

T06P03

June 2, 2020

1 T06P03 - MdSM

1.1 Enunciado

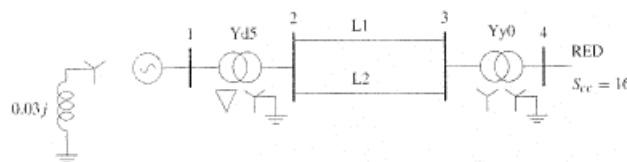
La red de la figura incluye los datos expresados en valores pu para una misma base. Determine:

1. Potencia de cortocircuito trifásica en el nudo 3.
2. Intensidades por las fases de la línea L1 tras una falta fase-fase en el nudo 2.
3. Módulo de la intensidad por el neutro del transformador T1 (Yd5) durante la falta anterior.

Tome como datos:

- Generador: $X_0 = 0.05$, $X_1 = X_2 = 0.10$.
- Transformador T1 (Yd5): $X_0 = 0.10$, $X_1 = X_2 = 0.12$.
- Transformador T2 (Yy0): $X_0 = X_1 = X_2 = X_{cc} = 0.15$.
- Línea L1: $X_0 = 0.16$, $X_1 = X_2 = 0.04$.
- Línea L2: $X_0 = 0.32$, $X_1 = X_2 = 0.08$.

Sistema eléctrico:



1.2 Resolución

```
[1]: # Importar librerías genéricas:
import numpy as np
import cmath as cm
import matplotlib.pyplot as plt

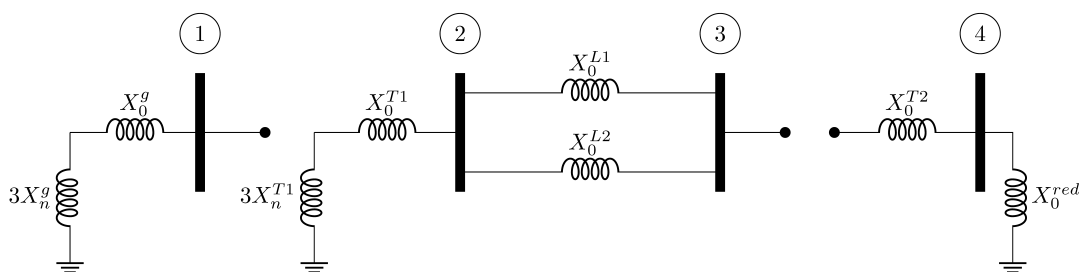
# Formato general de salida de resultados numéricos:
np.set_printoptions(precision=3, suppress=True)
```

1.2.1 Datos del problema

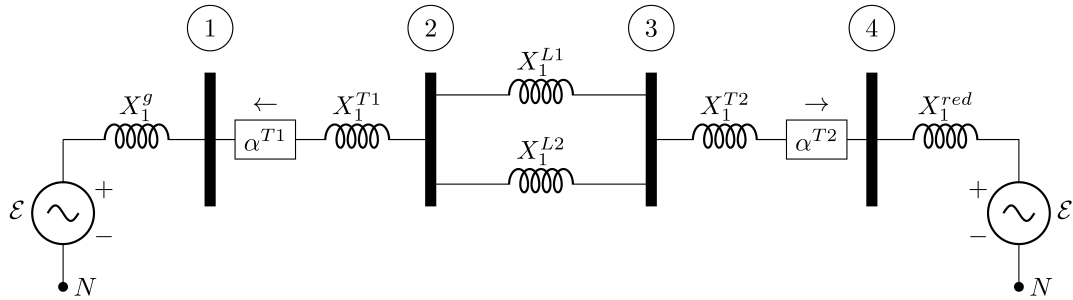
```
[2]: # Impedancias a las tres secuencias [homopolar=0, directa=1 e inversa=2] del
      ↪ generador [pu]:
Zg = np.array([0.05j,0.1j,0.1j])
# Impedancia de puesta a tierra del neutro del generador [pu]:
Zn = 0.03j
# Impedancias del transformador T1 [pu]:
Zt1 = np.array([0.1j,0.12j,0.12j])
# Impedancias absolutas de la línea L1 [pu]:
Zl1 = np.array([0.16j,0.04j,0.04j])
# Impedancias absolutas de la línea L2 [pu]:
Zl2 = np.array([0.32j,0.08j,0.08j])
# Impedancias del transformador T2 [pu]:
Zt2 = np.array([0.15j,0.15j,0.15j])
# Impedancia de cortocircuito de la red [pu]:
Ucc = 1
Scc = 16
Zcc = (Ucc**2/Scc)*1j
Zred = np.array([Zcc,Zcc,Zcc])
# Tensión del generador:
E = 1
# Desfase del transformador T1 [deg]:
DesfT1 = -150
# Desfase del transformador T2 [deg]:
DesfT2 = 0
```

1.2.2 Redes de secuencia del sistema

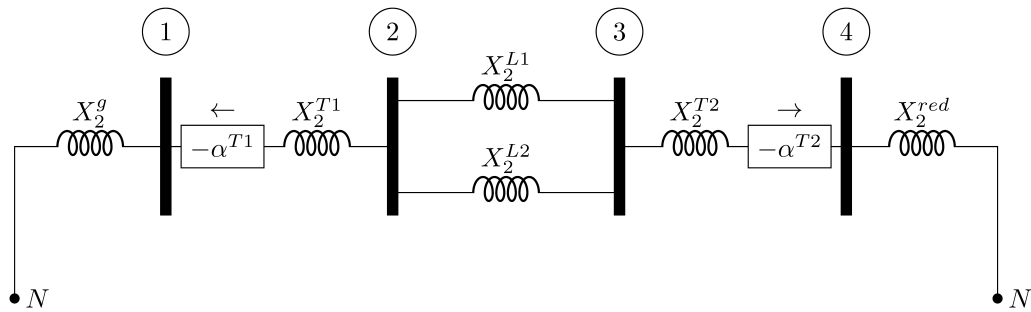
Red de secuencia homopolar/cero (RSH):



Red de secuencia directa (RSD):



Red de secuencia inversa (RSI):



1.2.3 Cálculo de los equivalentes de Thévenin de las redes de secuencia en el nudo 3

1.2.4 Red de secuencia homopolar/cero (RSH)

```
[3]: Z0 = Zt1[0]+((Zl1[0]*Zl2[0])/(Zl1[0]+Zl2[0]))
      print('Z0 = {:.3j}'.format(Z0))
```

Z0 = 0.207j

1.2.5 Red de secuencia directa (RSD)

```
[4]: # El equivalente entre el punto de análisis y tierra es el resultado del
      ↪ paralelo de dos ramas: izquierda (Rama A1) y derecha (Rama B1).
      RamaA1 = Zg[1]+Zt1[1]+((Zl1[1]*Zl2[1])/(Zl1[1]+Zl2[1]))
      RamaB1 = Zt2[1]+Zred[1]
      Z1 = RamaA1*RamaB1/(RamaA1+RamaB1)
      print('Z1 = {:.3j}'.format(Z1))
```

Z1 = 0.114j

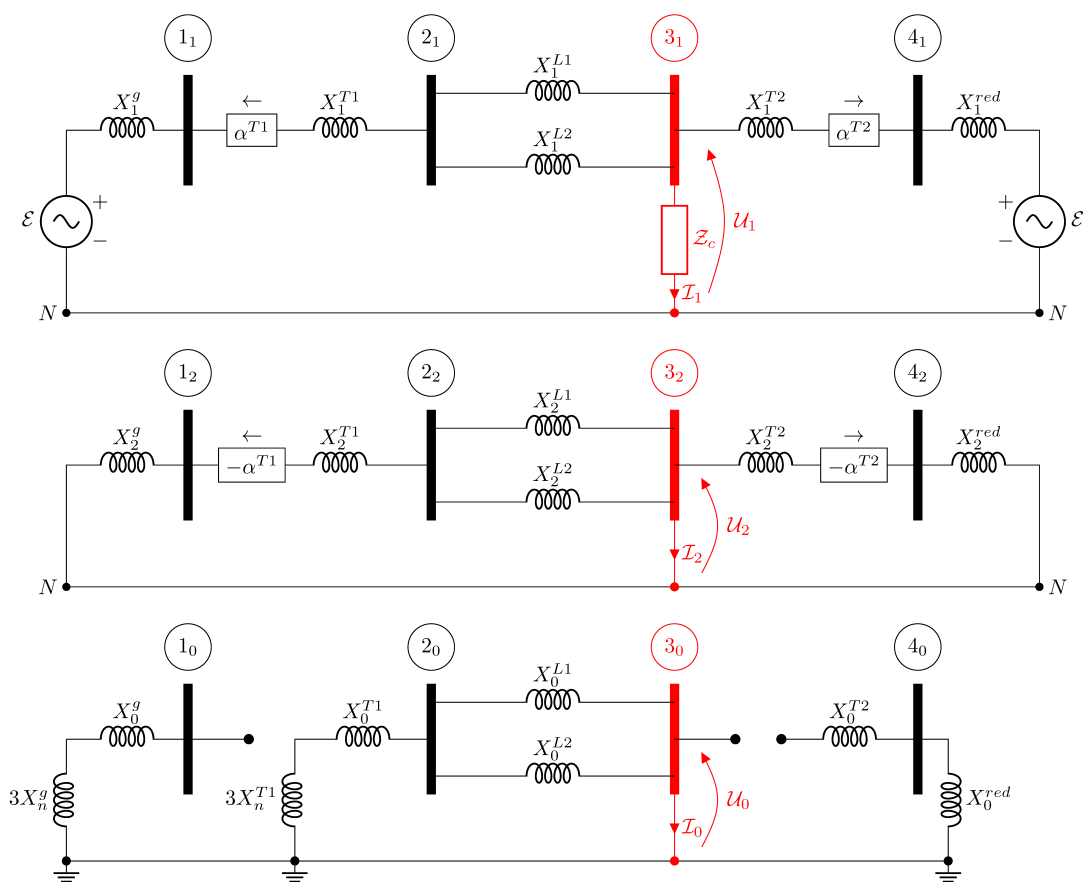
1.2.6 Red de secuencia inversa (RSI)

```
[5]: # Ídem que la RSD.
RamaA2 = Zg[2]+Zt1[2]+((Zl1[2]*Zl2[2])/(Zl1[2]+Zl2[2]))
RamaB2 = Zt2[2]+Zred[2]
Z2 = RamaA2*RamaB2/(RamaA2+RamaB2)
print('Z2 = {:.3}'.format(Z2))
```

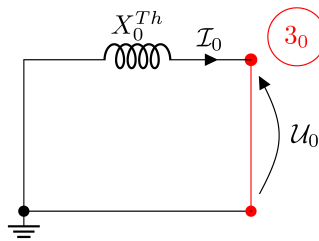
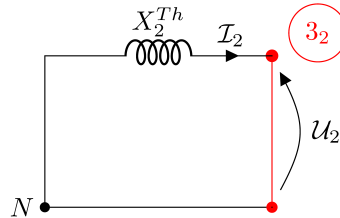
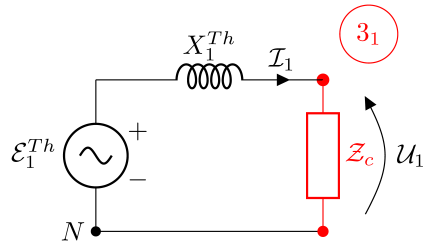
$Z2 = 0.114j$

1.2.7 Asociación de redes de secuencia (falta trifásica)

Ante falta trifásica a tierra en la barra 3, las tres redes de secuencia están desacopladas:



Considerando los equivalentes de Thévenin de las tres redes, el esquema queda simplificado:



[6]: *# Como es falta trifásica, las tres redes están desacopladas, y por lo tanto el*
→ cálculo de las corrientes de las tres redes de secuencia queda:

```
Z012 = np.array([Z0,Z1,Z2])
I0 = 0
I1 = E/Z012[1]
I2 = 0
I012 = np.array([I0,I1,I2])
print('I012 = ',I012)
U0 = 0
U1 = E-I1*Z012[1]
U2 = 0
U012 = np.array([U0,U1,U2])
print('U012 = ',U012)
```

```
I012 = [0.+0.j  0.-8.76j 0.+0.j  ]
U012 = [0.+0.j  0.+0.j 0.+0.j]
```

[7]: *## Transformación de las componentes simétricas a las componentes de fase*
→ (cambio de base)

```
# Matriz de Fortescue-Stovkis
a = cm.rect(1,120*cm.pi/180)
```

```

T = np.array([[1,1,1],[1,a**2,a],[1,a,a**2]])
# Corriente de falta en coordenadas rectangulares:
IRST = np.dot(T,np.transpose(I012))
print('IRST = ',IRST)
# Corriente de falta en coordenadas polares:
IRSTpolar=np.zeros([3,2])
for i in range(0,3):
    IRSTpolar[i] = cm.polar(IRST[i])
    IRSTpolar[i][1] = IRSTpolar[i][1]*180/cm.pi
print('Corriente de falta (coordenadas polares):',IRSTpolar)
# Tensiones en el punto de falta en coordenadas rectangulares:
URST = np.dot(T,np.transpose(U012))
print('URST = ',URST)
# Tensiones en el punto de falta en coordenadas polares:
URSTpolar=np.zeros([3,2])
for i in range(0,3):
    URSTpolar[i] = cm.polar(URST[i])
    URSTpolar[i][1] = URSTpolar[i][1]*180/cm.pi
print('Tensiones en el punto de falta (coordenadas polares):',URSTpolar)

```

```

IRST = [ 0. -8.76j -7.586+4.38j  7.586+4.38j]
Corriente de falta (coordenadas polares): [[ 8.76 -90. ]
 [ 8.76 150. ]
 [ 8.76  30. ]]
URST = [0.+0.j 0.+0.j 0.+0.j]
Tensiones en el punto de falta (coordenadas polares): [[0. 0.]
 [0. 0.]
 [0. 0.]]

```

Una vez con las magnitudes en valores reales, la potencia de cortocircuito trifásica se determina mediante:

$$S_{cc} = \mathcal{E}I_R^* + {}^2\mathcal{E}I_S^* + {}^1\mathcal{E}I_T^*.$$

En el caso de falta trifásica, dado que las corrientes de falta en las tres fases son iguales, se puede calcular también como:

$$S_{cc} = 3\mathcal{E}I_R^*.$$

Y se debe verificar que es igual a la suma de las potencias en el dominio de las componentes simétricas, es decir:

$$S_{cc} = 3\mathcal{E}I_1^*.$$

```

[8]: # Potencia de cortocircuito (trifásica) en la barra 3 calculada mediante las
      ↪ componentes de fase:

```

```

Scc3 = 3*E*IRST[0].conjugate()
print('Potencia de cortocircuito trifásica en la barra 3: {:.3} pu'.
      ↪format(abs(Scc3)))
# Potencia de cortocircuito (trifásica) en la barra 3 calculada mediante las
↪componentes simétricas:
Scc3sim = 3*E*I012[1].conjugate()
print('Potencia de cortocircuito trifásica en la barra 3 (coord. sim.): {:.3}
↪pu'.format(abs(Scc3sim)))

```

Potencia de cortocircuito trifásica en la barra 3: 26.3 pu

Potencia de cortocircuito trifásica en la barra 3 (coord. sim.): 26.3 pu

1.2.8 Cálculo de los equivalentes de Thévenin de las redes de secuencia en el nudo 2

1.2.9 Red de secuencia homopolar/cero (RSH)

```

[9]: Z0 = Zt1[0]
print('Z0 = {:.3}'.format(Z0))

```

Z0 = 0.1j

1.2.10 Red de secuencia directa (RSD)

```

[10]: # El equivalente entre el punto de análisis y tierra es el resultado del
↪paralelo de dos ramas: izquierda (Rama A1) y derecha (Rama B1).
RamaA1 = Zg[1]+Zt1[1]
RamaB1 = +((Zl1[1]*Zl2[1])/(Zl1[1]+Zl2[1]))+Zt2[1]+Zred[1]
Z1 = RamaA1*RamaB1/(RamaA1+RamaB1)
print('Z1 = {:.3}'.format(Z1))

```

Z1 = 0.115j

1.2.11 Red de secuencia inversa (RSI)

```

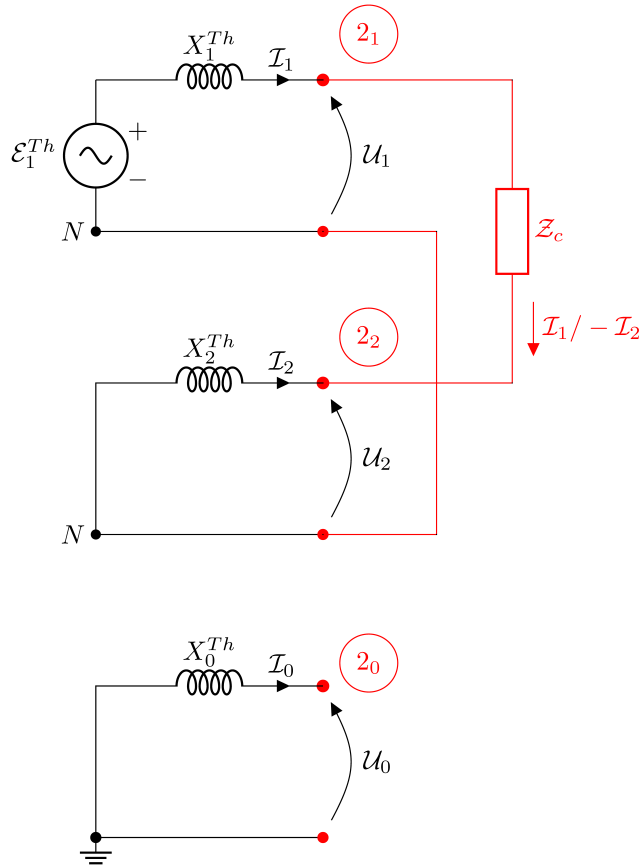
[11]: # Ídem que la RSD.
RamaA2 = Zg[2]+Zt1[2]
RamaB2 = ((Zl1[2]*Zl2[2])/(Zl1[2]+Zl2[2]))+Zt2[2]+Zred[2]
Z2 = RamaA2*RamaB2/(RamaA2+RamaB2)
print('Z2 = {:.3}'.format(Z2))

```

Z2 = 0.115j

1.2.12 Asociación de redes de secuencia (falta bifásica aislada)

Ante falta bifásica aislada en la barra 2, las tres redes de secuencia deben asociarse en paralelo y considerar la impedancia de contacto. Como la falta es aislada de tierra, puede observarse que la RSH queda igualmente aislada (en circuito abierto). Considerando los equivalentes de Thévenin de las tres redes, el esquema queda simplificado:

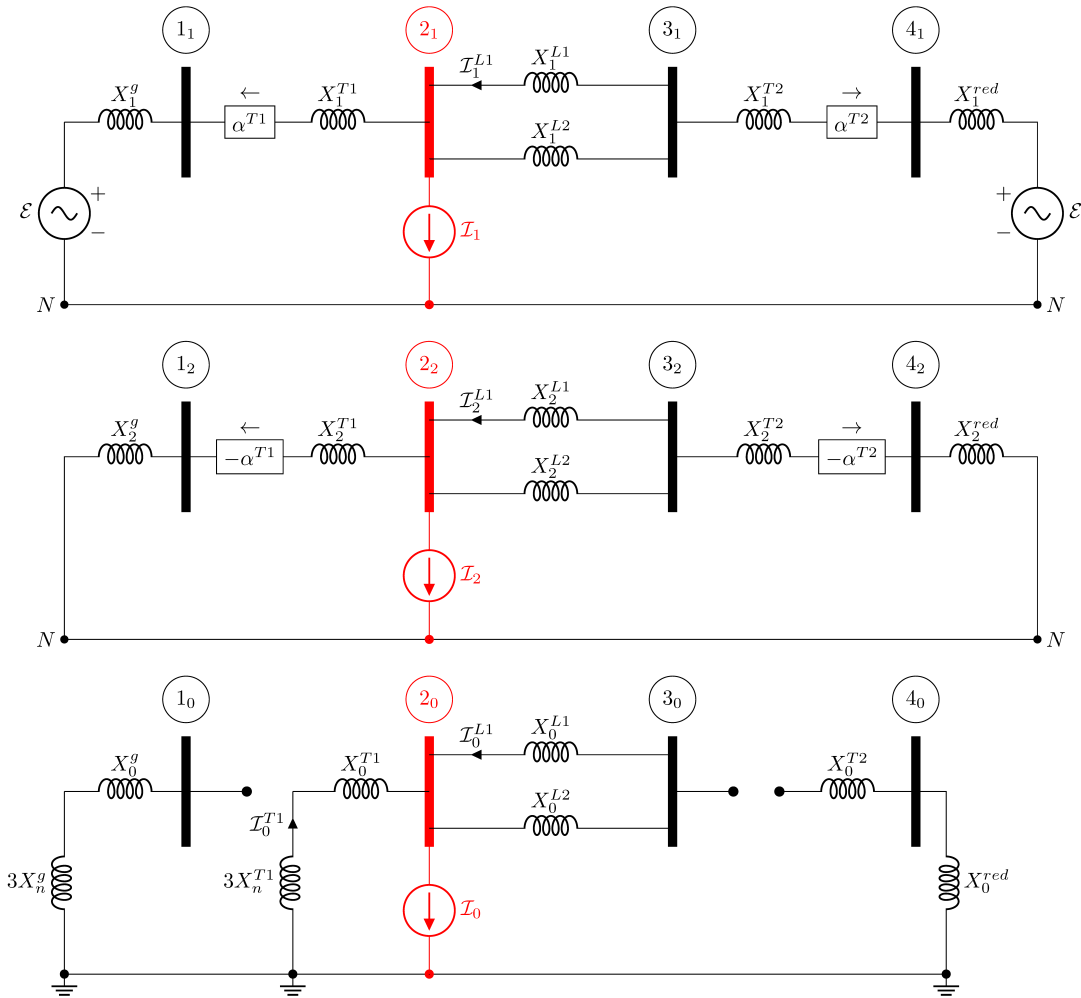


```
[12]: Z012 = np.array([Z0,Z1,Z2])
I0 = 0
I1 = E/(Z012[1]+Z012[2])
I2 = -I1
I012 = np.array([I0,I1,I2])
print('I012 = ',I012)
U0 = 0
U1 = E-I1*Z012[1]
U2 = -I2*Z012[2]
U012 = np.array([U0,U1,U2])
print('U012 = ',U012)
```

```
I012 = [ 0.+0.j      0.-4.363j -0.+4.363j]
U012 = [0. +0.j 0.5+0.j 0.5+0.j]
```

Una vez determinadas las corrientes y tensiones para cada una de las redes de secuencia en el punto de falta (barra 2 en este caso) se debe determinar las componentes simétricas de las variables deseadas (corriente en la línea L1 y en el neutro del transformador T1). Para ello se deberá aplicar el **TEOREMA DE SUSTITUCIÓN** por el cual, en cada una de las redes de secuencia, se sustituirá la falta por una fuente de intensidad (o de tensión si se prefiere) equivalente a la corriente (o tensión) producida por la falta. Una vez sustituida la falta por su equivalente se

operará el circuito hasta despejar la variable deseada y, finalmente, se obtendrán los valores en el dominio de fases mediante el cambio de base usando la Matriz de Fortescue ($[T]$).



1.2.13 Corrientes de fase de la línea L1 ante falta bifásica aislada en la barra 2

```
[13]: # Sustituimos el fallo por las fuentes de intensidad correspondientes y
      ↪ resolvemos el circuito resultante:

# RSD:
# Aplicamos el divisor de intensidad para determinar la contribución de la rama
      ↪ de la dcha. (Rama B1) a la corriente de cortocircuito.
IB1 = I012[1]*RamaA1/(RamaA1+RamaB1)
# Aplicamos nuevamente divisor de intensidad en la barra 2 para la corriente
      ↪ procedente de la rama de la dcha. (Rama B1).
I1L1 = IB1*Z12[1]/(Z11[1]+Z12[1])

# RSI:
# Ídem que RSD: Corriente rama B2.
```

```

IB2 = I012[2]*RamaA2/(RamaA2+RamaB2)
# Ídem que RSD: Corriente L1.
I2L1 = IB2*Z12[2]/(Z11[2]+Z12[2])

# RSH:
# Ídem que RSD: Corriente rama B0.
IB0 = I012[0]*0
# Ídem que RSD: Corriente L1.
I0L1 = IB0*Z12[0]/(Z11[0]+Z12[0])

# Componentes simétricas de la corriente por L1:
I012L1 = np.array([I0L1,I1L1,I2L1])
print('I012L1 [pu] =',I012L1)

# Corriente por L1 en el dominio de fases:
IRSTL1 = np.dot(T,np.transpose(I012L1))
print('IRSTL1 [pu] =',IRSTL1)

```

```

I012L1 [pu] = [0.+0.j    0.-1.394j 0.+1.394j]
IRSTL1 [pu] = [ 0.    +0.j -2.414+0.j  2.414-0.j]

```

1.2.14 Corriente por el neutro del transformador T1 ante falta bifásica aislada en la barra 2

```

[14]: # La corriente que circula por el neutro del transformador T1 sería la
      ↪ componente homopolar de la corriente.
I0nT1 = I012[0]

# Corriente por el neutro de T1 en el dominio de fases:
InT1 = 3*I0nT1
print('InT1 [pu] =',InT1)

```

```

InT1 [pu] = 0j

```