Київський національний університет
імені Тараса Шевченка
Факультет комп’ютерних наук та кібернетики

Лабораторна робота №2
з курсу
«Управління динамічними системами»
на тему:
«Аналітичне конструювання регуляторів.
Побудова фазових портретів.»

Виконав:
студент групи ІПС-21
факультету комп’ютерних наук та кібернетики
Ляшенко Матвій Олексійович

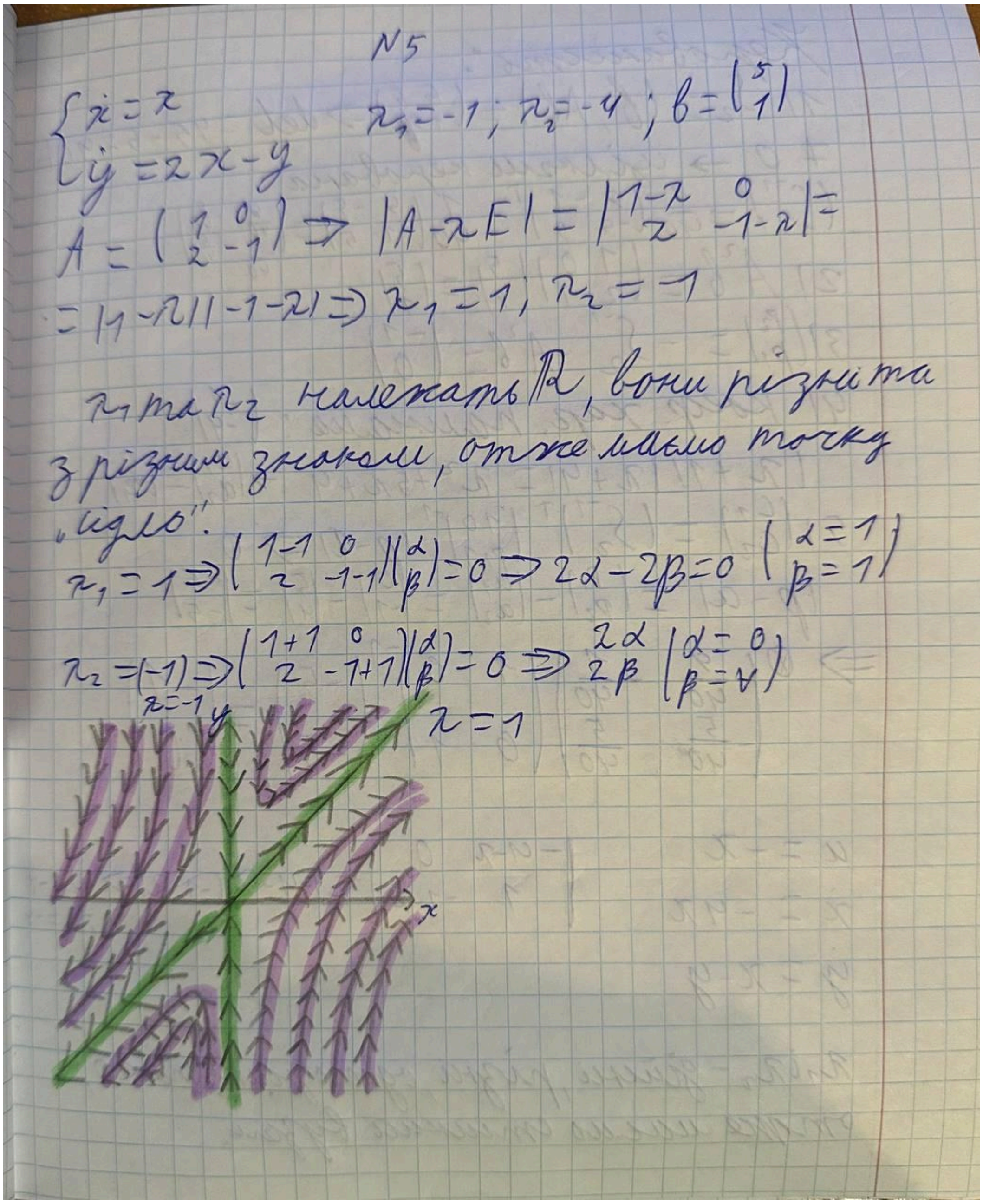
Київ – 2024

Варіант №5

Умова:

- Дослідити на стійкість задану систему. Визначити вигляд точки спокою. Намалювати фазовий портрет. (Все аналітично в зошиті).
- Розв’язати задачу модального керування, обравши одне керування з знайдених можливих. Визначити вигляд отриманої точки спокою. Намалювати фазовий портрет. (Все аналітично в зошиті).
- Зобразити фазові портрети особливих точок розімкненої системи та побудованої замкненої системи за допомогою програмних пакетів (бажано Sage). Траєкторії, сепаратриси, ізокліни (де треба) – різний колір та товщина.

Представлення розв’язку аналітично (в зошиті):



№5

$$\begin{cases} \dot{x} = x \\ \dot{y} = 2x - y \end{cases} \qquad \lambda_1 = -1,\ \lambda_2 = -4\ ;\ b = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 0 \\ 2 & -1 \end{pmatrix} \Rightarrow |A - \lambda E| = \begin{vmatrix} 1-\lambda & 0 \\ 2 & -1-\lambda \end{vmatrix} =$$

$$= |1-\lambda||-1-\lambda| \Rightarrow \lambda_1 = 1,\ \lambda_2 = -1$$

λ_1 та λ_2 належать $\mathbb{R}$, вони різні та з різним знаком, отже маємо точку „сідло".

$$\lambda_1 = 1 \Rightarrow \begin{pmatrix} 1-1 & 0 \\ 2 & -1-1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0 \Rightarrow 2\alpha - 2\beta = 0 \quad \begin{pmatrix} \alpha = 1 \\ \beta = 1 \end{pmatrix}$$

$$\lambda_2 = (-1) \Rightarrow \begin{pmatrix} 1+1 & 0 \\ 2 & -1+1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0 \Rightarrow \begin{matrix} 2\alpha \\ 2\beta \end{matrix} \quad \begin{vmatrix} \alpha = 0 \\ \beta = v \end{vmatrix}$$

Керованість:

1) $S_2 = (b, Ab) = \begin{pmatrix} 5 & 5 \\ 1 & 9 \end{pmatrix} \Rightarrow \det = 45 - 5 = 40 \neq$

$\neq 0 \to$ цілком керована.

$(S_2^{-1}) = \frac{1}{40}\begin{pmatrix} 9 & -1 \\ -5 & 5 \end{pmatrix}^T = \frac{1}{40}\begin{pmatrix} 9 & -5 \\ -1 & 5 \end{pmatrix} = \begin{pmatrix} \frac{9}{40} & -\frac{5}{40} \\ -\frac{1}{40} & \frac{5}{40} \end{pmatrix}$

2) $A^2 b = \begin{pmatrix} 1 & 0 \\ 2 & -1 \end{pmatrix}\begin{pmatrix} 5 \\ 9 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$

3) $\begin{pmatrix} p_2 \\ p_1 \end{pmatrix} = -S_2^{-1} \cdot A^2 b = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

4) коеф. хар. полінома: $\begin{pmatrix} -1 \\ -4 \end{pmatrix}$:

$(\lambda+1)(\lambda+4) = \lambda^2 + 5\lambda + 4 \to \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \end{pmatrix}$

5) $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = (S_2^{-1})^T \begin{pmatrix} 1 & 0 \\ p_1 & 1 \end{pmatrix}^{-1} (p - a) \Rightarrow ⊗$

$(p - a) = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} - \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} - \begin{pmatrix} 5 \\ 4 \end{pmatrix} = \begin{pmatrix} -5 \\ -5 \end{pmatrix}$

$$\Rightarrow ⊗ \begin{pmatrix} \frac{9}{40} & -\frac{1}{40} \\ -\frac{5}{40} & \frac{5}{40} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1}\begin{pmatrix} -5 \\ -5 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

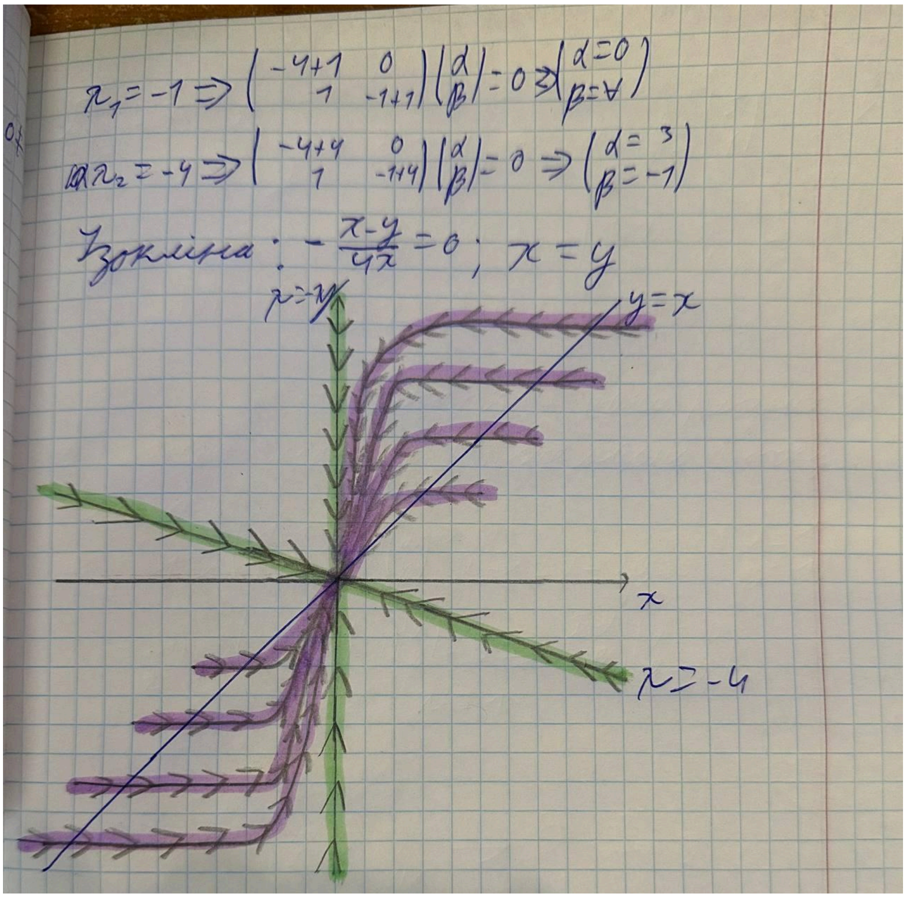
$u = -x$

$\dot{x} = -4x$

$\dot{y} = x - y$

$\begin{vmatrix} -4-\lambda & 0 \\ 1 & -1-\lambda \end{vmatrix} = (-4-\lambda)(-1-\lambda) \Rightarrow \lambda_1 = -1, \ \lambda_2 = -1$

λ_1 і λ_2 – дійсні, різні, одного знаку та < 0, отже маємо стійкий вузол.

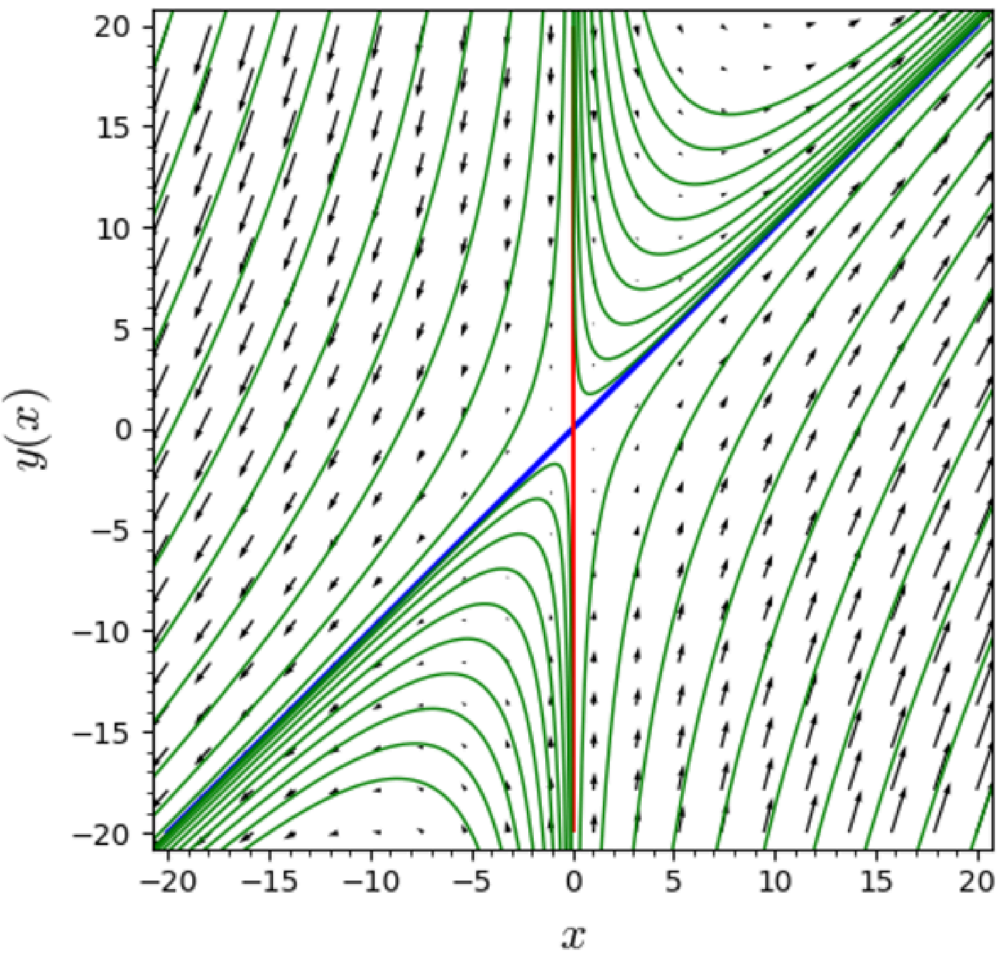


Код програми Sage:

```
# Коефіцієнти системи
m11, m12, m21, m22 = 1, 0, 2, -1  # Коефіцієнти
M = matrix([[m11, m12], [m21, m22]])

# Власні вектори
eigenVectors = M.eigenvectors_right()
v1, v2 = eigenVectors[0][1][0], eigenVectors[1][1][0]

# Змінні
x, y, t = var('x y t')
x1, y1 = function('x1')(t), function('y1')(t)

# Сепаратриси
if v1[0] != 0:
    separ1 = solve([v1[0] * y == v1[1] * x], y)[0].right()
else:
    separ1 = (x == 0)

if v2[0] != 0:
    separ2 = solve([v2[0] * y == v2[1] * x], y)[0].right()
else:
    separ2 = (x == 0)
```

```
# Система диференціальних рівнянь
dx = diff(x1, t) == m11 * x1 + m12 * y1
dy = diff(y1, t) == m21 * x1 + m22 * y1

# Діапазон графіку
plotRange = 20

# Поле напрямків
plt = plot_vector_field(
    [m11 * x + m12 * y, m21 * x + m22 * y],
    [x, -plotRange, plotRange],
    [y, -plotRange, plotRange],
    axes_labels=['$x$', '$y(x)$']
)

# Сепаратриси
if v1[0] != 0:
    plt += plot(separ1, (-plotRange, plotRange), color="blue", thickness=2)
else:
    plt += implicit_plot(separ1, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color="blue")

if v2[0] != 0:
    plt += plot(separ2, (-plotRange, plotRange), color="red", thickness=2)
else:
    plt += implicit_plot(separ2, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color="red")

# Центральні розв’язки задачі Коші
for i in range(-10, 10):
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics=[0, i / 2, i * 2])
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color="green")

# Додаткові лінії з боків
# Рівномірно розподіляємо початкові умови
additional_lines = 20  # Кількість додаткових ліній
step = plotRange / (additional_lines // 2)

for i in range(-plotRange, plotRange + 1, int(step)):
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics=[0, i, -i])
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color="green")

# Показати графік
plt.show(xmin=-plotRange, xmax=plotRange, ymin=-plotRange, ymax=plotRange)
```

Результат роботи програми Sage:



Малюнок №2

Код програми Sage:

```
# Коефіцієнти системи
m11, m12, m21, m22 = -4, 0, 1, -1  # Задайте потрібні коефіцієнти
M = matrix([[m11, m12], [m21, m22]])

# Власні вектори
eigenVectors = M.eigenvectors_right()
v1, v2 = eigenVectors[0][1][0], eigenVectors[1][1][0]

# Змінні
x, y, t = var('x y t')
x1, y1 = function('x1')(t), function('y1')(t)

# Сепаратриси
if v1[0] != 0:
   separ1 = solve([v1[0] * y == v1[1] * x], y)[0].right()
else:
   separ1 = (x == 0)

if v2[0] != 0:
   separ2 = solve([v2[0] * y == v2[1] * x], y)[0].right()
```

```
else:
    separ2 = (x == 0)

# Система диференціальних рівнянь
dx = diff(x1, t) == m11 * x1 + m12 * y1
dy = diff(y1, t) == m21 * x1 + m22 * y1

# Ізокліна
isocline = solve([(m21 * x + m22 * y) / (m11 * x + m12 * y)], y)[0].right()

# Діапазон графіку
plotRange = 20

# Поле напрямків
plt = plot_vector_field(
    [m11 * x + m12 * y, m21 * x + m22 * y],
    [x, -plotRange, plotRange],
    [y, -plotRange, plotRange],
    axes_labels=['$x$', '$y(x)$'], axes=True  # Додано осі
)

# Сепаратриси
if v1[0] != 0:
    plt += plot(separ1, (-plotRange, plotRange), color="blue", thickness=2)
else:
    plt += implicit_plot(separ1, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color="blue")

if v2[0] != 0:
    plt += plot(separ2, (-plotRange, plotRange), color="red", thickness=2)
else:
    plt += implicit_plot(separ2, (x, -plotRange, plotRange), (y, -plotRange, plotRange),
color="red")

# Ізокліна (чорна лінія продовжена до країв)
plt += plot(isocline, (-plotRange, plotRange), color="black", thickness=3)

# Лінії розв’язків задачі Коші
# Центральні лінії
for i in range(-10, 10):
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics=[0, i / 2, i * 2])
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color="green")

# Додаткові лінії по боках
for i in range(-plotRange, plotRange + 1, 2):
    x_t, y_t = desolve_system([dx, dy], [x1, y1], ics=[0, i, -i])
    plt += parametric_plot((x_t.rhs(), y_t.rhs()), (t, -plotRange, plotRange), color="green")
```

```
# Додано осі x і y як чорні лінії
plt += line([(-plotRange, 0), (plotRange, 0)], color="black", thickness=1)  # Вісь x
plt += line([(0, -plotRange), (0, plotRange)], color="black", thickness=1)  # Вісь y

# Показати графік
plt.show(xmin=-plotRange, xmax=plotRange, ymin=-plotRange, ymax=plotRange)
```

Результат роботи програми Sage: