

Lecture 13

CT System Analysis and Design using Laplace

C.L. Wyatt

2026-03-05

Today we look at three important applications of Laplace for the analysis, design, and implementation of CT systems.

Analyzing Electrical Circuits

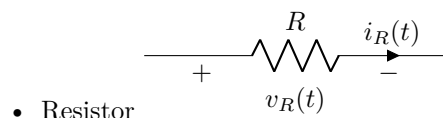
Recall from ECE 2714 that we can analyze electrical circuits that form LTI systems by

- Deriving a governing equation (LCDE)
- Finding the impulse response $h(t)$
- Using convolution to determine the output $y(t)$ given an input $x(t)$

Or alternatively using Fourier analysis

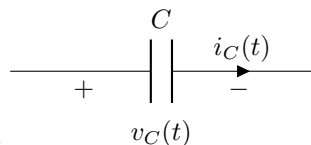
- Determine stability and derive the frequency response $H(j\omega)$
- use convolution theorem to find output $Y(j\omega)$ and use the inverse Fourier transform to find $y(t)$.

Now that we have Laplace as a tool we can simplify this process further. The idea is to go back to circuits and take the Laplace transform of each element's voltage-current model. Voltages and currents become expressed in the Laplace domain. The voltage-current relationship then becomes a linear function, with the generalized version of resistance, impedance, and the generalized version of inverse resistance, admittance.



$$v_R(t) = R i_R(t)$$
$$V_R(s) = R I_R(s)$$

Impedance R , Admittance $\frac{1}{R}$



- Capacitor

$$i_C(t) = C \frac{dv_C}{dt}(t)$$

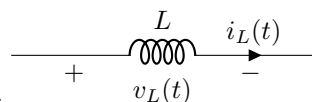
$$I_C(s) = C [sV_C(s) - v_C(0^-)]$$

$$I_C(s) = CsV_C(s) - Cv_C(0^-)$$

or rearranging

$$V_C(s) = \frac{1}{Cs} I_C(s) + \frac{v_C(0^-)}{s}$$

Impedance $\frac{1}{Cs}$, Admittance Cs



- Inductor

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

$$V_L(s) = L [sI_L(s) - i_L(0^-)]$$

$$V_L(s) = LsI_L(s) - Li_L(0^-)$$

or rearranging

$$I_L(s) = \frac{1}{Ls} V_L(s) + \frac{i_L(0^-)}{s}$$

Impedance Ls , Admittance $\frac{1}{Ls}$

This gives us new circuit models in Laplace domain with either constant voltage or current sources representing the terms containing auxiliary conditions at $t = 0^-$. Using KVL and KCL with these models gives us a convenient way to go from a circuit directly to a transfer function. Most upper level circuit courses will use these techniques. e.g. AC-circuits.

Capacitor Model

Using the above relationships for capacitor voltage and current after using the Laplace transform, we get two equivalent circuit models. Which one to use depends on the analysis being employed (KVL or KCL) and whether the initial capacitor voltage is zero.

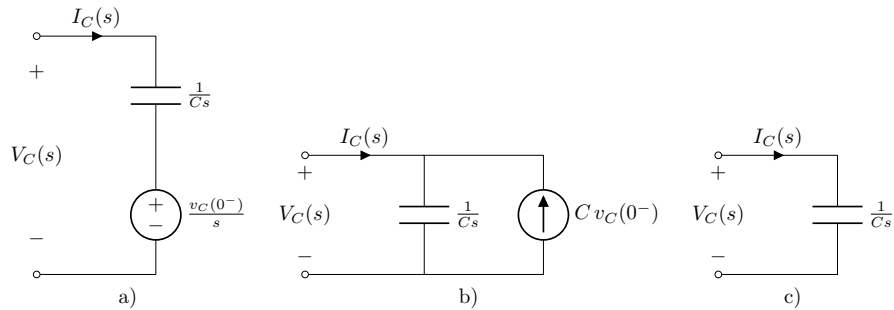


Figure 1: Equivalent circuit models for a capacitor in the Laplace domain. a) is most useful with a KVL analysis, b) is most useful with a KCL analysis, c) model when the initial capacitor voltage is zero.

Inductor Model

Using the above relationships for inductor voltage and current after using the Laplace transform, we get two equivalent circuit models. Which one to use depends on the analysis being employed (KVL or KCL) and whether the initial inductor current is zero.

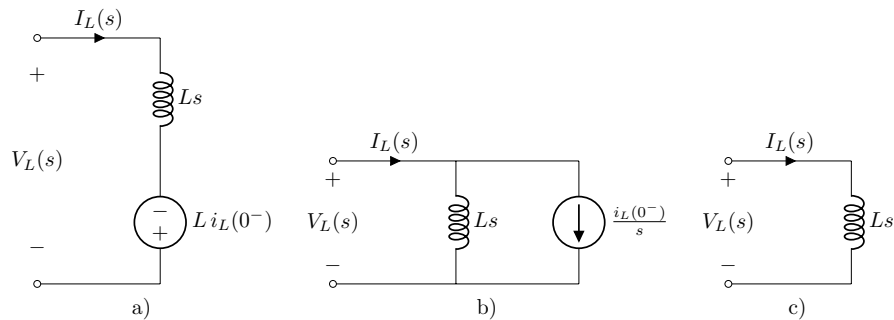


Figure 2: Equivalent circuit models for an inductor in the Laplace domain. a) is most useful with a KVL analysis, b) is most useful with a KCL analysis, c) model when the initial inductor current is zero.

Example 1

Consider the familiar series RC circuit in the figure below where the input voltage $x(t)$ and output voltage $y(t)$ are indicated. However now there is a non-zero initial capacitor voltage, $v_C(0^-) = V_0$. On the left we have the original circuit in the time-domain and on the right the transformed circuit in the Laplace domain. Since I plan to use a KCL analysis I choose the transformation that suits that approach best.

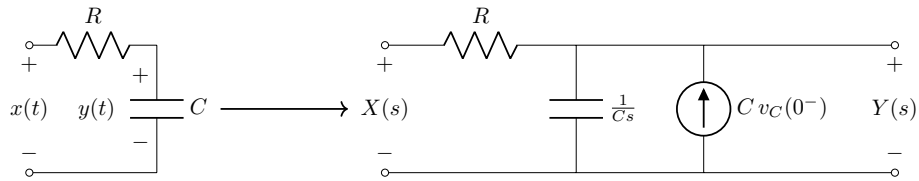


Figure 3: Circuit model before and after transformation using Laplace.

Using a KCL:

$$\frac{X(s) - Y(s)}{R} + Cv_C(0^-) = \frac{Y(s)}{\frac{1}{Cs}} = CsY(s)$$

Rearranging and substituting for $v_C(0^-)$ we get

$$(1 + RCs)Y(s) = X(s) + RCV_0$$

and

$$Y(s) = \frac{1}{1 + RCs}X(s) + \frac{RCV_0}{1 + RCs}$$

If $V_0 = 0$ then $Y(s) = \frac{1}{1 + RCs}X(s)$ and the transfer function is $H(s) = \frac{Y(s)}{X(s)} = \frac{1}{1 + RCs}$.

Example 2

Consider the series RLC with zero initial conditions for the inductor current and capacitor voltage.

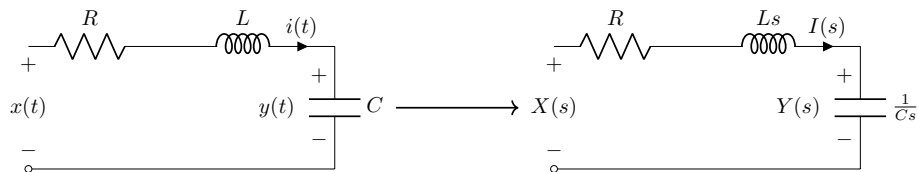


Figure 4: Circuit model before and after transformation using Laplace.

Using a KVL around the loop gives:

$$\begin{aligned}
X(s) &= RI(s) + LSI(s) + \frac{1}{Cs}I(s) \\
CsX(s) &= RCsI(s) + CLs^2I(s) + I(s) \\
CsX(s) &= (RCs + CLs^2 + 1) \cdot I(s) \\
I(s) &= \frac{Cs}{CLs^2 + RCs + 1} \cdot X(s)
\end{aligned}$$

The output is then

$$Y(s) = \frac{1}{Cs}I(s) = \frac{1}{CLs^2 + RCs + 1} \cdot X(s)$$

and the transfer function is

$$H(s) = \frac{Y(s)}{X(s)} = \frac{1}{CLs^2 + RCs + 1}$$

Example 3

This works with active (op-amp) circuits as well. Consider a second order circuit using the Sallen-Key topology.

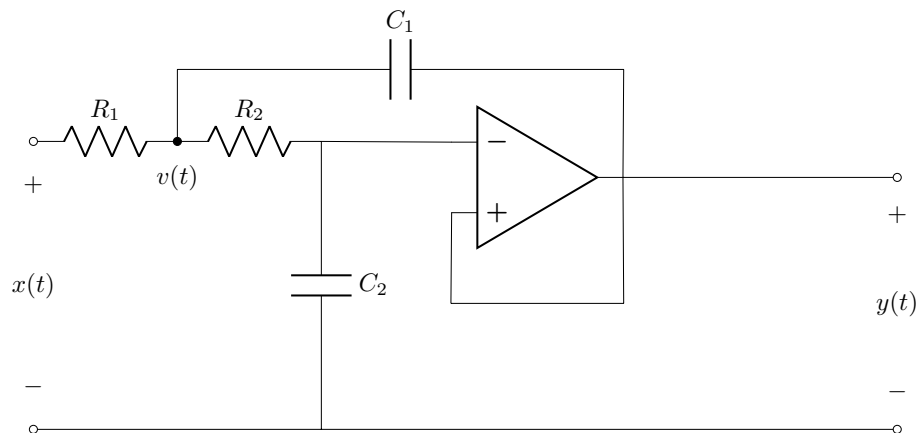


Figure 5: Circuit model in time domain.

If we assume the system is at rest when the input is applied, the initial capacitor voltages are zero and we can transform the circuit into the Laplace domain.

Doing a KCL at node V :

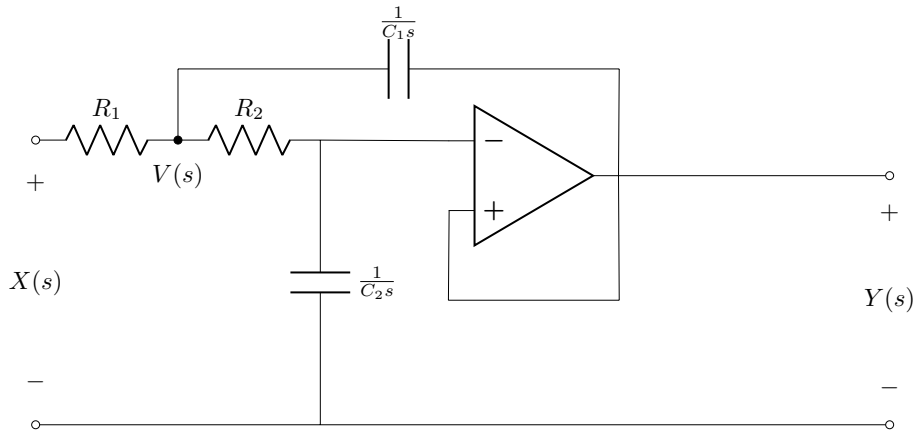


Figure 6: Circuit model in Laplace domain.

$$\frac{X(s) - V(s)}{R_1} + \frac{Y(s) - V(s)}{\frac{1}{C_1 s}} = \frac{V(s) - Y(s)}{R_2}$$

Doing a KCL at the non-inverting input of the op-amp:

$$\frac{V(s) - Y(s)}{R_2} = \frac{Y(s)}{\frac{1}{C_2 s}}$$

Given these two algebraic equations in X , Y , and V one can eliminate the the node voltage V and arrive at the transfer function. This is left as "an exercise for the reader".

Analyzing Block Diagrams

Recall from ECE 2714 that block diagrams can be used to model systems and implement/realize systems.

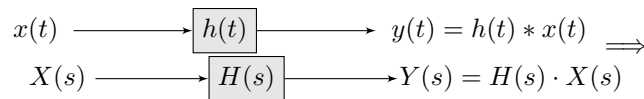


Figure 7: Transformation of input and output to Laplace domain (transfer function) for a single block.

- series connection, the overall transfer function $H(s) = H_1(s) \cdot H_2(s)$
- parallel connection, the overall transfer function $H(s) = H_1(s) + H_2(s)$



Figure 8: Transformation of input and output to Laplace domain for a series connection.

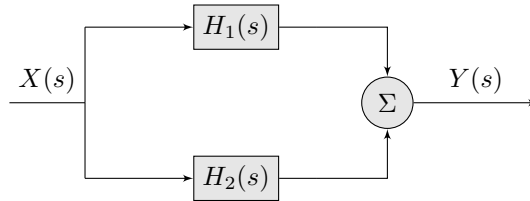


Figure 9: Transformation of input and output to Laplace domain for a parallel connection.

- feedback connection, the overall transfer function $H(s) = \frac{H_1(s)}{1+H_1(s) \cdot H_2(s)}$

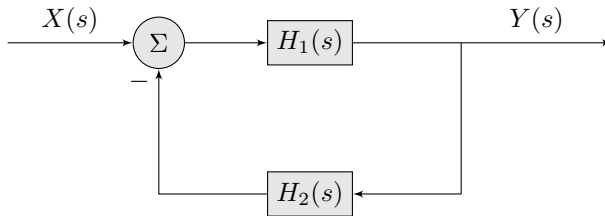


Figure 10: Transformation of input and output to Laplace domain for a feedback connection.

Note this is much easier than in the time domain.

We can use block diagrams to derive an overall transfer function from sub-components. To illustrate this consider three examples.

Example 4: PID Controller

The proportional-integral-derivative (PID) controller is the prototypical system in control theory and used extensively in applications. It can be described by the following block diagram:

where

- $H_1(s) = K_1$ for $K_1 \in \mathbb{R}$, is the proportional part of the controller;
- $H_2(s) = K_2 \cdot s$ for $K_2 \in \mathbb{R}$ is the derivative part of the controller;
- $H_3(s) = \frac{K_3}{s}$ for $K_3 \in \mathbb{R}$ is the integral part of the controller;

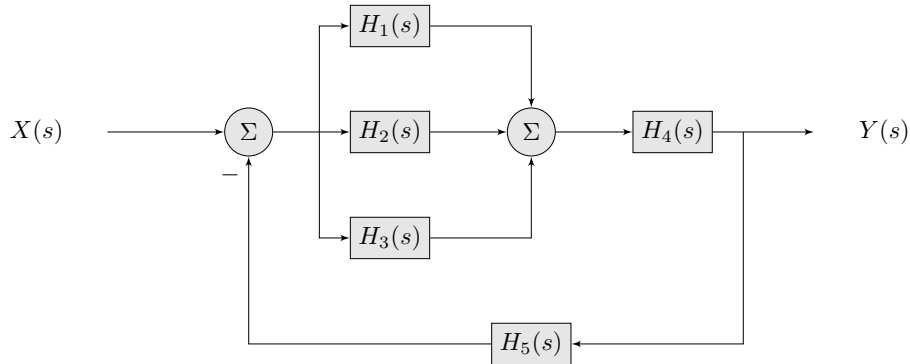


Figure 11: Block Diagram of PID controller.

- $H_4(s)$ is the transfer function that corresponds to the system being controlled, traditionally called the *plant*;
- $H_5(s)$ is the transfer function for the *measurement* part of the system.

The PID controller is made up of the three parallel blocks, H_1 , H_2 , and H_3 . It filters, or processes the difference in the input and the output after measurement, called the *error*, $X(s) - H_5(s) \cdot Y(s)$. The output of the controller is input to the system being controlled, H_4 , called the *control variable*, and its output, $Y(s)$, is the signal we are interested in controlling.

We can derive an overall transfer function for the controlled system using the rules for simplifying block diagrams. We might approach this as follows.

1. The controller itself consists of three parallel blocks. This can be replaced by a single block whose transfer function is the sum of the three:

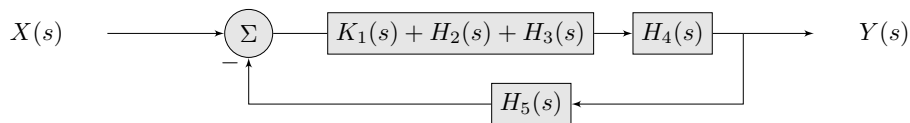


Figure 12: First simplification for the PID controller block diagram.

2. The simplified controller and the plant are in series, thus their combination can be replaced by their product:

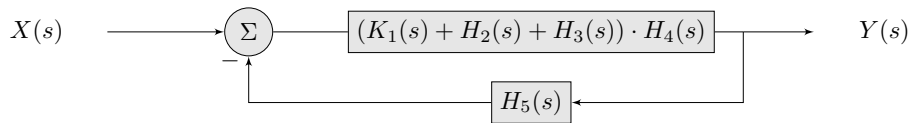


Figure 13: Second simplification for the PID controller block diagram.

3. Now we apply the feedback motif to arrive at the overall transfer function:

$$H(s) = \frac{(K_1(s) + H_2(s) + H_3(s)) \cdot H_4(s)}{1 + (K_1(s) + H_2(s) + H_3(s)) \cdot H_4(s) \cdot H_5(s)}$$

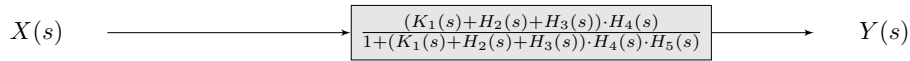


Figure 14: Final simplification for the PID controller block diagram.

After substituting for the PID block transfer functions and simplifying we arrive at:

$$H(s) = \frac{(K_2s^2 + K_1s + K_3) \cdot H_4(s)}{(K_2s^2 + K_1s + K_3) \cdot H_4(s) \cdot H_5(s) + s}$$

Choosing the best values for K_1 , K_2 , and K_3 (and defining what best means) is called "tuning" the controller. This depends on the plant transfer function $H_4(s)$ and the feedback transfer function $H_5(s)$, and is part of the control system design process. This process also takes into account measures of system stability as well as the rise and settling time of the output due to a step input.

Example 5: DC Motor Model

A DC motor converts an input voltage, $v_a(t)$, to a current, $i_a(t)$, through the windings, generating a torque on the motor shaft. An external torque and shaft mechanical dynamics give rise to rotational motion, i.e. a rotational velocity, ω , and it's integral, rotational position, α . This can be approximated by the linear system with block diagram:

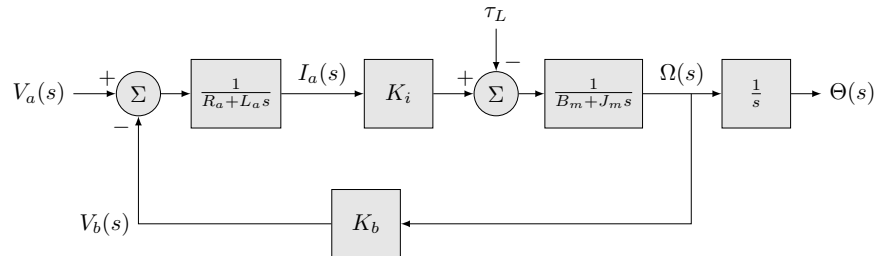


Figure 15: A block diagram for a DC motor.

where

- R_a and L_a are the armature resistance and inductance
- K_i is the torque gain

- K_b is the back EMF gain
- J_m and B_m are the rotor inertia and it's viscous friction coefficient

We will assume the load torque is zero (to make this a single input system). Then using the rules for simplifying the block diagram we combine the three series blocks modeling the armature, current-to-torque, and mechanical inertia and bearing friction. Then we use the feedback motif to combine the forward model and back EMF model. This is then combined in series with the rotational velocity to position integrator block to define the overall transfer function from input armature voltage to output shaft angle.

$$\begin{aligned}
 H(s) &= \frac{\Theta(s)}{V_a(s)} = \frac{\frac{K_i}{(R_a + L_a s) \cdot (B_m + J_m s)}}{s \cdot \left[1 + \frac{K_i \cdot K_b}{(R_a + L_a s) \cdot (B_m + J_m s)} \right]} \\
 &= \frac{K_i}{L_a J_m s^3 + (R_a J_m + L_a B_m) s^2 + (R_a B_m + K_i K_b) s}
 \end{aligned}$$

Determining the values for model parameters R_a , L_a , K_i , K_b , B_m , and J_m is a system identification problem. This is typically accomplished using a calibration step that collects data and uses a learning algorithm to adjust the parameters to fit the data.

Example 6: DC Servo

A servo-motor, or just "servo", is a electro-mechanical device whose input is a desired angle for a rotary shaft as an electrical signal and whose output is actual angle of the rotary shaft. We can build a linear system model for a DC servo using the previous two examples in combination.

Let the plant transfer function $H_4(s)$ from Example 4 be given by that of the DC motor in Example 5. That is the output of the PID block is the armature voltage to be applied to the motor, and the output is the rotation angle of the motor shaft. For simplicity assume the feedback transfer function has no gain or dynamics, so that $H_5(s) = 1$ (the angle is measured exactly and instantaneously). Essentially the system measures the error between the desired and actual angle, filters the error signal through the PID block, and applies that output as the armature voltage.

To arrive at the mathematical model, we can substitute the plant transfer function by that of the DC motor and simplify. This gives us the overall transfer function for the servo:

$$H(s) = \frac{X(s)}{\Theta(s)} = \frac{K_i K_2 s^2 + K_i K_1 s + K_i K_3}{L_a J_m s^4 + (R_a J_m + L_a B_m) s^3 + [(R_a B_m + K_i K_b) + K_i K_2] s^2 + K_i K_1 s + K_i K_3}$$

where X is the desired input angle and Θ is the actual angle. While this might look complicated, most the terms are constants. You should now be able to recognize it as a fourth-order system with two zeros and four poles, whose locations depend on both the DC motor parameters, which are typically fixed, and the PID parameter settings, which can be adjusted to ensure stability and maximize performance.

System Realization

We can also use block diagrams in the opposite way, to implement, or *realize*, a given transfer function (for example from a filter design) using three basic building blocks: summation, amplifier (gain), and integrator.

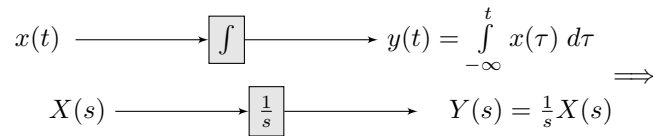


Figure 16: Transformation of an integrator to Laplace domain (transfer function).

The realization is not unique, with different canonical forms, each with advantages and disadvantages (e.g. reduced number of components and reduced sensitivity to component variation).

Direct Forms for Realization

It is best to use an example by way of explaining system realization. The extension to the general case is then easier to see.

Given a transfer function

$$H(s) = \frac{s + a}{s^2 + bs + c} = \frac{Y(s)}{X(s)}$$

for constants $a, b, c \in \mathbb{R}$, we wish to implement the corresponding system in terms of summations, amplifiers, and integrators.

We first use the definition of the transfer function to write the input output relationship in the Laplace domain as

$$Y(s) (s^2 + bs + c) = (s + a) X(s)$$

Next, we distribute the input and output signals over the polynomials

$$s^2Y(s) + bsY(s) + cY(s) = sX(s) + aX(s)$$

Then we divide through by s^2 to put each term in the form of an integrator

$$Y(s) + \frac{b}{s}Y(s) + \frac{c}{s^2}Y(s) = \frac{1}{s}X(s) + \frac{a}{s^2}X(s)$$

And solve for $Y(s)$

$$Y(s) = -\frac{b}{s}Y(s) - \frac{c}{s^2}Y(s) + \frac{1}{s}X(s) + \frac{a}{s^2}X(s)$$

We can then translate that directly to a block diagram

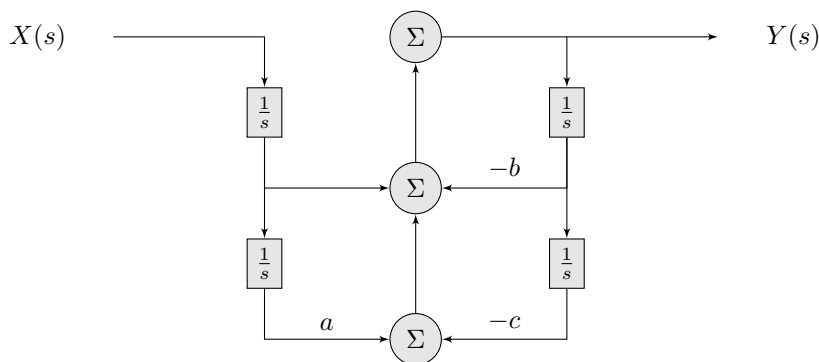


Figure 17: Direct Form I realization of the example transfer function.

The above is called the *Direct Form I* implementation/realization of the system. The major drawback of this form is that it requires four integrators to implement a second order system. Our intuition is that we should just need two. To see how to do this we note we can split the above block diagram into two stages in series by splitting the top summation and labeling the intervening signal as $W(s)$:

The first stage corresponds to the transfer function $H_1(s) = \frac{W(s)}{X(s)}$, while the second stage corresponds to the transfer function $H_2(s) = \frac{Y(s)}{W(s)}$, and the overall transfer function is $H(s) = H_1(s) \cdot H_2(s)$.

Recall from 2714 and the commutivity of multiplication that the order of two series connections does not affect the overall system. Thus if we reverse the two stages, i.e. $H(s) = H_2(s) \cdot H_1(s)$, we get the block diagram

Now we note that the signal $W(s)$ is duplicated through two identical paths with two integrators in series, giving duplicate outputs. This redundancy can

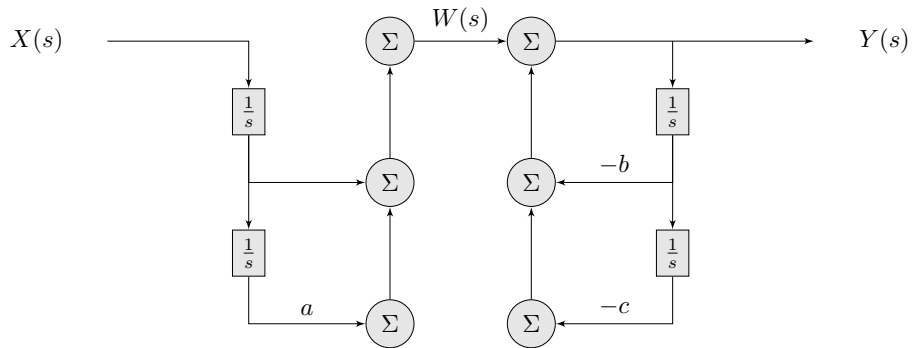


Figure 18: Direct Form I realization of the example transfer function, split into two stages.

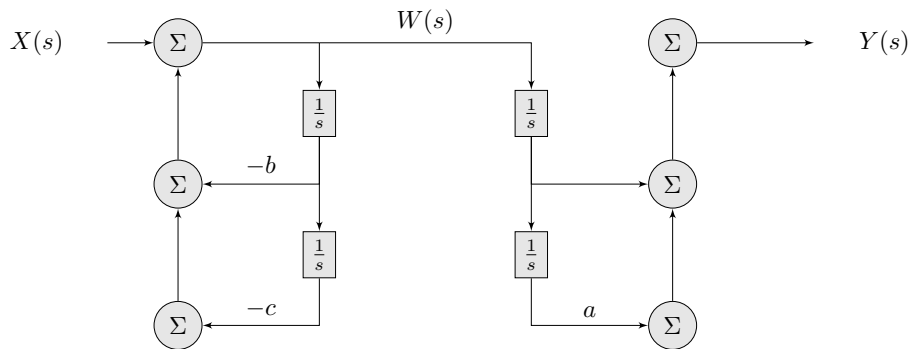


Figure 19: Direct Form I realization of the example transfer function, split into two stages, with the order reversed.

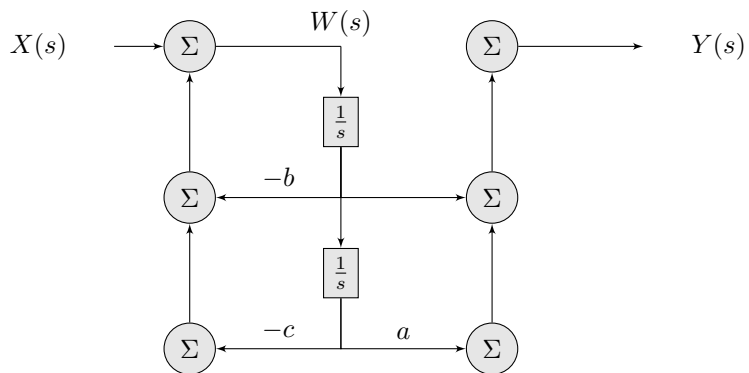


Figure 20: Direct Form II realization of the example transfer function.

be removed if we combine the integrators. This gives what is called the *Direct Form II* realization of the system.

A result from graph theory says that we can reverse the direction of signal flow and interchange the input and output in a block diagram without modifying the overall transfer function. This is called the *Transposed Direct Form II* realization and looks like

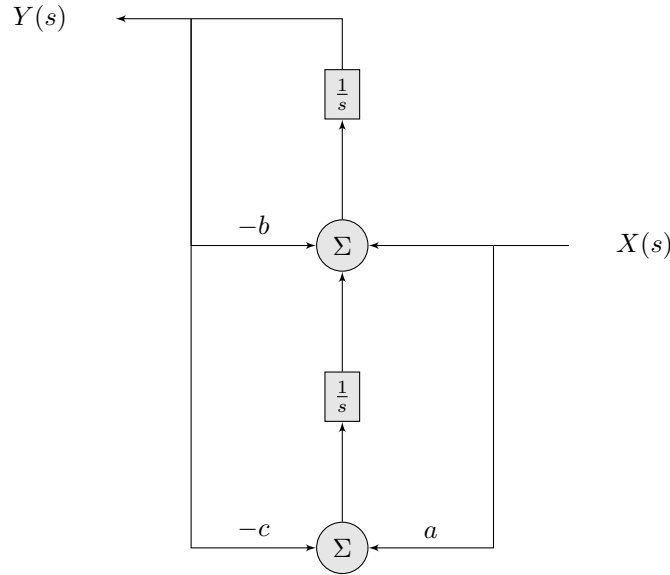


Figure 21: Transposed Direct Form II realization of the example transfer function.

Note the above block diagram can be mirrored left-right to give the normal signal flow of left-to-right. The transposed direct form II typically has the fewest blocks and summation nodes.

There are additional forms, including the *Lattice*, *Parallel*, and *Cascade of Second-Order Stages* (SOS). The Lattice is outside the course scope but the Parallel and SOS are easy to describe.

Parallel Forms for Realization

The parallel form requires real poles for first-order blocks. It can also use second-order blocks for system order greater than three with complex poles. To illustrate, consider a transfer function with real poles, e.g.

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

If we perform the partial fraction expansion then

$$H(s) = \frac{A}{s+1} + \frac{B}{s+2} = \frac{2}{s+1} + \frac{-2}{s+2}$$

We can then draw this as a parallel connection of two first-order systems

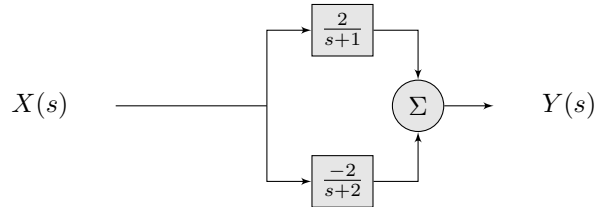


Figure 22: Partial parallel realization of the example transfer function.

We can then break down each parallel block into a transposed direct form II to arrive at the final parallel realization.

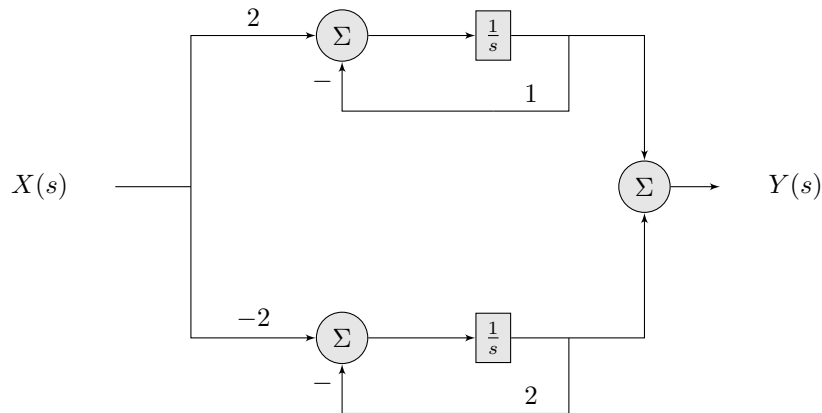


Figure 23: Parallel realization of the example transfer function.

Exercise: Confirm the block diagram for the final parallel realization has an overall transfer function we started with.

Suppose the system was fourth-order with two pairs of complex conjugate poles. Then the partial fraction expansion would be into the sum of two second order terms, and each parallel branch would be a direct-form I/II/transposed form.

Realization via Cascade of Second-Order Stages

The last common realization we will consider is the cascade of second-order stages. Here the transfer function is factored into multiplicative terms of second order

(with one first-order stage if the system has odd order). These are then placed in series, with each block typically implemented as a direct-form II, transposed direct-form II, or using a specific circuit topology like Sallen-Key.

For example, consider a transfer function

$$H(s) = \frac{25}{s^4 + 8s^3 + 26s^2 + 40s + 25}$$

which has four poles at $(-1 \pm j)$ and $(-1 \pm j2)$. If we group the conjugate poles together then the denominator can be factored into two second order polynomials with real coefficients

$$H(s) = \frac{25}{(s^2 + 4s + 5)(s^2 + 2s + 5)}$$

We can write this as the product of two second-order systems by choosing a factoring of the numerator

$$H(s) = \frac{5}{s^2 + 4s + 5} \cdot \frac{5}{s^2 + 2s + 5} = H_1(s) \cdot H_2(s)$$

This corresponds to a block diagram that is a cascade (series connection) of two second-order stages



Figure 24: Cascade of Second-Order Stages realization of the example fourth-order transfer function.

As in the parallel case, each block can be implemented as a parallel or direct form. For example using the direct form II implementation of each block

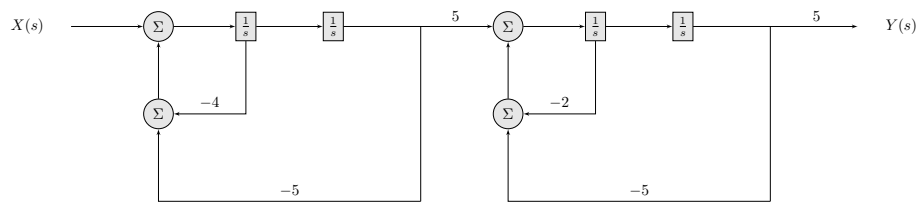


Figure 25: Cascade of Second-Order Stages realization of the example fourth-order transfer function.