

Модель системи у змінних стану

Процеси, що протікають у динамічному об'єкті, описуються диференціальними рівняннями наступного виду:

$$\sum_{i=1}^n a_i y^{(i)}(t) = \sum_{j=1}^m b_j u^{(j)}(t),$$

де a_i и b_j — постійні коефіцієнти, що зветься параметрами моделі;
 $u^{(j)}(t)$ — похідні вхідного сигналу $u(t)$;
 $y^{(i)}(t)$ — похідні вихідного сигналу $y(t)$;
 n — порядок моделі.

Введенням нових змінних, наприклад,

$$x_1(t) = y(t), x_2(t) = y'(t), x_3(t) = y''(t), \dots$$

диференціальне рівняння можна представити у нормальній формі Коші:

$$\dot{x}_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_{11}u_1 + \dots + b_{1m}u_m,$$

$$\dot{x}_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + b_{21}u_1 + \dots + b_{2m}u_m,$$

⋮

$$\dot{x}_n = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_{n1}u_1 + \dots + b_{nm}u_m,$$

Наведені рівняння зручно представляти у матричній формі:

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1m} \\ \vdots & & \vdots \\ b_{n1} & \dots & b_{nm} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}.$$

Компактно будемо представляти цей вираз так:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$

Це рівняння називають **рівнянням стану** динамічної системи.
Матриця-стовпець, що складається зі змінних стану, називається вектором стану і має вигляд:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Матриця **A** є квадратною розмірності $n \times n$, а матриця **B** має розмірність $n \times m$. Рівняння стану зв'язує швидкість зміни стану системи з самим станом і вхідними сигналами.

Аналогічно можна записати зв'язок вихідних сигналів лінійної системи з змінними стану і вхідними сигналами за допомогою **рівняння виходу:**

$$y = Cx + Du$$

Рівняння стану і рівняння виходу разом складають модель динамічної системи у просторі станів.

Описаним методом систему, що містить похідні вищого порядку, можна переформулювати як сукупність систем першого порядку шляхом введення додаткових невідомих – змінних стану системи. Вибір змінних стану може бути довільним, тому така операція не є однозначною. І нормальних форм Коші, в які перетворюється диференціальне рівняння, може бути безліч.

Оскільки вибір змінних стану неоднозначний, одній і тій ж передатній функції можуть відповідати різні моделі в просторі станів, але при зворотному переході всім цим моделям відповідає одна і та ж передатна функція .

Іноді ПФ називають зовнішньою моделлю системи, а уявлення в просторі станів - внутрішньою моделлю, оскільки ПФ враховує тільки входи і виходи динамічної системи, тоді як для побудови моделі у просторі станів можна обирати такі змінні стану, які можна реально спестерігати в системі.

Правильний вибір змінних стану динамічної системи визначає ефективність її математичного опису. Це набір змінних, які повністю описують стан системи в будь-який момент часу, дозволяючи передбачити її майбутнє поведінку. Важливо вибирати незалежні, фізично інтерпретовані змінні, що мінімізують розмір системи (наприклад, положення та швидкість у механічних системах). Неправильний вибір може призвести до надлишкових обчислень або невідповідності моделі реальності.

З огляду на те, що є безліч еквівалентних (з точки зору відносин входу-виходу) способів подання рівнянь стану системи, можна вибрати з них "найкращі" – найбільш зручні для використання в розглянутій задачі. Такі зручні для дослідника форми запису рівнянь називаються **канонічними**.

Прикладом є **канонічна форма керованості** динамічного об'єкта управління, яка описується рівнянням $dx(t)/dt = \mathbf{A}x(t) + \mathbf{B}u(t)$.

Матриці **A** і **B** мають вигляд:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},$$

де a_i - коефіцієнти характеристичного рівняння системи

$$s^n + a_1 s^{n-1} + \dots + a_n = 0,$$

яке в матричній формі записується так:

$$\det(\mathbf{I}s - \mathbf{A}) = 0.$$

Деякі визначення з матричної теорії

Розглянемо матрицю розміром $n \times m$:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}.$$

Задамося деяким числом k ($k \leq n$, $k \leq m$) і виберемо з матриці \mathbf{A} будь-які k рядків і k стовпців. Одержимо квадратну матрицю порядку k . Визначник такої матриці називається називається **мінором** k -го порядку, наприклад

$$M_k = \det \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix}.$$

Взагалі для матриці розміром $n \times m$ існує наступна кількість мінорів k -го порядку:

$$C_n^k \cdot C_m^k.$$

Тут C_n^k – число комбінацій із n елементів по k , що обчислюється за формулою:

$$C_n^k = \frac{n!}{k!(n-k)!}.$$

Рангом матриці називається найвищий з порядків відмінного від нуля мінорів цієї матриці. Щоб отримати значення рангу, треба знайти всі мінори матриці, відкинути ті, що дорівнюють нулю, взяти найбільший порядок з тих, що залишились.

Є також інший метод. Треба викинути з матриці рядки (або стовпці), які є пропорційними один одному – лінійно залежними. Кількість лінійно незалежних рядків, що залишились, є рангом матриці. Пошук рангу матриці великого розміру є складною трудомісткою задачею. Для її розв'язання в Scilab введена функція **rank**. Операція визначення рангу використовується для оцінки керованості систем управління.

Власний вектор (англ. eigenvector) квадратної матриці **A** з власним значенням (англ. eigenvalue) λ — це ненульовий вектор \mathbf{v} , для якого виконується співвідношення $\mathbf{A} \cdot \mathbf{v} = \lambda \cdot \mathbf{v}$, де λ це певний скаляр, тобто дійсне або комплексне число.

Власні вектори матриці **A** — це ненульові вектори, які під дією лінійного перетворення, що задається матрицею **A**, не міняють напрямку, але можуть змінювати свою довжину на коефіцієнт λ .

Провідну роль у розумінні власних значень матриць відіграє характеристичний поліном матриці. Власні значення матриці **A** розмірністю $n \times n$ є (і тільки вони є) коренями характеристичного полінома матриці **A**:

$$p_A(\lambda) = \det(A - I_n \lambda),$$

де I_n — одинична матриця порядку n .

$$I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

У Scilab з матрицями можливі наступні операції

Одинична матриця порядку $m \times n$ створюється командою `eye(m,n)`.

Визначник матриці **A** розраховується як `det(A)`.

Пошук рангу матриці: `rank(A)`.

Власні значення та власні вектори матриці **A** знаходяться з використанням функції **spec**. Синтаксис функції:

`evals = spec(A)`

`[R, diagevals] = spec(A)`

Тут `evals` – вектор власних значень матриці; `R` – матриця власних векторів; `diagevals` – матриця власних значень, розташованих по діагоналі.

Повернемось до представлення динамічної системи у канонічній формі керованності. Розглянемо модель системи у просторі станів:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}u$$

Для переведення у канонічну форму треба перетворити матрицю \mathbf{A} в матрицю \mathbf{A}_{ccf} , матрицю \mathbf{B} в матрицю \mathbf{B}_{ccf} , матрицю \mathbf{C} в матрицю \mathbf{C}_{ccf} . Для переважної більшості реальних систем матриця $\mathbf{D} = 0$.

Матриці \mathbf{A}_{ccf} і \mathbf{B}_{ccf} мають вигляд:

$$A_{\text{ccf}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}, \quad B_{\text{ccf}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Матриця \mathbf{C}_{ccf} створюється з коефіцієнтів b_i чисельника передатної функції системи:

$$W(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}.$$

Керованість та спостережуваність систем управління

Нехай існує система n -го порядку (з n компонентами вектора стану), m входами і l виходами:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}; \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \end{cases}$$

де \mathbf{x} – n -мірний вектор змінних стану:

\mathbf{y} – l -мірний вектор вихідних (регульованих) змінних

\mathbf{u} – m -мірний вектор дій, що управляють:

\mathbf{A} – матриця, що визначає динамічні властивості об'єкта, розмірності $n \times n$;

\mathbf{B} – матриця управління, розмірності $n \times m$;

\mathbf{C} – матриця виходу розмірності $l \times n$;

\mathbf{D} – матриця обходу розмірності $l \times m$.

Перше рівняння часто називають **рівнянням управління**, друге – **рівнянням спостереження**.

Система називається **повністю керованою**, якщо її можна перевести з будь-якого початкового стану $\mathbf{x}(0)$ в будь-який кінцевий стан $\mathbf{x}(t)$ за допомогою управління $\mathbf{u}(t)$ за кінцевий час.

Керованість системи визначається матрицею управління \mathbf{B} . Системи, в яких матриця $\mathbf{B} = 0$, є очевидно некерованими, оскільки вхід $u(t)$ не має ніякого впливу на вихід $y(t)$.

З лінійної системи $d\mathbf{x}/dt = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ за допомогою операцій диференціювання, при незмінних входах $\mathbf{u} = \text{const}$, можна одержати ціле сімейство еквівалентних систем:

$$d\mathbf{x}/dt = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u};$$

$$d^2\mathbf{x}/dt^2 = \mathbf{A}d\mathbf{x}/dt = \mathbf{A}^2\mathbf{x} + \mathbf{A}\mathbf{B}\mathbf{u};$$

$$d^3\mathbf{x}/dt^3 = \mathbf{A}^2d\mathbf{x}/dt = \mathbf{A}^3\mathbf{x} + \mathbf{A}^2\mathbf{B}\mathbf{u};$$

.....

$$d^n\mathbf{x}/dt^n = \mathbf{A}^{n-1}d\mathbf{x}/dt = \mathbf{A}^n\mathbf{x} + \mathbf{A}^{(n-1)}\mathbf{B}\mathbf{u}.$$

Об'єднаємо ці вирази в єдине матричне рівняння:

$$\begin{bmatrix} dx/dt \\ d^2x/dt^2 \\ d^3x/dt^3 \\ \dots \\ d^n x/dt^n \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \mathbf{A}^3 \\ \dots \\ \mathbf{A}^n \end{bmatrix} \cdot \mathbf{x} + \begin{bmatrix} \mathbf{B} \\ \mathbf{AB} \\ \mathbf{A}^2\mathbf{B} \\ \dots \\ \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \cdot \mathbf{u}.$$

Таким чином, маємо сімейство з n систем, яке можна розглядати як єдину систему з матрицею управління

$$\mathbf{Q} = [\mathbf{B}; \mathbf{AB}; \mathbf{A}^2\mathbf{B}; \dots; \mathbf{A}^{n-1}\mathbf{B}].$$

Цю матрицю називають **матрицею керованості** динамічної системи.

Згідно критерію керованості Калмана, система є повністю керованою, якщо ранг матриці керованості є рівним порядку системи n .

То ж система повністю керована тоді і тільки тоді, коли ранг матриці керованості дорівнює порядку системи: $\text{rank}(\mathbf{Q})=n$. Якщо $\text{rank}(\mathbf{Q})=0$, то система є повністю некерованою.

Можлива ситуація, коли ранг матриці керованості більше нуля, але менше порядку системи. Така система є **частково керованою**.

В пакеті Scilab матриця керованості може бути побудована за допомогою функції `cont_mat`, яка викликається однією з команд:

```
--> Q = cont_mat(A, B)
--> Q = cont_mat(sys)
--> Q = cont_mat(sys.A, sys.B)
```

Перевірити керованість системи можна командою

```
--> c = rank(cont_mat(A, B))
```

Якщо отримали результат $c = n$, система є керованою.

Тепер розглянемо друге рівняння стану (рівняння спостережуваності):

$$y = Cx + Du.$$

Система називається **спостережуваною**, якщо на кінцевому інтервалі часу, в кінці цього інтервалу, по виходу системи $y(t)$ при відомій керуючій дії $u(t)$ можна визначити всі початкові компоненти вектора стану $x(t)$.

Без втрати узагальненості при аналізі спостережуваності можна розглядати тільки таке рівняння: $y=Cx$, оскільки значення входу u завжди відоме, і ми можемо просто відняти з результатів вимірювань видного сигналу y добуток Du .

Як і в разі керованості, значення вектора x можна визначити не тільки за значенням виходу y , але і за швидкістю зміни y , тобто враховуючи похідні виходу. Вважаючи, що $u = \text{const}$ і продиференціювавши вихід за часом n разів, отримаємо:

$$d\mathbf{y} / dt = \mathbf{C}d\mathbf{x} / dt = \mathbf{C}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}) = \mathbf{C}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{B}\mathbf{u};$$

$$d^2\mathbf{y} / dt^2 = \mathbf{C}\mathbf{A}d\mathbf{x} / dt = \mathbf{C}\mathbf{A}^2\mathbf{x} + \mathbf{C}\mathbf{A}\mathbf{B}\mathbf{u};$$

$$d^3\mathbf{y} / dt^3 = \mathbf{C}\mathbf{A}^2d\mathbf{x} / dt = \mathbf{C}\mathbf{A}^3\mathbf{x} + \mathbf{C}\mathbf{A}^2\mathbf{B}\mathbf{u};$$

.....

$$d^n\mathbf{y} / dt^n = \mathbf{C}\mathbf{A}^{n-1}d\mathbf{x} / dt = \mathbf{C}\mathbf{A}^n\mathbf{x} + \mathbf{C}\mathbf{A}^{(n-1)}\mathbf{B}\mathbf{u}.$$

Об'єднаємо ці вирази в єдине матричне рівняння:

$$\begin{bmatrix} d\mathbf{y} / dt \\ d^2\mathbf{y} / dt^2 \\ d^3\mathbf{y} / dt^3 \\ \dots \\ d^n\mathbf{y} / dt^n \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \dots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \cdot d\mathbf{x} / dt.$$

Матриця

$$\mathbf{H} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \dots \\ \mathbf{CA}^{n-1} \end{bmatrix}$$

називається **матрицею спостережуваності**. Відповідно до критерію спостережуваності, якщо ранг матриці спостережуваності є рівний порядку n , система є спостережуваною.

Як і для керованості, в середовищі Scilab існує ще одна вбудована функція **obsv_mat(A,C)**, яка дозволяє розрахувати матрицю спостережуваності системи за допомогою матриць A та C.

Вця функція може викликатися в одному з варіантів :

- > `N = obsv_mat (A, C)`
- > `N = obsv_mat (sys)`
- > `N = obsv_mat (sys.A, sys.C)`

Спостережуваність системи перевіряється командою:

- > `rank (obsv_mat (A, C))`

Якщо отримаємо результат, рівний порядку системи, вона є повністю спостережуваною

Переведення диференціальних рівнянь у «Канонічну форму керованості (КФУ)» радикально полегшує аналіз та синтез систем управління.

1. Прямий зв'язок із передавальною функцією

КФУ дозволяє миттєво перейти від диференціального рівняння (або передавальної функції) до матричного вигляду. Коефіцієнти знаменника передавальної функції просто "розкладаються" в останньому рядку матриці стану A .

Якщо система задана рівнянням:

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1\dot{y} + a_0y = bu$$

то в КФУ матриця $A_{\text{сф}}$ матиме вигляд супутньої матриці, де нижній рядок - це коефіцієнти $-a_0, -a_1, \dots, -a_{n-1}$. Це робить структуру системи прозорою.

Матриці КФУ будуються з поліномів чисельника і знаменника передатної функції.

$$W(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}.$$

З коефіцієнтів полинома знаменника:

$$A_{\text{ccf}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}$$

Матриця B_{ccf} завжди однакова:

$$B_{\text{ccf}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Коефіцієнти чисельника передатної функції безпосередньо стають елементами матриці C_{ccf} :

$$C_{ccf} = [b_0, b_1, b_2, \dots]$$

2. Спрощення синтезу регуляторів

Це головна практична причина. Коли ми проектуємо закон управління за станом $u = -Kx$, нам потрібно знайти коефіцієнти посилення вектора K .

У загальному вигляді це вимагає вирішення складних систем рівнянь.

У КФУ євдання стає тривіальним. Оскільки структура матриць фіксована, ми можемо безпосередньо зіставити коефіцієнти бажаного характеристичного рівняння (де хочемо бачити наші полюса) з коефіцієнтами матриці замкнутої системи.

3. Гарантія керованості

Якщо систему вдалося привести до КФУ за допомогою невиродженого перетворення, це є автоматичним доказом того, що система є повністю керована.

У Scilab для отримання поліному знаменника передатної функції використовується характеристичне рівняння

$$\det(\mathbf{I}s - \mathbf{A}) = 0.$$

Тож треба виконати команду типу

```
-->den = det (s*eye (2, 2) - A) ;
```

Поліном чисельника треба вилучити з виразу передатної функції, що робиться за допомогою матричної формули

$$\text{Num}(W) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

Матриця \mathbf{M}^{-1} називається зворотною до матриці \mathbf{M} , якщо виконується така рівність:

$$\mathbf{M}^{-1} \mathbf{M} = \mathbf{I}.$$

Для матриць розміру 2x2 можна використати таку формулу для отримання поліному чисельника передатної функції:

$$\text{Num}(W) = \mathbf{C} \cdot \text{adj}(s\mathbf{I} - \mathbf{A}) \cdot \mathbf{B}$$

Тут $\text{adj}(s\mathbf{I} - \mathbf{A})$ – так звана приєднана матриця. Її отримаємо, якщо в матриці $(s\mathbf{I} - \mathbf{A})$ поміняємо місцями елементи головної та знаки побочної діагоналей.

Наприклад, маємо систему з матрицями:

$$A = \begin{bmatrix} -10 & 12 \\ -8 & 10 \end{bmatrix}, B = \begin{bmatrix} -9 \\ -6 \end{bmatrix}, C = [-1 \quad 2]$$

Тоді матриця $\mathbf{I}s - \mathbf{A}$:

$$\rightarrow M = s * \text{eye}(2,2) - A$$

$M = [2 \times 2 \text{ polynomial}] \text{ of } s$

$$10 + s \quad -12$$

$$8 \quad -10 + s$$

Далі треба перетасувати елементи цієї матриці. Але приєднану матрицю можна безпосередньо створити і з елементів матриці \mathbf{A} :

$$\rightarrow \text{adj_matrix} = [s - A(2,2), A(1,2); A(2,1), s - A(1,1)]$$

$\text{adj_matrix} = [2 \times 2 \text{ polynomial}] \text{ of } s$

$$-10 + s \quad 12$$

$$-8 \quad 10 + s$$

Чисельник передатної функції

$$\rightarrow \text{num_poly} = C * \text{adj_matrix} * B$$

$\text{num_poly} = [\text{polynomial}] \text{ of } s$

$$6 - 3s$$

Задача 4.3-1. Для записаних нижче моделей побудувати канонічні моделі з діагональними матрицями при змінних стану.

Для варіанта 20:

$$dx_1/dt = -10x_1 + 12x_2 - 9u;$$

$$dx_2/dt = -8x_1 + 10x_2 - 6u;$$

$$y = -x_1 + 2x_2 .$$

У редакторі сценаріїв Scilab вводимо наступні команди.

```
// 1. Задаємо початкові матриці системи
```

```
A = [-10, 12; -8, 10];
```

```
B = [-9; -6];
```

```
C = [-1, 2];
```

```
disp("--- Початкова система ---");
```

```
disp("A =", A, "B =", B, "C =", C);
```

```
// 2. Перевіряємо керованість
Q = cont_mat(A, B); // Вбудована функція матриці керованості
disp("--- Аналіз керованості ---");
disp("Матриця керованості Q =", Q);
disp("Ранг матриці керованості : " + string(rank(Q)));
if rank(Q) < 2 then
    disp(" УВАГА: Система некерована! Стандартний перехід
неможливий.");
    disp(" Будуємо канонічну форму на основі коефіцієнтів
поліномів.");
end
// 3. Знаходимо характеристичний поліном (знаменник)
s = poly(0, 's'); // Задаємо символічну змінну 's'
char_poly = det(s*eye(2,2) - A);
disp("--- Поліноми системи ---");
disp("Характеристичний поліном (знаменник):", char_poly);
```

```
// Вилучаємо коефіцієнти знаменника [a0, a1]
a = coeff(char_poly);
a0 = a(1);
a1 = a(2);
// 4. Знаходимо чисельник передатної функції
// Використовуємо приєднану матрицю adj(sI - A) для 2x2
adj_matrix = [s - A(2,2), A(1,2); A(2,1), s - A(1,1)];
num_poly = C * adj_matrix * B;
disp("Чисельник передатної функції :", num_poly);
// Вилучаємо коефіцієнти чисельника [b0, b1]
b = coeff(num_poly);
b0 = b(1);
b1 = b(2);
// 5. Формуємо матриці в канонічній формі керованості
A_ccf = [0, 1; -a0, -a1];
B_ccf = [0; 1];
C_ccf = [b0, b1];
```

```
disp("--- Підсумкова канонічна форма керованості ---");  
disp("A_ccf =", A_ccf);  
disp("B_ccf =", B_ccf);  
disp("C_ccf =", C_ccf);
```

Задача 4.3.2. Дослідити задані та канонічні моделі з метою визначення міри їхньої керованості за допомогою критеріїв Калмана

Розв'язування

У редакторі сценаріїв Scilab вводимо наступні команди.

```
// 1. Задаємо початкові матриці системи  
A = [-10, 12; -8, 10]; B = [-9; -6]; C = [-1, 2];  
disp("--- Початкова система ---");  
disp("A =", A, "B =", B, "C =", C);
```

```
// 2. Перевіряємо керованість
Q = cont_mat(A, B);
contr=rank(Q);
disp("--- Аналіз керованості ---");
disp("Матриця керованості Q =", Q);
disp("Ранг матриці керованості : ",contr);
select contr
case 2 then
disp("Матриця керована");
case 0 then
disp("Матриця не керована");
else
disp("Матриця частково керована");
end
```

Задача 4.3-3. Визначити підпростори керованості та області досяжності змінних стану досліджуваних моделей.

Розв'язування

1. Згідно критерію керованості Калмана, система є повністю керованою, якщо ранг матриці керованості є рівним порядку системи n .

Матриця керованості: $\mathbf{Q} = [\mathbf{B}; \mathbf{AB}; \mathbf{A}^2\mathbf{B}; \dots; \mathbf{A}^{n-1}\mathbf{B}]$.

Для нашої моделі другого порядку $\mathbf{Q} = [\mathbf{B}; \mathbf{AB}]$.

Будуємо матрицю керованості:

```
// 1. Задаємо початкові матриці системи
A = [-10, 12; -8, 10]; B = [-9; -6]; C = [-1, 2];
disp("--- Початкова система ---");
disp("A =", A, "B =", B, "C =", C);
// 2. Матриця керованості
Q = cont_mat(A, B);
disp("Матриця керованості Q =", Q);
```

У вікні консолі бачимо результат:

"Матриця керованості Q ="

-9. 18.

-6. 12.

У попередній задачі виявили, що ранг матриці керованості дорівнює 1. Другий стовпець матриці керованості дорівнює першому, помноженому на -2. Це означає, що вони лінійно залежні. Підпростір керованості — це не площина, а пряма лінія, що проходить через початок координат у напрямку вектора [3;2].

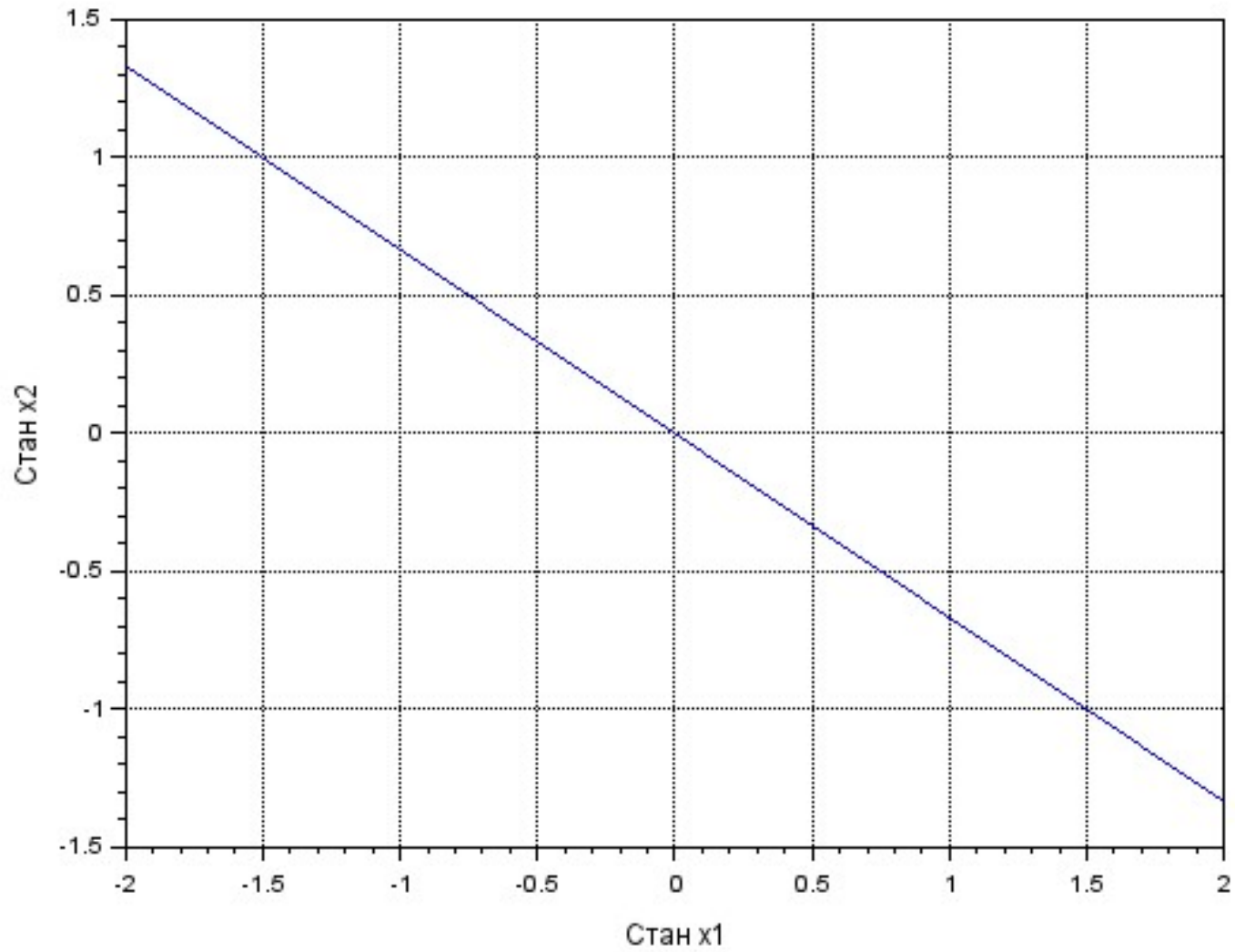
Рівняння підпростору: $2x_1 + 3x_2 = 0$ ($x_2 = -2x_1/3$).

Будуємо графік.

```
x1=-2:0.1:2;
```

```
plot(x1, -2*x1/3); xgrid
```

```
xlabel('Стан x1'); ylabel('Стан x2');
```



Ця лінія і зображує підростір керованості.

2. Область досяжності

Для лінійних неперервних систем область досяжності з нульового початкового стану $x(0) = 0$ за будь-який час $t > 0$ збігається з підпростором керованості.

Що це означає: Якщо ви стартуєте з «нуля», то якими б складними не були ваші маніпуляції з входом $u(t)$, ви ніколи не зможете покинути пряму $2x_1 + 3x_2 = 0$.

Весь інший простір площини для вас — «заборонена зона». Ви можете рухатися вперед і назад вздовж лінії, але не можете зробити ні кроку вбік.

Підсумок

Підпростір керованості: Одновимірною лінією $2x_1 + 3x_2 = 0$.

Область досяжності: Та сама лінія.

Висновок: Система має «жорстку» структуру, яка не дозволяє маневрувати в повному просторі станів.

Задача 4.3-4. Побудувати канонічну форму керованості для заданих моделей:

$$dx_1/dt = -10x_1 + 12x_2 - 9u;$$

$$dx_2/dt = -8x_1 + 10x_2 - 6u;$$

$$y = -x_1 + 2x_2 .$$

Розв'язування

1. Визначення матриці переходу (**T**)

Ми вводимо зв'язок між старим вектором стану **x** та новим **z** через невинроджену (таку, що може бути інвертована) матрицю **T**:

$$\mathbf{x}(t) = \mathbf{T} \cdot \mathbf{z}(t) \iff \mathbf{z}(t) = \mathbf{T}^{-1} \cdot \mathbf{x}(t).$$

2. Підстановка в рівняння стану

Підставимо $\mathbf{x}(t) = \mathbf{T} \cdot \mathbf{z}(t)$ в початкову систему:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u; \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}u, \end{cases}$$

$$\mathbf{T} \cdot \dot{\mathbf{z}}(t) = \mathbf{A} \cdot \mathbf{T} \cdot \mathbf{z}(t) + \mathbf{B}u.$$

Помножимо обидві частини рівняння зліва на T^{-1} :

$$\dot{z}(t) = (T^{-1} \cdot A \cdot T) z(t) + (T^{-1} \cdot B) u$$

3. Нові матриці системи

Після перетворення ми отримуємо нову систему рівнянь:

$$\begin{aligned}\dot{z}(t) &= \bar{A} z(t) + \bar{B} u(t), \\ y(t) &= \bar{C} z(t).\end{aligned}$$

з новими матрицями.

Нова матриця стану: $\bar{A} = T^{-1} \cdot A \cdot T$.

Новий вектор входу: $\bar{B} = T^{-1} \cdot B$.

Новий вектор виходу: $\bar{C} = C \cdot T$.

Алгоритм побудови матриці T

Крок 1: Обчислення матриці керованості

Для системи 2-го порядку $Q = [B; AB]$.

Крок 2: Знаходження характеристичного полінома

Обчислюємо $\det(sI - A) = s^2 + a_1 s + a_0$.

Крок 3: Будуємо допоміжну матрицю W :

а) побічна діагональ матриці заповнюється одиницями;

б) над цією діагоналлю розташовуються коефіцієнти характеристичного полінома;
в) все, що під цією діагоналлю, заповнюється нулями.
Для нашої системи

$$W = \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}$$

Крок 4: Формування матриці переходу **T**

Матриця переходу обчислюється як добуток: **T=Q·W**.

Виконаємо розв'язання у Scilab. Створюємо наступний сценарій.

```
// 1. Задаємо початкові матриці системи
```

```
A = [-10, 12; -8, 10]; B = [-9; -6]; C = [-1, 2];
```

```
disp("--- Початкова система ---");
```

```
disp("A =", A, "B =", B, "C =", C);
```

```
// 2. Матриця керованості
```

```
Q = cont_mat(A, B);
```

```
disp("Матриця керованості Q =", Q);
```

```
// 3. Знаходимо характеристичний поліном
s = poly(0, 's'); // Задаємо символну змінну 's'
char_poly = det(s*eye(2,2) - A);
disp("Характеристичний поліном:", char_poly);
// Вилучаємо коефіцієнти полінома [a0, a1]
a = coeff(char_poly);
a0 = a(1);
a1 = a(2);
// 4. Будуємо допоміжну матрицю
W=[a1, 1; 1,0];
disp("W=", W);
// Формування матриці переходу
T=Q·W;
disp("T=", T);
// Визначник матриці T
disp("Визначник T=",det(T));
```

Виконавши сценарій, отримуємо результат:

```
"T="
```

```
18. -9.
```

```
12. -6.
```

```
"Визначник T="
```

```
0.
```

Це означає, що матриця T є вироджена, тому використання стандартного методу розрахунку матриць \bar{A} , \bar{B} і \bar{C} є неможливим. Матриці КФУ слід розраховувати через передатну функцію, як це показано в задачі 4.3-1. Отримаємо результат:

```
"A_ccf ="
```

```
0. 1.
```

```
4. 0.
```

```
"B_ccf ="
```

```
0.
```

```
1.
```

```
"C_ccf ="
```

```
6. -3.
```

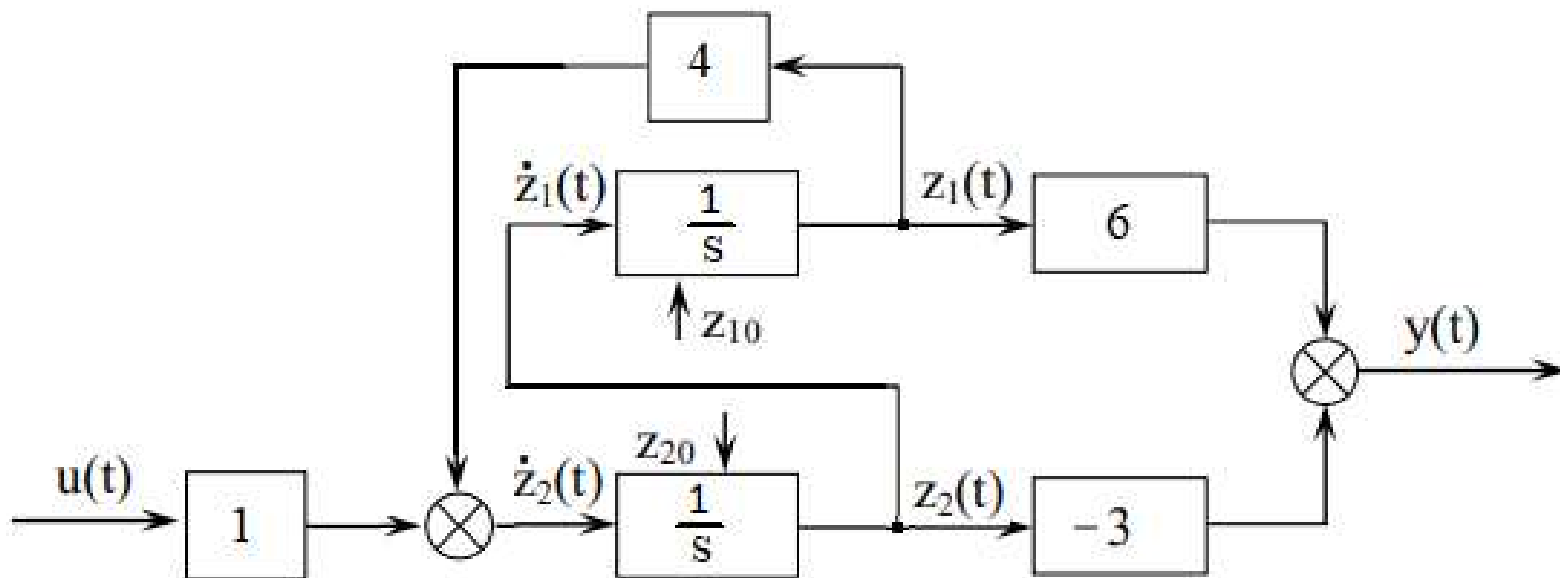
Загальний вигляд КФУ (матрична форма)

$$\begin{cases} \dot{z} = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \\ y = \begin{bmatrix} 6 & -3 \end{bmatrix} z. \end{cases}$$

Конкретні рівняння для нашої моделі

$$\begin{cases} \dot{z}_1(t) = z_2(t), & z_1(0) = z_{10} \\ \dot{z}_2(t) = 4z_1(t) + u(t), & z_2(0) = z_{20} \\ y(t) = 6z_1(t) - 3z_2(t) \end{cases}$$

Структурна схема



Задача 5.3-1. Для записаних нижче моделей побудувати канонічні моделі з діагональними матрицями при змінних стану.

$$dx_1/dt = -10x_1 + 12x_2 - 9u;$$

$$dx_2/dt = -8x_1 + 10x_2 - 6u;$$

$$y = -x_1 + 2x_2 .$$

Розв'язування

Знаходимо поліноми передатної функції, як в задачі 4.3-1.

// 1. Задаємо початкові матриці системи

$$A = [-10, 12; -8, 10]; B = [-9; -6]; C = [-1, 2];$$

// 2. Знаходимо характеристичний поліном (знаменник)

`s = poly(0, 's');` // Задаємо символну змінну 's'

`char_poly = det(s*eye(2,2) - A);`

`disp("Знаменник передатної функції", char_poly);`

// 3. Знаходимо чисельник передатної функції

// Використовуємо приєднану матрицю `adj(sI - A)` для 2x2

`adj_matrix = [s - A(2,2), A(1,2); A(2,1), s - A(1,1)];`

`num_poly = C * adj_matrix * B;`

```
disp("Чисельник передатної функції :", num_poly);  
// 4. Вилучаємо коефіцієнти поліномів  
a = coeff(char_poly);  
a0 = a(1); a1 = a(2);  
b = coeff(num_poly);  
b0 = b(1); b1 = b(2);  
disp("--- коефіцієнти поліномів ---");  
disp(sprintf("a0 = %g a1 = %g b0 = %g b1 = %g", a0, a1, b0, b1))
```

Маємо результат

Чисельник (6 - 3s): $b_1 = -3$, $b_0 = 6$.

Знаменник (-4 + s² = -4 + 0s + s²): $a_1 = 0$, $a_0 = -4$.

Побудова матриці стану (\mathbf{A}_o)

У КФС матриця стану \mathbf{A}_o є транспонованою відносно матриці КФУ. Коефіцієнти знаменника розташовуються у першому стовпці:

$$\mathbf{A}_o = \begin{bmatrix} -a_1 & 1 \\ -a_0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix}.$$

Побудова вектора входу (B_o)

У КФС вектор входу B_o містить у собі всі коефіцієнти чисельника передавальної функції. Це показує, як вхідний сигнал $u(t)$ розподіляється між інтеграторами:

$$B_o = \begin{bmatrix} b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} -3 \\ 6 \end{bmatrix}.$$

Побудова вектора виходу (C_o)

Ми домовляємося, що вихід системи — це просто перший стан z_1 :

$$C_o = [1 \ 0].$$

Підсумковий вигляд:

$$\begin{cases} \dot{z} = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} z + \begin{bmatrix} -3 \\ 6 \end{bmatrix} u \\ y = [1 \ 0] z \end{cases}$$

$$\begin{cases} \dot{z}_1(t) = z_2(t) - 3u(t) \\ \dot{z}_2(t) = 4z_1(t) + 6u(t) \\ y(t) = z_1(t) \end{cases}$$

Задача 5.3-2. Дослідити задані та канонічні моделі з метою визначення міри їхньої спостережуваності за допомогою критеріїв Калмана

Розв'язування

У редакторі сценаріїв Scilab вводимо наступні команди.

```
// 1. Задаємо початкові матриці системи
A = [-10, 12; -8, 10]; B = [-9; -6]; C = [-1, 2];
disp("--- Початкова система ---");
disp("A =", A, "B =", B, "C =", C);
// 2. Перевіряємо спостережуваність
N = obsv_mat(A, C);
observ = rank(N);
disp("--- Аналіз спостережуваності ---");
disp("Матриця спостережуваності N =", N);
disp("Ранг матриці спостережуваності : ", observ);
```

```
select observ
case 2 then
disp("Матриця спостережувана");
case 0 then
disp("Матриця не спостережувана");
else
disp("Матриця частково спостережувана");
end
```

Задача 5.3-3. Визначити підпростори спостережуваності досліджуваних моделей.

Розв'язування

Система називається спостережуваною, якщо на кінцевому інтервалі часу, в кінці цього інтервалу, по виходу системи $y(t)$ при відомій керуючій дії $u(t)$ можна визначити всі початкові компоненти вектора стану $x(t)$.

Матриця спостережуваності:

$$\mathbf{H} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \dots \\ \mathbf{CA}^{n-1} \end{bmatrix}$$

Для нашої моделі другого порядку

$$\mathbf{H} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ -6 & 8 \end{bmatrix}$$

У теорії систем підпростором спостережуваності називають простір станів, які «дають свій внесок» у вихідний сигнал.

Визначається як лінійна оболонка рядків матриці **H**.

Лінійна оболонка — це множина **всіх можливих** комбінацій векторів, що є рядками матриці **H**.

Маємо два вектори (рядки нашої матриці):

$$\bullet \quad r_1 = C = [-1 \quad 2]$$

$$\bullet \quad r_2 = CA = [-6 \quad 8]$$

Представимо матрицю як систему лінійних рівнянь:

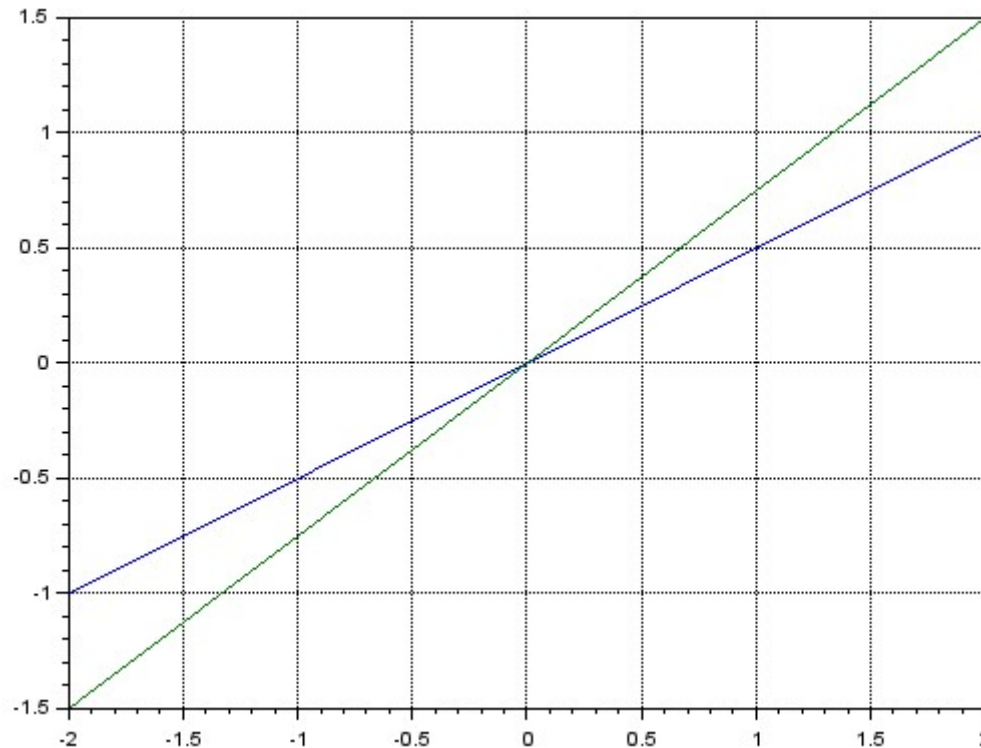
$$-1 \cdot x_1 + 2 \cdot x_2 = 0$$

$$-6 \cdot x_1 + 8 \cdot x_2 = 0$$

Представимо їх графіками на площині (x_1, x_2) :

```
--> x1=-2:0.1:2;
```

```
--> plot(x1, x1/2, x1, 6*x1/8); xgrid
```



Щоб визначити на цьому графіку простір спостережуваності, спочатку виділимо на ньому простір неспостережуваності.

Математично це ядро (kernel) матриці спостережуваності.

Ядро матриці \mathbf{A} — це множина усіх векторів x , які при множенні на матрицю дають нульовий вектор ($\mathbf{Ax}=0$). Отримаємо ядро за допомогою функції Scilab:

--> $W = \text{kernel}(N)$

$W =$

$[\]$

Оскільки ми отримали «пустий» результат, це означає, що простору неспостережуваності нема, тобто спостережуваною є вся площина станів системи (x_1, x_2) .

Задача 5.3-4. Побудувати канонічну форму спостережуваності для заданих моделей.

Розв'язування

Канонічна форма спостережуваності (КФС) є дзеркальним відображенням форми керованості (КФУ). У цій формі структура матриць дозволяє «відновити» внутрішні стани системи, маючи лише вхід u та вихід y .

1. Побудова матриць КФС

Для нашої системи з передатною функцією

$$G(s) = \frac{-3s+6}{s^2-4}$$

коефіцієнти знаменника ($a_1 = 0$, $a_0 = -4$) та чисельника ($b_1 = -3$, $b_0 = 6$) розподіляються в матрицях наступним чином:

Матриця стану \mathbf{A}_{obs} : Коефіцієнти знаменника записуються у перший стовпець:

$$A_{obs} = \begin{bmatrix} -a_1 & 1 \\ -a_0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix}$$

Матриця входу \mathbf{B}_{obs} : Містить коефіцієнти чисельника:

$$B_{obs} = \begin{bmatrix} b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} -3 \\ 6 \end{bmatrix}$$

Матриця виходу \mathbf{C}_{obs} : Має вигляд:

$$C_{obs} = [1 \quad 0]$$

Загальний вигляд моделі в КФС (матрична форма):

$$\begin{cases} \dot{z} = \begin{bmatrix} -a_1 & 1 \\ -a_0 & 0 \end{bmatrix} z + \begin{bmatrix} b_1 \\ b_0 \end{bmatrix} u \\ y = [1 \quad 0] z \end{cases}$$

Конкретні рівняння для нашої моделі:

$$\begin{cases} \dot{z}_1 = 0 \cdot z_1 + z_2 - 3u \\ \dot{z}_2 = 4 \cdot z_1 + 6u \\ y = z_1 \end{cases}$$

Кожне рівняння в КФС має свій фізико-математичний сенс для спостерігача.

Рівняння виходу ($y = z_1$). Ми прирівнюємо перший стан z_1 безпосередньо до того, що бачимо на датчику виходу y .

Перше динамічне рівняння. Показує, що зміна першого стану залежить від другого стану та безпосереднього впливу входу.

Друге динамічне рівняння замикає цикл зворотного зв'язку через стан z_1 .

2. Структурна схема КФС

Головна відмінність схеми КФС від КФУ полягає в тому, що сигнал входу $u(t)$ тепер подається на вхід кожного інтегратора через свої коефіцієнти b_i , а вихід $y(t)$ знімається безпосередньо з першого інтегратора.

