

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

Лабораторна робота №2(в. 9)
із курсу «Моделювання систем»

Виконала
студентка 3 курсу
групи ІПС-33
Постернак Таїсія

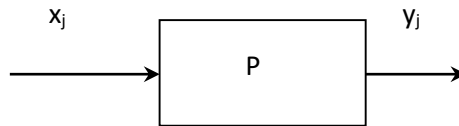
Київ, 2023

Тема.

Побудова лінійної моделі з допомогою псевдообернених операторів.

Постановка задачі.

Будемо вважати, що на вхід системи перетворення, математична модель якої невідома, поступають послідовно дані у вигляді $m-1$ вимірних векторів \mathbf{x}_j . На виході системи спостерігається сигнал у вигляді вектора \mathbf{y}_j розмірності P .



Для послідовності вхідних сигналів \mathbf{x}_j , $j=1,2,\dots,n$ та вихідних сигналів \mathbf{y}_j , $j=1,2,\dots,n$ знайти оператор P перетворення вхідного сигналу у вихідний.

Теорія.

Будемо шукати математичну модель оператора об'єкту в класі лінійних

операторів
$$\mathbf{A} \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix} = \mathbf{y}_j, \quad j=1,2,\dots,n.$$
 (1)

Невідома матриця \mathbf{A} математичної моделі об'єкту розмірності $p \times n$.

Систему (1) запишемо у матричній формі

$$\mathbf{A} \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ 1 & 1 & \dots & 1 \end{pmatrix} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n),$$

або $\mathbf{AX} = \mathbf{Y}$, де $\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ 1 & 1 & \dots & 1 \end{pmatrix}$ – матриця вхідних сигналів розмірності $m \times n$, $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$ – матриця вихідних сигналів розмірності $p \times n$.

Матрицю \mathbf{X} будемо інтерпретувати як двовимірне вхідне зображення, а матрицю \mathbf{Y} вихідне зображення.

Тоді $\mathbf{A} = \mathbf{YX}^+ + \mathbf{VZ}^T(\mathbf{X}^T)$, де матриця $\mathbf{V} = \begin{pmatrix} \mathbf{v}_{(1)}^T \\ \mathbf{v}_{(2)}^T \\ \vdots \\ \mathbf{v}_{(p)}^T \end{pmatrix}$, розмірності $p \times m$,
 $\mathbf{Z}(\mathbf{X}^T) = \mathbf{I}_m - \mathbf{XX}^+$.

Якщо для матриці \mathbf{A} відома псевдообернена (обернена) матриця \mathbf{A}^+ ,

то для розширеної матриці $\begin{pmatrix} \mathbf{A} \\ \mathbf{a}^T \end{pmatrix}$ справедлива формула Гревіля для псевдообернення матриці:

$$\begin{pmatrix} A \\ a^T \end{pmatrix}^+ = \begin{cases} \left(A^+ - \frac{Z(A)aa^T A^+}{a^T Z(A)a} : \frac{Z(A)a}{a^T Z(A)a} \right), & \text{if } a^T Z(A)a > 0 \\ \left(A^+ - \frac{R(A)aa^T A^+}{1 + a^T R(A)a} : \frac{R(A)a}{1 + a^T R(A)a} \right), & \text{if } a^T Z(A)a = 0 \end{cases},$$

де $Z(A) = E - A^+ A$, $R(A) = A^+ (A^+)^T$.

Для першого кроку алгоритму $(a_1^T)^+ = \frac{a_1}{a_1^T a_1}$, де $A = \begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{pmatrix}$.

Формула Мура-Пенроуза для знаходження оберненої (псевдооберненої) матриці:

$$A^+ = \lim_{\delta^2 \rightarrow 0} \left\{ (A^T A + \delta^2 E_n)^{-1} A^T \right\} = \lim_{\delta^2 \rightarrow 0} \left\{ A^T (A A^T + \delta^2 E_m)^{-1} \right\}.$$

Матриця A розмірності $m \times n$.

Хід роботи.

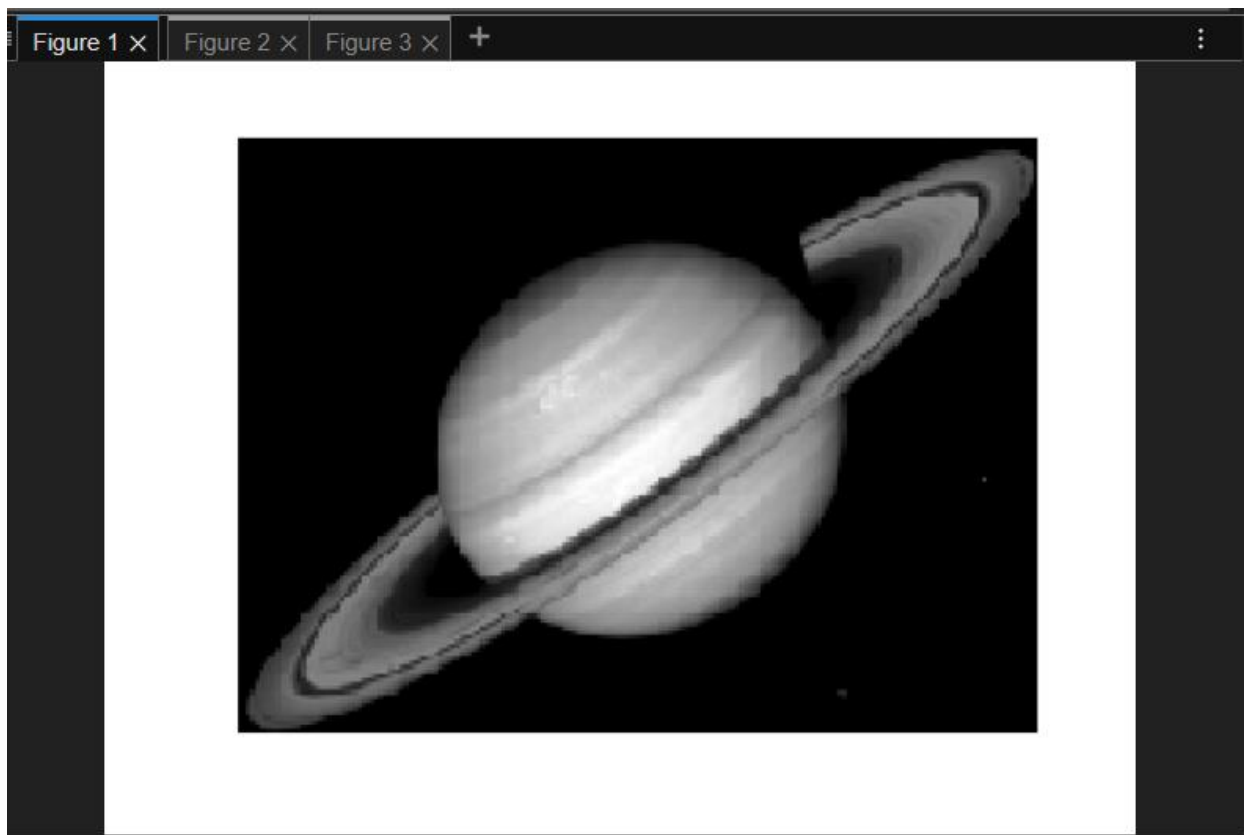
Зчитуємо дані та застосовуємо реалізовану формулу Гревіля. Далі перевіряємо правильність знаходження псевдооберненої матриці за допомогою теореми:

Матриця A^+ розмірності $n \times m$ є псевдооберненою матрицею до матриці A розмірності $m \times n$ тоді і тільки тоді, якщо виконуються такі умови:

- $AA^+A = A$;
- $A^+AA^+ = A^+$;
- AA^+ – симетрична матриця розмірності $m \times m$;
- A^+A – симетрична матриця розмірності $n \times n$.

Опісля будуємо лінійний оператор для бажаного перетворення із отриманої псевдооберненої матриці.

Вхідне зображення:



Вихідне зображення:



Отримане зображення:



Код програми (MATLAB).

```
clear

A = imread("x1.bmp");
B = imread("y9.bmp");
imshow(A)
figure
imshow(B)

A = double(A);
B = double(B);

pseudoinverseG = greville(A);
check(pseudoinverseG,A);

operatorG = B * pseudoinverseG;
resG = uint8(operatorG * A);

figure
imshow(resG);

disp(pseudoinverseG);
resMult = A*pseudoinverseG;
disp(resMult);
disp("Error: " + norm(resMult - eye(size(resMult,1))));

function result = greville(A)
    curr_a = A(1,:);
    curr = zeros(size(curr_a));
    curr = transpose(curr);
    if (nnz(curr_a(1) ~= 0))
        curr = transpose(curr_a) / (curr_a * transpose(curr_a));
    end
    n = size(A,1);

    for i = 2:n
        curr_a = A(i,:);
        z = eye(size(curr,1))-(curr*A(1:i-1,:));
        r = curr * transpose(curr);
        condition = (curr_a * z) * transpose(curr_a);
        if (nnz(condition) == 1)
            row_to_add = (z * transpose(curr_a)) / condition;
        else
            row_to_add = (r * transpose(curr_a)) / (1 + ((curr_a * r) *
transpose(curr_a)));
        end
        curr = curr -( row_to_add * (curr_a * curr));
        curr = cat(2,curr, row_to_add);
    end
    result = curr;
end

function check(Aplus,A)
    assert(isequal(round(Aplus,4),round(Aplus * A * Aplus,4)));
```

```
    assert(isequal(round(A,4),round(A*Aplus*A,4)));
    assert(issymmetric(round(A*Aplus,4)));
    assert(issymmetric(round(Aplus*A,4)));
end

function n = norm(A)
    n = sum(sum(abs(A)));
end
```