Assignment 1

Rohith Ingilela
EE19BTECH11005
Dept. of Electrical Engineering
IIT Hyderabad

September 10, 2020

Problem Statement

For each of the following transfer functions, write the corresponding differential equation.

(a) $\frac{X(s)}{F(s)} = \frac{7}{s^2+5s+10}$

(b) $\frac{X(s)}{F(s)} = \frac{15}{(s+10)(s+11)}$

(c) $\frac{X(s)}{F(s)} = \frac{s+3}{s^3+11s^2+12s+18}$

Solution

We know that, multiplication by "s" in the Laplace domain is equivalent to differentiation in the time domain.

$$\mathcal{L}^{-1}\{as^nX(s)\} = a\frac{d^nx}{dt^n}$$

(a)

$$\frac{X(s)}{F(s)} = \frac{7}{s^2+5s+10}$$

$$s^2X(s) + 5sX(s) + 10X(s) = 7F(s)$$

Applying inverse laplace transform, the corresponding differential equation is

$$\boxed{\frac{d^2x}{dt^2} + 5\frac{dx}{dt} + 10x(t) = 7f(t)}$$

(b)

$$\frac{X(s)}{F(s)} = \frac{15}{(s+10)(s+11)}$$

$$\frac{X(s)}{F(s)} = \frac{15}{s^2+21s+110}$$

$$s^2X(s) + 21sX(s) + 110X(s) = 15F(s)$$

Applying inverse laplace transform, the corresponding differential equation is

$$\boxed{\frac{d^2x}{dt^2} + 21\frac{dx}{dt} + 110x(t) = 15f(t)}$$

(c)

$$\frac{X(s)}{F(s)} = \frac{s+3}{s^3+11s^2+12s+18}$$

$$s^3X(s)+11s^2X(s)+12sX(s)+18X(s) = sF(s)+3F(s)$$

Applying inverse laplace transform, the corresponding differential equation is

$$\boxed{\frac{d^3x}{dt^3}+11\frac{d^2x}{dt^2}+12\frac{dx}{dt}+18x(t) = \frac{df}{dt}+3f(t)}$$

Code

Python code to convert a transfer function to differential equation:

```python
from sympy.integrals.transforms import inverse_laplace_transform, InverseLaplaceTransform
from sympy.abc import s, t
from sympy import pprint, fraction, Function, Wild, Derivative, Eq, Pow

def transfer_to_differential(tf, fun_X = Function('X'), fun_F = Function('F')):
    tf = fraction(tf)
    res = Eq(inverse_laplace_transform(tf[1] * fun_X(s), s, t), inverse_laplace_transform(tf[0] * fun_F(s), s, t))

    wf = Wild('w')
    ilw = InverseLaplaceTransform(wf, s, t, None)

    for exp in res.find(ilw):
        e = exp.match(ilw)[wf]
        args = e.args
        if len(args) == 2:
            p = 1 if not isinstance(args[0], Pow) else args[0].args[1]
            newexp = Derivative(Function(args[1].name.lower())(t), t, p)
            res = res.replace(exp, newexp)
        elif len(args) == 1:
            newexp = Function(e.name.lower())(t)
            res = res.replace(exp, newexp)

    return res
```

Full source with above solutions at:
`https://github.com/1R0H1TH/EE2101/blob/master/transferToDifferential.py`