7. Розглядається трьохгалузева модель еколого-економічного балансу (промисловість, сільське господарство, очисні споруди) з наступними параметрами:

$A_{11} = \begin{pmatrix} 0{,}6 & 0{,}1 \\ 0{,}2 & 0{,}1 \end{pmatrix}$, $A_{12} = \begin{pmatrix} 0{,}1 \\ 0{,}3 \end{pmatrix}$, $A_{21} = (0{,}4 \quad 0{,}3)$, $A_{22} = 0{,}3$, $B_1 = \begin{pmatrix} 0{,}7 & 0{,}8 \\ 0{,}1 & 0 \end{pmatrix}$,

$B_2 = \begin{pmatrix} 0{,}1 \\ 0.1 \end{pmatrix}$, $x_1(0) = \begin{pmatrix} 16 \\ 10 \end{pmatrix}$, $c_1(t) = \begin{pmatrix} 8 \\ 3 \end{pmatrix} e^{0{,}02t}$, $c_2 = 0{,}8$.

Графічно дослідити динаміку $x_1(t)$, $x_2(t)$. Також у фазовому просторі $x_1(t)$ зобразити траєкторію з технологічним темпом зростання, траєкторію замкненої системи та загальну траєкторію системи.

Будемо будувати динамічну систему Леонтєва-Форда

Будуємо модель: $x_1(t) = A_1 x_1(t) + Bx_1'(t) + c(t)$, де

$$A_1 = A_{11} + A_{12}(E - A_{22})^{-1} A_{21}$$
$$B = B_1 + B_2(E - A_{22})^{-1} A_{21}$$
$$c(t) = c_1(t) - c_2(A_{12}(E - A_{22})^{-1})^T$$

In [1]:
```
from sympy import *

A11 = Matrix([[0.6, 0.1],
              [0.2, 0.1]])
A12 = Matrix([0.1, 0.3])
A21 = Matrix([[0.4, 0.3]])
A22 = Matrix([0.3])

B1 = Matrix([[0.7, 0.8],
             [0.1, 0]])
B2 = Matrix([0.1, 0.1])

E = Matrix.eye(1)

c1 = lambda t: Matrix([8, 3]) * exp(0.02*t)
c2 = 0.8

A1 = A11 + A12 * (E - A22).inv() * A21
B = B1 + B2 * (E - A22).inv() * A21

c = lambda t: c1(t) - c2 * (A12 * (E - A22).inv())
```

In [2]:
```
A1.n(3)
```

Out[2]:

$$\begin{bmatrix} 0.657 & 0.143 \\ 0.371 & 0.229 \end{bmatrix}$$

In [3]:
```
B.n(3)
```

Out[3]:

$$\begin{bmatrix} 0.757 & 0.843 \\ 0.157 & 0.0429 \end{bmatrix}$$

Маємо диф. рівняння:

$$\frac{dx_1}{dt} = B^{-1}(E - A_1)x_1 - B^{-1}c(t)$$

Щоб знайти $x_1(t)$ знаходимо розв'язок рівняння з початковим значенням $x_1(0)$

Для знаходження x_2 використовуємо:

$$x_2 = (E - A_{22})^{-1} A_{21} x_1 - c_2$$

In [18]:
```
from sympy.solvers.ode.systems import dsolve_system

x10 = Function('x10')
x11 = Function('x11')
t = Symbol('t')
x1 = Matrix([x10(t), x11(t)])


eq = Eq(B.inv()*(Matrix.eye(2) - A1)*x1 - B.inv()*c(t), x1.diff(t))

solution = dsolve_system(
    [eq.lhs[0] - eq.rhs[0], eq.lhs[1] - eq.rhs[1]],
    [x10(t), x11(t)],
    ics={ x10(0): 16, x11(0): 10 }
)

s = lambda x: simplify(simplify(x))
x1 = Matrix([s(solution[0][0].rhs), s(solution[0][1].rhs)])

x1.xreplace({n : round(n, 3) for n in x1.atoms(Number)})
```

Out[18]:

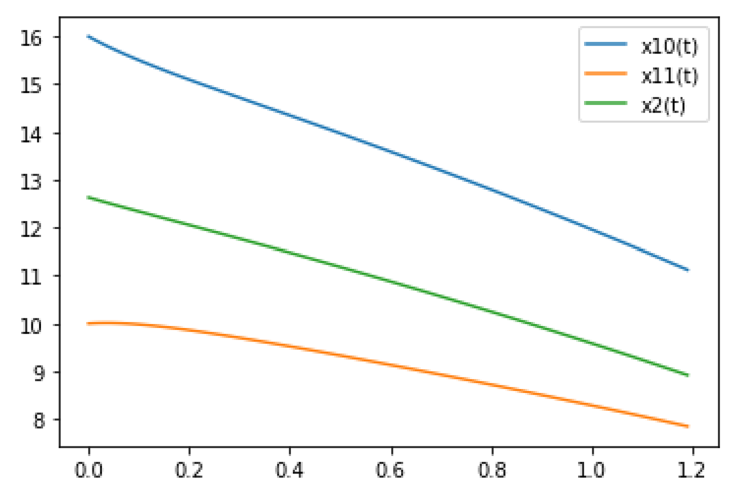
$$\begin{bmatrix} -\frac{18.091}{\left(e^{-0.221t}\right)^{1.0}} - \frac{0.924e^{-0.221t}}{\left(e^{-0.221t}\right)^{1.0}} + \frac{36.002e^{-0.201t}}{\left(e^{-0.221t}\right)^{1.0}} - 1.525e^{0.02t} + 0.276 + 0.263e^{-9.564t} \\ -\frac{9.643}{\left(e^{-0.221t}\right)^{1.0}} - \frac{0.493e^{-0.221t}}{\left(e^{-0.221t}\right)^{1.0}} + \frac{19.191e^{-0.201t}}{\left(e^{-0.221t}\right)^{1.0}} + 1.461e^{0.02t} - 0.264 - 0.252e^{-9.564t} \end{bmatrix}$$

In [23]:
```
x2 = (E - A22).inv() * A21 * x1 - Matrix([c2])
x2[0].xreplace({n : round(n, 3) for n in x2.atoms(Number)})
```

Out[23]:

$$-\frac{14.47}{\left(e^{-0.221t}\right)^{1.0}} - \frac{0.739e^{-0.221t}}{\left(e^{-0.221t}\right)^{1.0}} + \frac{28.797e^{-0.201t}}{\left(e^{-0.221t}\right)^{1.0}} - 0.245e^{0.02t} - 0.756 + 0.042e^{-9.564t}$$

In [15]:
```
import matplotlib.pyplot as plt
import numpy as np
ts = np.arange(0, 1.2, 0.01)
x10_t = [x1[0].subs({t: i}).n(4) for i in ts]
x11_t = [x1[1].subs({t: i}).n(4) for i in ts]
x2_t = [x2[0].subs({t: i}).n(4) for i in ts]
plt.plot(ts, x10_t, label='x10(t)')
plt.plot(ts, x11_t, label='x11(t)')
plt.plot(ts, x2_t, label='x2(t)')
plt.legend()
plt.show()
```



Замкнена динамічна система:

$$\frac{dy_1}{dt} = (E - A_1)B^{-1}y_1$$

In [20]:
```
y10 = Function('y10')
y11 = Function('y11')
y1 = Matrix([y10(t), y11(t)])

eq = Eq((Matrix.eye(2) - A1)*B.inv()*y1, y1.diff(t))

solution = dsolve_system(
    [eq.lhs[0] - eq.rhs[0], eq.lhs[1] - eq.rhs[1]],
    [y10(t), y11(t)],
    ics={ y10(0): 16, y11(0): 10 }
)

y1 = Matrix([s(solution[0][0].rhs), s(solution[0][1].rhs)])
y1.xreplace({n : round(n, 3) for n in y1.atoms(Number)})
```

Out[20]:

$$\begin{bmatrix} \left(19.09e^{9.785t} - 3.09\right) e^{-9.564t} \\ \left(2.848e^{9.785t} + 7.152\right) e^{-9.564t} \end{bmatrix}$$

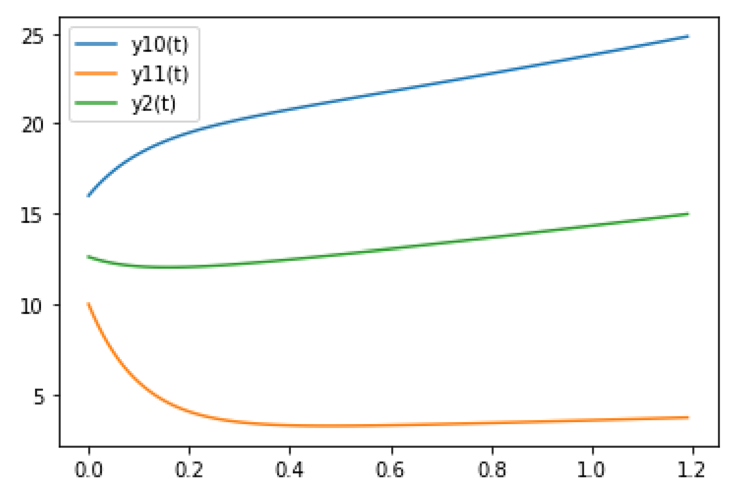
In [22]:
```
y2 = (E - A22).inv() * A21 * y1 - Matrix([c2])
y2[0].xreplace({n : round(n, 3) for n in y2.atoms(Number)})
```

Out[22]:

$$0.429 \left(2.848e^{9.785t} + 7.152\right) e^{-9.564t} + 0.571 \left(19.09e^{9.785t} - 3.09\right) e^{-9.564t} - 0.8$$

In [9]:
```
y10_t = [y1[0].subs({t: i}).n(4) for i in ts]
y11_t = [y1[1].subs({t: i}).n(4) for i in ts]
y2_t = [y2[0].subs({t: i}).n(4) for i in ts]

plt.plot(ts, y10_t, label='y10(t)')
plt.plot(ts, y11_t, label='y11(t)')
plt.plot(ts, y2_t, label='y2(t)')
plt.legend()
plt.show()
```



Коефіцієнт технологічного зростання:

In [10]:
```
eigenvals = ((Matrix.eye(2) - A1).inv() * B).eigenvals()
growth_factor = 1/max(eigenvals)
growth_factor.n(3)
```

Out[10]: 0.221

In [11]:
```
plt.plot(x10_t, x11_t, label='general')
plt.plot(y10_t, y11_t, label='closed')

g = np.exp(float(growth_factor)*ts)
plt.plot(g * 16, g * 10, label='techno')
plt.legend()
plt.show()
```