Block diagrams

notes

▪

Contents map

developed content units	**taxonomy levels**
block diagrams	u1, e1

prerequisite content units	**taxonomy levels**
transfer function	u1, e1

notes

▪

Main ILO of sub-module "Block diagrams"

Describe the purpose and advantages of using block diagrams to a peer unfamiliar with them

Identify standard blockdiagram components (e.g., sum, integral, derivative, multiplication, generic function) when interpreting or constructing block representations of time-domain systems, using the lecture-provided visual notation

Construct a block diagram representation for a first-order linear differential equation of the form $\dot{y} = ay + bu$, using basic functional blocks (integrators, multipliers, summations)

Derive the closed-loop transfer function for a system with negative feedback, using the algebraic manipulation of Laplace-domain expressions

notes

- By the end of this module, I want you to be able to confidently do all these things
- These are fundamental skills you'll use throughout your engineering career
- Don't worry if it seems abstract now - we'll build up to it step by step
- Pay special attention to the feedback loop derivation - it's one of the most important concepts in control theory

Roadmap

- recap of the diagrams in the time domain
- recap of the diagrams in the frequency domain
- rules for how to transform the diagrams
- examples

notes

- Today we're going to build on what you already know about block diagrams
- We'll start with a quick review of time domain representations - this should be familiar from your previous courses
- Then we'll add the frequency domain perspective, which is crucial for control systems analysis
- The transformation rules might seem tricky at first, but once you practice with the examples, they'll become second nature
- Remember, these diagrams are just visual representations of the equations we've been working with

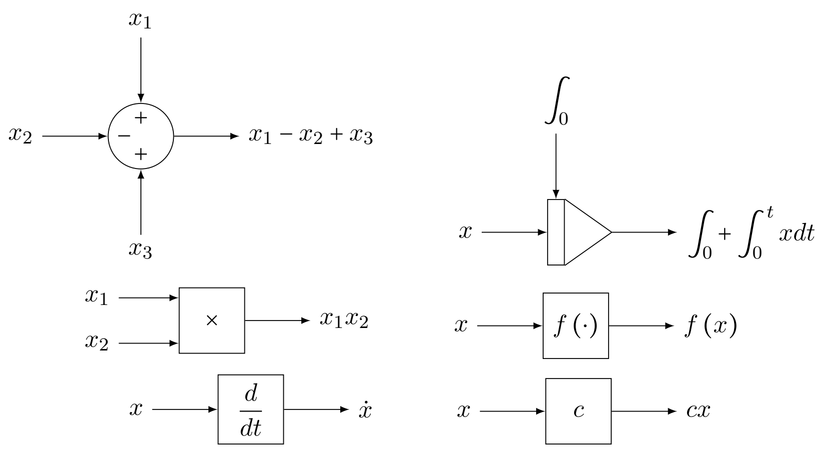
Block diagrams - why?

- used very often in companies
- aid visualization *(until a certain complexity is reached. . .)*
- enable "drag & drop" way of programming
- here primarily used for interpretations

notes

- Let me tell you why we're spending time on this - in industry, you'll see these diagrams everywhere
- They're like the universal language of control engineers
- While complex systems might become hard to visualize, for most practical applications they're incredibly useful
- Many control system design tools (like Simulink) use exactly this drag-and-drop approach
- But more importantly, they help us understand how different components interact in a system

Most common block diagrams in the time domain

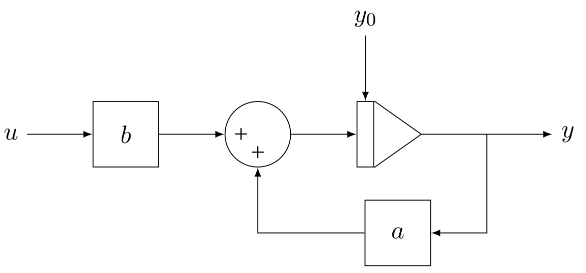


notes

- These are the building blocks of any control system diagram
- The sum block is how we represent adding or subtracting signals - notice the + and - signs
- The integral block is crucial - remember it always includes initial conditions
- Multiplication can be between two signals or by a constant - these are different operations
- The generic function block is powerful but dangerous - we'll see why soon
- Derivatives appear less often in practice because they amplify noise

Representing a first order DE with a block scheme

$$\dot{y} = ay + bu$$



notes

- This is a perfect example of how we combine basic blocks to represent differential equations
- Notice how the derivative appears on the left, but we implement it using integration
- The feedback path with gain 'a' creates the system dynamics
- The input 'u' gets scaled by 'b' before entering the system
- This structure is fundamental - you'll see variations of it constantly

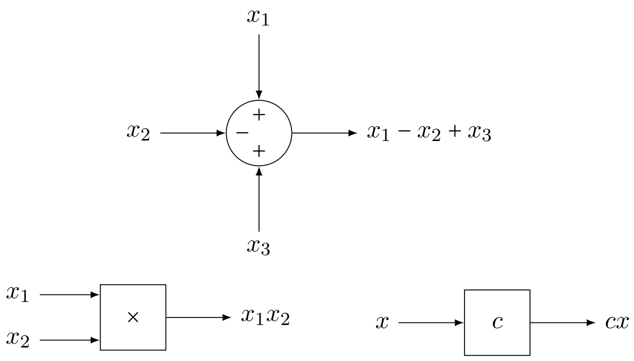
Discussion: how do we represent $\ddot{x} + \frac{f}{m}\dot{x} + \frac{k}{m}x = \frac{1}{m}u$?

notes

- Here's a challenge for you to think about - how would you extend this to second order systems?
- The procedure is similar but needs an additional integrator
- Remember that each derivative becomes an integrator in the diagram
- The damping term (with f/m) would create another feedback path
- If you're stuck, think about how you'd solve the equation for the highest derivative
- I'll show the solution in the video lecture, but try drawing it yourself first

Block diagrams that are equal in both time and frequency domains

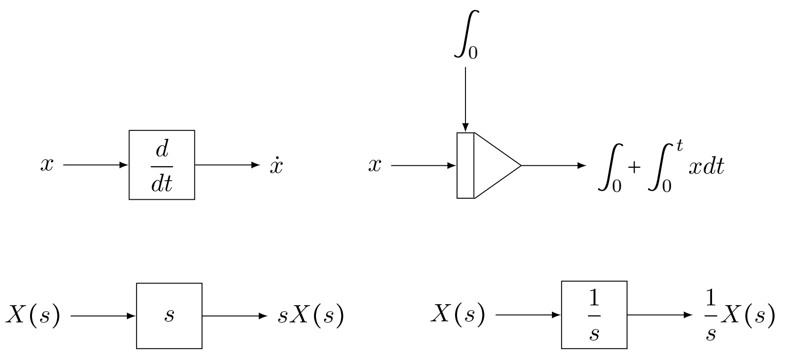
(Here we may use both $x(t)$ and $X(s)$)



notes

- Some operations look identical in both domains - these are the simplest cases
- Summation works the same whether we're adding time signals or their Laplace transforms
- Multiplication by a constant is equally simple in both domains
- But be careful - multiplying two signals is different from multiplying two transfer functions!
- These similarities make it easier to switch between time and frequency domains

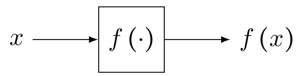
Block diagrams that are logically the same in both time and frequency domains



notes

- Here we see operations that are equivalent but look different
- Differentiation in time becomes multiplication by 's' in Laplace domain
- Integration becomes division by 's' plus initial conditions
- Notice how the initial conditions appear in the time domain but are often omitted in frequency domain
- This is why we prefer frequency domain for analysis - it turns calculus into algebra!

A block diagram that does not exist in the frequency domain



Discussion: why?

notes

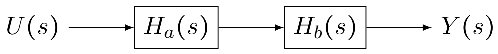
- This is a crucial limitation - nonlinear operations don't translate directly to frequency domain
- The Laplace transform can't easily handle arbitrary nonlinear functions
- Think about it - how would you represent sin(y(t)) or y(t)$^2 intermsofY(s)?Thisiswhyweoftenlinearizesystemsbeforeanalyzingthem$
- Remember: frequency domain methods assume linearity and time-invariance
- When you see this block, you'll need to either linearize it or stay in time domain

in the following:
rules for manipulating block diagrams in the frequency domain

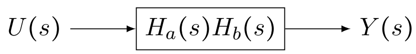
notes

- Now we get to the really useful part - the rules for simplifying diagrams
- These rules will help you reduce complex systems to simpler equivalent forms
- They're based on algebraic manipulations of the transfer functions
- I'll show you the formal rules first, then we'll practice applying them
- Mastering these will save you hours of work when analyzing systems

Series of transfer functions



is equivalent to

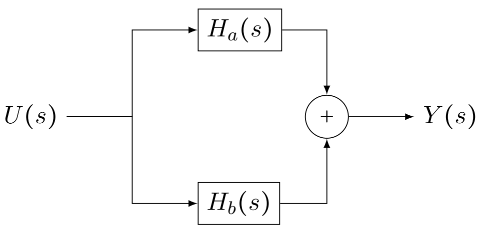


Discussion: why?

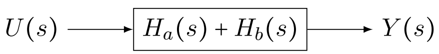
notes

- This is the simplest rule - series connection means multiplication
- Think of it like function composition: first apply H$_a, thenapplyH_btotheresult$
- Let's think through it: The output Y is H$_bappliedtotheintermediatesignal, whichisH_aappliedtoUSomathematically: Y = H_b(H_a(U)) = (H_b \mathring{u} H_a)U$
- The order matters in general, but for transfer functions it's commutative
- Remember this only works for LTI systems - nonlinear systems don't obey this rule

Parallel of transfer functions



is equivalent to

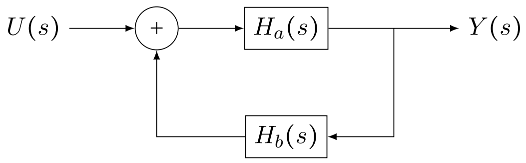


Discussion: why?

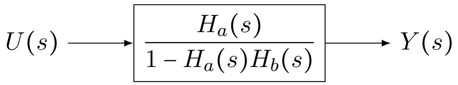
notes

- Parallel paths add their effects - this is superposition in action
- The same input goes to both systems, and their outputs are summed
- This works because of the linearity property of LTI systems
- The total output is the sum of each system's response to the input
- Notice this wouldn't work if the summing junction had different signs
- Also wouldn't work if there were nonlinear elements in the paths

Elimination of feedback loops



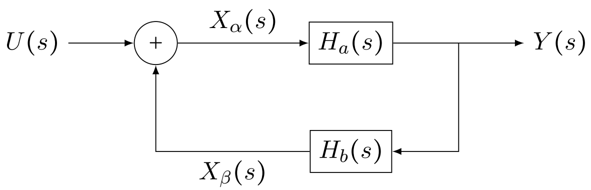
is equivalent to



notes

- This is the most important rule in control systems - feedback reduction
- The formula looks complex but has a beautiful symmetry to it
- Notice the denominator is 1 minus the loop gain (product of both transfer functions)
- This specific form is for positive feedback - we'll see negative feedback soon
- Memorize this pattern - you'll use it constantly in control system design

Elimination of feedback loops: how to remember the formula

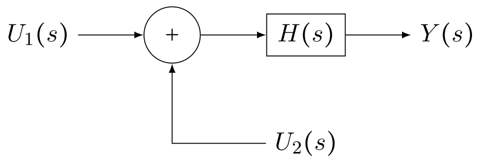


- $Y = H_a X_\alpha$
- $X_\alpha = U + X_\beta$
- $X_\beta = H_b Y$
- $\implies Y = H_a (U + H_b Y)$
- $\implies Y - H_a H_b Y = H_a U$

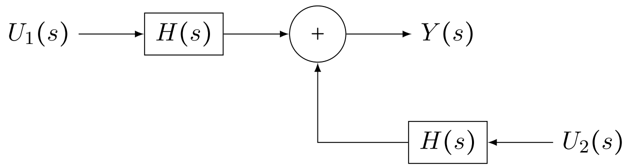
notes

- Let me show you how to derive this so you'll never forget it
- We introduce intermediate variables X$andXtorepresentsignalsThenwewriteequationsforeachrelationship$
- Finally, we substitute and solve for Y in terms of U
- This is much better than memorizing - you can re-derive it anytime
- Notice how the loop gain H$_a H_b appearsnaturallyinthealgebraFornegativefeedback, justchangethe + to - inthesummingjunction$

Moving blocks around sum operators



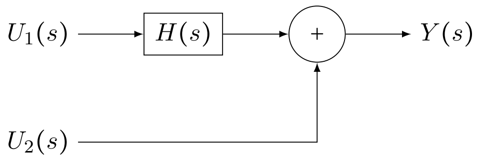
is equivalent to



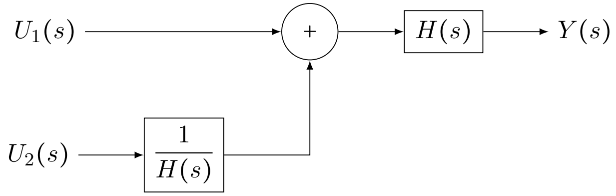
notes

- This rule shows how we can move blocks past summing junctions
- The key insight is that the same operation must be applied to all inputs
- Here we've moved H(s) to operate on both U1 and U2
- This is useful when you want to combine parallel paths
- Remember: what you do to one input, you must do to all inputs

Moving blocks around sum operators



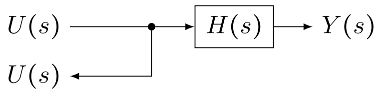
is equivalent to



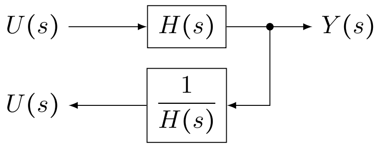
notes

- This is a more advanced version of the previous rule
- When moving a block past a summing junction, we need to compensate with its inverse
- Notice how U2 now passes through 1/H(s) to maintain equivalence
- This is useful when you need to combine signals before processing
- Be careful - this only works if H(s) is invertible (no zeros at infinity)

Moving blocks around connections



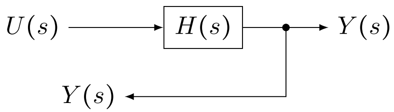
is equivalent to



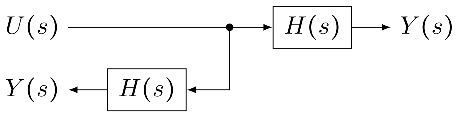
notes

- This rule shows how to move a branch point past a block
- Similar to the previous rule, we need to compensate with the inverse
- The key idea is that the branched signal must remain unchanged
- This is useful when you need to access the original input signal
- Again, this only works if H(s) is invertible

Moving blocks around connections



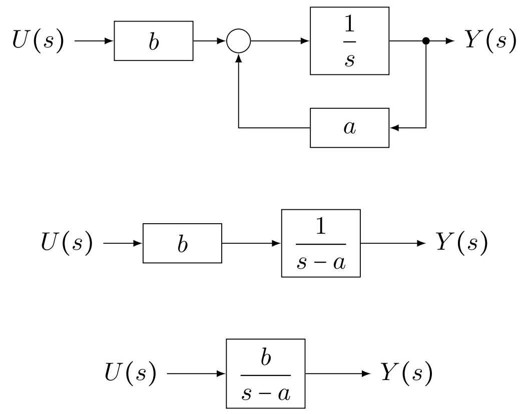
is equivalent to



notes

- This is the companion to the previous rule - now we're branching the output
- Here we need to apply the same transfer function to maintain equivalence
- The branched signal must still represent the output Y(s)
- This is useful when you need to feed the output forward
- Notice we don't need any inverses here - just duplicate the block

Combining the example above to model $\dot{y} = ay + bu$



notes

- Now let's see how all these rules come together in a practical example
- We start with the time-domain differential equation
- First we implement it directly with basic blocks
- Then we apply the feedback reduction rule to simplify
- Finally, we combine the series blocks
- Notice how we end up with exactly the transfer function we'd get by taking Laplace transforms
- This shows the consistency between time and frequency domain

Self-assessment material

notes

-

Question 1

transfer function Which of the following block operations has **no equivalent representation** in the frequency domain?

1. Multiplication by a constant
2. **(correct)** Nonlinear transformation by a generic function $f(\cdot)$
3. Summation of signals
4. Integration
5. I do not know

Solution 1: *Nonlinear operations like $f(y(t))$ do not generally have a direct equivalent in the frequency domain. In contrast, operations like multiplication, summation, and integration do, as they correspond to algebraic operations in the Laplace domain.*

Question 2

transfer function What is the equivalent transfer function of two blocks with transfer functions $H_a(s)$ and $H_b(s)$ connected **in series**?

1. $H_a(s) + H_b(s)$
2. $H_a(s) - H_b(s)$
3. **(correct)** $H_a(s) \cdot H_b(s)$
4. $H_a(s)/H_b(s)$
5. I do not know

Solution 1: *Blocks in series multiply: the output of the first is the input of the second, so the overall transfer function is $H_a(s) \cdot H_b(s)$.*

Question 3

transfer function Which of the following statements about the feedback loop formula $\dfrac{H_a(s)}{1 - H_a(s)H_b(s)}$ is **correct**?

1. It is only valid for time-domain systems
2. It represents the series connection of two systems
3. **(correct)** It gives the closed-loop transfer function for a negative feedback loop
4. It requires $H_b(s)$ to be zero
5. I do not know

Solution 1: *The formula $\dfrac{H_a(s)}{1 - H_a(s)H_b(s)}$ is the standard expression for the closed-loop transfer function in the case of negative feedback.*

Question 4

transfer function Which operation is **equivalent in both time and frequency domains** for block diagrams?

1. Integration
2. Derivation
3. **(correct)** Multiplication by a constant
4. Nonlinear transformation
5. I do not know

Solution 1: *Multiplication by a constant is equivalent in both time and frequency domains since it does not involve any transformation or differentiation/integration.*

Question 5

transfer function In a parallel connection of two transfer functions $H_a(s)$ and $H_b(s)$, the resulting system has the transfer function:

1. $H_a(s) \cdot H_b(s)$
2. $\frac{H_a(s)}{H_b(s)}$
3. **(correct)** $H_a(s) + H_b(s)$
4. $H_b(s) - H_a(s)$
5. I do not know

Solution 1: *In a parallel configuration, the outputs of $H_a(s)$ and $H_b(s)$ are added together. The resulting transfer function is therefore $H_a(s) + H_b(s)$.*

notes

- the most important messages of this module are likely these ones

Recap of the module "Block diagrams"

1. you are suppose to know how to work with block diagrams
2. the rules are quite simple, and can be re-derived by hands