in the closed-loop transfer function $T(s) = \frac{10}{s^5 + 7s^4 + 6s^3 + 42s^2 + 8s + 56}$
- We stop at the third row, since the entire row consists of zeros
- We return to the row immediately above the row of zeros and form an auxiliary polynomial, using the entries in that row as coefficients. The polynomial will start with the power of s in the label column and continue by skipping every other power of s. $P(s) = s^4 + 6s^2 + 8$ → كتبنا معادلة بناء عالسطر الثاني
- Next, we differentiate the polynomial with respect to s and obtain واشتقيناها
$$\frac{dP(s)}{ds} = 4s^3 + 12s + 0$$
- We use the coefficients of the differential to replace the row of zeros → أرقام المشتقة هي اللي بنعبيها بالسطر تاع s^3 وبكمل حل.
- All all entries in the first column are positive. Hence, there are no right–half-plane poles. We need to understand the concept of symmetry before moving forward in our analysis.

s^5	1	6	8
s^4	~~7~~ 1	~~42~~ 6	~~56~~ 8
s^3	~~0~~ ~~4~~ 1	~~0~~ ~~12~~ 3	~~0~~ ~~0~~ 0
s^2	3	8	0
s^1	$\frac{1}{3}$	0	0
s^0	8	0	0

السطر الأول بساوي السطر الثاني ↓ فهيك كل المحددات حيكون الناتج فيها 0 (اللي بسطر الـ S^3)

↓ الخط الأحمر اللي بنفصل فيه التحليل هو الخط اللي فوق المشتقة (S^4)

* طريقة التحليل بطلت زي أول لإنه تغيرت الـ Math (بطلنا نتطلع بس ع أول عمود)

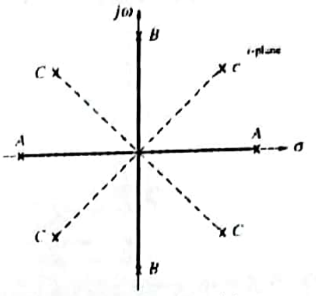
* سيتم تقسيم التحليل إلى قسمين قسم ما قبل الانشقاق وقسم ما بعده، التحليل أسفل الانشقاق بيشمل قواعد هاي الـ Case، أما أعلى الانشقاق بكون التحليل زي قبل.

* الفصل بين الأعلى والأسفل بناء عالمربع الأحمر (فوق سطر الانشقاق).

1 ؛ أسفل الاشتقاق ؛ مافي تغير
بالإشارة ← 4 poles in Img. axis

حل الـ Case السابقة بناء على التحليل ؛ 2 ؛ أعلى الاشتقاق ؛ مافي تغير
بالإشارة ← 1 Pole in LHP

More about Rows of Zeros



A: Real and symmetrical about the origin
B: Imaginary and symmetrical about the origin
C: Quadrantal and symmetrical about the origin

- Let us look further into the case that yields an entire row of zeros. An entire row of zeros will appear in the Routh table when a purely even or purely odd polynomial is a factor of the original polynomial. For example, $s^4 + 5s^2 + 7$ is an even polynomial; it has only even powers of s.
- Even polynomials only have roots that are symmetrical about the origin. This symmetry can occur under three conditions of root position:
  1. The roots are symmetrical and real,
  2. The roots are symmetrical and imaginary, or
  3. The roots are quadrantal.

← تحصل فقط بالـ cases اللي فيها سطر كامل بساوي ٥ وحليناه بطريقة المشتقة.

* الـ Symmetry ؛ ← * لو كانوا كلهم موجب بكونوا عالـ Imaginary part.

1- اذا مافي ولا تغير بالإشارة (كل العمود موجب) فالـ Poles بكونوا عالـ Imaginary بالزبط لا يمين ولا يسار.

↳ Something like this ؛ 0+2j
-0-3j

2- لو في تغيرات بالإشارة فعدد التغيرات بكون عدد الـ Poles اللي بالـ RHP ولكن بسبب خاصية الـ Symmetry فبكون في نفسهم عالـ LHP أيضًا.

Example I

- For the transfer function $T(s) = \dfrac{20}{s^8 + s^7 + 12s^6 + 22s^5 + 39s^4 + 59s^3 + 48s^2 + 38s + 20}$ how many poles are in the right half-plane, in the left half-plane, and on the jω -axis
- No sign changes exist from the s^4 row down to the s^0 row. Thus, the even polynomial does not have right–half-plane poles. Since there are no right–half-plane poles, no left–half-plane poles are present because of the requirement for symmetry. Hence, the even polynomial, must have all four of its poles on the jω -axis.
- Since the polynomial is of degree 8, four roots remain. From the s^8 row down to the s^4 row, there are two sign changes, this means, two poles in the RHP. The two remaining poles will therefore be in the LHP.

↳ بسبب خاصية الـ Symmetry

s^8	1	12	39	48	20
s^7	1	22	59	38	0
s^6	~~−10~~ − 1	~~−20~~ − 2	~~10~~ 1	~~20~~ 2	0
s^5	~~20~~ 1	~~60~~ 3	~~40~~ 2	0	0
s^4	1	3	2	0	0
s^3	~~4~~ ~~4~~ 2	~~0~~ ~~6~~ 3	~~0~~ ~~0~~ 0	0	0
s^2	~~$\frac{3}{2}$~~ 3	~~2~~ 4	0	0	0
s^1	$\frac{1}{3}$	0	0	0	0
s^0	4	0	0	0	0

سطر الاشتقاق

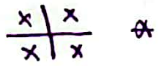
↳ بالامتحان بعطيك الـ table جاهز انت لازم تحلله وتكتشف هو أي Case وبناءً عليها تكمل تحليلك.

Example I

- For the transfer function $T(s) = \dfrac{20}{s^8 + s^7 + 12s^6 + 22s^5 + 39s^4 + 59s^3 + 48s^2 + 38s + 20}$ | how many poles are in the right half-plane, in the left half-plane, and on the jω -axis
- No sign changes exist from the s^4 row down to the s^0 row. Thus, the even polynomial does not have right–half-plane poles. Since there are no right–half-plane poles, no left–half-plane poles are present because of the requirement for symmetry. Hence, the even polynomial, must have all four of its poles on the jω -axis.
- Since the polynomial is of degree 8, four roots remain. From the s^8 row down to the s^4 row, there are two sign changes, this means, two poles in the RHP. The two remaining poles will therefore be in the LHP.

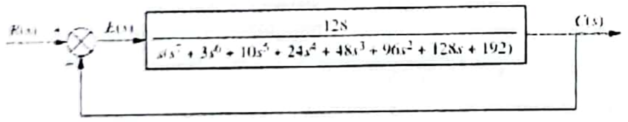
```
>> roots([1 1 12 22 39 59 48 38 20])

ans =

   0.5000 - 3.1225i ] → RHP
   0.5000 - 3.1225i ]
   0.0000 - 1.4142i *
   0.0000 - 1.4142i *
  -1.0000 - 0.0000i ] → LHP      → Imaginary
  -1.0000 - 0.0000i ]
  -0.0000 - 1.0000i *
  -0.0000 - 1.0000i *
```

* سؤال امتحان : بجيلك الجدول وبطلب تطلعله أماكن ال Poles وبعطيك 4 أشكال لأماكن ال Poles مثلاً وانت اختار الشكل الصح .



Example II

- Find the number of poles in the left half-plane, the right half-plane, and on the jω -axis for the system


R(s)
+
E(s)
$\dfrac{128}{s(s^7 + 3s^6 + 10s^5 + 24s^4 + 48s^3 + 96s^2 + 128s + 192)}$
C(s)
-

The closed Loop transfer function for the system is

$$T(s) = \frac{128}{s^8 + 3s^7 + 10s^6 + 24s^5 + 48s^4 + 96s^3 + 128s^2 + 192s + 128}$$

Example II - Continued

- Two sign changes exist from the s^6 row down to the s^0 row. Thus, the even polynomial has two right–half-plane poles. Since there are two right–half-plane poles, there are two left–half-plane due to symmetry.
- Since the degree of the even polynomial Is 6, and we have found four poles on the left and right, this means we have two poles remaining on the jω axis
- Since the degree of the overall polynomial is 8, and we have found six, we look at the sign changes from s^8 row down to the s^6 row, since there are no sign changes, this means we have all remaining poles in the LHP

s^8	1	10	48	128	128
s^7	~~3~~ 1	~~24~~ 8	~~96~~ 32	~~192~~ 64	
s^6	~~2~~ 1	~~16~~ 8	~~64~~ 32	~~128~~ 64	
s^5	~~0~~ ~~6~~ 3	~~0~~ ~~32~~ 16	~~0~~ ~~64~~ 32	~~0~~ ~~0~~ 0	
s^4	$\cancel{\frac{8}{3}}$ 1	$\cancel{\frac{64}{3}}$ 8	~~64~~ 24		
s^3	~~-8~~ -1	~~-40~~ -5			
s^2	~~3~~ 1	~~24~~ 8			
s^1	3				
s^0	8				

هون سطر المشتقة

because we have 2 changes

```
>> roots([1 3 10 24 48 96 128 192 128])

ans =
   1.0000 + 1.7321i        -1.0000 + 1.7321i
   1.0000 - 1.7321i        -1.0000 - 1.7321i
   0.0000 + 2.0000i        -2.0000 + 0.0000i
   0.0000 - 2.0000i        -1.0000 + 0.0000i
```

RHP ← [1.0000 + 1.7321i, 1.0000 - 1.7321i

Imaginary ← [0.0000 + 2.0000i, 0.0000 - 2.0000i

LHP ← [-1.0000 + 1.7321i, -1.0000 - 1.7321i, -2.0000 + 0.0000i, -1.0000 + 0.0000i

في منهم real وفي منهم Complex.

2 RHP
2 LHP → بسبب ال Symmetry
2 Imaginary] → بسبب ال symmetric

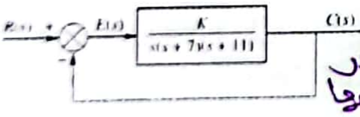
2 LHP] → للي فوق

لإنه طلعت 4 وبكون ضايلي 2 لإنه الجزء اللي تحت لازم يطلع 6 عدد ال Poles، فاللي ضايل بكون أكيد عال Imaginary.

Stability Design via Routh-Hurwitz Criterion

Amplifier → مضروب بالمعادلة لزيادة القوة → Increase current or voltage.

- Find the range of gain, K, for the system shown in the figure that will cause the system to be stable, unstable, and marginally stable. Assume K > 0.



- The closed-loop transfer function is

$$T(s) = \frac{K}{s^3 + 18s^2 + 77s + K}$$

محور K فيه Transfer funct. (لأنها نقطة تقاطع مع حالك)

استخدام ال Routh-Hurwitz بخليك تعرف range القيم ل K اللي بنقدر نستخدمها وما تخرب ال System

s^3	1	77
s^2	18	K
s^1	$\frac{1386 - K}{18}$	
s^0	K	

طلعوا من القوانين

Since K is assumed positive, we see that all elements in the first column are always positive except the s^1 row. This entry can be positive, zero, or negative, depending upon the value of K. $\boxed{\text{If } K < 1386}$ all terms in the first column will be positive, and since there are no sign changes, the system will have three poles in the left half-plane and be **stable**.

بتخلي قيم أول عمود موجبة فال System stable

$\boxed{\text{If } K > 1386}$ the s^1 term in the first column is negative. There are two sign changes, indicating that the system has two right–half-plane poles and one left– half-plane pole, which makes the system **unstable**

* بالماتلاب ممكن يكون في tools بتوجهك range قيم ال K المحتملة لبظل ال System stable دون الحاجة لحل قوانين وحل رياضي ، بالصور والرسم.

Stability Design via Routh-Hurwitz Criterion

- If $\boxed{K = 1386}$ we have an entire row of zeros, which could signify jω poles.
- Returning to the s² row and replacing K with 1386, we form the even polynomial

$$P(s) = 18s^2 + 1386$$

- Differentiating with respect to s, we have

$$\frac{dP(s)}{ds} = 36s + 0$$

- Replacing the row of zeros with the coefficients of the differential, we obtain the new Routh- Hurwitz table for the case of K = 1386

s^3	1	77
s^2	18	1386
s^1	~~0~~ 36	
s^0	1386	

هون بنحط الخط الأحمر (فوق الخط اللي حطينا فيه المشتقة)

Since there are no sign changes from the even polynomial (s^2 row) down to the bottom of the table, the even polynomial has its two roots on the jω -axis of unit multiplicity. Since there are no sign changes above the even polynomial, the remaining root is in the left half-plane. Therefore, the system is marginally stable.

undamped (Marginally stable)

Step 1 : أعمل Vector لـ K بـ for loop.

how we solve these Problems in Matlab :

```
for k = 1:2000
 R = roots ([1  18  77  K])
* my_roots = real(R)
  if max(my_roots) ≥ 0
    disp ("System unstable")
    K; break;
  end
```

لأنه الـ real هو اللي يعني لأعرف الـ Poles يمين ولا شمال.

قيمة الـ k اللي الـ sys صار عندها unstable.

* لو رسمت الـ Step response لقيم مختلفة لـ k صح بس بياخذ كثير وقت.

* لنفرض عند النجمة كانت الأجوبة :

−0.5 + 2j
−0.5 − 2j
−2
−1
2

اللي موجب بالـ real يكون عاليمين فبكون unstable

فبنوقف بقيم الـ k الـ Stable → unstable.
لغاية (k-1) لأنه عند k صار unstable

References

- The material in these slides are based on: Control Systems Engineering, Norman S. Nise, 7th Edition (2014), John Wiley And Sons
- **Chapter 6 – Stability**
  - Sections *6.1, 6.2, 6.3, 6.4*, (Students kindly note that some sections involve math derivations that we did not cover in class)

Steady State Errors

Dr. ASHRAF E. SUYYAGH

THE UNIVERSITY OF JORDAN

DEPARTMENT OF COMPUTER ENGINEERING
FALL 2022

*بيعني بهذا ال chapter أعرف أطبق عال (Matlab) .

* لما نصمم ال Control systems بيهمنا 3 شغلات :
ال Stability و ال transiant و Steady-state .

* ال Steady-state ← لما ال Time يوصل للمالانهاية ، كل ال transiant response بتموت وبضل ال Forced input وبدنا ال Steady-state-error يكون أقل ما يمكن .

Topics

* باللي درسناه لهلأ كان R(s) = $\frac{1}{s}$ (unit step) (كنت بـ State ورحت ع State ثانية) .

- Definitions and Signal Inputs
- Sources of Error
- The Final Value Theorem
- Static Error Constants k_p, k_v and k_a
- Steady State Errors in Closed-Loop Unity Feedback System
- Steady State Errors in Closed-Loop Unity Feedback System with Disturbance
- Steady State Errors in Closed-Loop Non-Unity Feedback System
- Steady State Errors in Closed-Loop Non-Unity Feedback System with Disturbance
- Extra Examples

*هنا محطة فيها Satellite بده يتتبع عدة أجرام فضائية أو هيك ،
استخدام ال unit step funct لما يكون عندي Satellite بده يتتبع
geostationary orbit (محل ما أطلقناه وحطيناه بضله بمكانه ال
Satellite) (مثل نايلسات) ، فعملية التتبع هاي بتكون on/off عادي.

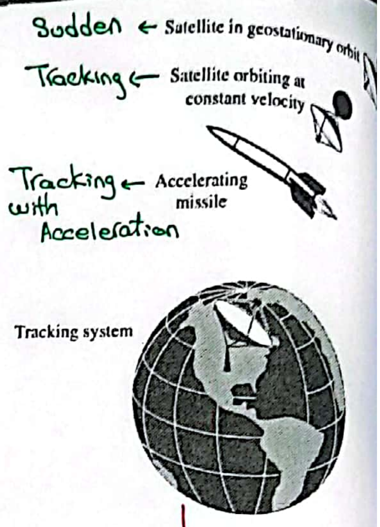
3 Definition and Test Inputs

Steady-state error is the difference between the input and the output for a prescribed test input as $t \to \infty$.

Step inputs represent constant position and thus are useful in determining the ability of the control system to position itself with respect to a stationary target, such as a satellite in geostationary orbit.

Ramp inputs represent constant-velocity inputs to a position control system by their linearly increasing amplitude. These waveforms can be used to test a system's ability to follow a linearly increasing input or, equivalently, to track a constant-velocity target. For example, a position control system that tracks a satellite that moves across the sky at a constant angular velocity

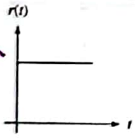
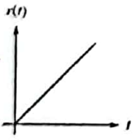
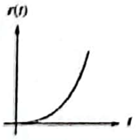
Parabolas, whose second derivatives are constant, represent constant acceleration inputs to position control systems and can be used to represent accelerating targets, such as the missile in, to determine the steady-state error performance.


Sudden ← Satellite in geostationary orbit
Tracking ← Satellite orbiting at constant velocity
Tracking with Acceleration ← Accelerating missile
Tracking system

*مرات ما بزبط يكون الـ System بطريقة الـ on/off مثل الـ Satellite المتحرك (محطة الفضاء
الدولية) ← بتلف كل 90 دقيقة بتلف لفة حوالين الأرض فلازم الـ Satellite يضل يتحرك
ويتبع ، هدول الـ Systems ما بنحطلهم Step funct بنحطلهم input متحرك بسرعة ثابتة
(عشان ما يضل يزيد لحد ما يطلع برا المدار) (constant velocity) هنا الـ input اسمه Ramp
↓
tu(t)

4 Test Waveforms

TABLE 7.1 Test waveforms for evaluating steady-state errors of position control systems

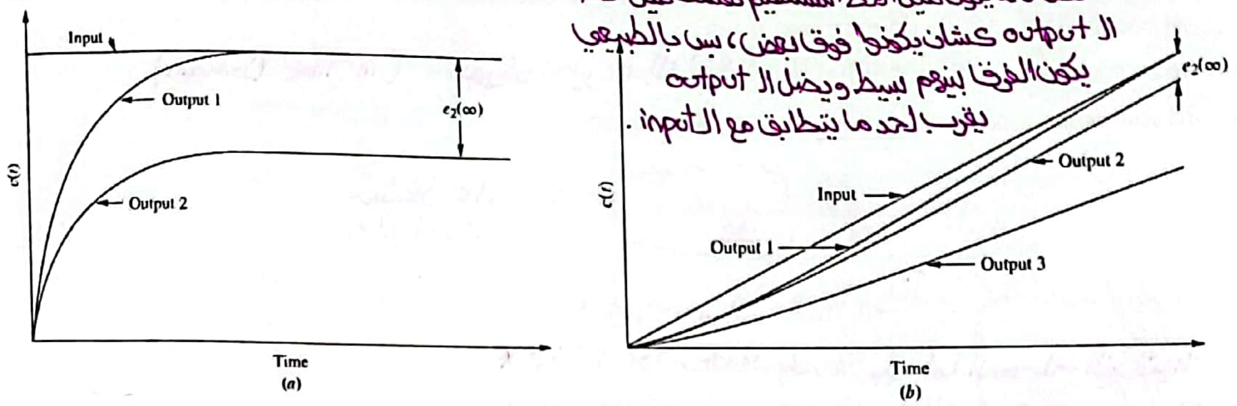
Waveform	Name	Physical interpretation	Time function	Laplace transform
	Step	Constant position	1	$\frac{1}{s}$
	Ramp	Constant velocity	t	$\frac{1}{s^2}$
	Parabola	Constant acceleration	$\frac{1}{2}t^2$	$\frac{1}{s^3}$

مثل لما كنا نحرك الديش لحد ما يلقط التلفزيون القناة.

← بدي بكل لحظة أكون قريب عالخط بهاي اللحظة أو وكل نقطة بهاي اللحظة موجودة عالخط.

* النوع الثالث من الـ Systems مثل الصواريخ (ما بتكون سرعتها ثابت بتحتاج تسارع فالـ Ramp
ما بكفيها فنستخدم الـ Parabola) (constant acceleration).
↲ $\frac{1}{2}t^2\underbrace{u(t)}$
↳ بنضيفه عشان نتأكد إنه ما في قيم Time سالبة.

Steady State Errors for Unit Step and Ramp Input



* أفضل حالة يكون ميل الخط المستقيم نفسه ميل خط ال output عشان يكونوا فوق بعض ، بس بالطبيعي يكون الفرق بينهم بسيط ويضل ال output يقرب لحد ما يتطابق مع ال input .

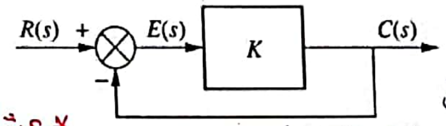
* لو ال output ضل يبعد عن ال input معناه ال error كل ماله عم بزيد .

* Control Systems → Proportional (gain (Amplification)) : اللي اشتغلنا عليه لحد الآن ، بكون فيها خطأ ولو بنسبة بسيطة .

→ Integral : وجودها بال System بتخليلي الأخطاء zero.

Sources of Errors

- Many steady-state errors in control systems arise from nonlinear sources, such as backlash in gears or a motor that will not move unless the input voltage exceeds a threshold.
- The steady-state errors we study here are errors that arise from the configuration of the system itself and the type of applied input. Consider a system with only a gain, and a system with an Integrator (Laplace of Integration has division by s)



$$e_{\text{steady-state}} = \frac{1}{K} c_{\text{steady-state}}$$

Sources of errors :

← أول سبب ، إنه الطبيعة non-linear فكل شغلنا افتراضات عال linear (approx. LTI system) ، هاي ال errors ما حنحلها لإنه كثير معقدة .

← ثاني سبب هو التصميم نفسه ومرات بنكون مجبورين نعمل هاي الشغلة وهذا اللي حنحله (inherent errors) .

* عشان يكون ال error قليلة جدًا بال system لازم تكون K = ∞ (كبيرة جدًا) ومستحيل تكون ∞ ، الأ ما يكون في error لنا ← بندخل هاي الشغلة بإضافة Integrator block بتعدل هاي الأخطاء وهاي ال block هي أساس ال PID controller .

($\frac{1}{s}$ = Integrator block) "هاي مش unit step"

→ Ex: * Laplace $\int f(t) = \frac{F(s)}{S}$ ↲ عمليات تكامل بال System / * المشتقة : SF(S) ↲ عمليات اشتقاق بال System

بتربط علاقة الـ limit مابين
الـ Time Domain والـ frequency domain

The Final Value Theorem

- From the definition of the steady-state error, steady-state error is the difference between the input and the output for a prescribe test input as t → ∞.
- We are interested in $e(\infty) = \lim_{t\to\infty} e(t)$ → مفروض تصير 0 بالنهاية (in Time Domain)
- But since we are working in the s-domain, the final-value theorem is helpful

$$e(\infty) = \underline{\lim_{t\to\infty}} e(t) \boxed{= \lim_{s\to 0} sE(s)}$$

→ حنشتغل على هاي احنا.

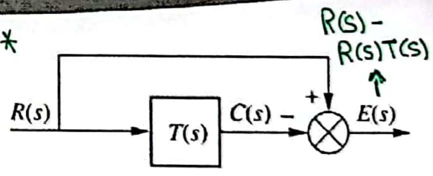
* كيفية حساب الخطأ 8

* عرفنا الـ Steady-state error بأنه لما المحصلة النهائية للـ System تروح لـ ∞ بدي أوصل لإنه الـ error يكون 0 .

Example for Forward Path System

* لازم الـ System يكون Stable لو ما كان Stable فكل افتراضاتنا بتكون خاطئة .

- Find the steady-state error for the system shown if
T(s) = $5/(s^2 + 7s + 10)$ and R(s) is a unit step response 1/s



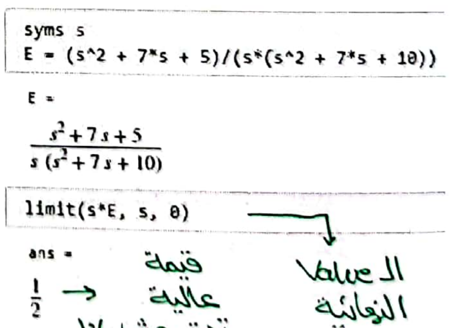
$$E(s) = \frac{s^2 + 7s + 5}{s(s^2 + 7s + 10)}$$

- To find the error e(∞), we apply $\lim_{s\to 0} sE(s)$
- We solve the limit in MATLAB:

$$* R(S) * (1 - T(S)) \rightarrow R(S) * \left(1 - \frac{5}{S^2+7S+10}\right)$$

$$\rightarrow R(S) * \left(\frac{S^2+7S+5}{S^2+7S+10}\right) \rightarrow \frac{1}{S} * \left(\frac{S^2+7S+5}{S^2+7S+10}\right)$$

$$= \frac{S^2+7S+5}{S(S^2+7S+10)}$$


*
```
syms s
E = (s^2 + 7*s + 5)/(s*(s^2 + 7*s + 10))
```

E =

$$\frac{s^2 + 7s + 5}{s\,(s^2 + 7s + 10)}$$

*
```
limit(s*E, s, 0)
```

ans =

$\frac{1}{2}$ → قيمة عالية تعتبر مش 1% ولا 5%

الـ Value النهائية بعيدة عن الـ Value اللي بدي ياها بنسبة 50%

* عشان نطلع ال Steady-state-error ← Final value theorem :

$$\lim_{s \to 0} s \times \left(\frac{s^2+7s+5}{s(s^2+7s+10)} \right)$$

استخراج ناتجها باستخدام ال Matlab في Command اسمه limit بشتغل عال Symbolic :

Slide 8 ← وفي طريقة ثانية أسهل numeric حناخدها لقدام .

9

Errors in Closed-Loop Unity Feedback System I

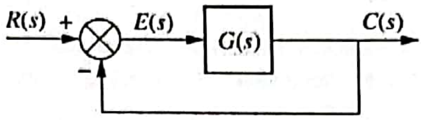
* حنشتغل على : Closed-loop system with unity feedback (1)

$$E(s) = R(s) - C(s)$$

$$C(s) = E(s)G(s)$$

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

$$\boxed{e(\infty) = \lim_{s \to 0} \frac{sR(s)}{1+G(s)}}$$



يومنا تكون صفر وهذا هو ال Idle case بس

بالواقع في حالات لل error :

↳ Finite error

↳ Infinite error ← بضل يكبر

If the limit equals zero, this means there is no steady state error, otherwise we could have:

- a finite (small)error and we need to check if it is within the required error band
- or the error grows to infinity!

(±)

* كيف بعوض ال Matlab ال limit : إذا قيمة النهاية من اليمين = قيمة النهاية من اليسار فبطلع الجواب للنهاية طبيعي أما لو مش متساوين بيعطي NAN .

بنعوض R(s) بالحالات الثلاثة : Step / Ramp / Parabola

* حنشوف إنه كل ما نضيف Integrator لل System حيتحول من ∞ ال Finite ال 0 .

10

Errors in Closed-Loop Unity Feedback System II

R(s) could be a unit step, ramp, or parabola, therefore:

Step Input. with $R(s) = 1/s$, we find

$$\boxed{e(\infty) = e_{\text{step}}(\infty) = \lim_{s \to 0} \frac{s(1/s)}{1+G(s)} \overset{*}{=} \frac{1}{1+\lim_{s \to 0} G(s)}}$$

بحتاج بمقامها S

Ramp Input. with $R(s) = 1/s^2$, we obtain

$$\boxed{e(\infty) = e_{\text{ramp}}(\infty) = \lim_{s \to 0} \frac{s(1/s^2)}{1+G(s)} = \lim_{s \to 0} \frac{1}{s+sG(s)} \overset{*}{=} \frac{1}{\lim_{s \to 0} sG(s)}}$$

بحتاج بمقامها S^2 عشان يضل عندي S

Parabolic Input. with $R(s) = 1/s^3$, we obtain

$$\boxed{e(\infty) = e_{\text{parabola}}(\infty) = \lim_{s \to 0} \frac{s(1/s^3)}{1+G(s)} = \lim_{s \to 0} \frac{1}{s^2+s^2G(s)} \overset{*}{=} \frac{1}{\lim_{s \to 0} s^2G(s)}}$$

بحتاج بمقامها S^3 عشان يضل عندي S

* **Static Error Constants**

$$G(s) \equiv \frac{(s+z_1)(s+z_2)\cdots}{s^n(s+p_1)(s+p_2)\cdots}$$

* K_p → Position

$$\boxed{\lim_{s \to 0} G(s)} = \infty$$

G(s) denominator must have at least an s^1 at the denominator for the limit to be infinity (one integration)

* K_v → Velocity

$$\boxed{\lim_{s \to 0} sG(s)} = \infty$$

G(s) denominator must have an s^2 at the denominator for the limit to be infinity (two integrations)

* K_a → acceleration

$$\boxed{\lim_{s \to 0} s^2G(s)} = \infty$$

G(s) denominator must have an s^3 at the denominator for the limit to be infinity (three integrations)

Triple Integration ← double Integration ← one integration

$$G(s) = \frac{(s+z_1)(s+z_2)\cdots}{s^n(s+p_1)(s+p_2)\cdots}$$

* يهمني قيم Kp ، Kv ، Ka يكونوا ∞ عشان $\frac{1}{\infty}$ = 0 فبتقل نسبة الخطأ لـ 0 .

* لتكون G(s) = ∞ لازم يكون بمقامها 0 لما S=0 ← G(s) = ∞ ← e=0

* وجود S في المقام يعني في Integration .

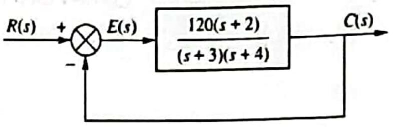
* لازم يكون الـ feedback عبارة عن Unity عشان أقدر أحل.

Example: Steady-State Errors for Systems with No Integrations

Find the steady-state errors for inputs of 5u(t), 5tu(t), and $5t^2u(t)$ to the system shown assuming the system is stable.

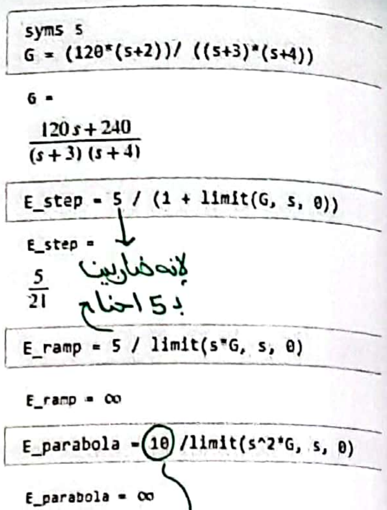
unit step ↰ Ramp↲ ↳Parabola

*ماقي داعي أطلع E(s) لأنه المعطيات بدلالة G(s) و G(s) موجودة عندي.



Taking a look at E(s), we notice that the denominator has no standalone s term. We expect that we have finite error for the unit step but infinite for the ramp and parabola inputs.

R(s) = 5/s R(s) = 5/ s^2 R(s) = 10/ s^3



```
syms s
G = (120*(s+2))/ ((s+3)*(s+4))
```

G =

$\frac{120\,s+240}{(s+3)\,(s+4)}$

```
E_step = 5 / (1 + limit(G, s, 0))
```

E_step =

$\frac{5}{21}$

لأنه ضاربين بـ 5 احنا

```
E_ramp = 5 / limit(s*G, s, 0)
```

E_ramp = ∞

```
E_parabola = 10 /limit(s^2*G, s, 0)
```

E_parabola = ∞

من الـ laplace لـ $5t^2$

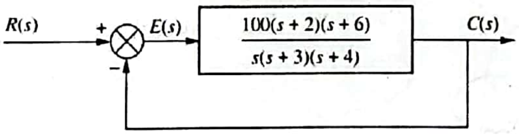
*هون وضحلك إنه في error لأنه ماقي Integratrons *

* يعني هنا الـ System يصلح إدخال unit step عليه فقط ومع ذلك بكون فيه نسبة خطأ.

Example: Steady-State Errors for Systems with One Integration

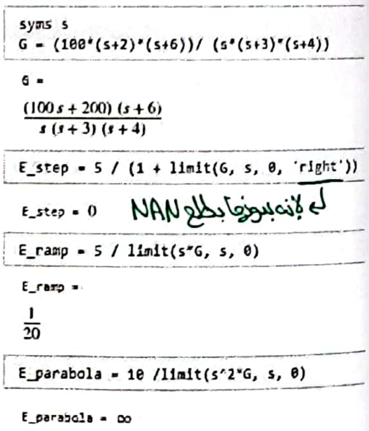
Find the steady-state errors for inputs of 5u(t), 5tu(t), and $5t^2u(t)$ to the system shown assuming the system is stable.

.



Taking a look at E(s), we notice that the denominator has one term of s with a degree of s^1 That is one integrator, we expect that we have zero for the unit step but infinite for the ramp and parabola inputs.

R(s) = 5/s R(s) = 5/ s^2 R(s) = 10/ s^3



```
syms s
G = (100*(s+2)*(s+6))/ (s*(s+3)*(s+4))
```

G =

$\frac{(100\,s+200)\,(s+6)}{s\,(s+3)\,(s+4)}$

```
E_step = 5 / (1 + limit(G, s, 0, 'right'))
```

E_step = 0

لأنه بدونها بطلع NAN

```
E_ramp = 5 / limit(s*G, s, 0)
```

E_ramp =

$\frac{1}{20}$

```
E_parabola = 10 /limit(s^2*G, s, 0)
```

E_parabola = ∞

*هون بوضحلك إنه إضافة Integrator واحد (s وحدة) بخلي الـ error بالـ unit step = 0 وبالـ Finite = Ramp وبالـ Parabola لساه ∞

بدل ماكانت Infinite بدل ماكانت Finite

* لازم نفحص الSystem اذا Stable ولا لأ عشان نعرف نكمل ولا نوقف :

Example: Unstable Systems

* الStability تحسب للSystem كامل فلازم نطلع معادلة الSystem :

A unity feedback system has the following forward transfer function:

$\frac{G(S)}{1+G(S)}$

$$G(s)=\frac{10(s+20)(s+30)}{s^2(s+25)(s+35)(s+50)}$$

Find the steady-state error for the following inputs: 15u(t), 15tu(t), and 15t²u(t)

Notice, that unlike the previous examples, we did not assume or tell you that the system is stable, so we need to check if the system is stable, otherwise, nothing matters.

This is a feedback system with a transfer function Sys given as:

```
s = tf('s' );
G = (10 *(s + 20)*(s + 30)) / (s^2 *(s + 25)*(s + 35)*(s + 50));
Sys = feedback (G, 1)
```

→ معادلة الSystem

لاستخدامها لازم يكون التعريف tf('s') مش syms

```
Sys =

              10 s^2 + 500 s + 6000
  ----------------------------------------------------
  s^5 + 110 s^4 + 3875 s^3 + 43760 s^2 + 500 s + 6000

Continuous-time transfer function.
```

```
roots ([1, 110, 3875, 43760, 500, 6000])
```

```
ans = 5x1 complex
  -50.0064 + 0.0000i
  -34.9959 + 0.0000i
  -24.9984 + 0.0000i
    0.0004 + 0.3703i
    0.0004 - 0.3703i
```

⇒ System unstable فما بكمل.

استخدام طريقة الNumeric مش الSymbolic .

Finding Static Error Constants using MATLAB

```
numg=500*poly([-2 -5 -6]); % Define numerator of G(s).
deng=poly([0 -8 -10 -12]); % Define denominator of G(s).
G=tf(numg,deng); % Form G(s).
' Check Stability' % Display label.
```

Poly Command بوصف المعادلة باستخدام الroots تبعتها لأنها مكتوبة بهاي الطريقة.

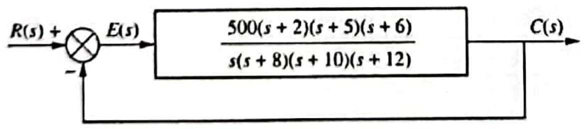
```
ans = ' Check Stability'
```

Evaluate the static error constants and find the expected error for the standard step, ramp, and parabolic inputs.

```
T=feedback(G,1) % Form T(s).
```

```
T =

     500 s^3 + 6500 s^2 + 26000 s + 30000
  -------------------------------------------
  s^4 + 530 s^3 + 6796 s^2 + 26960 s + 30000

Continuous-time transfer function.
```

First, we need to know if the system is stable or not



```
poles=pole(T) % Display closed-loop poles.
```

→ Pole Command

أسهل من الroots

لحاله بيعطيك الroots دغري ال(real part) منها.

```
poles = 4x1
  -516.9544
    -5.7623
    -5.4278
    -1.8554
```

```
roots ([1 530 6796 26960 30000])
```

```
ans = 4x1
  -516.9544
    -5.7623
    -5.4278
    -1.8554
```

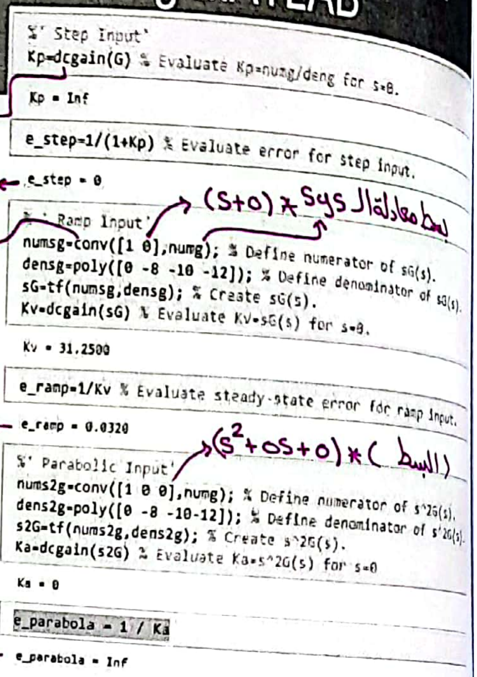
* أول خطوة نفحص الSystem اذا Stable ولا لأ.

لما تأكدت إنه Stable فبلش حل.

Finding Steady-State Error Constants using MATLAB

Static Error Constants

K_p	$\lim_{s\to 0} G(s)$
K_v	$\lim_{s\to 0} sG(s)$
K_a	$\lim_{s\to 0} s^2G(s)$



```
%' Step Input'
Kp=dcgain(G) % Evaluate Kp=numg/deng for s=0.

Kp = Inf

e_step=1/(1+Kp) % Evaluate error for step input.

e_step = 0

%' Ramp Input'
numsg=conv([1 0],numg); % Define numerator of sG(s).
densg=poly([0 -8 -10 -12]); % Define denominator of sG(s).
sG=tf(numsg,densg); % Create sG(s).
Kv=dcgain(sG) % Evaluate Kv=sG(s) for s=0.

Kv = 31.2500

e_ramp=1/Kv % Evaluate steady-state error for ramp input.

e_ramp = 0.0320

%' Parabolic Input'
nums2g=conv([1 0 0],numg); % Define numerator of s^2G(s).
dens2g=poly([0 -8 -10-12]); % Define denominator of s^2G(s).
s2G=tf(nums2g,dens2g); % Create s^2G(s).
Ka=dcgain(s2G) % Evaluate Ka=s^2G(s) for s=0

Ka = 0

e_parabola = 1 / Ka

e_parabola = Inf
```

بعطيني قيمة K دغري.

لأنه في Integration واحد

بضرب S في G(S)

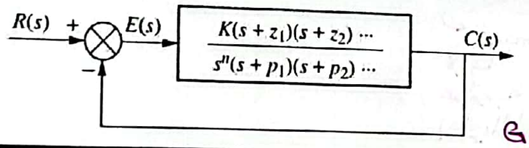
بعمل معادلة ال Sys * (S+0)

وجود Integration واحد حولها من ∞ لـ Finite

(البسط) * (S^2+0S+0)

ضلت ∞ لأنه هو محتاجه 3 Integration لتصير 0

System Types Summary



The number of integrations determine the number and value of errors we have in the system

Systems with no Integrators (المقام لـ G(s) مافيه S) ← Type 0

فيها Integration وحدة (مقام G(s) فيه S وحدة) ← Type 1

فيها 2 Integrations (مقام G(s) فيه S^2) ← Type 2

		Type 0		Type 1		Type 2	
Input	Steady-state error formula	Static error constant	Error	Static error constant	Error	Static error constant	Erro
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	$\frac{1}{K_a}$

* كل ما نزيد بعدد ال Integrators بقل ال error.

* لو ما في ولا Integrator فحيكون أكيد عنّا نسبة خطأ.

Interpretation Example 1

* بيجي زيـــه بالامتحان دوائر*

Interpreting the Steady-State Error Specification

← بما إنه Constant فبكون Type0 يعني Step input.

PROBLEM: What information is contained in the specification $K_p = 1000$?

SOLUTION: The system is stable. The system is Type 0, since only a Type 0 system has a finite K_p. Type 1 and Type 2 systems have $K_p = \infty$. The input test signal is a step, since K_p is specified. Finally, the error per unit step is

$$e(\infty) = \frac{1}{1+K_p} = \frac{1}{1+1000} = \frac{1}{1001}$$

* أولخطوة لازم تعرفها
إنه مدام طلع عندك قيم
للerror والشغل تبعك
معناه الsystem
stable. لوما كان
stable ما بنكمل الشغل.

Interpretation Example II

*system is stable. *

- What conclusions do we draw if a system has $K_v = 1000$

بما إنه Constant
Ramp ← Type 1 ←
input

1. The system is stable.
2. The system is of Type 1, since only Type 1 systems have K_v's that are finite constants. Recall that $K_v = 0$ for Type 0 systems, whereas $K_v = \infty$ for Type 2 systems.
3. A ramp input is the test signal. Since K_v is specified as a finite constant, and the steady-state error for a ramp input is inversely proportional to K_v, we know the test input is a ramp.
4. The steady-state error between the input ramp and the output ramp is $1/K_v$ per unit of input slope.

Designing for Gain, Error Specification and Stability

PROBLEM: Given the control system in Figure find the value of *K* so that there is 10% error in the steady state.

ضربنا البسط بـ S

SOLUTION: Since the system is Type 1, the error stated in the problem must apply to a ramp input; only a ramp yields a finite error in a Type 1 system. Thus,

```
num=[1 5]; % Define numerator of G(s)/K.
den=poly([0 -6 -7 -8]); % Define denominator
GdK=tf(num,den); % Create G(s)/K.
num=conv([1 0],num); % Define numerator of sG(
GdK=tf(num,den); % Create sG(s)/K.
e=0.1
```

```
e = 0.1000
```

```
kv = 1 / e
```

```
kv = 10
```

Kv اللي طلعناها

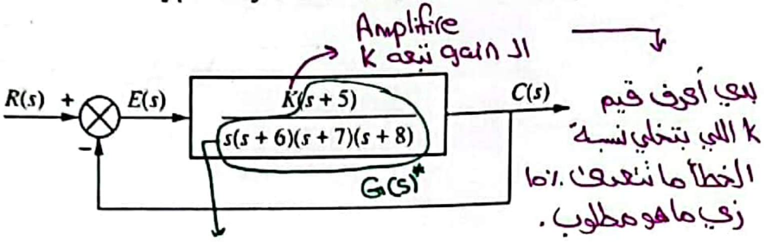
```
K = kv / dcgain(GdK)
```

اللي عندي

```
K = 672
```

```
%Check Stability ✓
T=feedback(K*GdK,1); % Form T(s).
poles=pole(T) % Display closed-loop poles.
```

```
poles = 4x1 complex
    0.0000 + 0.0000i
   -7.9956 +25.9610i
   -7.9956 -25.9619i
   -5.0088 + 0.0000i
```



بدي أعرف قيم k اللي بتخلي نسبة الخطأ ما تتعدى 10% زي ما هو مطلوب.

في s بالمقام فإذا

Ku ← Type1

* e = 10% → $K_v = \frac{1}{e} = 10$

* المشكلة عندي هلأ إنه قيمة K مجهولة ← if $G(s) = k\left(\frac{(s+5)}{s(s+6)(s+7)(s+8)}\right)$

فرح اشتغل هيك: $\frac{G(s)}{k}$ = System equation ← لأقدر أكتب الـSystem بدون مجاهيل. ← لازم أضمن بعد كل هذا إنه الـSystem ضل Stable

Steady-State Errors with Disturbances I

* أهم مصدرين للخطأ: التصميم نفسه والـ Systems كلها تقريبا non-linear ، بس فعليا بالواقع أهم الأخطاء بتكون من مصادر خارجية (Disturbances)

- Feedback control systems are used to compensate for disturbances or unwanted inputs that enter a system. The advantage using feedback is that regardless of these disturbances, the system can be designed to follow the input with small or zero er
- The figure shows a feedback control system with a disturbance, D(s), injected between the controller and the plant. We now derive the expression for steady-state error with the disturbance included. By setting each input D(s) and R(s) to zero one at time, the transform of the output is given by:

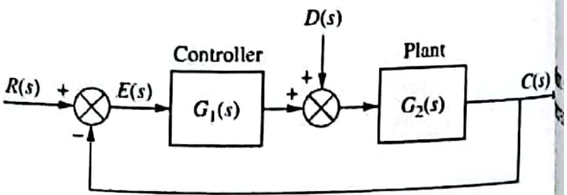
$$C(s) = E(s)G_1(s)G_2(s) + D(s)G_2(s)$$

$$C(s) = R(s) - E(s)$$

$$E(s) = \underbrace{\frac{1}{1+G_1(s)G_2(s)}R(s)}_{\text{الشق الأول هو زي كل اشي حللناه من البداية وزي معادلة الـfeedback.}} - \underbrace{\frac{G_2(s)}{1+G_1(s)G_2(s)}D(s)}_{\text{الشق الثاني هو الخطأ الناتج من الـDisturbance.}}$$

↓ حنصير ناخذه بعين الاعتبار ونشوف تأثيره على Sys

بعد تعويض المعادلتين ببعض طلعتلي معادلة E(S)



* وجود الـ Disturbance ما بأثر على الأشياء اللي كانت عنا قبل (الأخطاء الناتجة من الـ input والـ non-linearity والـ configuration والـ Ka ، Kp ، Kv).

Steady-State Errors with Disturbances II

- To find the steady-state value of the error, we apply the final value theorem

$$e(\infty) = \lim_{s\to 0} sE(s) = \lim_{s\to 0}\frac{s}{1+G_1(s)G_2(s)}R(s) - \lim_{s\to 0}\frac{sG_2(s)}{1+G_1(s)G_2(s)}D(s)$$
$$= e_R(\infty) + e_D(\infty)$$

$$e_R(\infty) = \lim_{s\to 0}\frac{s}{1+G_1(s)G_2(s)}R(s)$$

$$e_D(\infty) = -\lim_{s\to 0}\frac{sG_2(s)}{1+G_1(s)G_2(s)}D(s)$$

* للتبسيط حنفرض إنه D(s) عبارة عن Step function.

$(D(S) = \frac{1}{s})$

عشان S و D(s) يلغوا بعض.
بعدين بنقسم البسط والمقام
على G₂(s) بطلع معادلة Slide22

- Notice that the first term relating the error to the input is the same as the one derived and analysed before (see slide 8), where G1 and G2 are in series and can be replaced with G(s), and the input can be a unit step, a ramp, or a parabola

Steady-State Errors with Disturbances III

- Now, we only focus on the error due to the disturbance D(s). To simplify the analysis, we only consider a unit step error, that is D(s) = 1/s. Substituting this value of D(s) into the previous equation will yield:

$$e_D(\infty) = -\frac{1}{\lim\limits_{s\to 0}\frac{1}{G_2(s)} + \lim\limits_{s\to 0} G_1(s)}$$

← نسبة الخطأ الناتجة عن ال
Disturbance ع

- This equation shows that the steady-state error produced by a step disturbance can be reduced by increasing the dc gain of G_1(s) or decreasing the dc gain of G_2(s).

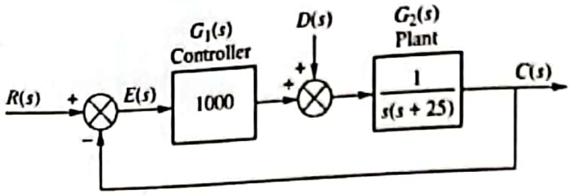
← قيمة ال (dcgain)

* زيادة G₁(s) بقلل من نسبة الخطأ الخارجي.
* زيادة G₂(s) بزيد من نسبة الخطأ الخارجي.

← فما بزبط نزيد التنين
لازم نزيد G₁ ونقلل G₂
لنقلل نسبة الخطأ الخارجي.

Steady-State Error Due to Step Disturbance Example

- Find the steady-state error component due to a step disturbance for the system of the above figure

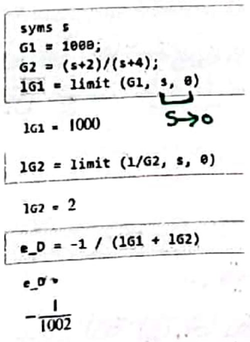


$$* \quad e_D(\infty) = -\frac{1}{\lim\limits_{s\to0}\frac{1}{G_2(s)} + \lim\limits_{s\to0} G_1(s)} = -\frac{1}{0+1000} = -\frac{1}{1000}$$

Steady-State Error Due to Step Disturbance Example

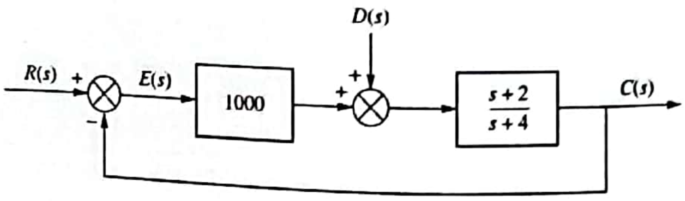
« حنحل بالماتلاب »

- Evaluate the steady-state error component due to a step disturbance for the system shown in the figure using MATLAB. Assume the system is stable.



```
syms s
G1 = 1000;
G2 = (s+2)/(s+4);
lG1 = limit (G1, s, 0)

lG1 = 1000

lG2 = limit (1/G2, s, 0)

lG2 = 2

e_D = -1 / (lG1 + lG2)

e_D =
-1/1002
```

S→0



* إشارة السالب بالناتج بتدل على إنه :
لنفرض ال Signal كانت 5 ، لو ال error
موجب بكون الخطأ = ٤ لو كانت سالبة بكون
الخطأ أعلى من هيك .

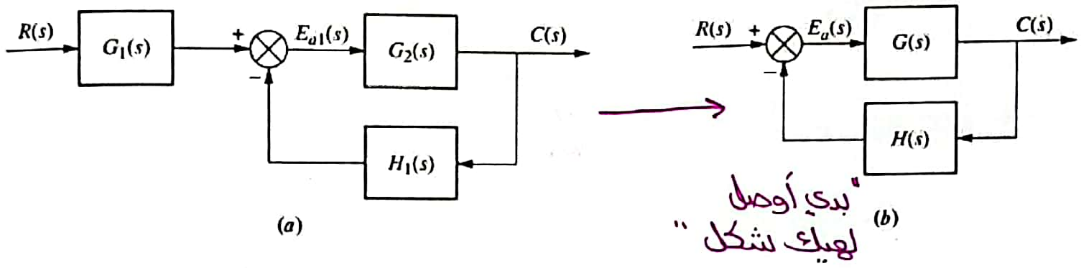
* هون ال error طالع
بس لل Disturbance .

* لو بدك تطلعي ال Total error
بتطلعي ال error العادي زي أول وبتجمعيه
ال Disturbance error .

* بدك ال feedback يكون unity gain زي أول .

Steady-State Error for Nonunity Feedback Systems

- Control systems often do not have unity feedback because of the compensation used to improve performance or because of the physical model for the system. The feedback path can be a pure gain other than unity or have some dynamic representation.
- A general feedback system, showing the input transducer, G1(s), controller and plant,G2(s), and feedback,H1(s),is shown in the Figure (a) below, now we go through the steps to make it look like unity feedback systems



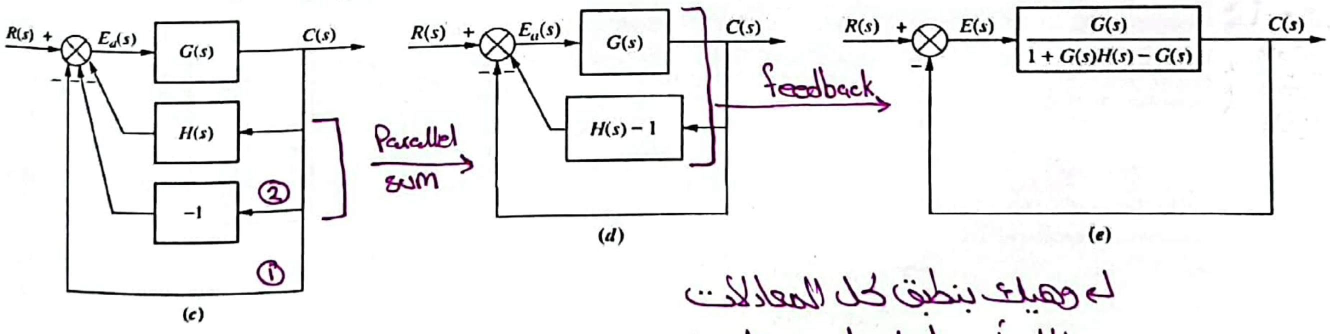
"بدي أوصل لهيك شكل"

Steady-State Error for Nonunity Feedback Systems II

تحويل ال System لـ unity gain بإضافة خطين للـ System

① 1

② -1



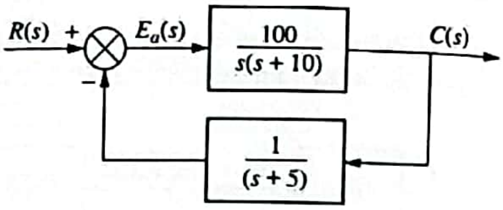
← وهيك بنطبق كل المعادلات اللي أخذناها عليه عادي.

Steady-State Error for Nonunity Feedback Systems Example

- For the system shown in the figure, find the system type, the appropriate error constant associated with the system type, and the steady-state error for a unit step input. Assume input and output units are the same.

$$G(s)=\frac{100}{s(s+10)}$$

$$H(s)=\frac{1}{(s+5)}$$



$$G_e(s)=\frac{G(s)}{1+G(s)H(s)-G(s)}=\frac{100(s+5)}{s^3+15s^2-50s-400}$$

Type 0

$$K_p=\lim_{s\to 0}G_e(s)=\frac{100\times 5}{-400}=-\frac{5}{4}$$

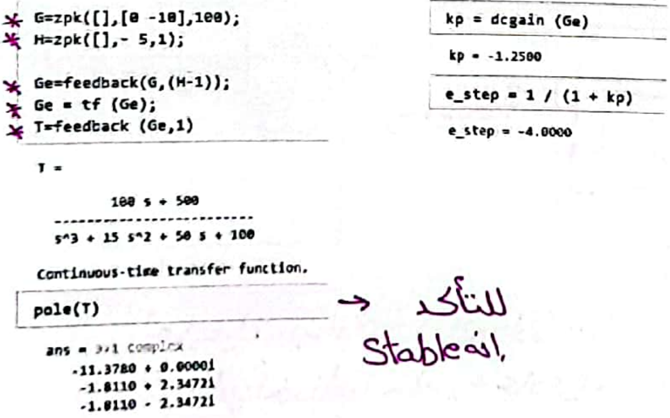
$$e(\infty)=\frac{1}{1+K_p}=\frac{1}{1-(5/4)}=-4$$

يعني لو الـSteady-state
كان مفروض تكون 5 فهو
أعطاني 20 فأعطاني ناتج أكبر من
اللي بدي ياه.

Steady-State Error for Nonunity Feedback Systems Example

- For the system shown in the figure, find the system type, the appropriate error constant associated with the system type, and the steady-state error for a unit step input. Assume input and output units are the same, and the system is stable. Use MATLAB

في ← [
طرق
كثيرة
للكتابة



```
* G=zpk([],[0 -10],100);
* H=zpk([],- 5,1);

* Ge=feedback(G,(H-1));
* Ge = tf (Ge);
* T=feedback (Ge,1)
```

```
T =

         100 s + 500
  ----------------------
  s^3 + 15 s^2 + 50 s + 100

Continuous-time transfer function.
```

```
pole(T)
```

→ للتأكد انه Stable.

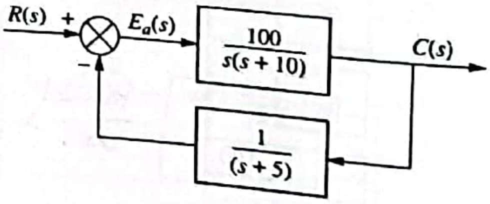
```
ans = 3x1 complex
  -11.3780 + 0.0000i
   -1.8110 + 2.3472i
   -1.8110 - 2.3472i
```

```
kp = dcgain (Ge)
```

```
kp = -1.2500
```

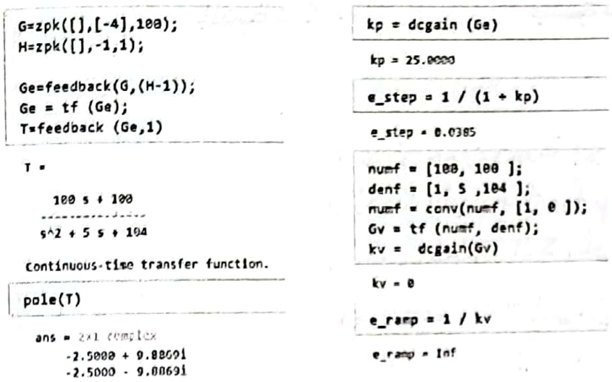
```
e_step = 1 / (1 + kp)
```

```
e_step = -4.0000
```



Steady-State Error for Nonunity Feedback Systems Example 2

- Find the steady-state error for a unit step input given the nonunity feedback system. Repeat for a unit ramp input. Assume input and output units are the same.



```
G=zpk([],[-4],100);
H=zpk([],-1,1);

Ge=feedback(G,(H-1));
Ge = tf (Ge);
T=feedback (Ge,1)
```

```
T =

   100 s + 100
  --------------
  s^2 + 5 s + 104

Continuous-time transfer function.
```

```
pole(T)
```

```
ans = 2x1 complex
    -2.5000 + 9.8869i
    -2.5000 - 9.8869i
```

```
kp = dcgain (Ge)
```

```
kp = 25.0000
```

```
e_step = 1 / (1 + kp)
```

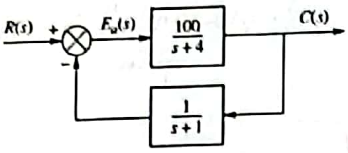
```
e_step = 0.0385
```

```
numf = [100, 100 ];
denf = [1, 5 ,104 ];
numf = conv(numf, [1, 0 ]);
Gv = tf (numf, denf);
kv =  dcgain(Gv)
```

```
kv = 0
```

```
e_ramp = 1 / kv
```

```
e_ramp = Inf
```



بنعرف إنه عنا أخطاء بسبب R وأخطاء بسبب D (disturbances).

Steady-State Error for Nonunity Feedback Systems with Disturbances

- To continue our discussion of steady-state error for systems with nonunity feedback, let us look at the general system shown in the figure, which has both a disturbance and nonunity feedback. We will derive a general equation for the steady-state error and then determine the parameters of the system in order to drive the error to zero for step inputs and step disturbances.

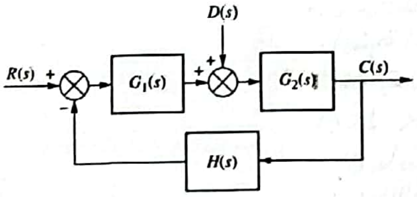
$$e(\infty) = r(\infty) - c(\infty)$$

لازم يكون بين r و c = 0 بناء على ذلك اشتق المعادلات.

r(∞)

$$e(\infty) = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\left\{\left[1-\frac{G_1(s)G_2(s)}{1+G_1(s)G_2(s)H(s)}\right]R(s) - \left[\frac{G_2(s)}{1+G_1(s)G_2(s)H(s)}D(s)\right]\right\}$$

$$e(\infty) = \lim_{s\to 0} sE(s) = \left\{\left[1-\frac{\lim_{s\to 0}[G_1(s)G_2(s)]}{\lim_{s\to 0}[1+G_1(s)G_2(s)H(s)]}\right] - \left[\frac{\lim_{s\to 0} G_2(s)}{\lim_{s\to 0}[1+G_1(s)G_2(s)H(s)]}\right]\right\}$$



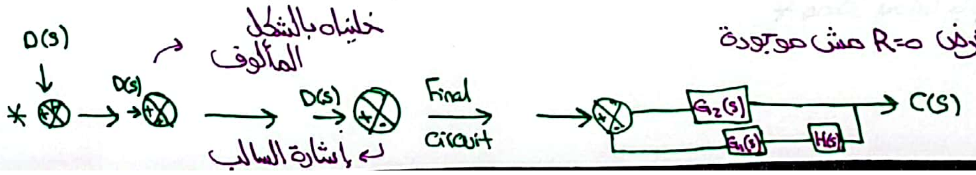
For zero error,

$$\frac{\lim_{s\to 0}[G_1(s)G_2(s)]}{\lim_{s\to 0}[1+G_1(s)G_2(s)H(s)]} = 1 \quad \text{and} \quad \frac{\lim_{s\to 0} G_2(s)}{\lim_{s\to 0}[1+G_1(s)G_2(s)H(s)]} = 0$$

* ليطلع معادلة r بده يفرض D = 0 مش موجودة كأنها.

* ليطلع معادلة D بده يفرض R=0 مش موجودة كأنها. ←

خليناه بالشكل المألوف



← إشارة السالب

Steady-State Error for Nonunity Feedback Systems with Disturbances

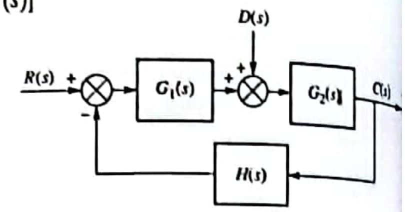
- To continue our discussion of steady-state error for systems with nonunity feedback, let us look at the general system shown in the figure, which has both a disturbance and nonunity feedback. We will derive a general equation for the steady-state error and then determine the parameters of the system in order to drive the error to zero for step inputs and step disturbances.

For zero error, → بالنسبة لـ r

→ بالنسبة لـ D

* وزعت الـ limit عالبسط والمقام.

$$\frac{\lim_{s\to0}[G_1(s)G_2(s)]}{\lim_{s\to0}[1+G_1(s)G_2(s)H(s)]}=1 \quad \text{and} \quad \frac{\lim_{s\to0}G_2(s)}{\lim_{s\to0}[1+G_1(s)G_2(s)H(s)]}=0$$



- The two equations can always be satisfied if
- 1 (1) the system is stable,
- 2 (2) G1(s) is a Type 1 system, → لازم أشوف s بالمقام واحدة
- 3 (3) G2(s) is a Type 0 system, and → مافي s بالمقام
- 4 (4) H(s) is a Type 0 system with a dc gain of unity.

* عوض كمان قيمة R(s) و D(s) اللي هم Step funct. ($\frac{1}{s}$) فبلغوا الـ s اللي بالبسط.

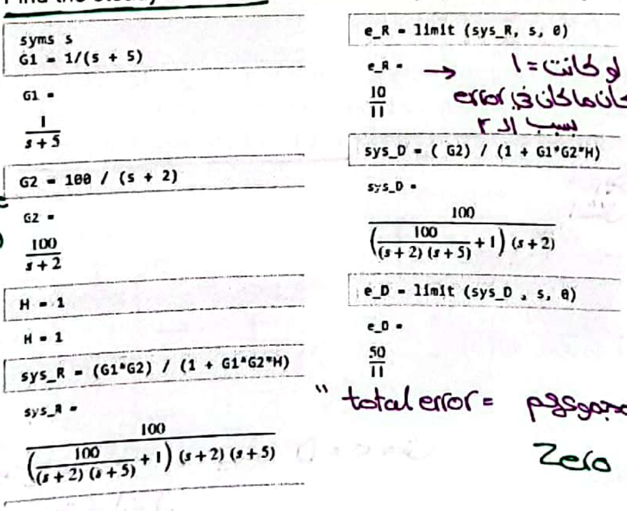
فنوصل للمعادلات النهائية للـ error بالـ Nonunity Feedback Systems with Disturbance.

→ كشان الـ error يكون بساوي ⓪ وتحقق المعادلتين لازم تكون الشروط التالية 1, 2, 3, 4 (معطاه بالامتحان)

Example

- Find the steady-state error for the following system assuming that it is stable, and that both the input and disturbance are unit step:

لو حكالك بالنسبة للDisturbance بتحل بس الDisturbance و لو حكالك بالنسبة للinput بتحلها بس لـ R وإنا ما حكالك بتحلها للاثنين



```
syms s
G1 = 1/(s + 5)
```

G1 =

$$\frac{1}{s+5}$$

```
G2 = 100 / (s + 2)
```

G2 =

$$\frac{100}{s+2}$$

```
H = 1
```

H = 1

```
sys_R = (G1*G2) / (1 + G1*G2*H)
```

sys_R =

$$\frac{100}{\left(\frac{100}{(s+2)\,(s+5)}+1\right)(s+2)\,(s+5)}$$

```
e_R = limit (sys_R, s, 0)
```

e_R = → لو كانت = 1 كان ما كان في error بسبب الـ r

$$\frac{10}{11}$$

```
sys_D = ( G2) / (1 + G1*G2*H)
```

sys_D =

$$\frac{100}{\left(\frac{100}{(s+2)\,(s+5)}+1\right)(s+2)}$$

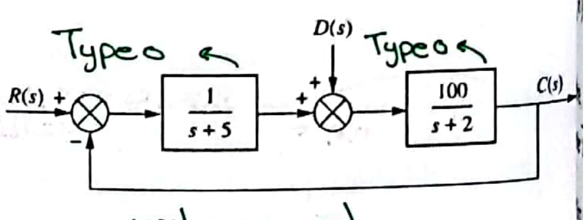
```
e_D = limit (sys_D , s, 0)
```

e_D =

$$\frac{50}{11}$$

" total error = مجموعهم "

لـ شرحنا فقط هاي الحالة.



Type0 ←

Type0 ←

unity gain ←

* ما تحققت كل الشروط فبعرف إنه المحصلة النهائية الخطأ ما حيكون Zero حيكون اله قيمة.

* ممكن نحلها بالإيد مرات يكون أسرلنا.

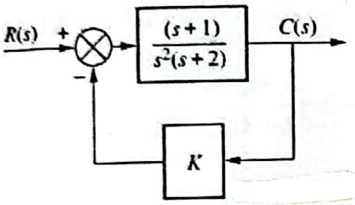
Additional Example I

Given the system shown below, find the **system type** and the value of K to yield 0.1% error in the steady state → ramp or Step or Parabola ?

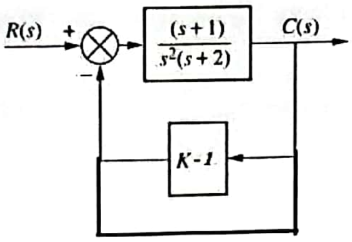
First step is to convert this system to a unity feedback system as follows:

Second step is to evaluate the upper feedback loop which yields:

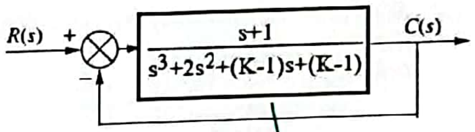
unity كشان يصير



ما بزبط نطرح لو range ال input مش نفس ال range اللي ماخذ منه ال ~~feed~~back. يعني ما بزبط ادخل Volt بين 0 و 5 تتحكم بال Motor وال Sensor بيرجع قيم بال mV مش بال V.



→ منحط ال amplifier مشان نخلي range ال Sensor ملائم لل input فنقدر نعمل عملية الطرح



ماكو s منفصلة أو Integrator فحيكون
step ← Type 0 → K_p

→ بدنا نلاقي قيمة ال K المناسبة بشرط تكون 0.1% بعيدة عن ال steady-state error.

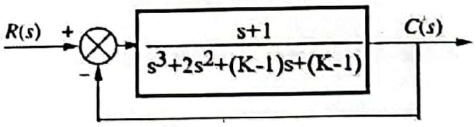
*مافي disturbance بال System فبنحل → زي ما أخذنا بالبداية، نستخرج K_p, K_v, K_a وهكذا

Additional Example I - Continued

Analysing G(s), it is clear that that the system is Type 0, therefore, the static error constant mentioned in the question relates to K_p

$$e_{step}(\infty) = 0.001 = \frac{1}{1+K_p}$$

→ مش نفسها k اللي بال System اتبزوي



Therefore, K_p is 999 and we know that K_p is the limit of G(s) as s goes to 0

It is easier to solve this mathematically, by substituting ***s*** by 0

$$K_p = \lim_{s\to 0}\left(\frac{s+1}{s^3+2s^2+(K-1)s+(K-1)}\right) = \frac{1}{(K-1)}$$ = 999, then K = 1.001 and G(s) is $\frac{s+1}{s^3+2s^2+0.001s+0.001}$

Check if the system is stable (remember to retrieve the transfer function for the whole system first):

```
roots ([1 2 1.001 1.001])
```

```
ans =

  -1.7546 + 0.0000i
  -0.1227 + 0.7453i
  -0.1227 - 0.7453i
```

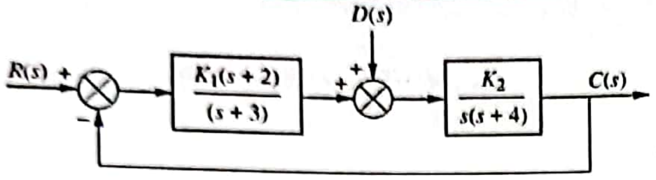
*لازم نطلع ال Stability لل System كامل.

(بيجي زي هيك بالامتحان)

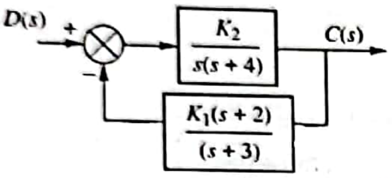
Additional Example II

- Design the values of K_1 and K_2 in the system of the adjacent figure such that the steady-state error component due to a unit st disturbance is 0.00001, and the steady-state error component due to a unit ramp input is 0.002.

→ R(S)



- When analysing the steady-state error due to disturbance only, we assume that R(s) is equal to zero, rearranging the shape:



We know that:

$$e_D(\infty) = -\lim_{s\to 0}\frac{sG_2(s)}{1+G_1(s)G_2(s)}D(s)$$

" صفرنا R(s) لنحل الDisturbance "

Additional Example II - Continued

- Since D(s) is assumed as a unit step disturbance, the equation becomes:

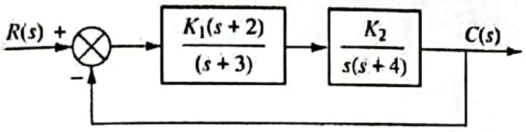
$$e_D(\infty) = -\lim_{s\to 0}\frac{G_2(s)}{1+G_1(s)G_2(s)}$$

- Substituting: e_D, G_1, G_2 and rearranging

- $0.00001 = \lim_{s\to 0}(\quad \dfrac{\dfrac{K_2}{s(s+4)}}{1+\dfrac{K_1K_2(s+2)}{s(s+3)(s+4)}} \quad \lim_{s\to 0}(\quad \dfrac{K_2(s+3)}{s(s+3)(s+4)+K_1K_2(s+2)} \quad \dfrac{3}{2K_1} = 0.00001$, thus $K_1 = 150{,}000$

Additional Example II - Continued

- When analysing the steady-state error due to input only, we assume that D(s) is equal to zero, and the system becomes:



- Since the input is ramp, the error due to only the input is $e_R(\infty) = \frac{1}{K_V}$ = 0.002 and we know that $K_v = \lim_{s\to 0}(sG(s)) = \lim_{s\to 0}\left(s \frac{K_1(s+2)}{s+3} \frac{K_2}{s(s+4)}\right) = \lim_{s\to 0}\left(\frac{K_1(0+2)}{0+3} \frac{K_2}{(0+4)}\right) =$ $K_1K_2/6$

So, $6/(K_1K_2) = .002$ and we already got K_1 = 150,000. therefore, $K_2 = 0.02$

أصعب أنواع الأسئلة → فيها دمج مع اللي أخذناه (Time response)

Additional Example III

A second-order, unity feedback system is to follow a ramp input with the following specifications: the steady-state output position shall differ from the input position by 0.01 of the input velocity; the natural frequency of the closed-loop system shall be 10 rad/s. Find the following:

→ K_V

- * The system type

  Since we are talking about steady state-error from input velocity, and the error is finite (not infinite and not zero) → Type 1

- * The exact expression for the forward-path transfer function ← لأنه Type1 ففيه Integrator

  Since we have a type 1 system, the simplest assumption about G(s) is a pole with an integrator $G(s) = \frac{K}{s(s+\alpha)}$ → أبسط صيغة لـ System فرضناها.

  For a Type 1 system, the error is $1/K_v$. The limit will give K/α so the error is $1/(K/\alpha) = 0.01$, or $K/\alpha = 100$.

  The overall transfer function for the unity feedback system with system with $G(s) = \frac{K}{s(s+\alpha)}$ is $T(s) = \frac{G(s)}{1+G(s)} = \frac{K}{s^2+\alpha s+K}$.

  Since it is given that ω_n is 10, this means that K = 100, and therefore $\alpha = 1$

  * $\frac{\omega_n^2}{s^2+2\zeta\omega_n+\omega_n^2}$

- * The closed-loop system's damping ratio $2\zeta\omega_n = \alpha = 1$. Thus, $\zeta = \frac{1}{20}$

Additional Example III

- Now, as a new exercise, find K and a, that will yield K_v of 1000 and an overshoot of 20%

$$\boxed{\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100}$$

For a 20% overshoot, this means $\zeta = 0.456$

② $K_v = 1000 = \frac{K}{a}$ $T(s) = \frac{K}{s^2+as+K}$ $\omega_n = \sqrt{K}$ $2\zeta\omega_n = a$

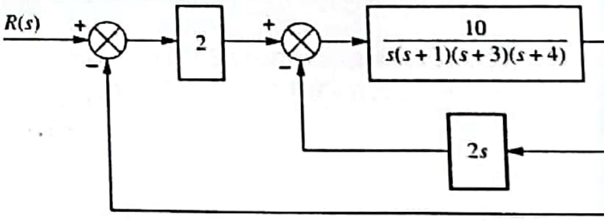
① $a = 0.912\sqrt{K}$

* عوض معادلة ①
و ② ببعض بطلع
معك K و a

Substituting a in the Kv equation, this will yield K = 831744 and a = 831.744

Additional Example IV

Given the system in the adjacent figure, find the following:



- The closed-loop transfer function:

$$G_1(s) = \frac{\frac{10}{s(s+1)(s+3)(s+4)}}{1+\frac{20}{(s+1)(s+3)(s+4)}} = \frac{10}{s(s^3+8s^2+19s+32)}$$

$$\text{Type 1} \longleftarrow G_e(s) = \frac{20}{\underline{s}(s^3+8s^2+19s+32)}$$

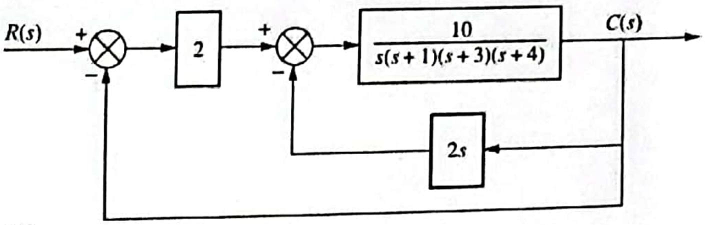
$$T(s) = \frac{G_e(s)}{1+G_e(s)} = \frac{20}{s^4+8s^3+19s^2+32s+20}$$

- The system type

Since there is one integrator in $G_e(s)$, this means a Type 1 system

Additional Example IV

Given the system in the adjacent figure, find the following:



* - The steady-state error for an input of 5u(t)

E_Step = 0 ⟶ لأنه Type1

* - The steady-state error for an input of 5tu(t)

From $G_e(s)$, $K_v = \lim_{s\to 0} sG_e(s) = \frac{20}{32} = \frac{5}{8}$. Therefore, $e_{ss} = \frac{5}{K_v} = 8$.

* - Discuss the validity of your answers to Parts c and d.

```
>> roots ([1 8 19 32 20])

ans =

   -5.4755 + 0.0000i
   -0.7622 + 1.7526i
   -0.7622 - 1.7526i
   -1.0000 + 0.0000i
```

معاهل الSystem
Stable ولا لأ؟ ولازم
للSystem كامل يكون.

References

- The material in these slides are based on:
  Control Systems Engineering, Norman S. Nise, 7th Edition (2014), John Wiley And Sons
- **Chapter 7 – Steady State Errors**
- Sections *7.1, 7.2, 7.3, 7.4, 7.5, and 7.6*, (Students kindly note that some sections involve math derivations that we did not cover in class

Quick Intro to PID Controllers

Proportional, Derivative and Integral Control

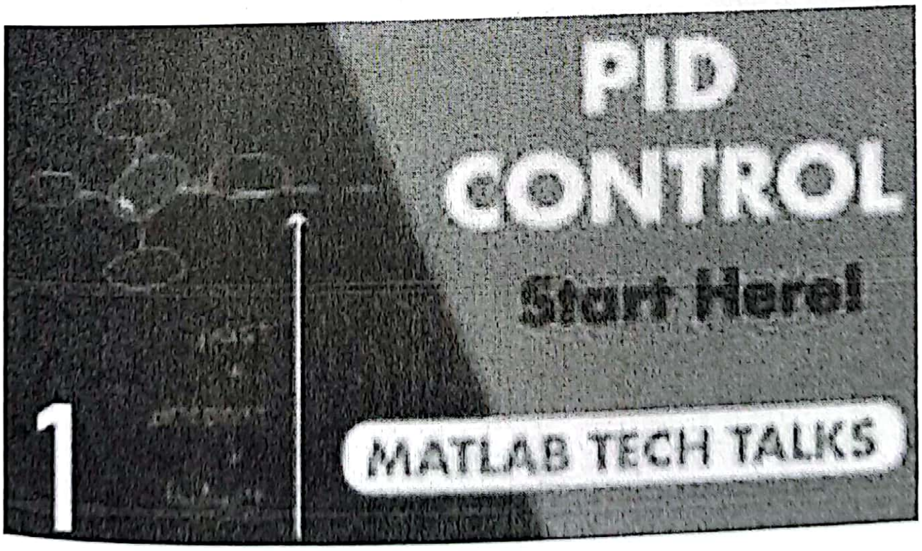
Dr. ASHRAF E. SUYYAGH

THE UNIVERSITY OF JORDAN

DEPARTMENT OF COMPUTER ENGINEERING
FALL 2022

PID Control – Intuitive Explanation

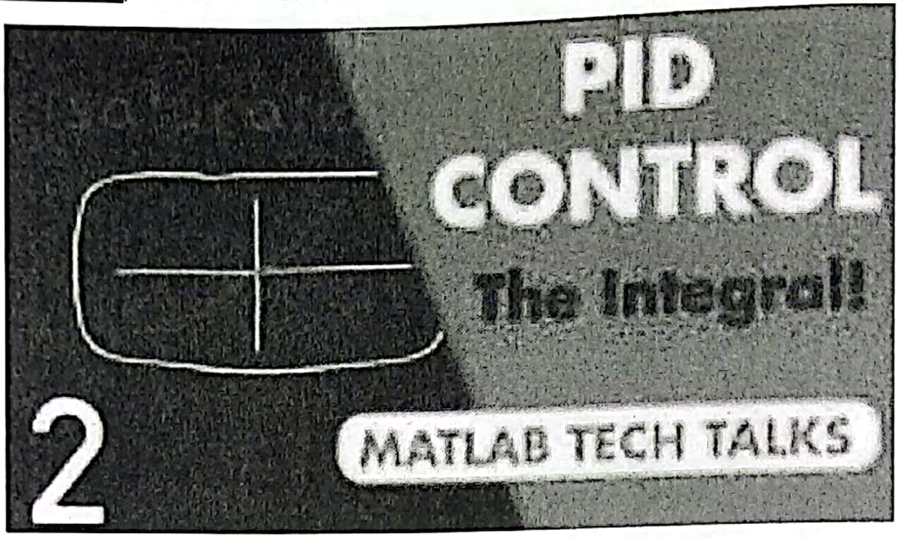
https://www.youtube.com/watch?v=wkfEZmsQqiA&list=PLn8PRpmsu08pQBgjxYFXSsODEF3Jqmm-y
Or in ***presentation mode***, click on the video below



3

Integral Windup Problem– Intuitive Explanation

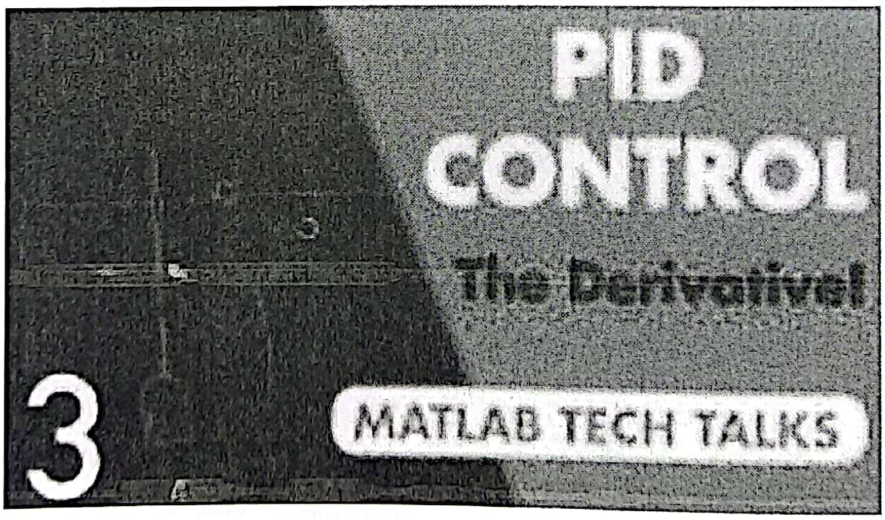
https://www.youtube.com/watch?v=NVLXCwc8HzM&list=PLn8PRpmsu08pQBgjxYFXSsODEF3Jqmm-y&index=2
Or in ***presentation mode***, click on the video below



4

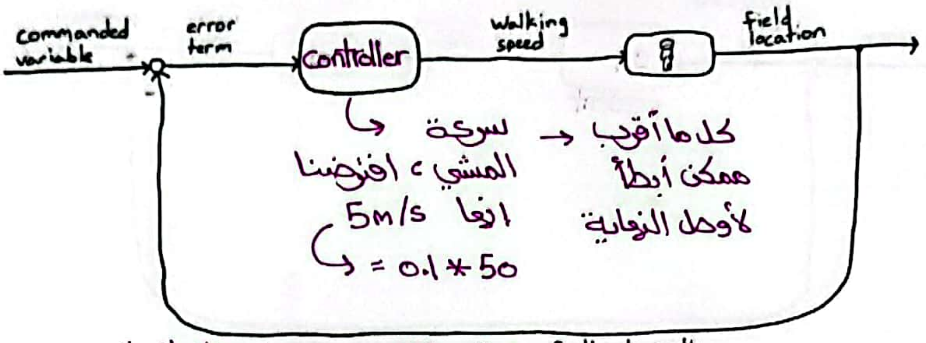
Noise Filtering in PID Control– Intuitive Explanation

https://www.youtube.com/watch?v=7dUVdrs1e18&list=PLn8PRpmsu08pQBgjxYFXSsODEF3Jqmm-y&index=3
Or in ***presentation mode***, click on the video below



PID Controller
Part 1

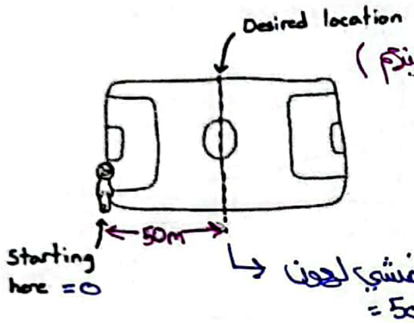
→ أشهر أنواع الـ Controller (linear systems)



سرعة المشي ، افترضنا انها 5m/s
= 0.1 * 50

→ كل ما أقرب ممكن أبطأ لأوصل النهاية

feedback path ← قديه أنا بعيد عالمسافة النهائية (وين بدي أروح ووين أنا موجود الفرق بيلزم)



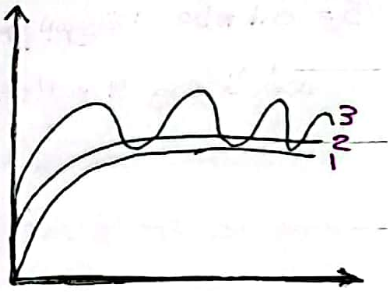
بدنا نمشي لهون = 50 m

بتقل كل ما يقرب لـ Desired location

* حنفرض إنه سرعة الإنسان

* عند نقطة البداية مقدار الخطأ = 0

* حضل أمشي لحد ما يصير الفرق = 0



← بحب أشوف الـ Transient response تبعه زي 2 (يكون الطلوع Smoothe)
والأهم انشي ما يكون undamped ولا يكون بطلع بسرعة قوية ممكن تأثر سلبًا.

* وظيفتنا الآن حنضيف controller سواء كان SW أو HW نصير نغير من الشكل تبع الـ System حتى نحصل عالشكل اللي أنا بدي إياه. (حنشوفه نظريًا)

* PID controller هم مجموعة من الـ Controllers :

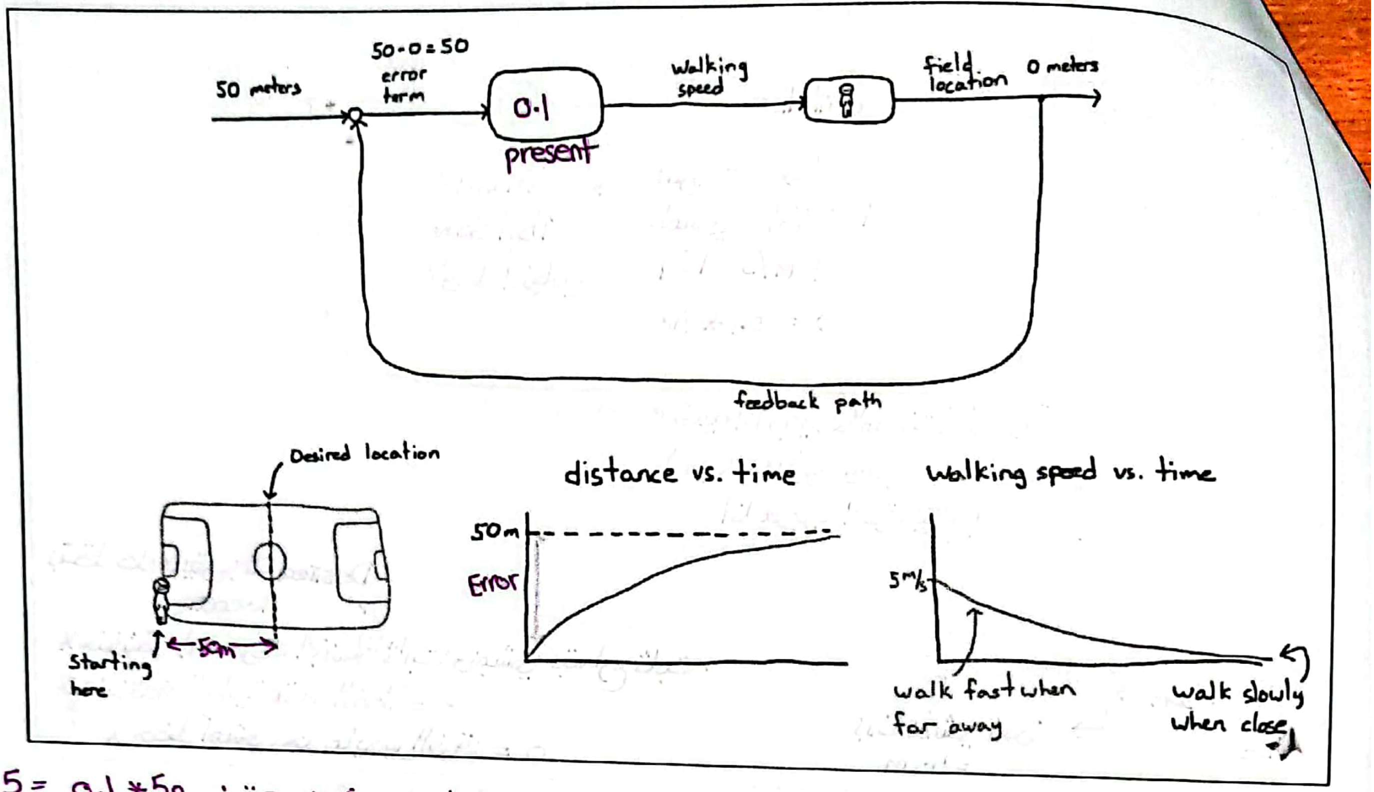
- P : اللي كنا نشوفه طول الوقت (الـ gain / K) (كنا نقول عنه Amplifer)
- I : Integrator
- D : Derivative

* دائمًا بكون عندي يا إما P / PI / PD / PID

* الـ P وجودها أساسي.

* أخذنا إنه دائمًا عنا بالـ feedback sys فيه Plant يقسم لأجزاء (actuator / Process / ---).

* بهاي الصورة بنشوف كيف الـ P controllers بتنجح بحالات بس بعدين ممكن تفشل فشل ذريع بحالات ثانية لو كانت لحالها.

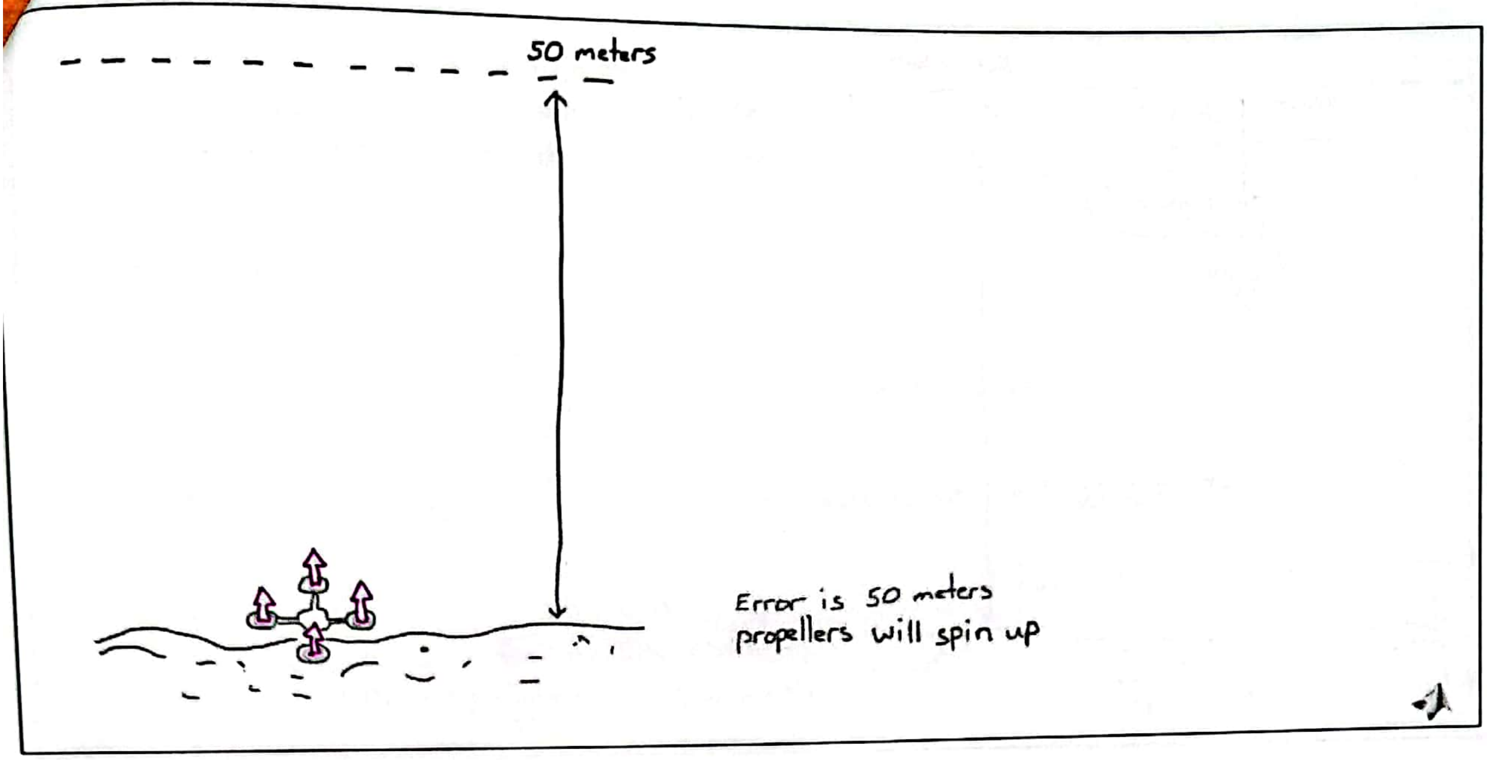


← هون الـ Controller تبعي فقط عبارة عن gain بيساوي 0.1 ، بالبداية بمشي بسرعة : 5 = 0.1 *50

بعدين كل ما أقرب للـ desired location ببطئ : 0.1*45 = 4.5 ، 0.1 * 40 = 4 وهكذا لحد

ما أوصل للمكان المطلوب.

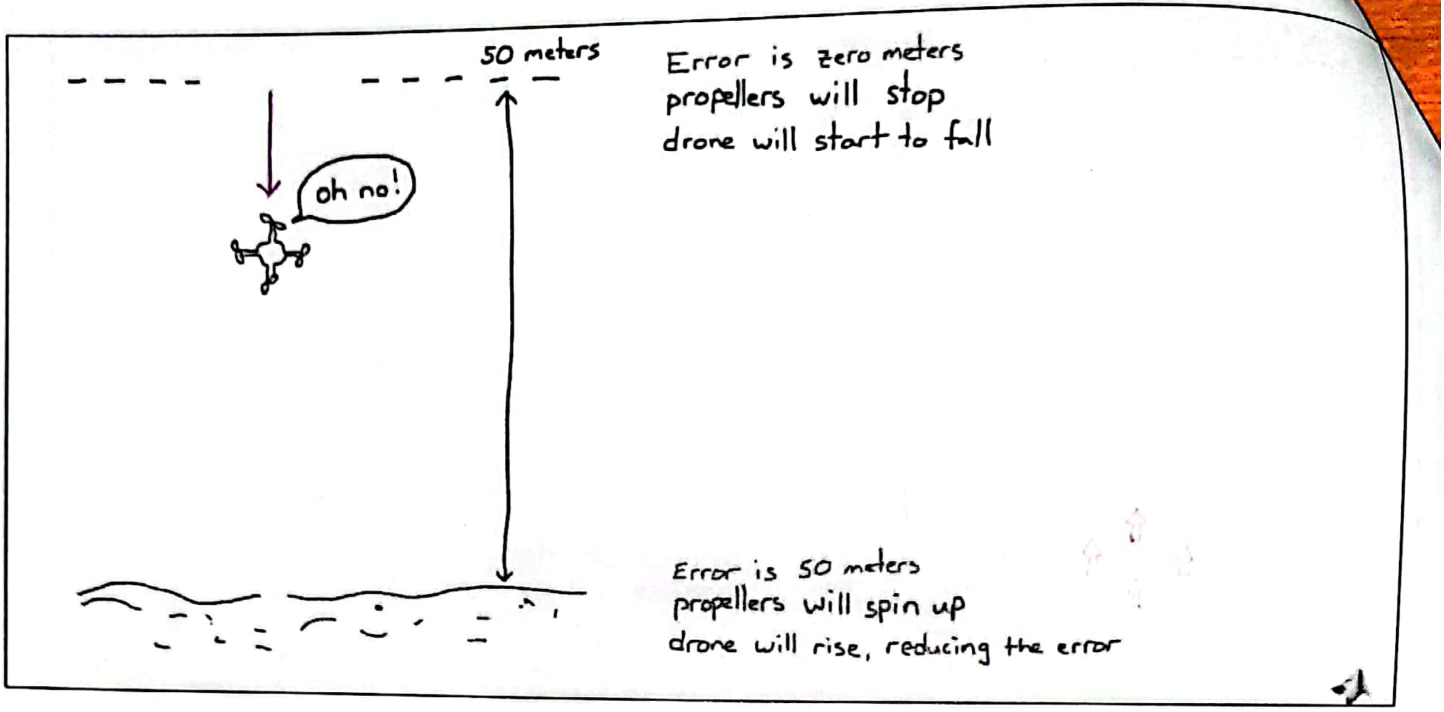
* هون الـ P لحاله كان كافي.



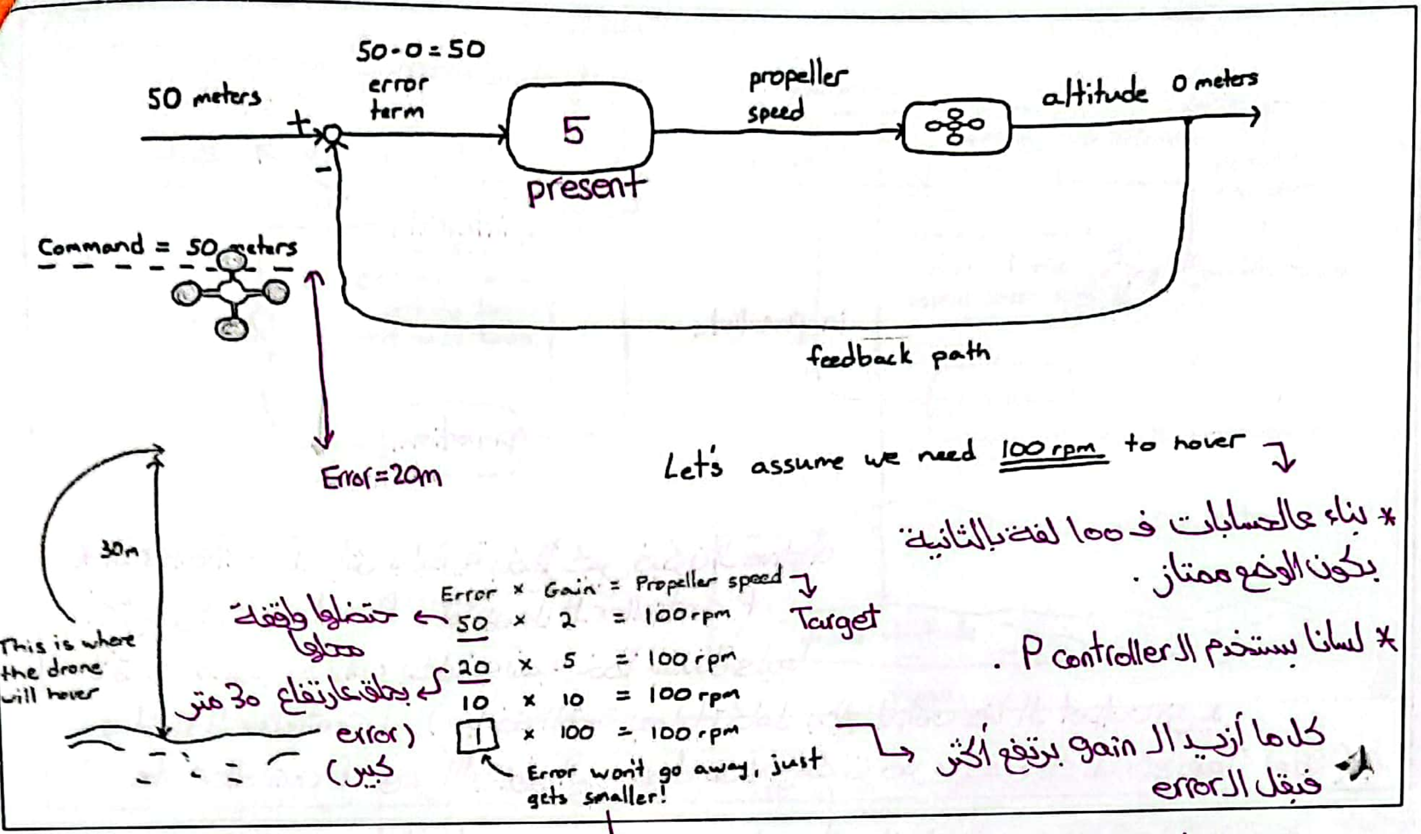
* غيرنا هون الـ App وبدل الإنسان حطينا drone بدنا إياه يوصل من الأرض لارتفاع 50m .

* الـ drone اله مراوح بتشتغل بسرعة معينـــة ، بالبداية الخطأ بكون 0 بتشتغل بتبلش تطير بسرعة معينة وكل ما نقرب بتقل سرعتها لحد ما توصل الـ target . الـ error بكون = 0 فلما وصل الـ target طفا الـ Controller لإنه الفرق أصبح 0 فالـ control signal = 0 فبوقع الـ drone عالأرض .

* فالـ P controller لحالها هون ما اشتغلت صح .



*حنستخدم هون الفيزياء شوي بحيث إنه أنا بدي هذا الـ drone يطير ويوقف بمكان معين
يعني محصلة القوة عليه لازم تكون zero . محصلة القوة الموجودة عالـdrone هون هي الجاذبية الأرضية
فلازم يكون عندي قوة رفع مساوية إلها كشان تصير المحصلة = 0 و يضل واقف مكانه.



Let's assume we need 100 rpm to hover

* بناء عالحسابات فـ 100 لفة بالثانية بكون الوضع ممتاز .

* لسانا بنستخدم الـ P controller .

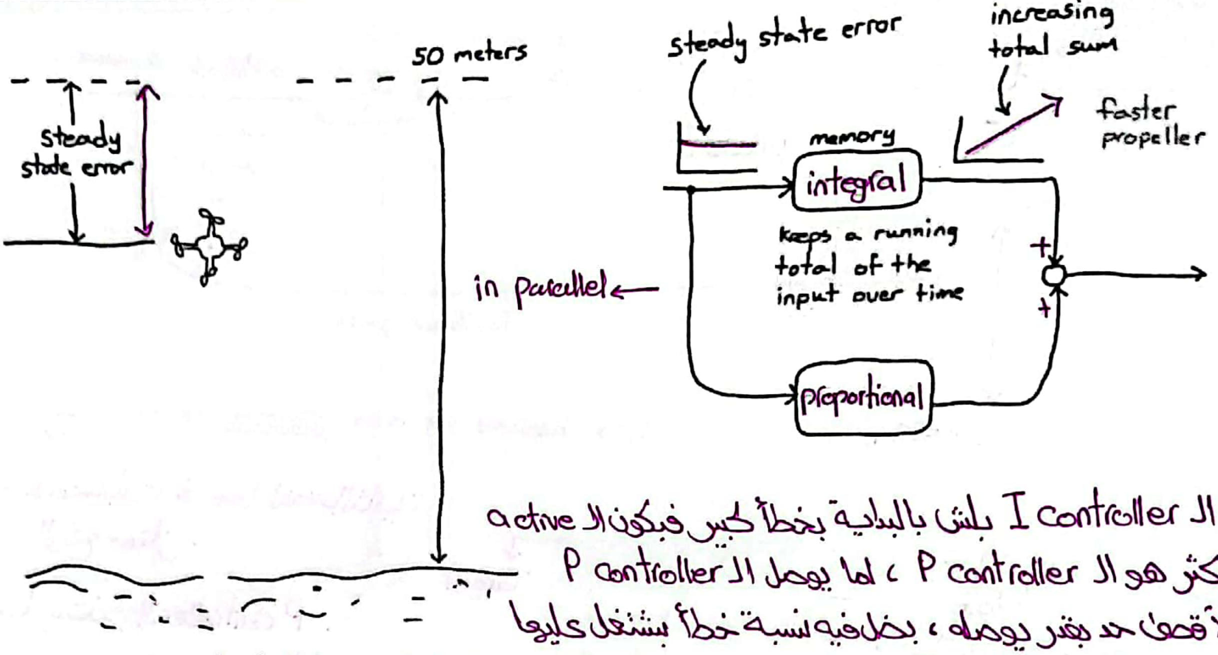
Error	×	Gain	=	Propeller speed
50	×	2	=	100 rpm
20	×	5	=	100 rpm
10	×	10	=	100 rpm
$\boxed{1}$	×	100	=	100 rpm

* كل ما أزيد الـ gain برتفع أكثر فبقل الـ error

* حتى يكون الـ error =0 الـ gain لازم تكون ∞
وهذا مستحيل يصير

* فأنا المغزى هون إنه باستخدام الـ P controller فقط بوصلنا الـ App حيكون الخطأ دائما Finite (حيضل
فيه error) فأنا مش كافي لحاله فطلع الـ I controller .

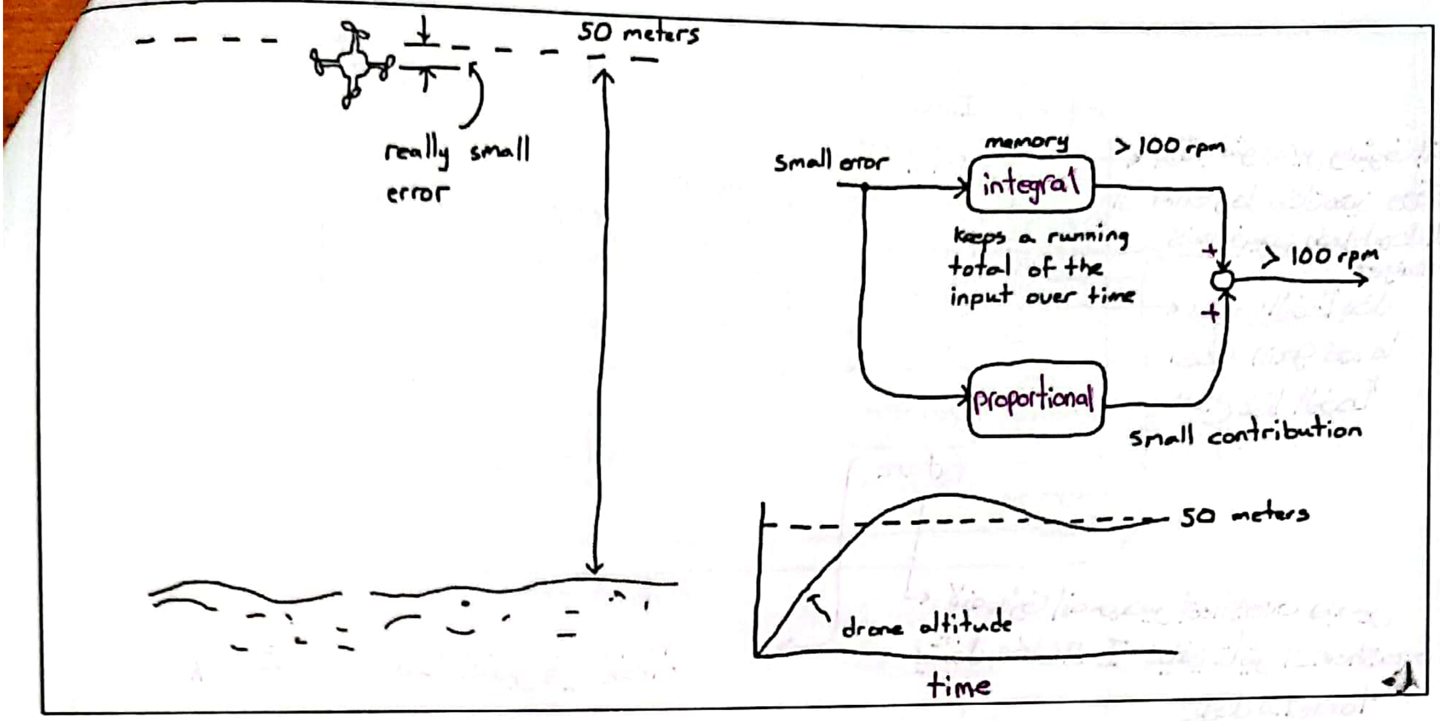
* إضافة Integrator يعني إضافة block بالـ hardware أو Code بالـ Software أو موجودة بمكتبات
جاهزة.



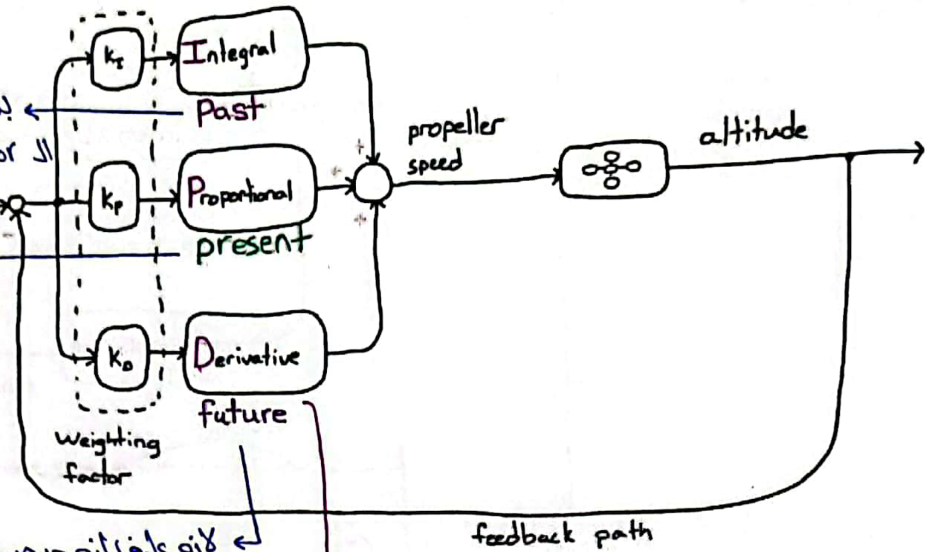
* ال I controller بلش بالبداية بخطأ كبير فبكون ال active
أكثر هو ال P controller ، لما يوصل الـ P controller
لأقصى حد بقدر يوصله ، بضل فيه نسبة خطأ بشتغل عليها
ساعتها الـ I controller (بياخذ القيمة بعملها تكامل بطلع output هنا ال output بزيد
عال P controller وهو اللي برفع ال drone من آخر مكان قدر يوقف فيه لل Target (50)).

* هيك بصير الـ Steady-state-error = 0 وصار الواقع بطابق الرياضيات.

* هون وجود ال I مع ال P كان أفضل من إنها تكون لحالها.



* هون بنوضح مشكلة ال I controller : لما يبلش يشتغل ليوصلنا لل target (50) بيقفز فجأة فوقه
بيعملي overshoot ، مش بس وصل ال steady-state لأ تعداها كمان. بالأخير حنوصل لل steady-state
أكيد بس ال overshoot مشكلة يعتبر لإنه رفع ال drone ل 52 فصار الخطأ بساوي 2- (يعني
مثلاً بدل ما السرعة تكون مما خلاها 120) فهذا حيضل يعمل overshoot / undershoot لحد ما يوصل
وهذا اشي مش منيح فحله كان بإضافة ال D controllers .



براقب شو بصير وبشوف إنه ال error وصل لمكان معين ووقف عنده فبصير يكمل ليوصل ال target

بشوف الخطأ بضل أعطيه gain لحد ما يروح هذا الخطأ

لإنه عارف إنه حيصير overshoot فبصير يشتغل عكس ال I ليلغي تأثير ال overshoot ويوصل للtarget بطريقة Smoothe

* بتشتغل على معدل التغير في الخطأ.

* فلو بشوف إنه معدل التغير كبير وسريع بتقلله وإذا كان بطيئ بسرعه.

* شغلها معاكس للـ I controller عشان تنحل المشاكل.

* فهيك بتلغي تأثيرات الـ overshoots اللي نتجت من ال Integral اللي خلاّ ال drone يقفز بسرعـة.

* وجود الـ 3 controllers هدول مع بعض بحلّ ٩٠٪ تقريباً من المشاكل لإنه كلهم بشتغلوا مع بعض.

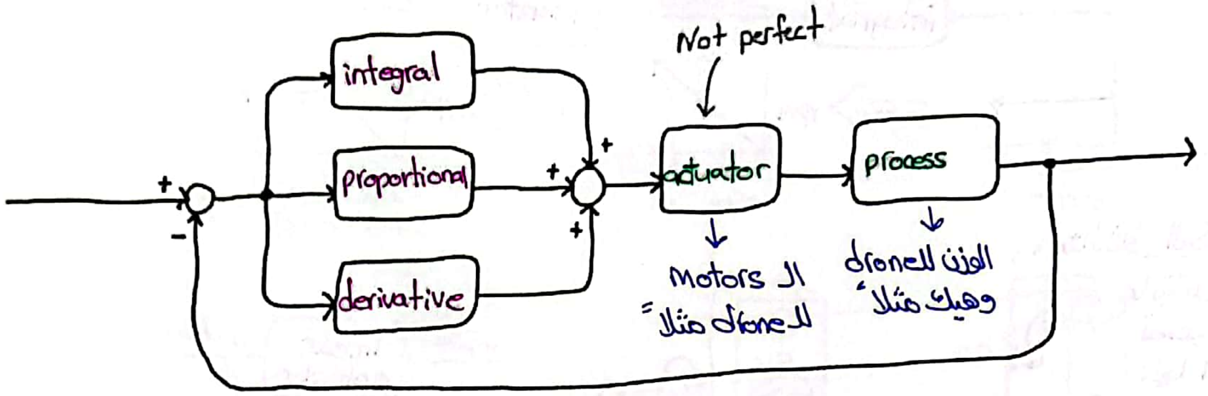
* مش شرط بكل ال Apps يكونوا ال 3 موجودين ، احنا بنقرر الأفضل بناءً على ال App وتصميمه.

* احنا من مواقع ال Zero وال Pole كنا نعرف شو نوع ال System (damped / Underdamped / over damped / Critically damped) ، بالكتاب بيشرح عن اشي اسمه root locus بال Matlab في tool بتغنينا عن هاد ال root locus.

* root locus بشكل عام كانوا يرسموا diagrams ويلعبوا بال Zero وال Poles بحيث يوصلوا للي بدهم ياه ، كله بالرياضيات. أما احنا حنشتغل بإنه نعرف ال gain وال Parameters اللي لازم أحطها بال Sys سواء SW أو HW عشان تعطيني الشكل اللي بدي ياه. يعني بالآخر نطلع (Kp, KD, KI) ← هدول مو تلوين الخطأ تبعون ال Grains.

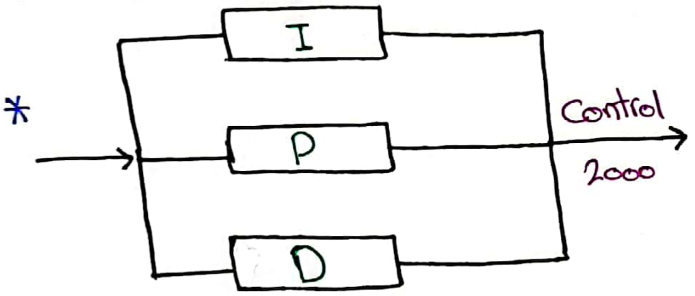
Part 2

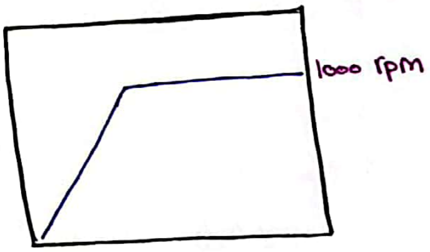
PID control



* منمشاكلنا بالواقع إنه أنا مفترض إنه الـ System عندي ideal و linear بس أول مشكلة بتواجهنا بالواقع هي الـ Physical limitations of the sys و كنا مفترضين إنه الـ PID controllers أي Signal بتطلع منهم قادر إنه يستجيب إلها الـ Motor .

*

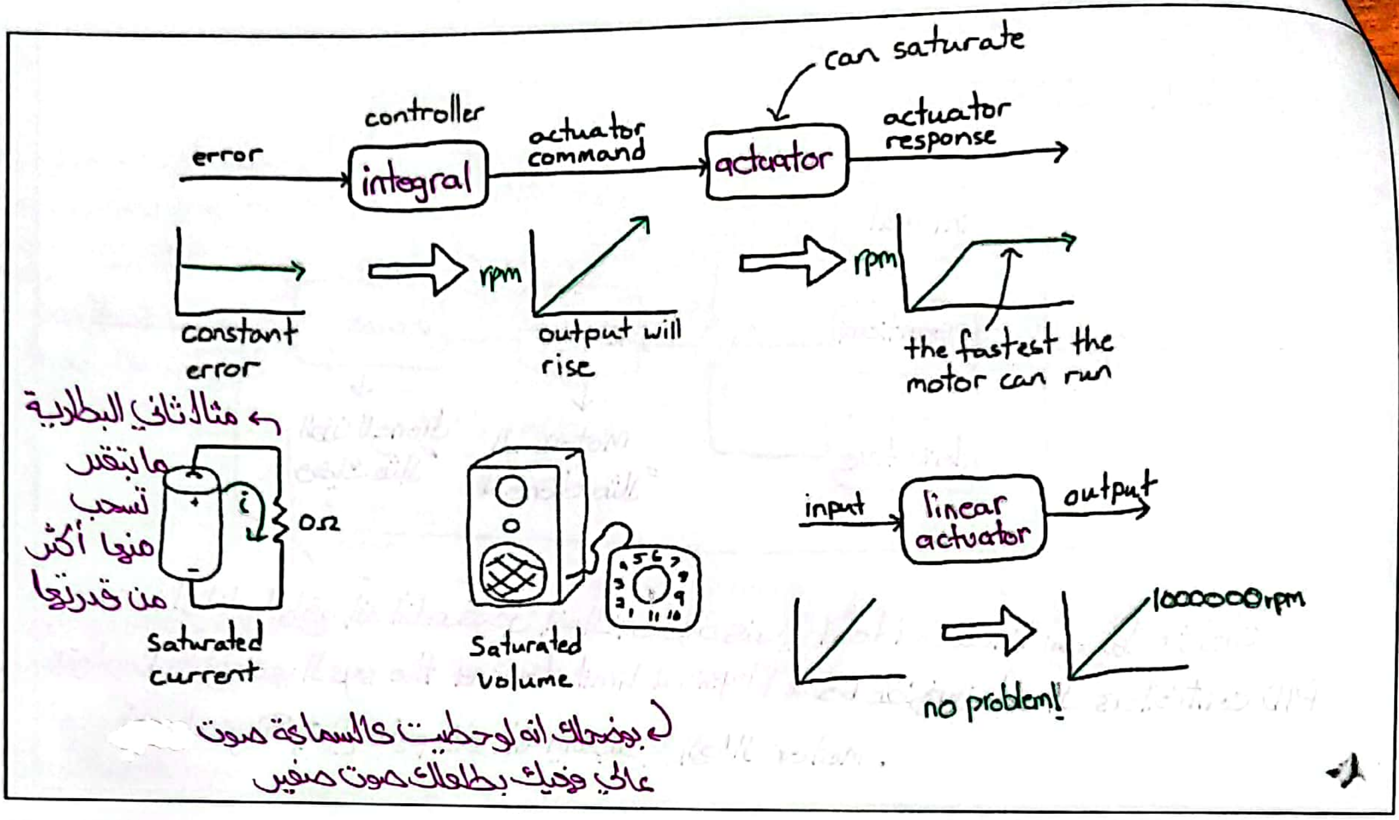




* لومفترضين إنه linear الـ actuator وهذا مش واقع (أي قيمة لـ X بلاقيلها قيمة لـ Y) فعليا هو ما بصير هيك .

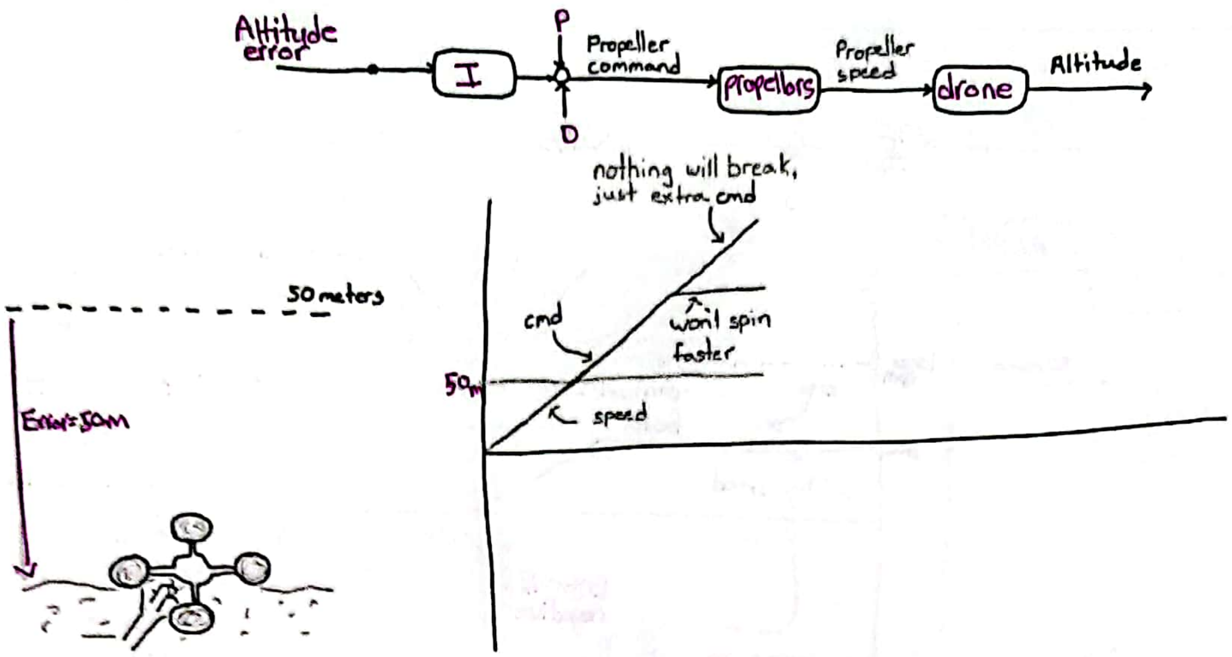
* فعليا الـ Motors زي ما هو مرسوم إلها سرعة قصوى شو ما أعطيته أكثر منها حيضل يلف عليها بس (إلها Saturation) ، فعدم استجابته للـ Control إنه يكون 2000 هذه مشكلة .

* بالوضع الطبيعي لو ما في Disturbance عادي بس بالواقع مش هيك الوضع .



مثال ثاني البطارية
ما بتقدر
تسحب
منها أكثر
من قدرتها

بوضحلك انه لو حطيت عالسماعة صوت
عالي وفيك بطلعلك صوت ضعيف



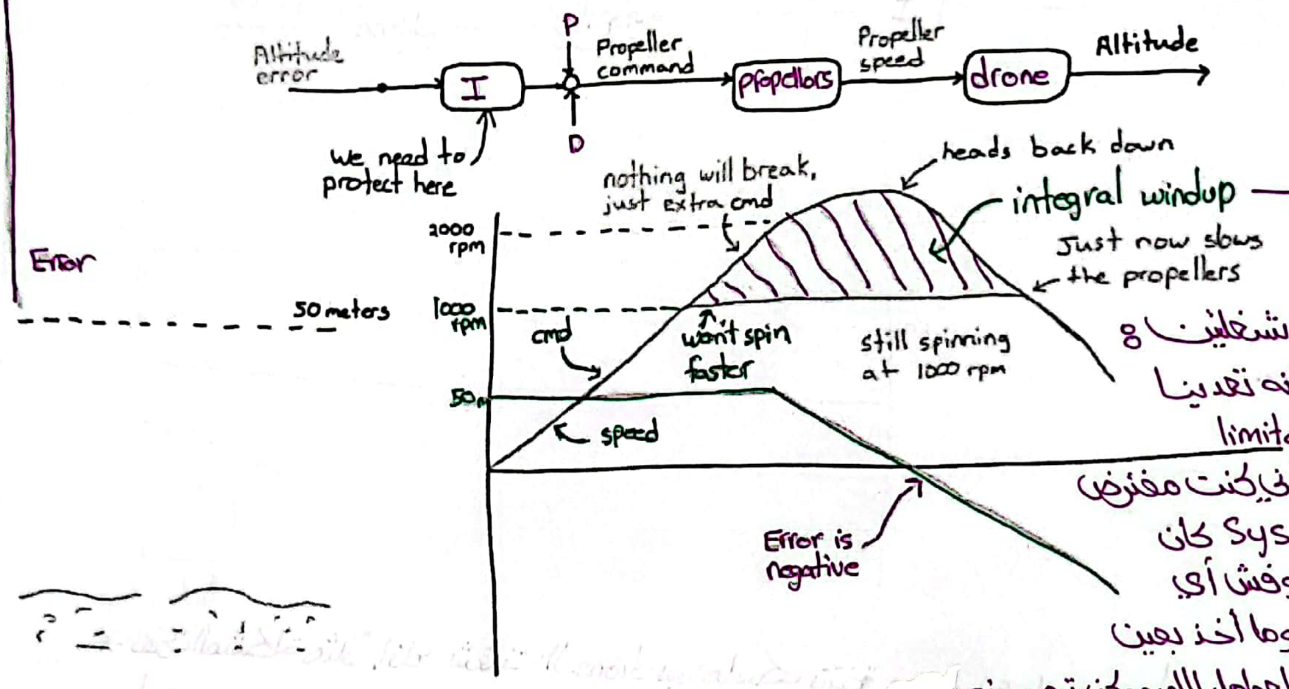
* هون المشكلة مثلاً ، انك شغلت الـ drone بس ماسكه بتنفقده بإيدك لسبب ما .

← بالـ P : شايف إنه في فرق وقاعد بيعطي الـ Signal أمر إنه تطير .

← بالـ I : واقف براقب الماضي وشايف إنه الوقت بمضي والـ error مازال نفسه ففكر انه الشغل
هلأ عليه فبستلمه وبزيد سرعة الـ motor ومازال الـ drone ماهو يتحرك لإنه ماسكه
بإيدي . بس أقرر أتركه يطلع ففجأة بلاقيه طلع بسرعة مهولة جدًا وسرعته كانت فوق اللي
مطلوب وماشاف الـ steady-state وطلع زيادة .

← مع إني بزيد بسرعته لإنه كان ممسوك بالبداية بس السرعة المحددة هنا لازم تكون 1000 كحد أقصى
(مشي كسرعة 1000 رغم إنه الـ order كان 2000 مثلاً) فهلا الـ Integral لما يشوف هذا الحكي
بصير يعطي أوامر لـ drone إنه ينزل من السرعة لـ 1900 بعدين 1800 وهكذا . فهذا الوقت اللي
بقضيه الـ drone منحد ما طلع كثير لحد ما ينزل للحد الأعلى للسرعة بيسمى بالـ windup (الصورة المقابلة)

صار
يعكس
شغله



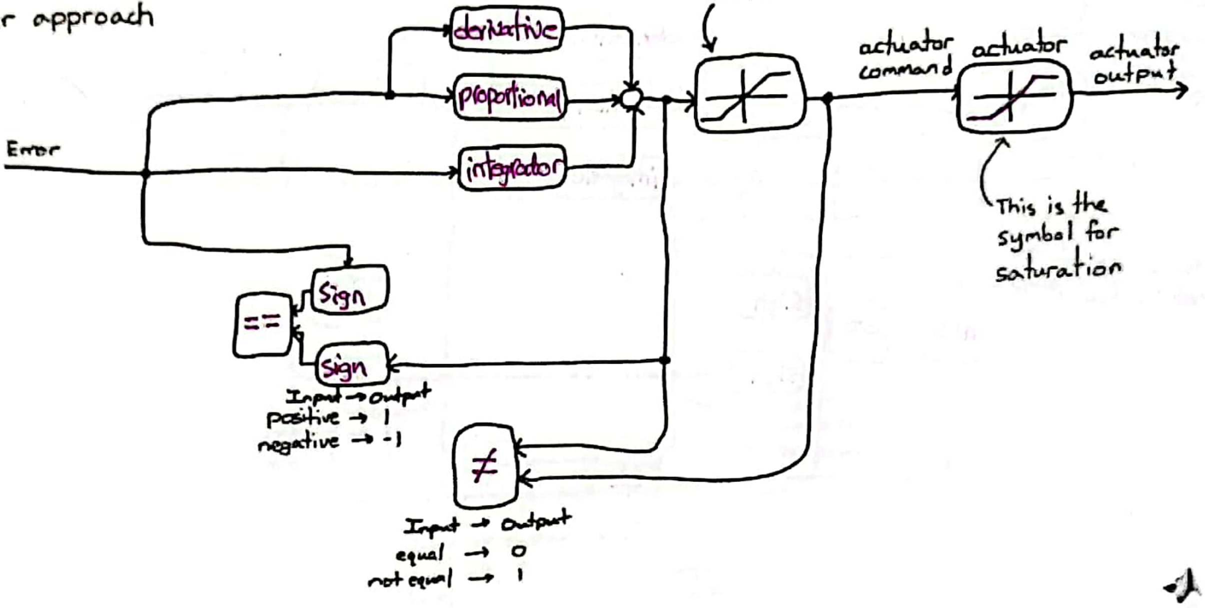
لسببها شغلتين :
الأولى إنه تعدينا
الـlimitations
والثانية اني كنت مفترض
إنه الـ Sys كان
linear وفش أي
مشاكل وما أخذ بعين
الاعتبار العوامل اللي ممكن تصير زي
إنه أكون ماسك الـ drone أو
يخبط فيه عصفور أو هيك .

Integrator anti-windup – Keep the integrated value from increasing past some specified limit.

Clamping – turning the integrator off

Back-calculation

Observer approach



* الحل لمشكلة الـ windup ؟ إضافة circuit أو (if statement) SW أو Comparator بقارن الـ order مع الواقع .

* متى بعرف إنو فيه Saturation ؟ لما يكون يعطي أمر إنه طير بسرعة 50 وفعليا الـ output = 50
فما عندي مشكلة . (اللي بعطيه ياه هو نفسه اللي بطلع) .

* متى بعرف إنه تجاوزت الـ limitations ؟ لما أعطيه مثلاً طير s 1200 وبعطيني 1000 ولا 1300
وبرضه بعطيني 1000 وهكذا .

* فلما مايكون الـ Control اللي بطلع جوابه مساوي لـ actual بعرف إنه أنا بتعدى الـ Physical limitations .

* لو الـ order مع الواقع متساويين الوضع تمام لو مو متساويين بعرف إني وصلت حد الـ Saturation .

* المشكلة الثانية إنه الـ error موجب (عندي خطأ) ومازال الـ Controller بيعطي أمر طيران .

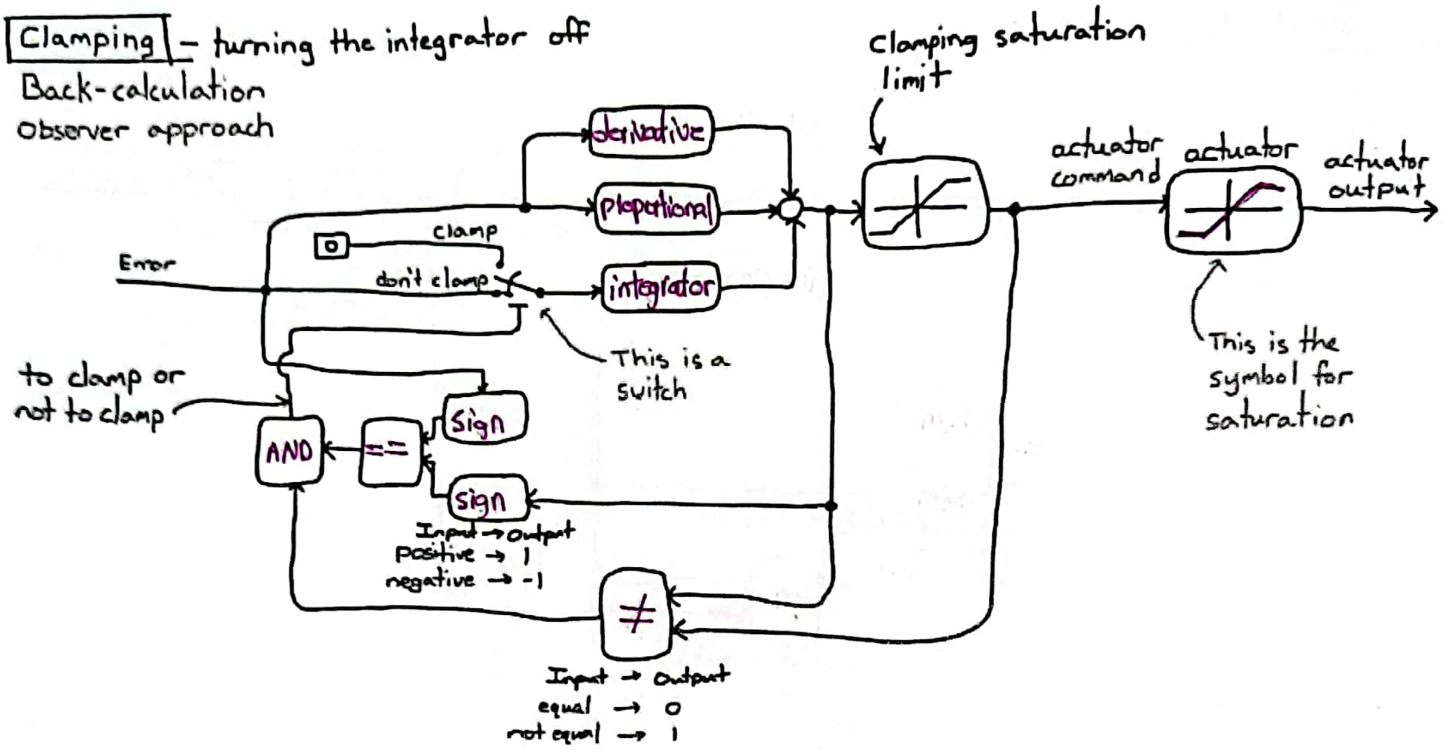
* ففعليا لو عندي windup ومازال الـ Controller شغال وبعطي أمر طلوع أو نزول (الـ error والـ
Controller نفس الإشارة معناته بيعطو signal order للطيران أو هناك شغال يا Controller) كلو تحققوا هالأمرين
بنعرف إنه في عنا مشكلة بحلوها بإضافة Switch كال Integrator . (باللحظة اللي بنوصل فيها الـ Saturation
والـ Controller شغال أفصل لإنه كل المشكلة منه لـ Integrator) هذا بيسمى (clamping)

Integrator anti-windup - keep the integrated value from increasing past some specified limit.

Clamping - turning the integrator off

Back-calculation

Observer approach

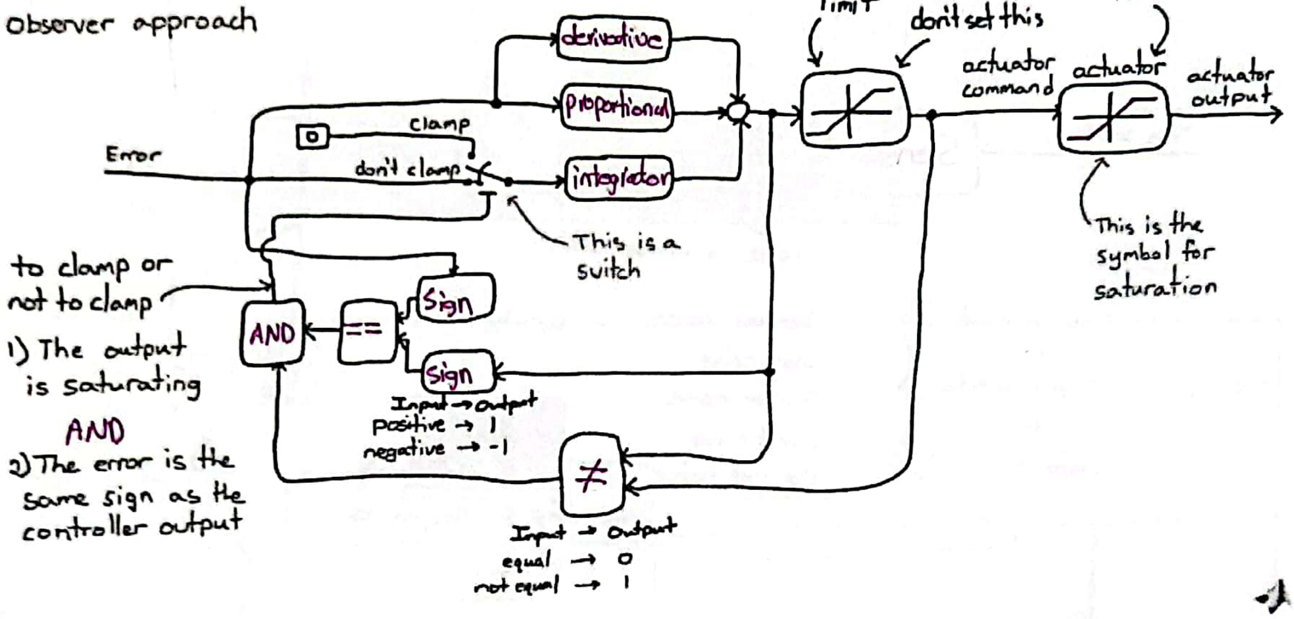


Integrator anti-windup - keep the integrated value from increasing past some specified limit.

Clamping - turning the integrator off

Back-calculation

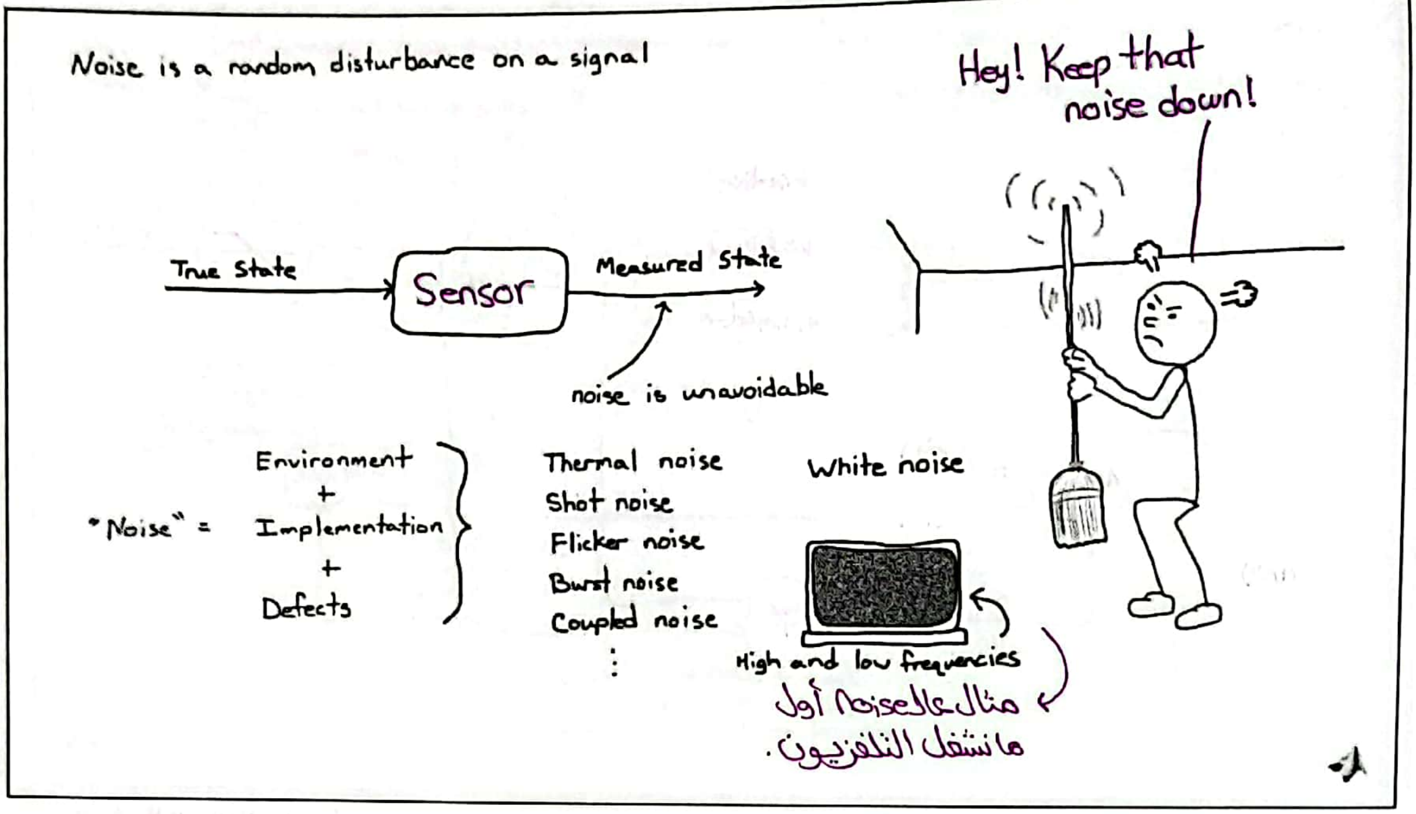
Observer approach



* شو القيمة اللي لازم أحطها للـ Clamping ؟ مثلاً أنا حاكي إنه الطاقة الاستيعابية لمراوح الـ drone هي 1000 rpm أقصى اشي ، فهل الـ Clamping أحطها 1000 ولا أعلى ولا أقل ؟ دايما بنحطه أقل (900) مثلاً عشان في إله تقدم بالعمر فمارح يعطيك قد ما كان يعطيك أول ما اشتغل فحطه أقل من العمر اللي محطوطه بالبدايـــة.

Part 3

* المشاكل اللي بنواجها الـ derivative controller وكيف بنحلها.

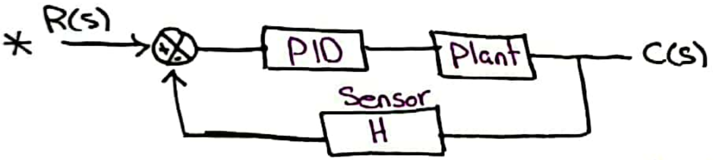


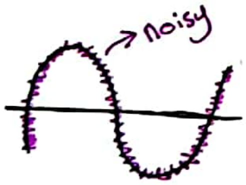
Noise is a random disturbance on a signal

* الـ feedback بنقيس فيه الـ out وبنرجعه لجهة الـ in مشان نشوف الفرق بين الـ desired والـ out الـ actual signal ونعدل بناءً عهذا الناتج ، وهذا الـ feedback بنحصله بتركيب Sensors.

* في مشاكل بالقراءة من الـ Sensor مثل الـ noise أو الـ thermal noise اللي بتعتمد عالـ Circuit وقوتها وتصميمها.

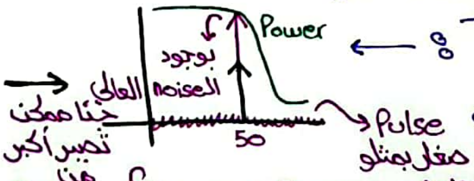
*





Time Domain

50 Hz → Nyquist = 100 Hz



frequency Domain الـNoise

*مشكلتنا ع الـDerivative بتشتغل كمعدل التغير بالخطأ والـProportional بضرب الخطأ بـ gain والـ Integral بكامل الخطأ وبتشتغل عليه فلو لنشوف وجود الـ nois بالـTime/freq. domain ؛ ←

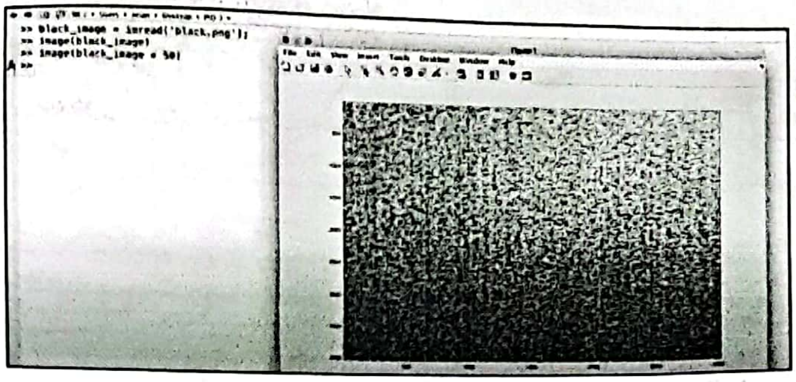
* الـ Pcontroller بقلل من وجود الـ Noise بالـSensor

بس المشكلة هنا بالـD controller اللي بتشتغل كمشتقة الخطأ ، لما الـ Signal يجي عليها noise سوا (low freq) فمشتقته قليلة ، بينما لو كان الـNoise كثير عالي فالمشتقة بتكون كبيرة فبتأثر بشكل كبير فهذا بنحله باستخدام الـ Filters (مابلغي الـ noise بخفف من تأثيرها)

$\frac{de}{dt}$

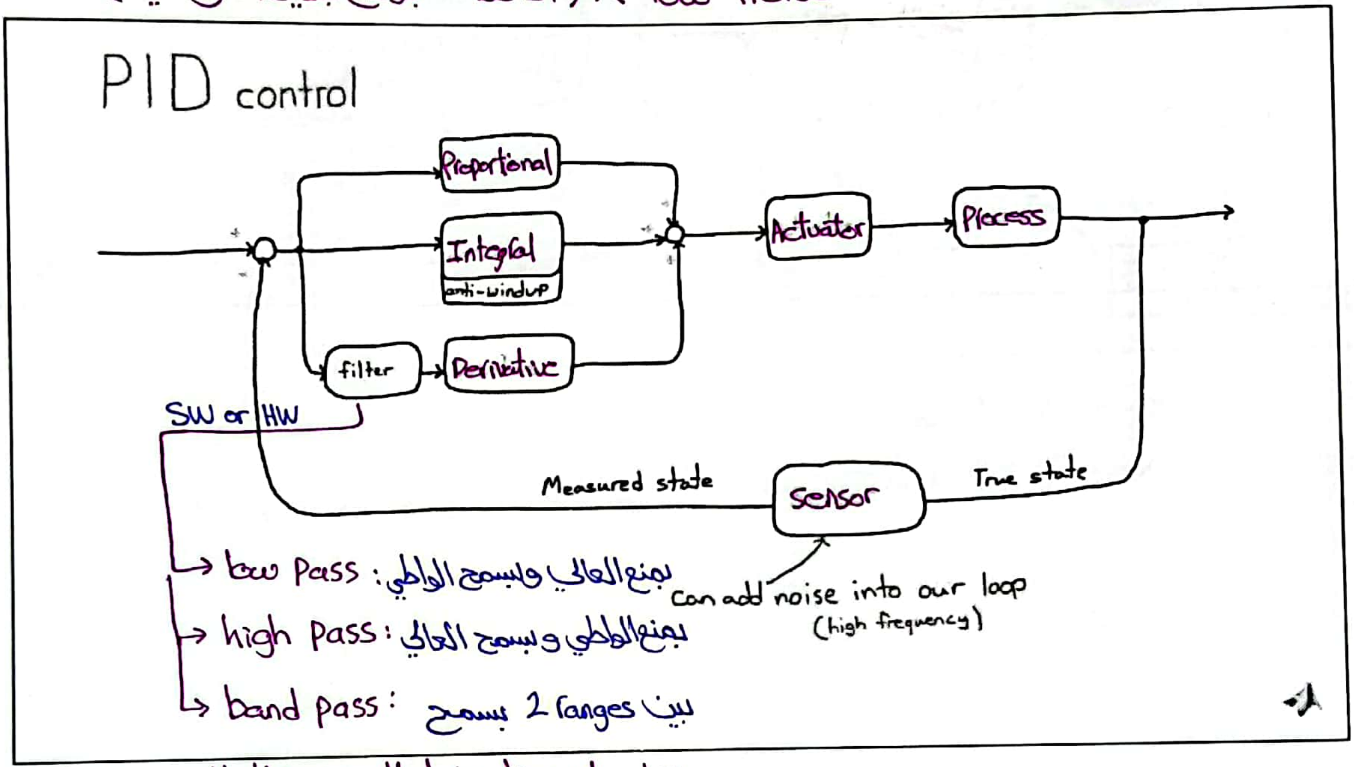
← بدلما يعطيني قيم صحيحة بصير يعطيني قيم خاطئة ← الـ Cutoff تبعه احنا بنقرر حسب الـ System . هون مثلاً 50

ما بتعتمد عالـ Signal بتعتمد عالـNoise.



```
>> black_image = imread('black.png');
>> image(black_image)
>> image(black_image + 50)
>>
```

→ تجربه كاميرا التلفون عند تصوير شاشة سوداء وزيادة الـ Brightness لأعلى اشي لوما ظهور حبيبات فجودة الكاميرا بتكون جيدة (low noise). (الـ noise بوضح بالليل أكثر اشي)



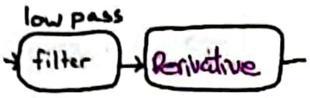
→ low Pass : يمنع العالي ويسمح الواطي

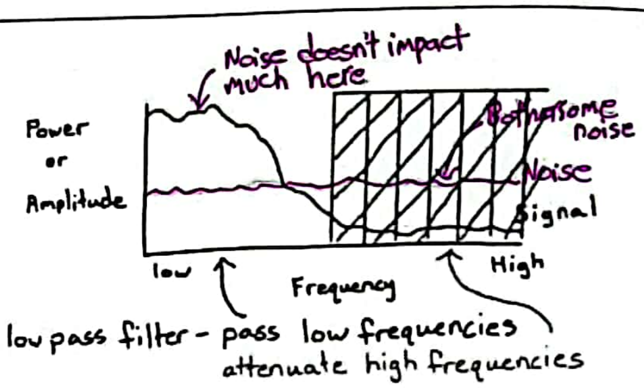
→ high Pass : يمنع الواطي ويسمح العالي

→ band pass : بين 2 ranges بسمح

* احنا بهمنا بنشيل الـ noise العالي قدر الإمكان قبل ما توصل للـ D controller .

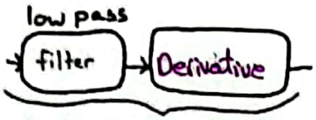
* يعني بنحتاج low pass Filter .



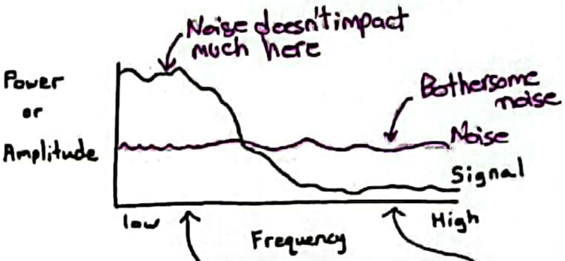


* الفكرة الأساسية هنا ال Signal اللي
بدي ياها بحط عليها cuttof freq.
فاللي بدي ياها وقبل بيمرق واللي بعد
بتم فلترتها.

«كيفية تصميم ال Filter» ← تصميمه بال Frequency Domain كتير أسهل

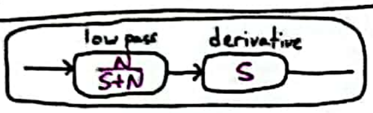


let's look at the structure of this



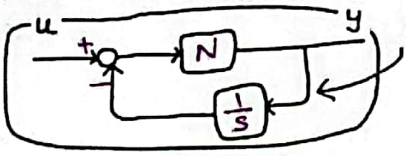
low pass filter - pass low frequencies
attenuate high frequencies

Laplace domain transfer function	what it is
S	derivative
$\frac{1}{S}$	integral
$\frac{N}{S+N}$	low pass filter with cut off at N rad/s



$$\rightarrow \boxed{\frac{SN}{S+N}} \rightarrow \quad \Leftarrow \text{equal} \Rightarrow \quad \rightarrow \boxed{\frac{SN}{S+N}} \rightarrow$$

An alternative approach:



An integral in the feedback path

$$\rightarrow \boxed{\frac{N}{1+N\frac{1}{s}}} \rightarrow$$

$$y = N\left(u - \frac{y}{s}\right)$$

$$y + \frac{Ny}{s} = Nu$$

$$y = \frac{N}{1+N\frac{1}{s}}\, u$$

$$\boxed{\frac{y}{u} = \frac{N}{1+N\frac{1}{s}}}$$

* بنعرف من أول بـ laplace إنه المشتقة بنضرب بـ S والتكامل بنقسم S S .

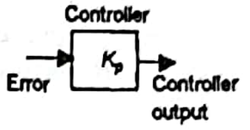
* وجدوا إنه تصميم ال Filter المناسب هو : $\frac{N}{S+N}$ وال N هو ال Cut off freq لل low pass Filter.

* فبقدر أصمم ال low pass مع ال Derivative باستخدام ال Integral بال Feedback زي ما مشروح فوق.

P-Controller (Proportional Control) → بمثال الملعب شفناه

(كان حالة خاصة، إنه ال error =0 بالعادة يكون فيه نسبة خطأ دائمًا لو P لحاله)

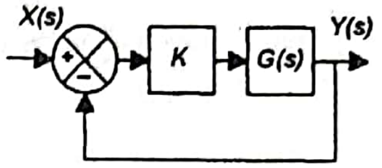
In proportional control, the controller produces a control action that is proportional to the error. There is a constant gain K_p acting on the error signal e and so:



controller output = $K_p e$ → * gain مضروبة بالخطأ *

We have steady-state error! How the P-control gain K_p affects the stead-state errors?

For a closed loop-system with a process transfer function G(s), and unity feedback:



$$Y(s) = \frac{G(s)K_P}{1+G(s)K_P}X(s)$$

→ حطوا شو مابلك هون فرضوا 1/(S+1)

We need to determining the value of the output as the time t tends to an infinite value. To do this in the s-domain, we use the *final value theorem (which if the limit exists):*

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

P-Controller (Proportional Control)

$$Y(s) = \frac{G(s)K_P}{1+G(s)K_P}X(s) \quad \rightarrow \quad y_{ss} = \lim_{s\to 0} s\frac{G(s)K_P}{1+G(s)K_P}X(s)$$

If x(t) = u(t), then X(s) = 1/s

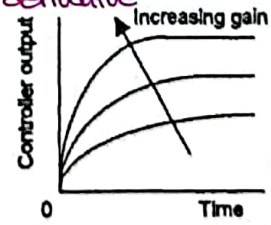
$$y_{ss} = \lim_{s\to 0} s\frac{K_P/(s+1)}{1+K_P/(s+1)}\frac{1}{s}$$

$$= \lim_{s\to 0}\frac{K_P}{s+1+K_P}$$

Suppose G(s) = 1/(1+s) for this demo example

As the limit goes to zero → $\underline{y_{ss} = K_P/(1+K_P)}$ → الصيغة النهائية بعد ما كملنا derivative

Kp	Yss	Offset
1	0.5	0.5
4	0.8	0.2
10	0.91	0.09



* شوما أحط قيمة لل gain بضل في عنا نسبة خطأ.

* شو مازدنا بال gain بتقل نسبة الخطأ بس مستحيل توصل لـ 0 إلا لو كانت ال gain infinity وهذا مستحيل.

Increasing the proportional gain decreases the steady state error, but it does not eliminate it to zero

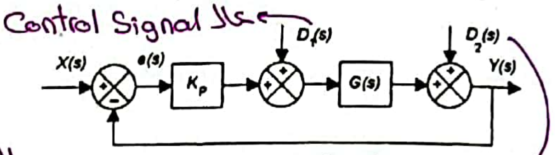
* أهمية وجود ال P بكل أنواع ال Controller لازم تكون لإنه وجدنا إنه هيك أفضل اشي ليقلل من مشاكل ال Disturbance (بـ Slide 7).

P-Controller (Proportional Control and Disturbance Rejection)

7

Previously we considered the effect of a disturbance on the performance of a closed-loop control system. We have seen that closed-loop control systems are better at minimizing disturbances than an open-loop system.

Consider this closed-loop control system with two possible sources of disturbances, one being a disturbance affecting the input to the process and the other affecting its output.



هاد الـ Control Signal

هاد الـ actuator signal

$Y(s)$ is $K_PG(s)e(s) + G(s)D_1(s) + D_2(s)$

Error = $X(s) - Y(s)$. → الفرق بين الـ Input والـ output

$$Y(s) = K_PG(s)[X(s) - Y(s)] + G(s)D_1(s) + D_2(s)$$

$$Y(s)[1 + K_PG(s)] = K_PG(s)X(s) + G(s)D_1(s) + D_2(s)$$

$$Y(s) = \frac{K_PG(s)}{1+K_PG(s)}X(s) + \frac{G(s)}{1+K_PG(s)}D_1(s) + \frac{1}{1+K_PG(s)}D_2(s)$$

Increasing the proportional gain reduces the effect of disturbances.

The first term is the normal expression for the closed-loop system with no disturbances. The other terms are the terms arising from the two disturbances. The factor 1/[1 + K_pG(s)] is thus a measure of how much the effects of the disturbances are modified by the closed-loop.

*كل ما أزيد قيم الـ k_p الـ D_1 والـ D_2 بقل (يعني بقل الـ disturbance)

سمضروبة بـ Constant

PD-Controller (Proportional and Derivative Control)

8

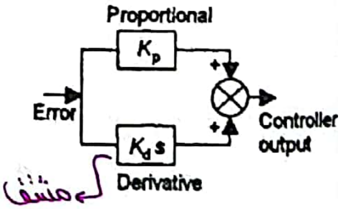
In derivative control, the controller produces a control action that is proportional to the rate at which the error is changing.

$K_d\frac{de}{dt}$ where K_d is the derivative gain

Derivative control is not used alone but always in conjunction with proportional control and, often, also integral control.

$$\text{controller output} = K_pe + K_d\frac{de}{dt}$$

The proportional element has an input of the error **e** and an output of K_p**e**. The derivative element has an input of **e** and an output which is proportional to the derivative of the error with time



مشتق الخطأ

In Laplace form: controller output $(s) = (K_p + K_ds)E(s)$

Can be rewritten as: controller output $(s) = K_P(1 + T_ds)E(s)$

*where $T_d = K_d/K_p$ and is called the ***derivative time constant***.

هاي فقط استخدم بدلالة T_d بدل K_d

*احنا وظيفتنا نلاقي قيم Kp و Kd و K_I ونطبقها للController.

فائدته :
بغير شكل ال Signal وبقيم ال oscillation ويعمل
عملية Smoothing لل Signal وهذا اشي كثير مرغوب ومفيد .

PD-Controller (Proportional and Derivative Control)

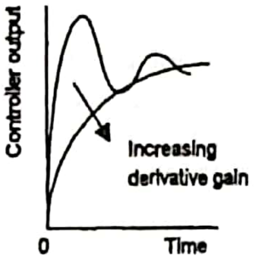
Derivative control has the controller effectively ***anticipating*** the way an error signal is growing and responding as the error signal begins to change (This is why we say that the derivative controller looks for the future).

A problem with this is that noise can lead to quite large responses.

Adding derivative control to **only** proportional control still leaves the output steady-state error and does not eliminate it.

Changing the amount of derivative control in a closed-loop system will change the damping ratio since increasing K_d Increases the damping ratio.

فبقل ال oscillation
وال under-damped
effects.



→ constant / s

PI-Controller (Proportional and Integral Control)

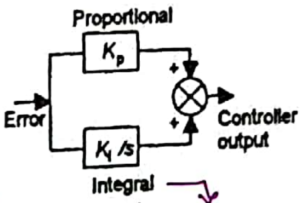
In integral control, the controller produces a control action that is proportional to the integral of the error with time.

$K_i \int e\, dt$ where K_i is the integrating gain

Integral control is not used alone but always in conjunction with proportional control and, often, also derivatve control.

$$\text{controller output} = K_p e + K_i \int e\, dt$$

The proportional element has an input of the error **e** and an output of K_p**e**. The integral element has an input of **e** and an output which is proportional to the integral of the error with time



→ تكامل الخطأ.

وظيفتها الأساسية تقيم الخطأ من ال System وال Steady-state

In Laplace form: $\text{controller output}(s) = \left(K_p + \frac{K_i}{s}\right)E(s)$

Can be rewritten as: $\text{controller output}(s) = \frac{K_p}{s}\left(s + \frac{1}{T_i}\right)E(s)$

where $T_i = K_p/K_i$ and is called the ***integral time constant.***

"بدلالة Ti بدل Ki"

The presence of integral control **eliminates** steady-state errors and this is generally an important feature required in a control system.

* ال P بتقلل الDisturbance وال D بتقلل ال oscillation وال I بقيم الخطأ اللي ضل .

* لنصمم ال Controller بالشكل المناسب صعب نضل نجرب عشوائي لازم نعرف نقاط البدايـــة .

PID-Controller (Proportional, Derivative and Integral Control)

The basic form is a *three-term controller.*

$$\text{output} = K_p e + K_i \int e\,dt + K_d \frac{de}{dt}$$

In Laplace form:

$$\text{output}(s) = K_p\left(1 + \frac{K_i}{K_p s} + \frac{K_d}{K_p} s\right)E(s)$$

$$\text{output}(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)E(s)$$

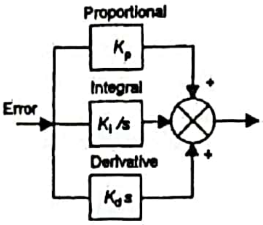


Table 7.4 Effect of Increasing the PID Gains K_P, K_D, and K_I on the Step Response

سرعة الresponse ←

الفرق الزمني لتوصل ال Steady-State ←

PID Gain	Percent Overshoot	Settling Time	Steady-State Error
Increasing K_P	Increases	Minimal impact	Decreases
Increasing K_I	Increases	Increases	Zero steady-state error
Increasing K_D	Decreases	Decreases	No impact

→ Zero بس ما بوصل لـ

مشكلته بزيد ال overshoot لل Signal فبعمل مشاكل ال damping

برضه بزيد ال overshoot

بهملوه لأنه ما إله تاثير كبير

يوصلني لنسبة خطأ = 0

لأنه زاد ال overshoot فبحتاج فترة أطول

هو اللي بقلل ال overshoot

لأنه قل ال overshoot فبحتاج وقت أقل

PID-Controller (Proportional, Derivative and Integral Control)

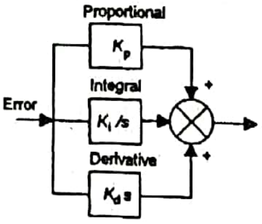
The basic form is a *three-term controller.*

$$\text{output} = K_p e + K_i \int e\,dt + K_d \frac{de}{dt}$$

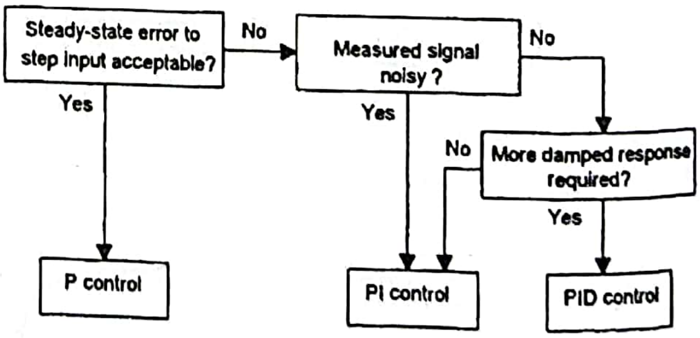
In Laplace form:

$$\text{output}(s) = K_p\left(1 + \frac{K_i}{K_p s} + \frac{K_d}{K_p} s\right)E(s)$$

$$\text{output}(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)E(s)$$



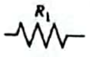
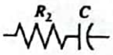
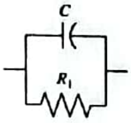
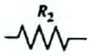
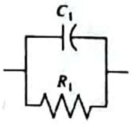
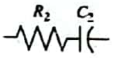
Controller Selection →

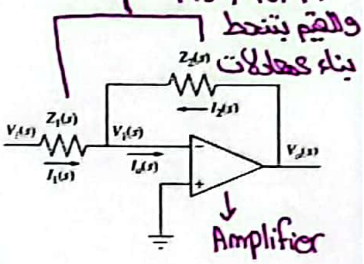

Steady-state error to step input acceptable?
No
Yes
Measured signal noisy ?
No
Yes
More damped response required?
No
Yes
P control
PI control
PID control

How are PID Controllers Realized (Classical Techniques)

13

This slide is just to have a quick idea, do not memorize the shapes or equations!
There are equations used to get the values for the resistors and capacitors from K_p, K_i, and K_d

Function	$Z_1(s)$	$Z_2(s)$	$G_c(s) = -\frac{Z_2(s)}{Z_1(s)}$
PI controller			$-\frac{R_2}{R_1}\frac{\left(s+\frac{1}{R_2C}\right)}{s}$
PD controller			$-R_2C\left(s+\frac{1}{R_1C}\right)$
PID controller			$-\left[\left(\frac{R_2}{R_1}+\frac{C_1}{C_2}\right)+R_2C_1s+\frac{\frac{1}{R_1C_2}}{s}\right]$



شو مايحط مكانها هنا اللي بحدد شكل ال Controller
PID / PD / PI
والقيم بتنحط بناء عمعادلات

Amplifier

* بس بوضحلك تصميم ال Controllers بال Circuit .

* مش مطلوب منك حفظ .

* ال Controllers ممكن يكونوا Circuit أو block of SW .

* بال Circuit بنعرف قيم Kp و Kd و Ki وبنحطها بال Circuit تبعتك بس احنا بدنا نحل بال SW

من ال Matlab وبعدين بنطلع قيم R و C .

* احنا يهمنا ال Software مش ال Circuit .

How are PID Controllers Realized (Computer Control)

14

We use libraries and functions, because the controller and system equations are all realized in software, so is the PID controller

Example, ARM DSP library has a function for PID Controller

* مكتبة ال CMIS - DSP (مكتبة فيها أشياء بتخص ال PID control) .

https://www.keil.com/pack/doc/CMSIS/DSP/html/group__PID.html

* The function takes as input two parameters S and in: → PID motor controller جاهزها

```
__STATIC_FORCEINLINE float32_t arm_pid_f32 (arm_pid_instance_f32 * S, float32_t in )
```

in is the input signal → object → فيه قيم kp و kd و ki → هم اللي يهمنا نجيبهم.

And S is a struct which has the gains and the values of Kp, Kd, Ki

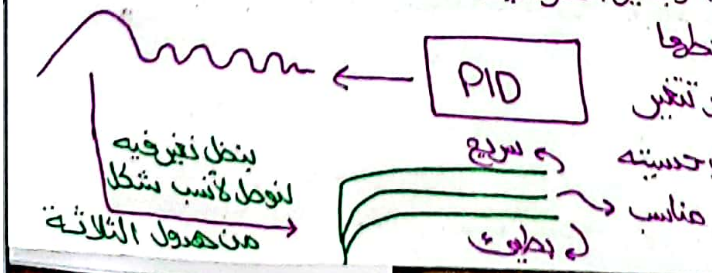
https://www.keil.com/pack/doc/CMSIS/DSP/html/structarm__pid__instance__f32.html

Data Fields	
float32_t	A0
float32_t	A1
float32_t	A2
float32_t	state [3]
float32_t	Kp
float32_t	Ki
float32_t	Kd

But where and how do we get the values of Kp, Kd, Ki?

MATLAB PID Tuner (DEMO in class)

* بكتب ال System بال S domain وبشغل ال block وبطلع رسمة ال response وبصير اختار أي controller استخدم وشو القيم اللي أحطها وبناء عالقيم الرسمة بتصير تتغير فبتوقف عند الشكل اللي حسيته مناسب.



حنعمله → بال Simulink ب self learning 9

*بالطريقة الثانية لازم تضل عينك دايما عال Closed-loop

- stability ايمتا تكون Stable.

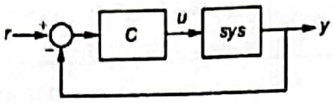
* هيك مثال بال Continuous واكيد في Version discrete.

*الحلو بال Simulink أنه بشتغل In real time يعني ال Motors

Initial Automatic PID Tuning

وال Controls موجودة بالحقيقة وال Simulink مشبوك عليها وبتحكم بالقيم ، فالاشي موجود بالحقيقة مش بس الشغل ك Model وبس.

MATLAB has the ***pidtune*** command which takes in as input the plant system as input, and the type of controller you want to use. Notice, that MATLAB assumes a unity feedback design, so, if need be, transform your system to match a unity feedback design.

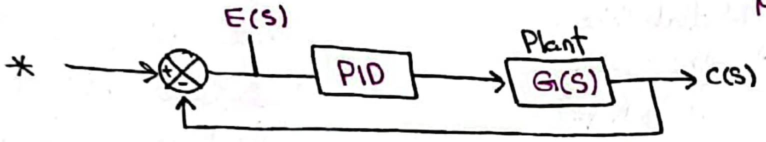


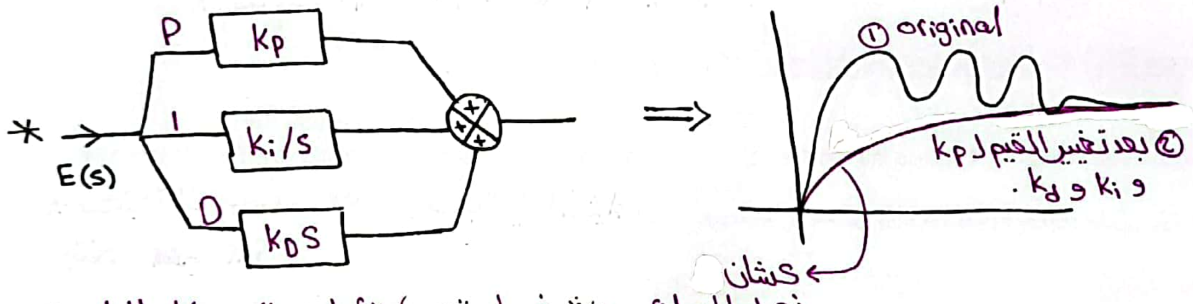
sys is the transfer function, it could be constructed using any of the techniques we have used before (tf, zpk)
C could be any of the following controllers (in this course, we only cover 1-DOF types, and only those in red)

- 'P' — Proportional only
- 'I' — Integral only
- 'PI' — Proportional and integral
- 'PD' — Proportional and derivative
- 'PDF' — Proportional and derivative with first-order filter on derivative term
- 'PID' — Proportional, integral, and derivative
- 'PIDF' — Proportional, integral, and derivative with first-order filter on derivative term

محاضرة 5/1/2023
كيفية تطبيق اللي
شرحناه عال Matlab
والSimulink.

*شرح الMatlab وال Simulink *





عشان نحولها لهيك حنشوف طريقتين الأولى Manually والثانية الطريقة الأفضل والأسرع .

*فرض معادلة ال Sys : $\frac{100}{S^2+15S+100}$ → underdamped *فتح الماتلاب وعمل Script وكتب المعادلة :

s=tf('s') ← sys=100/(s^2+15*s+100) بعدين عمل معادلة ال Sys كاملة (tot=feedback(sys,1))

بعدين Stepplot(sys) ← بعدين بستخدم ال Command PIDtune (هاي الطريقة مش مرغوبة لأنها Manual)

بعطيك قيم kp, ki, kd ← [c,info]=pidtune(sys,'PID') ← Mycontroller= 2.25+15.8/s+0.0781*s

← new_sys =mycontroller *sys; ← Stepplot(new_sys) ← لنشوف تأثير ال Controller عال Sys ، بدون feedback لأنه ال feedback استخدمناه للStability بس. ← بنصير نغير قيم kp ، kd ، ki لنوصل للاشي الأنسب.

* الطريقة الثانية المستخدمة : Apps ← PIDTuner ← بنعمل Import for sys ← بنختار ال Type ال Controller

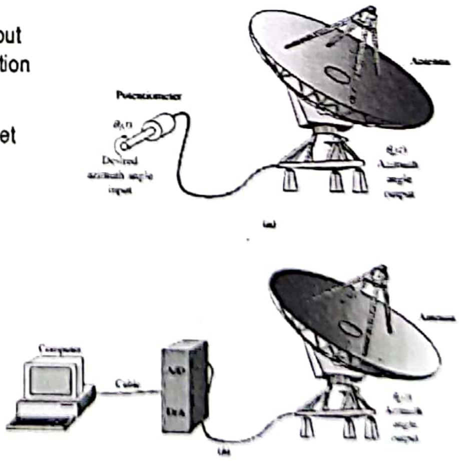
* ال Parallel form بوصف المعادلات زي ما بنعرف (kp, kd, ki) أما ال Standard بوصفهم ب To و Ti

* Show Parameters بتظهرلي المعلومات وبصير أغير Visually القيم لأوصل اللي بدي اياه. (تكملة فوق النوتس)

Brief Introduction to Digital Control

Introduction

- In chapter I, we learnt about the advantages of digital systems over analogue systems. We also learnt in this course and the embedded systems course about how to transform analogue data to digital data through sampling and quantization (A/D).
- We quickly reviewed how to determine the proper sampling rates in order to get digital representations that can faithfully represent the analogue signal.
- These days, many sensors are digital, that is, they have the A/D built-in an provide the sampled data at certain configured rates to the digital controller. These sensors can be used in the feedback loop or just to monitor the plant.
- In other cases, the A/D is built-in inside the controller chipset (SoC) and you to configure it to take samples in digital form at a required rate.


Antenna
Potentiometer
$\theta_i(t)$
Desired azimuth angle input
$\theta_o(t)$ Azimuth angle output
(a)
Antenna
Computer
Cable
A/D
D/A
$\theta_o(t)$ Azimuth angle output
(b)

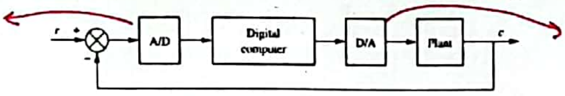
* الثلاث أدوار الأساسية للDigital computer: ① ال Control نفسه يكون جزء من الSystem. ② ماتكون الها علاقة بال Control، موجودة بس بتعمل مراقبة وعرض (supervision) ومابتقدر تتحكم بالقيم اللي بتعرض. ③ اللي موجود بالواقع هو الـSCADA اللي فيها جزئين ← Supervisory بتقرأ كل ال Data وبتعرضها

↳ Data Acquisition بتتيح للعامل وهو من غرفة بعيدة انه يتحكم بهذا الSystem

3

Digital Computer Role in Control Systems

- A digital computer might not take a direct control role at all. The whole plant might by controlled through conventional classical and analogue controllers, however, computers take a **supervisor role**. That is, they simply collect data about the system operation, creating logs, or showing real-time sensory data input to remote screens where engineers can monitor the plant and see if there are hazards or safety actions that need to be taken.
- A digital computer can replace many analogue controllers and be placed in the forward path, where it takes inputs, feedback data in digital form, process them through software code, equations, libraries, software PID, and issue commands to the plant for **direct control**. Programmer logic controllers (PLC) are common industrial examples, SoCs for other applications.



Analog to Digital

Digital to Analog

- In Industrial automation, the computer role can be **hybrid**; that is, it can take both a supervisory role and direct control role. These systems are famous and are referred to as SCADA (Supervisory Control and Data Acquisition). They provide Human Machine Interfaces (HMI) to control PLCs, read sensors, log data, and issue alarms.
https://inductiveautomation.com/resources/article/what-is-scada

* مثال عوجود ال SCADA: المفاعلات النووية اللي مربوطة بغرف تحكم بهاي المفاعلات.

* مثال تاني ال PLCs "Programm Logic controllers"

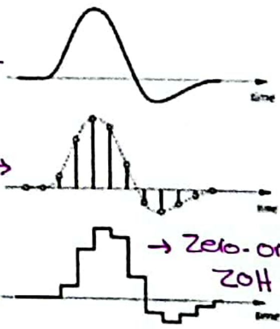
* ال Sampling rate نفسه بال Digital control ممكن يخليلي ال System unstable، بال Analog control كان يهمنا قيم ال gain مثلاً لإنها كانت مرات تخلي الSystem unstable.

* بتم التعبير عن ال A/D و ال D/A ب Switch

4

Review of the Sample and Hold

- In the embedded systems course, you learnt about the sample and hold circuit (SOH) which is the circuit that samples the analogue signal and holds the analogue value as steady as possible (unfortunately, we have drooping), then the A/D circuit converts this analogue value to a digital binary representation based on the resolution and voltage range of the ADC (quantization).
- Upon the complexity of this Sample and Hold circuit, they can be modeled and referred to as zero-order, first-order. In this course, we will assume the simplest circuit which is zero-order sample and hold, or "ZOH". You know this circuit from the embedded course as it generates the stair-step approximation.



→ Zero-order ZOH

after Sampling

SOH أبسط أنواع ال ← Stair-Step "زي الدرج"

ال Sampling هو الخطوة الأولى من ال A to D، هو بحوّل الTime domain من Continuous ل Discrete بعدين بندخل عال quantization اللي بمسك قيمة ال Sample ال analog وبيعطيها قيمة بال binary

* ال Sampling بفصل ال Circuit عن ال Signal وبياخذ آخر قيمة لل Capacitor (Slide 14 embedded).

* كل ما كان ال Sample rate عالي بتكون الدرجات أصغر فبتكون أقرب للشكل الحقيقي.

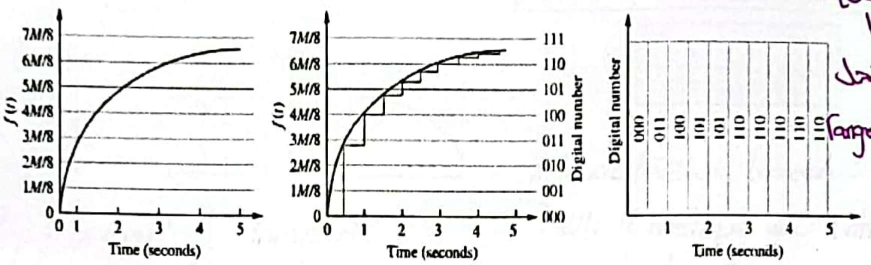
Voltage References and Quantization

- We have voltage references that the ADC must use to know the range of the analogue signal, V_{ref+} and V_{ref-}, and we know the resolution of the ADC to be for example N bits, and this means we have $(V_{ref+} - V_{ref-}) / 2^N$ ranges, and each range will take one of the 2^N possible binary representations.

↗ 0.1 , 0.2, ..., 0.9 → ٥٥٥ = رح تاخذ نفس قيمة ال

- There is always an error that is called the quantization error. No matter how much we increase the resolution, this error is always there and equals $\frac{1}{2} \times (V_{ref+} - V_{ref-}) / 2^N$ for a non-uniform quantizer

فمعناه
عندي quantization
error وهو دائمًا
موجود عمومًا بينحل
وقيمته = $\frac{1}{2}$ * الrange
الي طلعناها
$((V_{ref+} - V_{ref-})/2^N)$



* quantization: بكون عندي range (max / min) بقسمه s 2^N (resolution)

فلو مثلاً عندي A/D بياخذ ref. Volt = -5 و Max Volt = 5 معناه بيقدر ياخذ sin wave من -5 ل 5 فال total range = 10 ، لو ال A/D بيعطيني النتيجة بشكل مثلاً 16 bits فبقسم ال 10 هاي ع 2^{16} فبعرف عدد ال ranges.

Modeling the Sampler

- The simplest model for the sampler is that we have a pulse train (a series of rectangular pulses, on and off) multiplied by the signal. Effectively, this means we are reding the signal when we have a pulse (sampling it), or not reading the signal when the pulse is off.

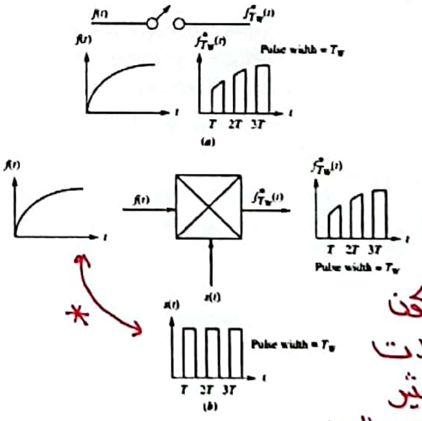
ال Signal الأصلية مضروبة كل مرة ب Impulse

$$f^*_{T_W}(t) = T_W \sum_{k=-\infty}^{\infty} f(kT)\delta(t - kT)$$

كل Impulse ↗
مزيح عالتاني
بقيمة T (Period) (Sample rate)

- Ideally, the width of the rectangular possible should be so small, more of an impulse train rather than a rectangular train. This ideal sampler can be mathematically described as

$$f^*(t) = \sum_{k=-\infty}^{\infty} f(kT)\delta(t - kT)$$



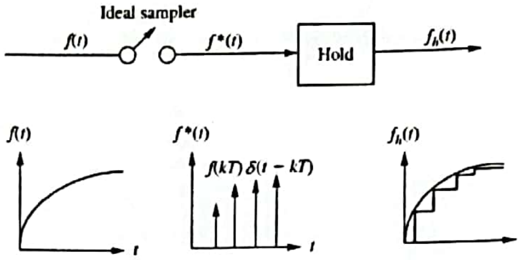
بهمنا يكون
المستطيلات
رفيعة كثير
بحيث In theory
بعبروا عنه ب dirac /
(δ (t)) delta function
(Impulse)

* لما نمثل المعادلات بال Time Domain بنستخدم T
أما بال Discrete domain بنستخدم n
↳ بياخذ قيم Discrete
زي 0.5 , 0.25 بس ما بكون range.

Modelling the Zero Hold Circuit

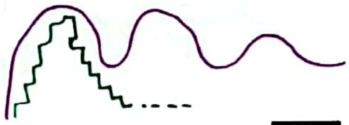
- Once we have taken a sample, we need to pass it to the quantizer to start converting it to a digital binary form. For this to happen, we need to keep the sampled value steady for the entire duration (in practice it drops a little due to capacitor discharge, *i.e.* drooping effect)
- The laplace transform for the zero-hold circuit is given by

$$\boxed{G_h(s) = \frac{1 - e^{-Ts}}{s}}$$



* المشكلة عندي إنه دخلت Computor ومش كل
أجزاء الـ System هي digital ، الـ Motor مثلاً
مش digital ، الـ Sensors جزء منها Analog جزء
digital ، الـ Circuit والـ Voltage devidor مش
digital وهكذا. فحنشوف blocks بالـ Analog
ونعبر عنها بالـ S Domain زي قبل ، و blocks
بالـ Digital بنعبر عنها بالـ Z Domain .

* قبل كنا نهتم بالـ Transiant
response وهلأ حيكون نفسه بس
باختلاف الرسمة حتظهر زي
اللون الأخضر :



أما المفاهيم نفسها ، الاشي الجديد موضوع الـ Samples. وحنشوف إنه سرعة الـ Samples
ممكن تخلي
الـ System
unstable
لإنه الـ Sys
مابلحق
كليا
فحيصير في
حدود للـ
Sampling
وقديه نعمله.

Digital Systems Modelling

- In this course, we mainly dealt with classical analogue control systems with continuous time-domain inputs and outputs. Once digital controllers (computers) are involved, many blocks in the system are using discrete time-domain samples of the original signal. As such, the Laplace transform can no longer be used to represent the system in the frequency domain.
- The discrete counterpart of the Laplace transform for digital control is the z-transform, and the new models we will develop in MATLAB will take into consideration the sampling rate F_s or sampling time T_s
- We have already learnt how to properly choose a sampling rate based on the sampled signal characteristics (maximum frequency component) and applying Shannon theorem of sampling and Nyquist rate. We also studied the concepts of undersampling and oversampling.
- Now, whereas the stability and transient response of analog systems depend upon gain and component values, sampled-data system stability and transient response also depend upon **sampling rate.** So now, we have new things to consider when choosing the sampling rate, that is the system stability.

The z-transform

- Similar to the laplace transform, the z-transform converts the function from the sampled time-domain to the frequency domain.
- In the time-domain, we no longer use the variable ***t*** to represent time, since ***t*** is associated with continuous time signals. Instead, we use the variable ***n*** to denote sampled time instances. This is the default for MATLAB commands as well.
- MATLAB has two commands to move between the sampled-time domain and the z-domain, **ztrans** and **iztrans**

بدل t

```
syms n
f = sin(n);
ztrans(f)
```

```
ans =
(z*sin(1))/(z^2 - 2*cos(1)*z + 1)
```

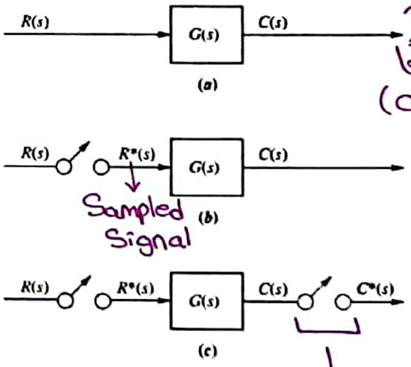
بحول الـ Samples (n) لأشكال أبسط لإنه صعب اتعامل معهم

```
syms z
F = 2*z/(z-2)^2;
iztrans(F)
```

```
ans =
2^n + 2^n*(n - 1)
```

The Transfer Function

- In this course, we have learnt how to obtain the transfer functions for classical systems as shown in Figure a. We have continuous time input R(s) going to an analogue system represented in the s-domain, and giving a continuous output.
- In Figure b; however, we have a sampled input R*(s) instead of a continuous time input R(s). In this example, we assume that the output is continuous.
- However, in many cases, we might only be interested in seeing the output for those sampled inputs and not in between, in Figure c, an imaginary (phantom) sampler is added at the output (coloured) to denote that we are interested to see the discrete output to the discrete inputs only. This assumption further simplifies future analysis.



اللي تعودنا عليه سابقًا (continuous)

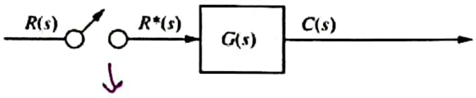
داخل Digital وطالع Analog

داخل Digital وطالع digital

لونه مختلف بيسموه Phantom (Imaginary)(وهمي)،
تمت إضافته عشان يعبر إنه مع الـ Input عندي يطلع discreted
وشكل الـ output عند النقاط اللي أنا مهتم فيها كمان discrete.
فهو مش اشي Physically موجود بس انحط عشان يذكرك
إنه اللي بهمنا لما تيجي القراءة discrete بدي أشوف الـ output اللي صدر عن هاي القراءة
الـ discrete واللي بضل الـ Step output.

Converting G(s) in Cascade with Z.O.H. to G(z)

- This figure below shows a discrete sampled input using a sample and hold circuit that goes into the s-domain block. Since the entire system is using sampled data, we need to find the overall new block G(z) that represents this system.
- We can convert G(s) in cascade with a zero-order hold (z.o.h.) to $G_n(z)$ using MATLAB's G_n=c2d (G, T,'zoh') command, where G1 is an LTI continuous-system object and G is an LTI sampled-system object. T is the sampling interval and 'zoh'is a method of transformation that assumes G(s) in cascade with a z.o.h. We simply put G(s) into the command (the z.o.h. is automatically taken care of) and the command returns G(z).



بداله نحط

ال transfer funct.

$$\frac{1-e^{-Ts}}{S} = \text{ال SOH}$$

Example I

Given a z.o.h. in cascade with $G_1(s) = (s+2)/(s+1)$ or

$$G(s) = \frac{1-e^{-Ts}}{s}\boxed{\frac{(s+2)}{(s+1)}} \rightarrow \text{لازم نحوله للـZ-Domain}$$

بزبطش يضل بالـ S

find the sampled-data transfer function, $\underline{G(z)}$, if the sampling time, T, is 0.5 second.

الإضافة من قبلي ←

```
T=0.5;                 % Input sampling interval.
numgs=[1 2];           % Define numerator of G(s).
dengs=[1 1];           % Define denominator of G(s).
Gs=tf(numgs,dengs)  % Create G(s) and display.
```

أحدد الـSample rate لإنه الـSys discrete

```
Gs =

  s + 2
  -----
  s + 1

Continuous-time transfer function.
```

```
Gn=c2d(Gs,T,'zoh')
```

$\rightarrow \approx \frac{1-e^{-Ts}}{s}$

↳ Continuous to discrete

```
Gn=c2d(Gs,T,'zoh')
```

```
Gn =

  z - 0.2131
  ----------
  z - 0.6065

Sample time: 0.5 seconds
Discrete-time transfer function.
```

→ الـ Sys كامل بالـ Z-Domain

Example II

* بنرجع رجوع بال System ← بنرجع ال Sys كيف كان قبل ال Sampling.

- Given G(z) of a certain function, can we retrieve back the original block G(s)? Let us do this by reversing the previous example.
- Notice that we can use the familiar **zpk** command to represent transfer functions in the z-domain, the only change is that we need to specify the sampling period so that it knows it is dealing with sampled data, and give a representation in ***z*** instead of ***s***

```
num=0.2131;
den=0.6065;
K=1;
T=0.5;
Gz=zpk(num,den,K,T)

Gz =

  (z-0.2131)
  ----------
  (z-0.6065)

Sample time: 0.5 seconds
Discrete-time zero/pole/gain model.
```

Zero ← num

Pole ← den

gain ← K

هاي الزيادة الوحيدة عشان يميز انه discrete مش Continuous

```
Gs=d2c(Gz,'zoh')

Gs =

  (s+2)
  -----
  (s+1)

Continuous-time zero/pole/gain model.
```

discrete to Continuous

زي بقله إذا كنت مستخدم مع ال Sys ال ZoH فافصله عن الـ Sys الأساسي واعطيني ياه شو كان بال S-Domain

Creating Digital Transfer Functions Polynomial Form

- A digital transfer function can be expressed as a numerator polynomial divided by a denominator polynomial, that is, F(z) = N(z) / D(z) . The numerator, N(z), is represented by a vector, numf, that contains the coefficients of N(z). Similarly, the denominator, D(z), is represented by a vector, denf, that contains the coefficients of D(z). We form F(z) with the command, F=tf(numf, denf, T), where T is the sampling interval. F is called a linear time-invariant (LTI) object. This object, or transfer function, can be used as an entity in other operations, such as addition or multiplication. We demonstrate with $F(z) = 150(z^2 + 2z + 7)/(z^2 - 0.3z + 0.02)$
- We MUST use an unspecified sampling interval, T= [ ] since we are already representing a digital system and not sampling a continuous one. It should be empty so that it creates a digital transfer function not a continuous one

```
numf=150*[1 2 7];    % Store 150(z^2+2z+7)  in numf and
denf=[1 -0.3 0.02]; % Store(z^2-0.3z+0.02) in denf and

F=tf(numf ,denf,[])

F =

  150 z^2 + 300 z + 1050
  ----------------------
     z^2 - 0.3 z + 0.02

Sample time: unspecified
Discrete-time transfer function.
```

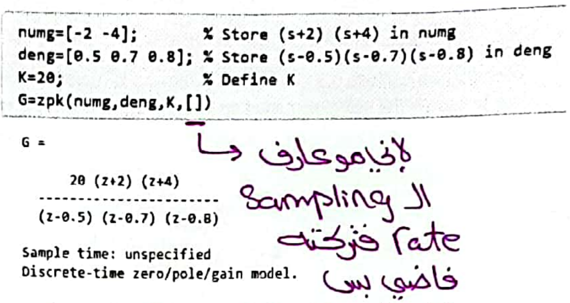
إضافة عشان نميز انه احنا بال Z-Domain (ضروري يكون)

Creating Digital Transfer Functions Directly Vector Method

- ★ We also can create digital LTI transfer functions if the numerator and denominator are expressed in factored form. We do this by using vectors containing the roots of the numerator and denominator. Thus,

$$G(z) = 20(z+2)(z+4)/[(z-0.5)(z-0.7)(z-0.8)]$$

- ★ can be expressed as an LTI object using the command, G=zpk (numg, deng, K, T), where numg is a vector containing the roots of N(z), deng is a vector containing the roots of D(z), K is the gain, and T is the sampling interval.



```
numg=[-2 -4];        % Store (s+2) (s+4) in numg
deng=[0.5 0.7 0.8]; % Store (s-0.5)(s-0.7)(s-0.8) in deng
K=20;                % Define K
G=zpk(numg,deng,K,[])
```

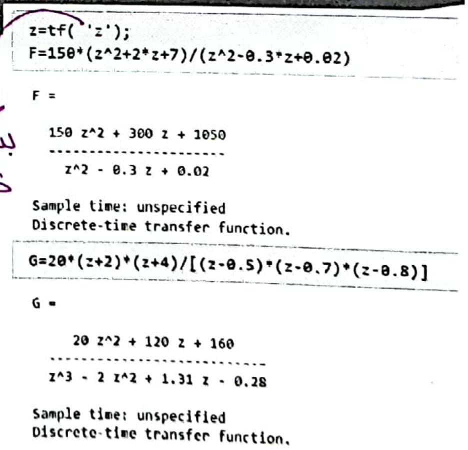
```
G =

      20 (z+2) (z+4)
  ----------------------
  (z-0.5) (z-0.7) (z-0.8)

Sample time: unspecified
Discrete-time zero/pole/gain model.
```

لأنا مو عارف ال Sampling rate فتركته فاضي بس

لازم يكون اله مكان عشان نميز إنه Z-Domain.

Creating Digital Transfer Functions using the z-method

- Similar to before where we defined s = tf ('s'), we can use z = tf ('z'), then write the digital transfer function directly



```
z=tf( 'z');
F=150*(z^2+2*z+7)/(z^2-0.3*z+0.02)
```

```
F =

  150 z^2 + 300 z + 1050
  ----------------------
    z^2 - 0.3 z + 0.02

Sample time: unspecified
Discrete-time transfer function.
```

```
G=20*(z+2)*(z+4)/[(z-0.5)*(z-0.7)*(z-0.8)]
```

```
G =

     20 z^2 + 120 z + 160
  ---------------------------
  z^3 - 2 z^2 + 1.31 z - 0.28

Sample time: unspecified
Discrete-time transfer function.
```

بدل 's'

عشان نميز انه احنا بال Z-Domain

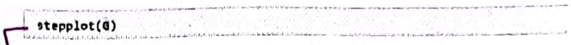
وبعدين بنتكلم عادي لو بدك نعمل Series

او feedback او هيك زي ما كنا نعمل قبل.

* في مكتبة بال Python بنستخدم نفس Command

ال Matlab.

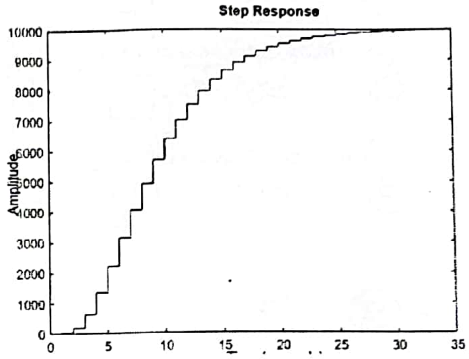
Transient Response



```
stepplot(G)
```

- Similar to before, we can observe the step response using the stepplot command

نفس قبل.



Block Diagram Reduction Important Note

✱ ▶ Our objective here is to be able to find the closed-loop sampled-data transfer function of an arrangement of subsystems that have a computer in the loop

✱ ▶ When manipulating block diagrams for sampled-data systems, you must be careful to remember the definition of the sampled-data system transfer function to avoid mistakes.

✱ ▶ For example, $z\{G_1(s)G_2(s)\} \neq G_1(z)G_2(z)$.

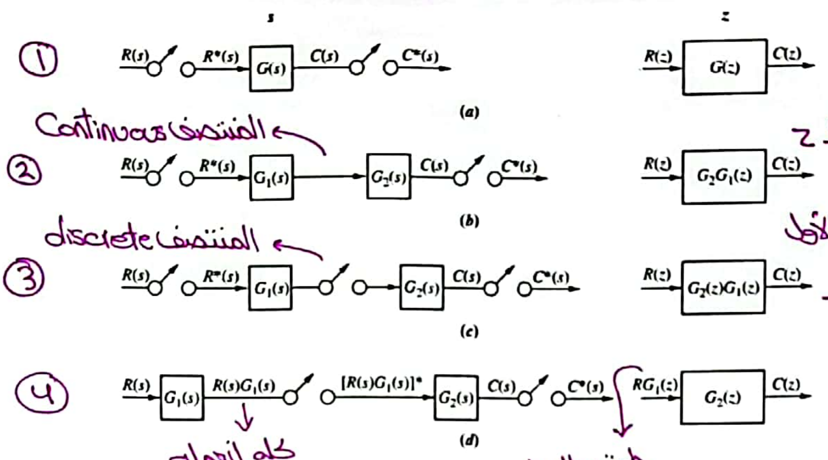
✱ ▶ The s-domain functions have to be multiplied together before taking the z-transform. In the ensuing discussion, we use the notation $G_1G_2(s)$ to denote a single function that is $G_1(s)G_2(s)$

✱ ▶ $z\{G_1(s)G_2(s)\} = z\{G_1G_2(s)\} = G_1G_2(z) \neq G_1(z)G_2(z)$

* نفس القوانين اللي عملناها قبل بال block Manipulation

بس حنعمل قبل تحضيرات ونشيل ال Switches ونحط مكانهم Combined blocks

Block Diagram Reduction Equivalent Blocks



أسهل واحدة داخل ال discrete و طالع discrete فبحوله كله من s لـ z

Continuous المنتصف ←

الفكرة هو بنضربهم بالأول بعين المضروب هو اللي بدخله عال c2d

discrete المنتصف ←

بس كإضافة ← ④

كله انعمله Sampling

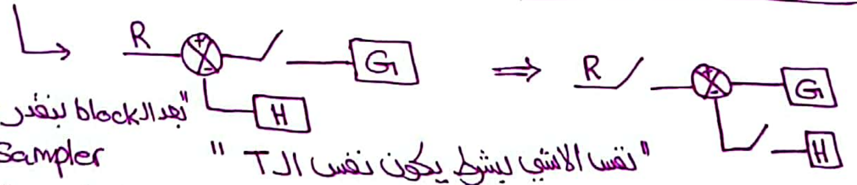
طبق عالقاعدة الثانية.

هون بدخل كل واحد لحاله عال c2d بعين بتضرب الناتج ببعض

Rules for Adding Phantom Sampler

لتسهيل ال Simplification ←

- A phantom sampler is an imaginary sampler that can be added to simplify block diagram reduction.
- **Rule I:** You can place a phantom sampler at the output of any subsystem that has a sampled input, provided that the nature of the signal sent to any other subsystem is not changed. For example, one can add phantom samplers at the output C(s). The justification for this, of course, is that the output of a sampled-data system can only be found at the sampling instants anyway, and the signal is not an input to any other block.
- **Rule II:** Another operation that can be performed is to add phantom samplers at the input to a summing junction whose output is sampled. The justification for this operation is that the sampled sum is equivalent to the sum of the sampled inputs, provided, of course, ***that all samplers are synchronized***

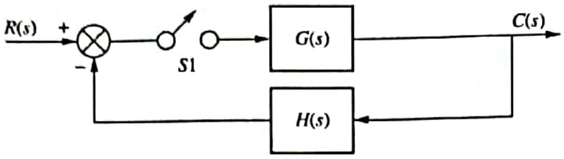


"بعد ال block بنقدر نضيف Sampler وهمي اذا كانت هاي ال block مش مشتركة على block ثانية " ذكرنا ما (كملنا بعد ال block ال output

"نفس الاشي بشرط يكون نفس ال T "

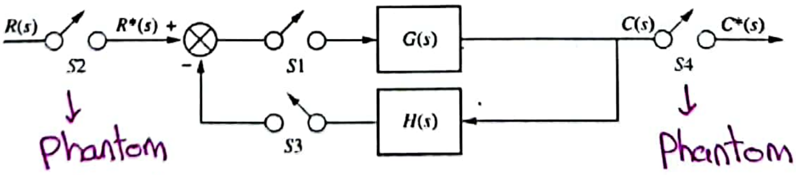
→ من أجل التبسيط

Digital Block Simplification Example



Digital Block Simplification Example – cont.

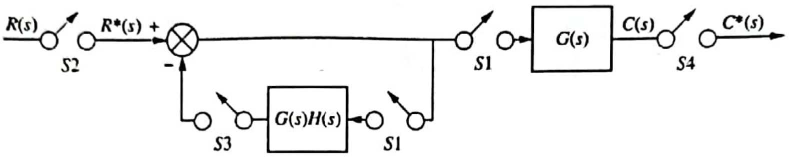
- Adding Phantom samplers per the rules



↓ Phantom Sampler

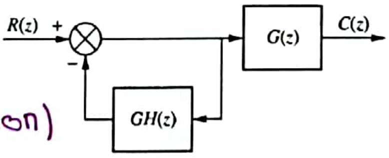
↓ Phantom Sampler

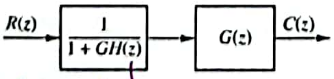
Digital Block Simplification Example – cont.

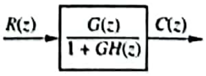


Digital Block Simplification Example – cont.

"كملنا زي قبل بعد ما شلنا الـSwitch
(block Manipulation)."







لازم نضل نحطها عشان تميز عن
الـ S-Domain

* بالـS-Domain كنا نرسم ال Poles ونشوفهم لو بال LHP فهو Stable ولو بال RHP فهو unstable ولو جاي عا jw-axis بكون Marginstable.

* كيف أعرف ال System اذا كان stable أو لا بال Z-Domain

25

Digital System Stability via the z-Plane I

- The glaring difference between analog feedback control systems and digital feedback control systems, is the effect that the sampling rate has on the transient response. Changes in sampling rate not only change the nature of the response from overdamped to underdamped, but also can turn a stable system into an unstable one.
- In the s-plane, the region of stability is the left half-plane. If the transfer function, G(s), is transformed into a sampled-data transfer function, G(z), the region of stability on the z-plane can be evaluated from the definition, z = e^{Ts}. Letting s =, we obtain

Complex محطين S محل ← ; constant ←

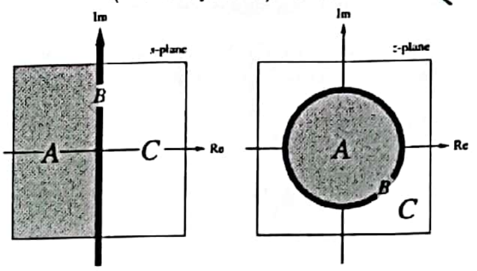
$$z = e^{Ts} = e^{T(\alpha+j\omega)} = e^{\alpha T}e^{j\omega T}$$
$$= e^{\alpha T}(\cos\omega T + j\sin\omega T)$$
$$= e^{\alpha T}\angle\omega T$$

↳ معادلة ال circle

$$(\cos\omega T + j\sin\omega T) = 1\angle\omega T.$$

→ العلاقة اللي بتربط ال Z مع ال S (للاستفادة)

لما S=0 ف Z=1



* بالـZ Domain مابنحكي LHP و RHP بنحكي عن unit circle *

*صرنا نعرف Stable أو لا بناءً عشغلة برا الدائرة أو جوا الدائرة أو عمحيط الدائرة .

↳ أي نقطة برة الدائرة كأنها RHP فال Sys بكون Unstable .

↳ أي نقطة داخل الدائرة كأنها LHP فال Sys بكون Stable .

↳ أي نقطة عحافة الدائرة :

→ لو ال root متكررات فوق بعض ما بكون Stable .

↳ لو كانوا ال root مختلفات عن بعض بكونوا Margin stable .

26

Digital System Stability via the z-Plane II

→ abs command → بيعطيني ال magnitude لعدد ال Complex num + Pole command ↓ بطلع ال Poles

- * If the magnitude e^{aT} > 1 , this means that the poles are outside the unit circle. Any point outside the circle means the system is unstable. This is equivalent to having a pole in the right half plane in the s-domain
- * If the magnitude e^{aT} = 1 , the pole lies exactly on the circle. This is equivalent to having a pole on the jw axis in the s-domain.
- * If the magnitude e^{aT} < 1 , the pole lies inside the circle. This is equivalent to having a pole on the left half plane in the s-domain
- * Stability criteria:

✓(1) stable if all poles of the closed-loop transfer function, T(z), are inside the unit circle on the z-plane,

✓(2) unstable if any pole is outside the unit circle and/or there are poles of multiplicity greater than one on the unit circle, and

✓(3) marginally stable if poles of multiplicity one are on the unit circle and all other poles are inside the unit circle.

بعدين باخذله ال abs وبعدين بقارن قيمهم لو أكبر من 1 فال Sys بكون unstable (لو لقيت عالأقل Pole واحدة أكبر من 1 فال System بكون unstable)

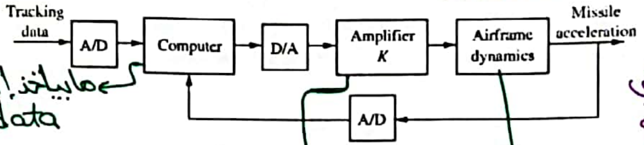
Example

- The missile shown can be aerodynamically controlled by torques created by the deflection of control surfaces on the missile's body. The commands to deflect these control surfaces come from a computer that uses tracking data along with programmed guidance equations to determine whether the missile is on track. The information from the guidance equations is used to develop flight control commands for the missile. A simplified model is shown as well.
- Here the computer performs the function of controller by using tracking information to develop input commands to the missile. An accelerometer in the missile detects the actual acceleration, which is fed back to the computer.
- Find the closed-loop digital transfer function for this system and determine if the system is stable for K = 20 and K = 100 with a sampling interval of T = 0.1 second.

بياخذ قراءة كل 0.1 second

Control
surfaces

قطع مستخصين للتحكم بحركة الصاروخ ووين بتوجه فالهم Controllers للتحكم.



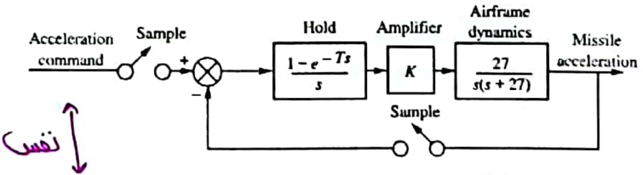
مابياخذ إلا digital data

بعمل تضخيم للVolt عشان تتناسب مع الـ Sys.

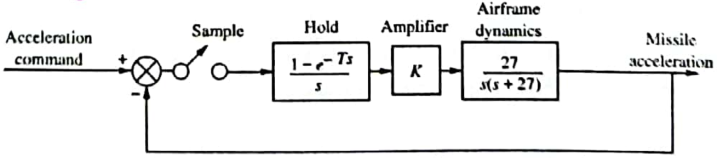
الاشي اللي حنتحكم فيه للتغيير بحركة الصاروخ

أي منهم اللي حيخلي الـ System Stable واللي حنستخدمها.

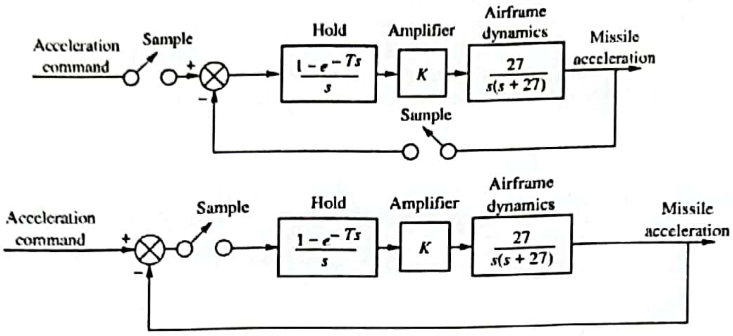
Example – cont.



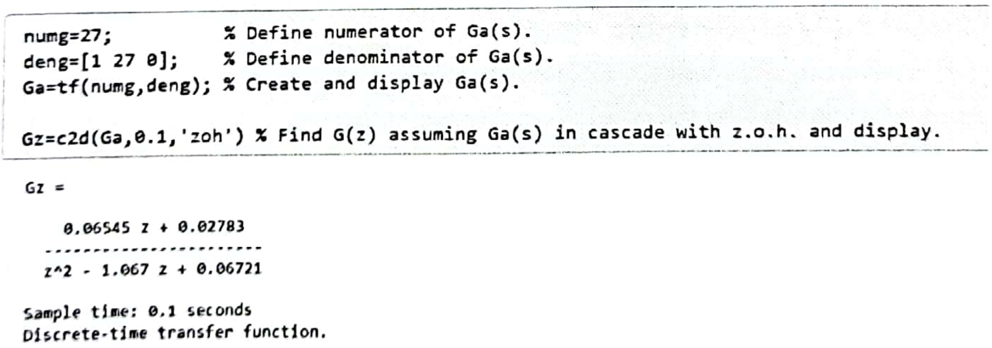
نفس بعض بشرط T تكون نفسها عالاثنين عشان أقدر أخليه زي تحت



Example – cont.



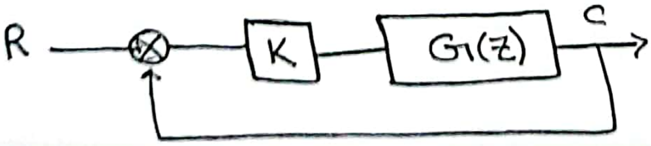
Example MATLAB Solution I



```
numg=27;           % Define numerator of Ga(s).
deng=[1 27 0];     % Define denominator of Ga(s).
Ga=tf(numg,deng); % Create and display Ga(s).

Gz=c2d(Ga,0.1,'zoh') % Find G(z) assuming Ga(s) in cascade with z.o.h. and display.
```

```
Gz =

   0.06545 z + 0.02783
  ----------------------
  z^2 - 1.067 z + 0.06721

Sample time: 0.1 seconds
Discrete-time transfer function.
```

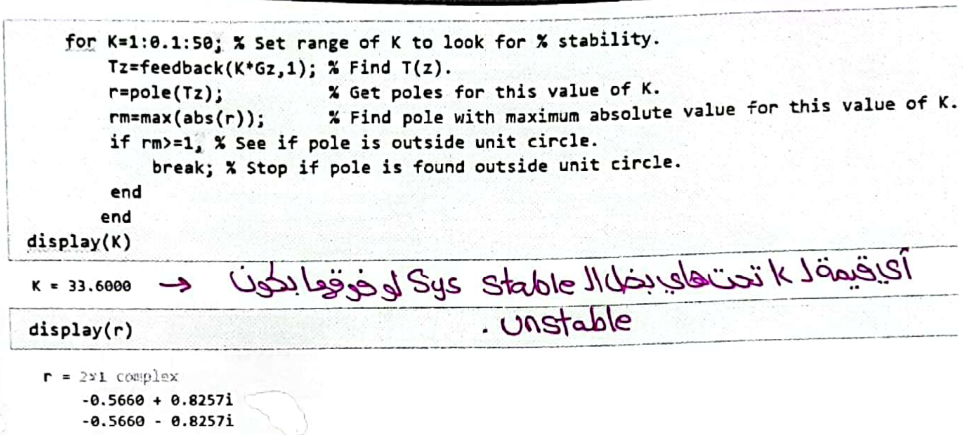
بعد هاي الخطوة صفّى الـSystem زي "هيك" :



" feedback System "

Example MATLAB Solution II

* بدي ألف أشوف شو قيم k اللي بتخلي ال System يكون Stable (ممكن تفحص k=20 و k=100 لحال وممكن تطلع قيم k اللي بكون تحتها Stable واللي أكبر منها unstable



```
    for K=1:0.1:50; % Set range of K to look for % stability.
        Tz=feedback(K*Gz,1); % Find T(z).
        r=pole(Tz);          % Get poles for this value of K.
        rm=max(abs(r));      % Find pole with maximum absolute value for this value of K.
        if rm>=1, % See if pole is outside unit circle.
            break; % Stop if pole is found outside unit circle.
        end
      end
display(K)
```

```
K = 33.6000
```

→ أي قيمة لـ k تحتها بضل ال Sys stable لو فوقها بكون unstable.

تشبه مبدأ ال Poles وال Zeros تقريبا

```
display(r)
```

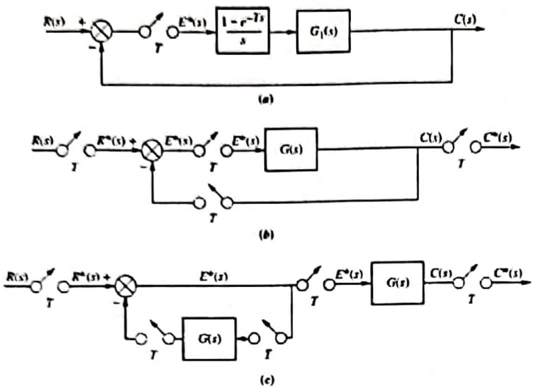
```
r = 2x1 complex
    -0.5660 + 0.8257i
    -0.5660 - 0.8257i
```

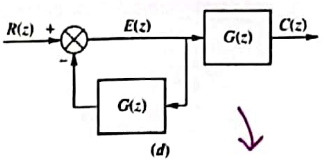
Steady-State Errors in Digital Systems

* نفس مفاهيم ال S-Domain تقريبا طريقة استخراج kp ، kv ، ka لوجود ال Sampling

- Any general conclusion about the steady-state error is difficult because of the dependence of those conclusions upon the placement of the sampler in the loop.
- Remember that the position of the sampler could change the open-loop transfer function.
- In the discussion of analog systems, there was only one open-loop transfer function, G(s), upon which the general theory of steady-state error was based and from which came the standard definitions of static error constants.
- For digital systems, however, the placement of the sampler changes the open-loop transfer function and thus precludes any general conclusions. In this chapter, we assume the typical placement of the sampler after the error and in the position of the cascade controller, and we derive our conclusions accordingly about the steady-state error of digital systems

Assumed System Model for Derivation





* هنا ال System
الي بتقدر تطبق عليه
المعادلات الي تحت *

" In Z-Domain "

Steady State Error Equations

The final value theorem for discrete signals states that → هاي الي اختلفت
عن ال S-Domain

* if S→o So Z→1
because $Z = e^{Ts}$

$$\boxed{e^*(\infty) = \lim_{z\to1}(1 - z^{-1})E(z)}$$

كانت هيك
$\lim_{S\to0} S\,E(S)$
بال S-Domain

The error due to a unit step input is defined as $\boxed{e^*(\infty) = \frac{1}{1+K_p}}$ $\boxed{K_p = \lim_{z\to1} G(z)}$ → ما اختلفت
عن قبل

The error due to a ramp input is defined as $\boxed{e^*(\infty) = \frac{1}{K_v}}$ $\boxed{K_v = \frac{1}{T}\lim_{z\to1}(z-1)G(z)}$

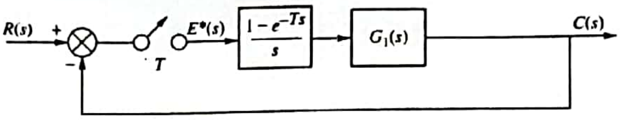
The error due to a parabola input is defined as $\boxed{e^*(\infty) = \frac{1}{K_a}}$ $\boxed{K_a = \frac{1}{T^2}\lim_{z\to1}(z-1)^2G(z)}$

→ اختلفوا عن
قبل

Example - MATLAB

- For step, ramp, and parabolic inputs, find the steady-state error for the feedback control system shown in the figure if

$$G_1(s) = \frac{20(s+3)}{(s+4)(s+5)}$$



- And the sampling rate is 0.1 sec, 0.5 sec

- We can use MATLAB's command dcgain(Gz) to find steady-state errors. The command evaluates the dc gain of Gz, a digital LTI transfer function object, by evaluating Gz at z = 1. We use the dc gain to evaluate, Kp, Kv, and Ka.

بطابق الشكل اللي أخذناه أو لأ، وهل لو ما بطابق بقدر أخليه يطابق
ولا لأ، لازم أجاوب السؤالين هدول قبل ما أحل.

MATLAB Solution (T = 0.1)

```
T = 0.1; % Input sampling interval.
G1s= zpk(-3, [-4 -5], 20)
```

```
G1s =

   20 (s+3)
  -----------
  (s+4) (s+5)

Continuous-time zero/pole/gain model.
```

```
Gz=c2d(G1s,T,'zoh') % Convert G1(s) and z . o. h. to G(z)
```

```
Gz =

   1.4994 (z-0.7405)
  ---------------------
  (z-0.6703) (z-0.6065)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

```
Tz=feedback(Gz, 1) % Create and display T(z).
```

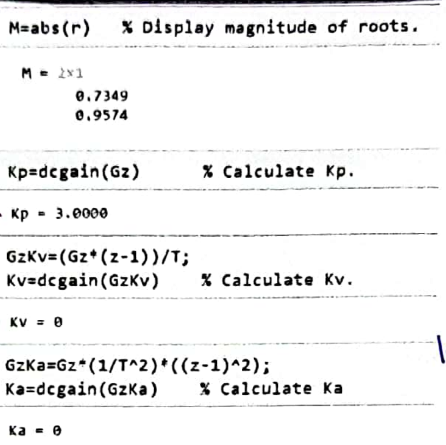
```
Tz =

    1.4994 (z-0.7405)
  ---------------------
  (z-0.7349) (z+0.9574)

Sample time: 0.1 seconds
Discrete-time zero/pole/gain model.
```

```
r=pole(Tz)          % Check stability.
```

```
r = 2x1
     0.7349
    -0.9574
```

أقل من 1
فال sys
stable
هلأ بقدر أحسب
الأخطاء Kp ، Ka ، Kv

MATLAB Solution (T = 0.1) - cont



```
M=abs(r)    % Display magnitude of roots.

  M = 2x1
        0.7349
        0.9574

Kp=dcgain(Gz)        % Calculate Kp.

  Kp = 3.0000

GzKv=(Gz*(z-1))/T;
Kv=dcgain(GzKv)      % Calculate Kv.

  Kv = 0

GzKa=Gz*(1/T^2)*((z-1)^2);
Ka=dcgain(GzKa)      % Calculate Ka

  Ka = 0
```

e=Finite ← Kp

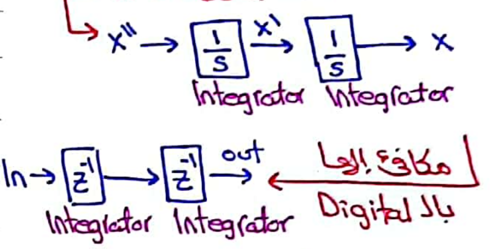
$e=\infty$ ← Kv

$e=\infty$ ← Ka

*note in Simulink:

$x'' = bx' + 3$

بده يحولها ل x



⇓

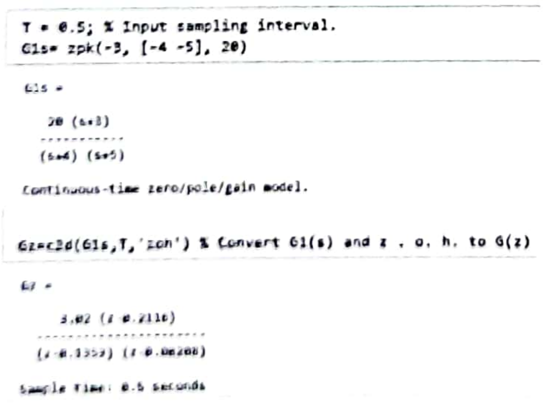
ولو في حدود دنيا
وكليا للتكامل
كنا بنعمل double
click عال block
وبنكتب هاي الحدود
اللي بتعبر عن Physical
limits

بنك تقيم ال error ولأنه
كمان كانت s بالمقام
منفصلة

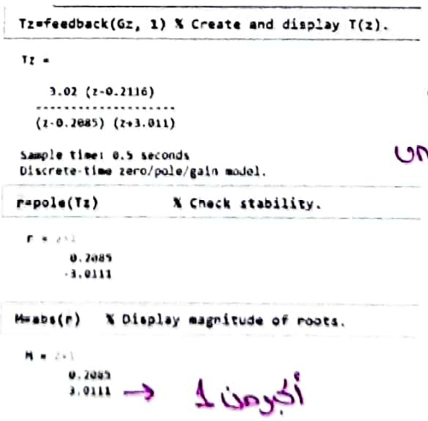
⇒ Type 0

*المكافئ لل Integrator بال Z-Domain = $\frac{1}{z}$ = z^{-1} ، فبرضو مش
موجود بمعادلة ال Z-Domain ف Type 0 .

MATLAB Solution (T = 0.5) (UNSTABLE)



```
T = 0.5; % Input sampling interval.
G1s= zpk(-3, [-4 -5], 20)

G1s =

    20 (s+3)
  -----------
  (s+4) (s+5)

Continuous-time zero/pole/gain model.

Gz=c2d(G1s,T,'zoh') % Convert G1(s) and z . o. h. to G(z)

Gz =

       3.02 (z-0.2116)
  ----------------------
  (z-0.1353) (z-0.08208)

Sample time: 0.5 seconds
```



```
Tz=feedback(Gz, 1) % Create and display T(z).

Tz =

     3.02 (z-0.2116)
  -------------------
  (z-0.2085) (z+3.011)

Sample time: 0.5 seconds
Discrete-time zero/pole/gain model.

r=pole(Tz)          % Check stability.

  r = 2x1
        0.2085
       -3.0111

M=abs(r)    % Display magnitude of roots.

  M = 2x1
        0.2085
        3.0111
```

3.0111 → أكبر من 1

فصار ال Sys unstable .

فما بنكمل حل.

*غيرنا ال T
هون وشفنا
إنه ال System
صار unstable

References

- The material in these slides are based on:
  Control Systems Engineering, Norman S. Nise, 7th Edition (2014), John Wiley And Sons
  • **Chapter 13 – Digital Control Systems**
  Sections *13.1, 13.2, 13.3, 13.4, 13.5, 13.6, 13,7* (Students kindly note that these sections involve lots of math, and we only described the ideas as we will use MATLAB instead)