UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA ELÉTRICA E ELETRÔNICA

Disciplina: Controle Digital Aplicado em Eletrônica de Potência

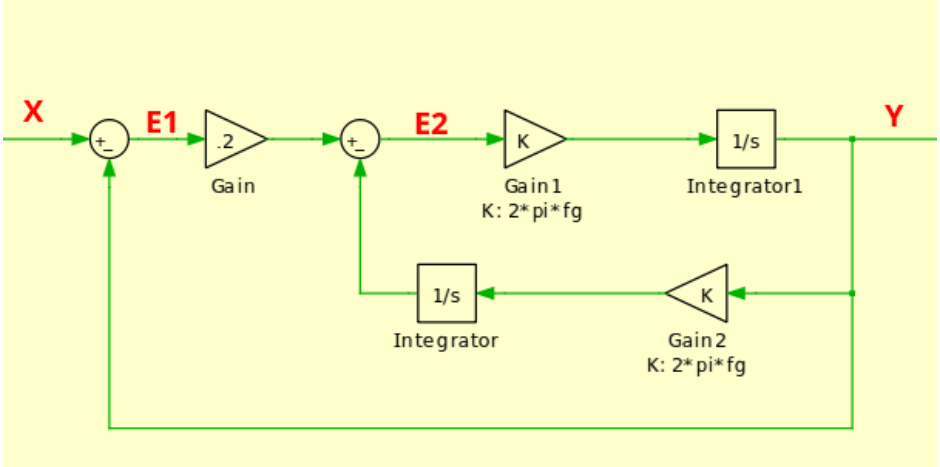
Discente: Eduardo Eller Behr

Planilha de cálculo do SOGI-PLL

In [ ]:
```
import sympy as sp
from sympy import pi, Eq, solve
```

In [ ]:
```
Xs, Ys, Xz, Yz, fg, K, s, z, Ts = sp.symbols('X(s) Y(s) X(z) Y(z) f_g K s z T_s')
```

Dedução da função de transferência do SOGI-PLL no domínio s



In [ ]:
```
E1 = Xs-Ys
E2 = K*E1-(2*pi*fg/s)*Ys
eq_y = sp.Eq(Ys, 2*pi*fg*E2/s); eq_y
```

Out[ ]:
$$Y(s) = \frac{2\pi f_g \left(K\left(X(s) - Y(s)\right) - \frac{2\pi Y(s) f_g}{s}\right)}{s}$$

In [ ]:
```
eq_tf_s = Eq(Ys/Xs, solve(eq_y, Ys)[0]/Xs)
print("Função de transferência em s do SOGI-PLL:"); display(eq_tf_s)
```

Função de transferência em s do SOGI-PLL:

$$\frac{Y(s)}{X(s)} = \frac{2\pi K f_g s}{2\pi K f_g s + 4\pi^2 f_g^2 + s^2}$$

Conversão da função de transferência do SOGI-PLL para o domínio z

In [ ]:
```
def z_transform(eq: Eq):
    rhs = eq.rhs.subs({s: (2/Ts)*(z-1)/(z+1)}).simplify()
    lhs = Yz/Xz
    return Eq(lhs, rhs)
```

In [ ]:
```
eq_tf_z = z_transform(eq_tf_s)
print("Função de transferência em z do SOGI-PLL:"); display(eq_tf_z)
```

Função de transferência em z do SOGI-PLL:

$$\frac{Y(z)}{X(z)} = \frac{\pi K T_s f_g \left(z-1\right)\left(z+1\right)}{\pi K T_s f_g \left(z-1\right)\left(z+1\right) + \pi^2 T_s^2 f_g^2 (z+1)^2 + (z-1)^2}$$

In [ ]:
```
rhs_numerator = eq_tf_z.rhs.as_numer_denom()[0]
rhs_denominator = eq_tf_z.rhs.as_numer_denom()[1]
lhs_numerator = eq_tf_z.lhs.as_numer_denom()[0]
lhs_denominator = eq_tf_z.lhs.as_numer_denom()[1]

eq_expanded = Eq(lhs_numerator*rhs_denominator, rhs_numerator*lhs_denominator).expand().simplify()
eq_expanded
```

Out[ ]:
$$\pi K T_s X(z) f_g \left(z^2 - 1\right) = Y(z) \left(\pi K T_s f_g z^2 - \pi K T_s f_g + \pi^2 T_s^2 f_g^2 z^2 + 2\pi^2 T_s^2 f_g^2 z + \pi^2 T_s^2 f_g^2 + z^2 - 2z + 1\right)$$

In [ ]:
```
eq_offset = Eq(eq_expanded.lhs/z**2, eq_expanded.rhs/z**2).expand().simplify(); eq_offset
```

Out[ ]:
$$\frac{\pi K T_s X(z) f_g \left(z^2 - 1\right)}{z^2} = \frac{Y(z) \left(-\pi K T_s f_g + \pi^2 T_s^2 f_g^2 + z^2 \left(\pi K T_s f_g + \pi^2 T_s^2 f_g^2 + 1\right) + 2z \left(\pi^2 T_s^2 f_g^2 - 1\right) + 1\right)}{z^2}$$

In [ ]:
```
eq_subs = eq_offset.subs({
    Ts: 1e-4,
    K: 0.2,
    fg: 50
}).evalf(6)
eq_subs
```

Out[ ]:
$$\frac{0.00314159 X(z) \left(z^2 - 1.0\right)}{z^2} = \frac{Y(z) \left(1.00339 z^2 - 1.99951 z + 0.997105\right)}{z^2}$$