

СЕКЦИЯ 2.
Моделирование динамических систем

Часть 2.

Председатель:
к. т. н., доцент В. В. Анохин

Оглавление

Поршнеv С. В. ИССЛЕДОВАНИЕ В МАТЛАВ ОСОБЕННОСТЕЙ ВЫНУЖДЕННЫХ КОЛЕБАНИЙ ЦЕПОЧКИ СВЯЗАННЫХ ГАРМОНИЧЕСКИХ ОСЦИЛЛЯТОРОВ	665
Прокопенко М. Н., Окунева Г. Л., Прасол Д. А. МОДЕЛИРОВАНИЕ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ ТЕПЛОНОСИТЕЛЯ В КАМЕРНОЙ СУШИЛКЕ	682
Спирина А. Н. ВЫЧИСЛЕНИЕ КОРРЕЛЯЦИОННОЙ РАЗМЕРНОСТИ ВРЕМЕННОГО РЯДА КОТИРОВОК ВАЛЮТ	688
Степанов А. В. ОТЫСКАНИЕ ПЕРИОДИЧЕСКИХ РЕШЕНИЙ ОДНОЙ ДИСКРЕТНОЙ СИСТЕМЫ С ГИСТЕРЕЗИСОМ.....	696
Тимченко А. А., Снитюк В. Е., Оксамытная Л. П., Ланских Е. П. ИДЕНТИФИКАЦИЯ КРИТЕРИАЛЬНОЙ ФУНКЦИИ НА БАЗЕ САМООРГАНИЗАЦИИ МОДЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ СИМВОЛЬНОЙ МАТЕМАТИКИ В СРЕДЕ МАТЛАВ.....	706
Фатуев В. А., Понятский В. М., Каргин А. В. ПАКЕТ ПРОГРАММ СТРУКТУРНОЙ И ПАРАМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ ЛИНЕЙНЫХ ОДНОМЕРНЫХ ДИНАМИЧЕСКИХ СИСТЕМ	715
Халин А. Л., Халина Н. А. ПОСТРОЕНИЕ СИММЕТРИЧЕСКИХ БИФУРКАЦИОННЫХ МНОГООБРАЗИЙ АНАЛИТИЧЕСКИХ ДИФФЕРЕНЦИАЛЬНЫХ СИСТЕМ	763
Черников С. А., Самер аль-Салек ИСПОЛЬЗОВАНИЕ ПАКЕТА МАТЛАВ/SIMULINK, ДЛЯ МОДЕЛИРОВАНИЯ И ИССЛЕДОВАНИЯ ГИРОСТАБИЛИЗАТОРА С АКТИВНЫМ ДИНАМИЧЕСКИМ ГАСИТЕЛЕМ КОЛЕБАНИЙ	773
Юмагулов М. Г., Нуров И. Д. Шарафутдинов И. В. АЛГОРИТМЫ ПРИБЛИЖЕННОГО ИССЛЕДОВАНИЯ ЗАДАЧИ О БИФУРКАЦИИ АНДРОНОВА-ХОПФА	775

УДК 530.18

ИССЛЕДОВАНИЕ В МАТЛАВ ОСОБЕННОСТЕЙ ВЫНУЖДЕННЫХ КОЛЕБАНИЙ ЦЕПОЧКИ СВЯЗАННЫХ ГАРМОНИЧЕСКИХ ОСЦИЛЛЯТОРОВ

Поршнев С. В.

Уральский государственный технический университет, Екатеринбург,
e-mail: sergey_porshnev@mail.ru

1. Введение

Модели линейных цепочек (рис. 1), совершающих вынужденные колебания, применяются в различных областях физики: физике твердого тела, радиофизике и т. п. (Например, в [1] для описания системы трех связанных электрических колебательных контуров использована модель, представляющая собой систему шариков с массами m_1, m_2, m_3 , связанных между собой пружинками одинаковой жесткости k .) Несмотря на то, что общие методы решения данной задачи известны [2], их практическая реализация сопряжена с рядом трудностей. Во-первых, аналитические решения уравнений движения длинных линейных цепочек ($N > 3$) могут быть получены только для относительно небольшого числа случаев [2]:

- 1) $k_0 = k_1 = \dots = k_{N-1}, m_0 = m_1 = \dots = m_{N-1}$;
- 2) $k_0 = k_2 = k_4 = \dots, k_1 = k_3 = k_5 = \dots; m_0 = m_1 = \dots = m_{N-1}$;
- 3) $k_0 = k_1 = \dots = k_{N-1}, m_0 = m_2 = m_4 = \dots, m_1 = m_3 = m_5 = \dots$;
- 4) $k_i = k \neq k_N, i = 0, 1, \dots, N-1, m_0 = m_1 = \dots = m_{N-1}$,
- 5) $k_i = k, i = 0, 1, \dots, N, m_i = m \neq m_{N-1}, i = 0, 1, \dots, N-2$.

Во-вторых, большинство этих решений оказываются весьма громоздкими и для их последующего анализа приходится использовать ПК.

Однако даже в этом случае решение удастся найти только для установившегося режима, и для частот, далеких от резонанса. Для переходных процессов при совпадении частоты внешней вынуждающей силы с одной из резонансных частот известно аналитическое решение для малого количества осцилляторов (2–3), однако, оно оказывается настолько громоздким, что для его анализа приходится применять ПК. В этих условиях для исследования линейных цепочек связанных осцилляторов использовать численные методы.

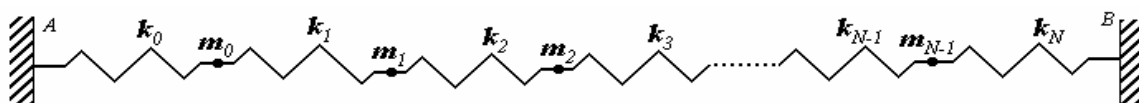


Рис. 1.

В статье описывается методика использования MATLAB для анализа особенностей вынужденных колебаний цепочки связанных линейных гармонических осцилляторов.

2. Математическая модель системы связанных гармонических осцилляторов и ее реализация в MATLAB

Будем считать, что внешняя вынуждающая сила приложена к крайнему левому шарiku (рис. 1). Тогда система дифференциальных уравнений, описывающих вынужденные колебания связанных линейных осцилляторов, имеет следующий вид:

$$\begin{cases} \ddot{x}_0 = -\omega_{00}^2 x_0 - \omega_{10}^2 (x_0 - x_1) + \frac{F(t)}{m_0}, \\ \ddot{x}_i = -\omega_{ii}^2 (x_i - x_{i-1}) - \omega_{i+1,i}^2 (x_i - x_{i+1}), \quad i = 1 \dots N-2, \\ \ddot{x}_{N-1} = -\omega_{N-1,N-1}^2 (x_{N-1} - x_{N-2}) - \omega_{NN-1}^2 x_{N-1}, \end{cases} \quad (1)$$

где $\omega_{i+1,i}^2 = k_{i+1,i} / m_i$, m_i — масса i -го шарика, $k_{i+1,i}$ — жесткость пружины, соединяющей i -ый и $i+1$ -ый шарики, $F(t)$ — вынуждающая сила.

Для нахождения численного решения системы ДУ в MATLAB используется следующий алгоритм:

1) задать вектор-функцию, возвращающую значения первых производных системы ДУ (размерность функции $2N$);

2) задать вектор, содержащий начальные условия $(x_i(0), \dot{x}_i(0))$, $i = 1, 2, \dots, N-1$);

3) обратиться к одной из функций, возвращающих таблицу, содержащую численное решение системы ДУ, например функции **ode45**;

4) провести визуализацию полученных численных решений.

Описание функции, возвращающей значения первых производных системы ДУ (1), мы разместили в файле **Euler2.m**, листинг которого приводится ниже.

% листинг файла Euler2.m

function z=euler2(t,x)

% описание функции, возвращающей значения первых производных

global k m A Omega

N=length(m); % число тел колебательной системы

z=zeros(2*N,1);

z(1)=x(2);

z(2)=-(k(1)/m(1))*x(1)-(k(2)/m(1))*(x(1)-x(3))+F(t,A,Omega);

K=3;

for i=2:N-1

z(K)=x(K+1);

z(K+1)=-(k(i)/m(i))*(x(K)-x(K-2))-(k(i+1)/m(i))*(x(K)-x(K+2));

K=K+2;

end;

```
z(2*N-1)=x(2*N);
z(2*N)=-(k(N)/m(N))*(x(2*N-1)-x(2*N-3))-k(N+1)/m(N)*x(2*N-1);
```

```
function f=F(t,A,Omega)
```

```
% функция, описывающая внешнюю вынуждающую силу
```

```
f=A*sin(Omega*t);
```

Далее для нахождения и визуализации численного решения системы ДУ (1), описывающих систему, совершающую свободные колебания, необходимо выполнить следующую последовательность команд:

```
>> clear all;
>> global k m A Omega
>> N=16; % число тел колебательной системы
>> n=1:N;
>> m(n)=1; % задание масс тел
>> i=1:N+1;
>> k(i)=1; % задание жесткостей пружин колебательной системы
>> n=1:2*N;
% начальные условия
>> R0(n)=0;
>> R0(1)=0.5
>> A=0; % амплитуда внешней вынуждающей силы
>> Omega=0.0; % частота внешней вынуждающей силы
>> Tfin=50*pi; правая граница временной сетки
>> Np=2^13-1; % число узлов временной сетки
>> [T,M]=ode45('euler2',[0:Tfin/Np:Tfin],R0);
>> subplot(4,4,1);plot(T,M(:,1));axis([0 40 —0.5 0.5]);title('n=1');
>> subplot(4,4,5);plot(T,M(:,3));axis([0 40 —0.5 0.5]);title('n=2');
>> subplot(4,4,9);plot(T,M(:,5));axis([0 40 —0.5 0.5]);title('n=3');
>> subplot(4,4,13);plot(T,M(:,7));axis([0 40 —0.5 0.5]);title('n=4');
>> subplot(4,4,2);plot(T,M(:,9));axis([0 40 —0.5 0.5]);title('n=5');
>> subplot(4,4,6);plot(T,M(:,11));axis([0 40 —0.5 0.5]);title('n=6');
>> subplot(4,4,10);plot(T,M(:,13));axis([0 40 —0.5 0.5]);title('n=7');
>> subplot(4,4,14);plot(T,M(:,15));axis([0 40 —0.5 0.5]);title('n=8');
>> subplot(4,4,3);plot(T,M(:,17));axis([0 40 —0.5 0.5]);title('n=9');
>> subplot(4,4,7);plot(T,M(:,19));axis([0 40 —0.5 0.5]);title('n=10');
>> subplot(4,4,11);plot(T,M(:,21));axis([0 40 —0.5 0.5]);title('n=11');
>> subplot(4,4,15);plot(T,M(:,23));axis([0 40 —0.5 0.5]);title('n=12');
>> subplot(4,4,4);plot(T,M(:,25));axis([0 40 —0.5 0.5]);title('n=13');
>> subplot(4,4,8);plot(T,M(:,27));axis([0 40 —0.5 0.5]);title('n=14');
>> subplot(4,4,12);plot(T,M(:,29));axis([0 40 —0.5 0.5]);title('n=15');
>> subplot(4,4,16);plot(T,M(:,31));axis([0 40 —0.5 0.5]);title('n=16');
```

Результат выполнения приведенной выше последовательности команд представлен на рис. 2.

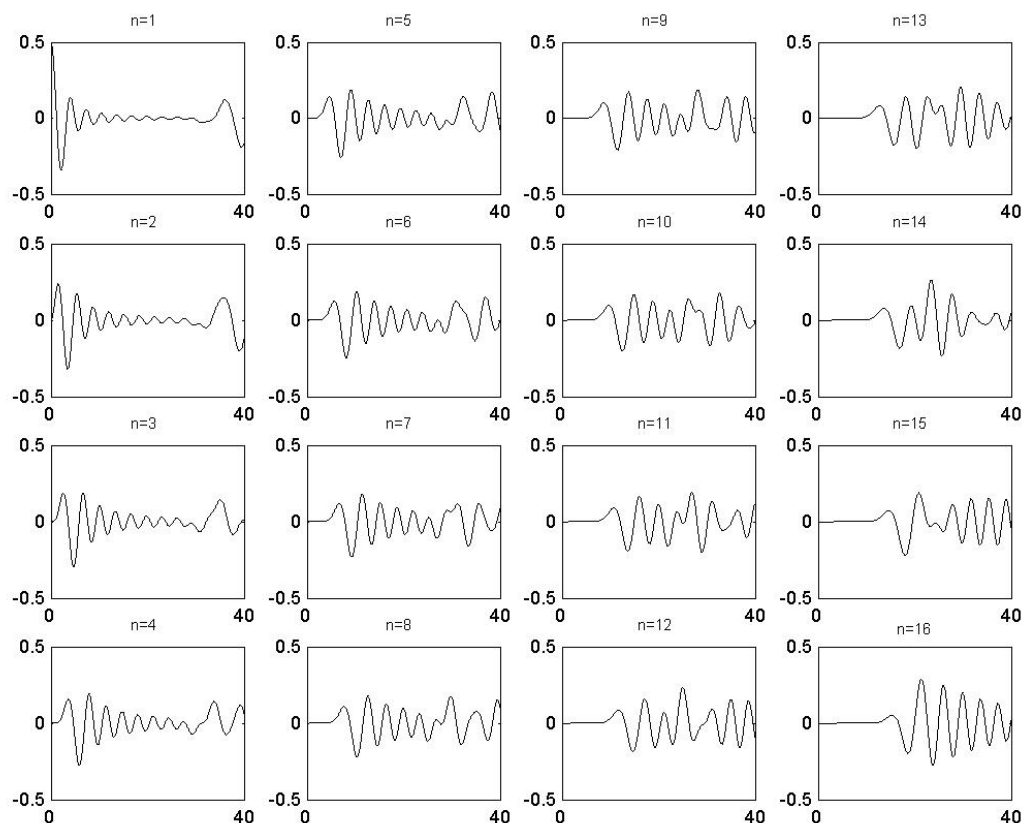


Рис. 2. Зависимость мгновенных значений смещения тел колебательной системы от времени.

Одной из основных проблем численного решения ДУ и систем ДУ является проблема выбора шага интегрирования Δt , поскольку при достаточно большом шаге интегрирования возникают неустойчивые решения, т. е. решения, погрешность которых начинает возрастать во времени экспоненциально быстро. Один из способов проверки устойчивости метода заключается в контроле величины полной энергии, которая в случае свободных колебаний должна сохраняться, поэтому для проверки правильности выбора шага интегрирования Δt можно использовать следующий алгоритм:

- 1) Задать начальные смещения и скорости тел цепочки связанных осцилляторов.
- 2) Задать временной интервал, на котором ищется решение системы ДУ.
- 3) Задать число точек, в которых ищется численное решение системы ДУ.
- 4) Найти решение системы ДУ.
- 5) Вычислить значения энергии системы связанных осцилляторов в каждый момент времени.

6) Проанализировать изменение энергии системы во времени на заданном временном интервале и оценить точность выполнения закона сохранения энергии.

7) При неудовлетворительной точности решения повторить пп. 3–6.

8) При удовлетворительной точности решения перейти к анализу вынужденных колебаний.

Как очевидно, для реализации описанного выше алгоритма необходимо уметь вычислять энергии каждого из тел системы в заданные моменты времени. Для решения данной задачи можно использовать функцию **En**, описание которой мы сохранили в файле **En.m**.

% листинг файла **En.m**

function E=En(N,m,k,M)

K=length(M); % число строк матрицы решений

K1=1;

for j=1:N

for i=1:K

if j>1

if j<N

$e(i)=0.5*m(j)*M(i,2*j).^2+...$
 $0.25*k(j)*(M(i,2*j-1)-M(i,2*j-3)).^2+...$
 $0.25*k(j+1)*(M(i,2*j-1)-M(i,2*j+1)).^2;$

end;

end;

if j==1

$e(i)=0.5*m(1)*M(i,2)^2+0.5*k(1)*M(i,1)^2+...$
 $0.25*k(2)*(M(i,1)-M(i,3))^2;$

end;

if j==N

$e(i)=0.5*m(N)*M(i,2*N)^2+0.5*k(N+1)*M(i,2*N-1)^2+...$
 $0.25*k(N)*(M(i,2*N-1)-M(i,2*N-3))^2;$

end;

end;

en(:,K1)=e';

K1=K1+1;

end;

E=en;

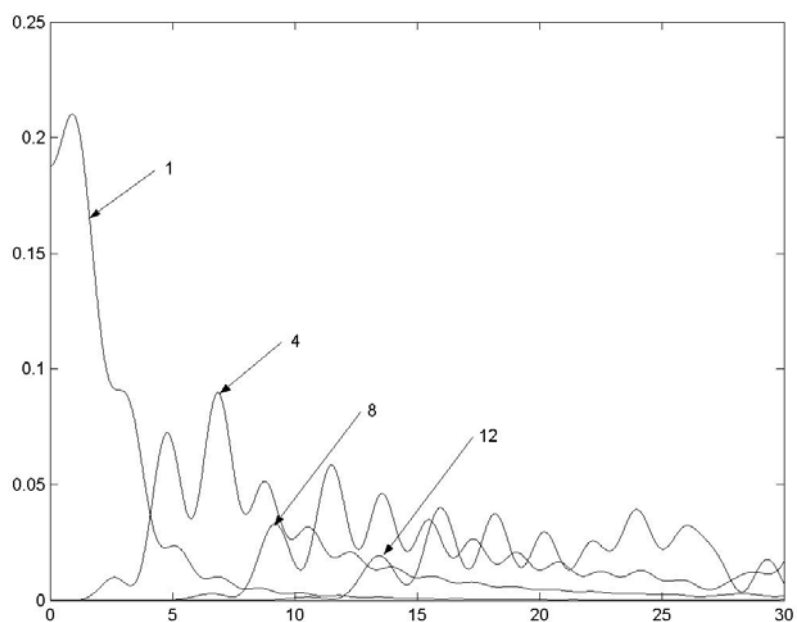


Рис. 3. Мгновенные значения энергий выбранных тел колебательной системы.

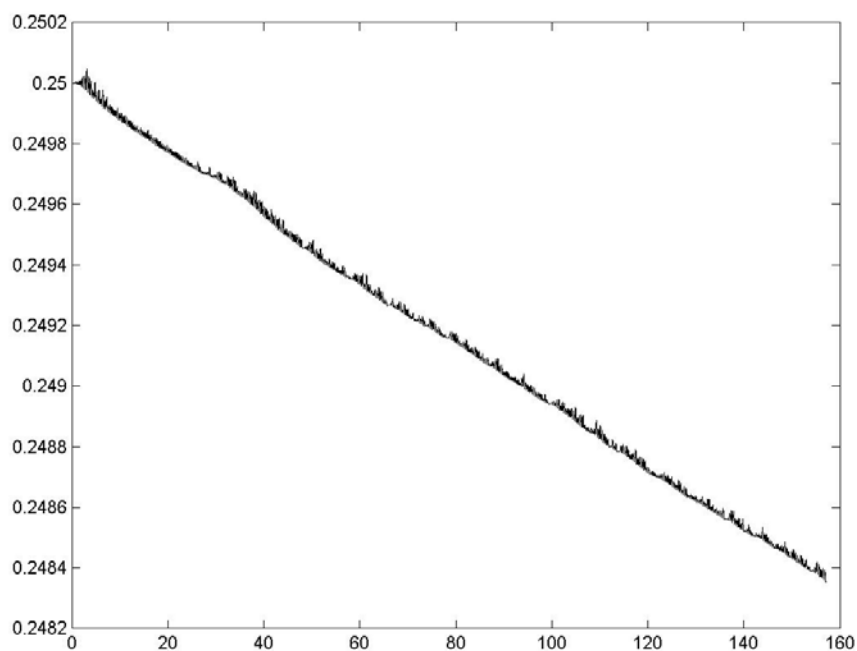


Рис. 4. Зависимость полной энергии системы от времени.

Для вычисления мгновенных значений энергии тел колебательной системы и ее полной энергии необходимо выполнить следующую последовательность команд:

```
>> clear all;
>> global k m A Omega
>> N=16;
>> n=1:N;
```



```

>> m(n)=1;
>> i=1:N+1;
>> k(i)=1;
>> n=1:2*N;
>> R0(n)=0;
>> R0(1)=0.5;
>> A=0.0;
>> Omega=0.0;
>> Tfin=50*pi;
>> Np=2^13-1;
>> [T,M]=ode45('euler2',[0:Tfin/Np:Tfin],R0);
>> E=En(N,m,k,M);
>> plot(T,E(1,:),T,E(8,:),T,E(16
>> for i=1:Np+1
    s=E(i,:);
    Efull(i)=sum(s);
end;
>> figure(1);plot(T,E(:,1),T,E(:,4),T,E(:,8),T,E(:,12))
>> axis([0 30 0 0.25]);
>> figure(2);plot(T,Efull)

```

Результаты выполнения приведенной выше последовательности команд представлены на рис. 3, 4. Анализ зависимости мгновенной энергии колебательной системы от времени, представленная на рис. 4, показывает, что полная энергия отклоняется от своего первоначального значения по линейному закону. При этом максимальная величина отклонения, характеризующая точность численного решения, на выбранном временном интервале не превосходит 0.24%.

3. Исследование особенностей вынужденных колебаний линейной цепочки связанных осцилляторов с одинаковыми массами

Рассмотрим результаты моделирования колебательной системы, состоящей из пяти шариков ($N = 5$) одинаковой массы ($m = 1$), соединенных пружинками одинаковой жесткости ($k_i = 1$, $i = 1, 2, \dots, 5$) под действием внешней вынуждающей силы, частота которой совпадает с одной из собственных частот изучаемой системы. Собственные частоты данной системы ω_α являются решением характеристического уравнения

$$|B - \omega^2 E| = 0, \quad (3)$$

где E — единичная матрица, B — трехдиагональная матрица, элементы которой вычисляются по следующим правилам:

$$\begin{aligned}
 B_{11} &= 2\omega_0^2, \quad B_{12} = -\omega_0^2, \\
 B_{ii-1} &= -\omega_0^2, \quad B_{ii} = 2\omega_0^2, \quad B_{ii+1} = -\omega_0^2, \quad i = 2, 3, \dots, N-1, \\
 B_{NN-1} &= -\omega_0^2, \quad B_{NN} = 2\omega_0^2.
 \end{aligned} \quad (4)$$

Значения собственных частот при выбранных параметрах системы равны 1.0, 1.4142136, 1.7320508, 1.9318517, 0.5176381. Зависимости смещений тел колебательной системы от времени для $\gamma=1.0$, 1.4142 представлены на рис. 5,6, соответственно.

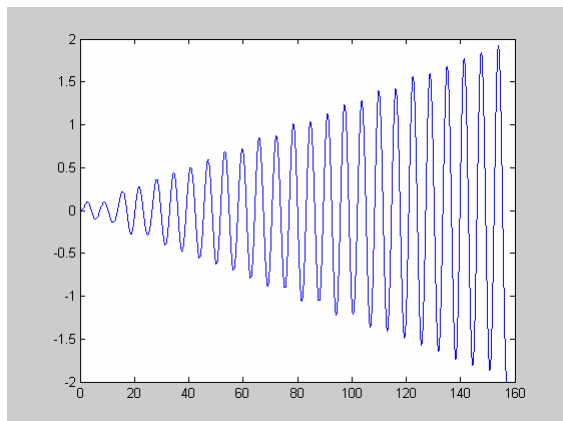


Рис. 5, а. Зависимость смещения первого шарика линейной цепочки от времени ($\gamma=1.0$).

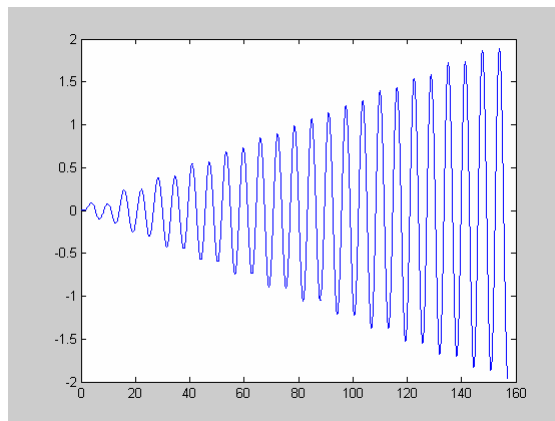


Рис. 5, б. Зависимость смещения второго шарика линейной цепочки от времени ($\gamma=1.0$).

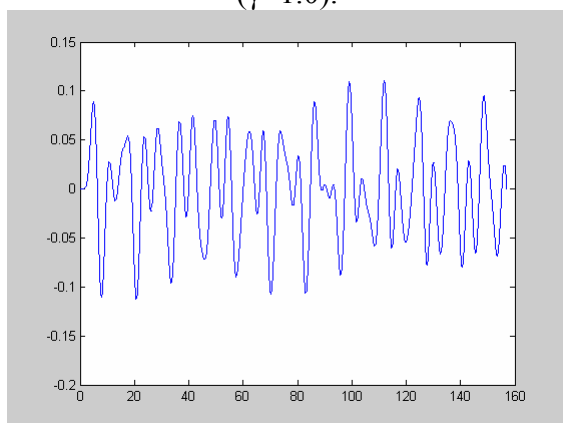


Рис. 5, в. Зависимость смещения третьего шарика линейной цепочки от времени ($\gamma=1.0$).

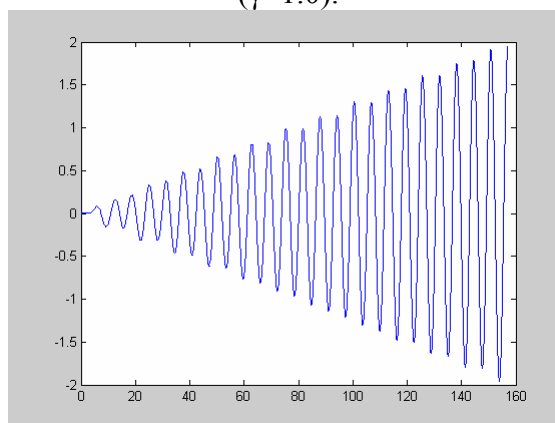


Рис. 5, г. Зависимость смещения четвертого шарика линейной цепочки от времени ($\gamma=1.0$).

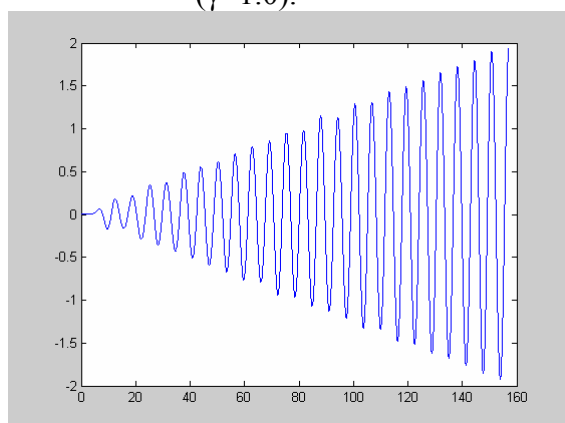


Рис. 5, д. Зависимость смещения пятого шарика линейной цепочки от времени ($\gamma=1.0$).

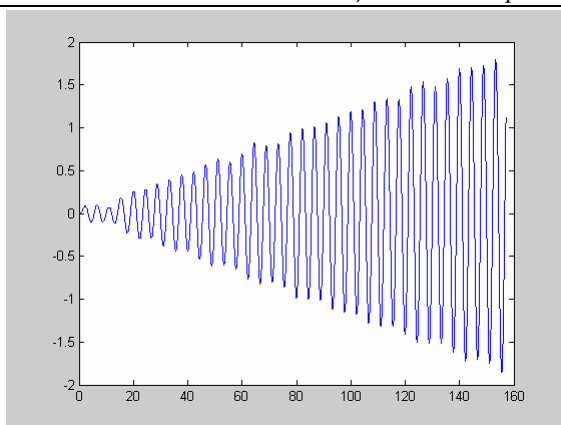


Рис. 6, а. Зависимость смещения первого шарика линейной цепочки от времени ($\gamma=1.4121$).

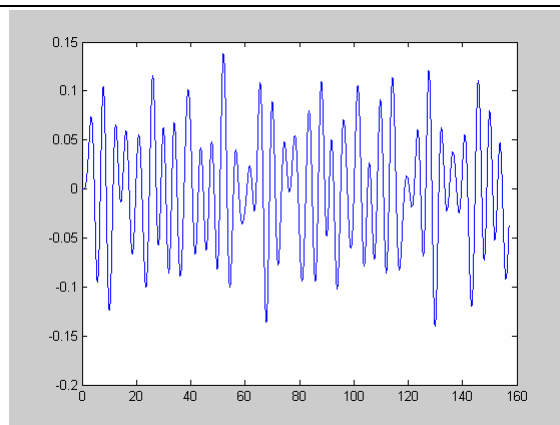


Рис. 6, б. Зависимость смещения второго шарика линейной цепочки от времени ($\gamma=0.4121$).

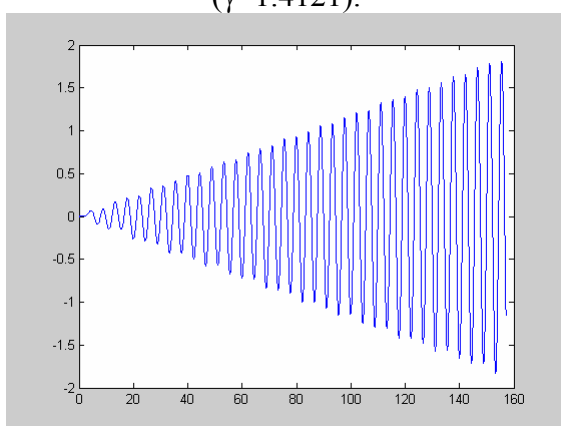


Рис. 6, в. Зависимость смещения третьего шарика линейной цепочки от времени ($\gamma=1.4121$).

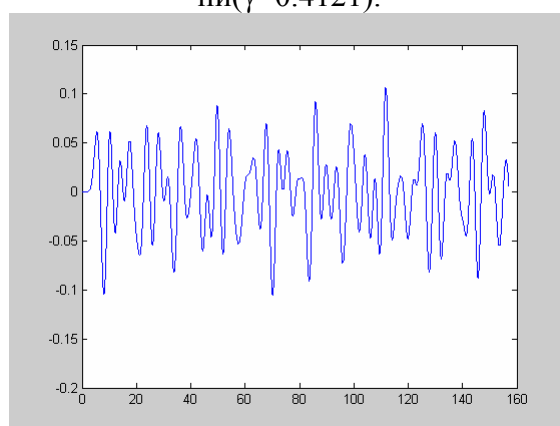


Рис. 6, г. Зависимость смещения четвертого шарика линейной цепочки от времени ($\gamma=1.4121$).

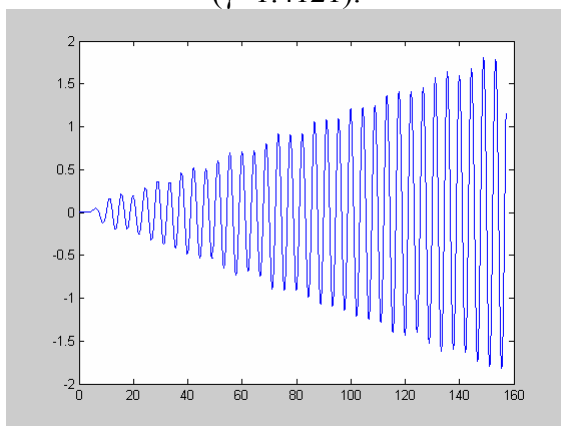


Рис. 6, д. Зависимость смещения пятого шарика линейной цепочки от времени ($\gamma=1.4121$).

Из рис. 5, 6 видно, что при возбуждении на резонансной частоте в системе есть шарик, движение которых не является резонансным. Для случая $\gamma=1$ таковым является шарик № 3, для случая $\gamma=1.4142136$ — шарик № 2, 4. Полученный результат, оказывающийся на первый взгляд неожиданным, требует дополнительных пояснений.

Система (1) является системой линейных неоднородных дифференциальных уравнений второго порядка. Как известно [3], общее решение такой системы является суммой общего решения соответствующей однородной системы и какого-либо частного решения неоднородной. Решение однородной системы является суперпозицией нормальных гармонически колебаний с собственными частотами ω_α , являющимся решением характеристического уравнения (3). Каждому собственному числу ω_α^2 , $\alpha = 1, 2, \dots, N$ соответствует собственный вектор Ξ_α , являющийся решением уравнения

$$\mathbf{B} \cdot \Xi_\alpha = \omega_\alpha^2 \cdot \Xi_\alpha. \quad (5)$$

Общее решение однородного уравнения представляется в виде:

$$x(t) = \sum_{\alpha=1}^N C_\alpha \Xi_\alpha e^{i\omega_\alpha t}, \quad (6)$$

где C_α — произвольные комплексные постоянные, определяемые из начальных условий.

В том случае, когда задача о собственных колебаниях решена для нахождения частного решения СДУ в установившемся режиме можно использовать метод разложения по собственным формам [4]. Следуя данному методу, представим вектор внешних сил в виде:

$$\tilde{\mathbf{F}}(t) = \sum_{\alpha=1}^N \tilde{f}_\alpha(t) \mathbf{M} \Xi_\alpha. \quad (7)$$

Здесь \mathbf{M} — матрица масс, являющаяся в нашем случае диагональной, а $\tilde{f}_\alpha(t)$ — проекции вектора внешних сил на вектор нормальных колебаний. Из (6), (7), видно, что вектор внешних сил и искомый вектор смещения могут быть представлены в комплексном виде:

$$\tilde{\mathbf{F}} = \begin{pmatrix} \tilde{F}_1 \\ \vdots \\ \tilde{F}_N \end{pmatrix} \exp(i\gamma t); \quad \tilde{\mathbf{X}} = \begin{pmatrix} \tilde{A}_1 \\ \vdots \\ \tilde{A}_N \end{pmatrix} \exp(i\gamma t), \quad (8)$$

где $\tilde{F}_i = F_i \exp(i\psi_i)$, $\tilde{A}_i = A_i \exp(i\varphi_i)$ — комплексные амплитуды.

Можно показать [4], что коэффициенты разложения \tilde{f}_α удовлетворяют равенству:

$$\tilde{f}_\alpha = \frac{\Xi_\alpha \cdot \tilde{\mathbf{F}}}{\Xi_\alpha \cdot \mathbf{M} \Xi_\alpha}. \quad (9)$$

При этом частное решение СДУ представляется в виде:

$$\tilde{\mathbf{X}}_s = \tilde{\mathbf{A}}_s \exp(i\gamma t), \quad (10)$$

$$\text{где } \tilde{A}_s = \sum_{\alpha=1}^N \frac{\tilde{f}_\alpha \Xi_{s\alpha}}{\omega_\alpha^2 - \gamma^2}.$$

При нулевых начальных условиях, то есть при $x_i(0) = 0$, $\dot{x}_i(0) = 0$ начальные фазы колебаний всех шариков, следовательно, и мнимые части комплексных амплитуд \tilde{A}_s , также будут равны нулю.

Из (10) видно, что для резонансного возбуждения какой-либо моды в системе с большим числом степеней свободы необходимо не только обеспечить резонансное соотношение между частотой этой моды и частотой внешней силы, но и создать такие условия, чтобы воздействие силы на разные элементы системы не оказалось взаимно скомпенсированным. При невыполнении данных условий в исследуемой системе, возбуждаемой на одной из собственных частот, резонировать будет только часть шариков. Например, как видно из формулы (10), шарик с номером s будет резонировать на частоте ω_α , если $\tilde{f}_\alpha \Xi_{s\alpha} \neq 0$. Таким образом вычислив матрицу $R_{s\alpha} = \tilde{f}_\alpha \Xi_{s\alpha}$ мы сможем без численного решения СДУ сделать выводы о поведении системы при $\gamma = \omega_\alpha$.

Проверим приведенные выше рассуждения, вычислив матрицу $R_{s\alpha}$. Для вычисления данной матрицы оказывается удобным создать две m-функции, листинги которых приведены ниже.

% листинг файла B4.m

function A=B4(k,m);

% функция, возвращающая матрицу B в соответствии с (4)

Nm=length(m);

Nk=length(k);

Omega=zeros(Nk,Nm);

for j=1:Nk

for i=1:Nm

Omega(j,i)=k(j)/m(i);

end;

end;

A=zeros(Nm,Nm);

for i=1:Nm

if i==1

A(1,1)=Omega(1,1)+Omega(2,1);

A(1,2)=-Omega(2,1);

end;

if (i>1)&(i<Nm)

A(i,i-1)=-Omega(i,i);

A(i,i)=Omega(i,i)+Omega(i+1,i);

A(i,i+1)=-Omega(i+1,i);

end;

if i==Nm

A(i,i-1)=-Omega(i,i);

A(i,i)=Omega(i,i)+Omega(i+1,i);

end;

end;

% листинг файла F4.m

function R=F4(k,m,F);

% функция, возвращающая матрицу $R_{s\alpha} = \tilde{f}_{\bar{o}} O_{s\bar{o}}$ в соответствии с (9) и

% собственные частоты колебаний линейной цепочки

B=B4(k,m);

N=length(m);

[Sigma,Teta]=eig(B);

Teta=Teta.^0.5;

M=diag(m);

N=length(F);

for i=1:N

tmp=Sigma(:,i);

H(i)=dot(tmp,F)/dot(tmp,M*tmp);

end;

for i=1:N

for j=1:N

R(i,j)=H(i)*Sigma(j,i);

end;

end;

for i=1:N

tmp(i)=Teta(i,i);

end;

R=[R,tmp];

Для вычисления матрицы $R_{s\alpha} = \tilde{f}_{\alpha} \Xi_{s\alpha}$ рассматриваемой системы необходимо выполнить следующую последовательность команд:

>> N=5; % задаем число шариков

>> n=1:N;

>> m(n)=1; % задаем массы шариков

>> j=1:N+1;

>> k(j)=1; % задаем жесткости пружин

>> F(n)=0;F(1)=1; % задаем вектор внешней силы

>> Q=F4(k,m,F) % вычисляем матрицу $R_{s\alpha} = \tilde{f}_{\bar{o}} O_{s\bar{o}}$

Q =

```

1.8660  3.2321  3.7321  3.2321  1.8660  0.5176
-1.5000 -1.5000  0.0000  1.5000  1.5000  1.0000
1.0000 -0.0000 -1.0000 -0.0000  1.0000  1.4142
-0.5000  0.5000 -0.0000 -0.5000  0.5000  1.7321
0.1340 -0.2321  0.2679 -0.2321  0.1340  1.9319

```

Из полученного результата видно, что при частоте внешней вынуждающей силы, равной 0.5176, будут находиться в резонансе шарики, при частоте 1.0000 — не в резонансе будет средний шарик (как и было обнаружено при численном решении СДУ (1), см. рис. 5), 1.4142 — не в резонансе будут второй и четвертый шарики (как и было обнаружено при численном решении СДУ (1), см. рис. 6), 1.7321 — не в резонансе будет средний шарик, 1.9319 — все шарики будут находиться в резонансе. Зависимо-

сти смещений шариков от времени при $\gamma = 0.5176, 1.7321, 1.9319$ представлены на рис. 7, 8, 9, соответственно.

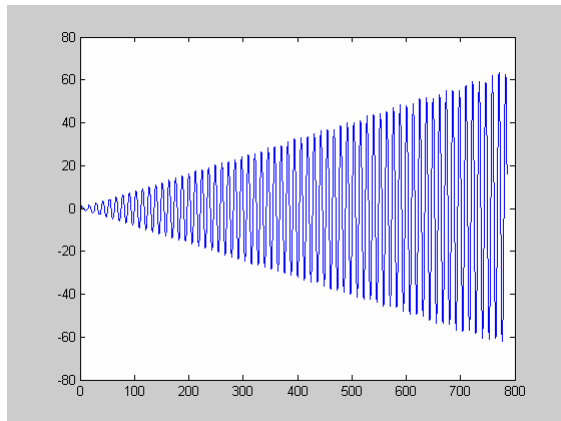


Рис. 7, а. Зависимость смещения первого шарика линейной цепочки от времени ($\gamma = 0.5176$).

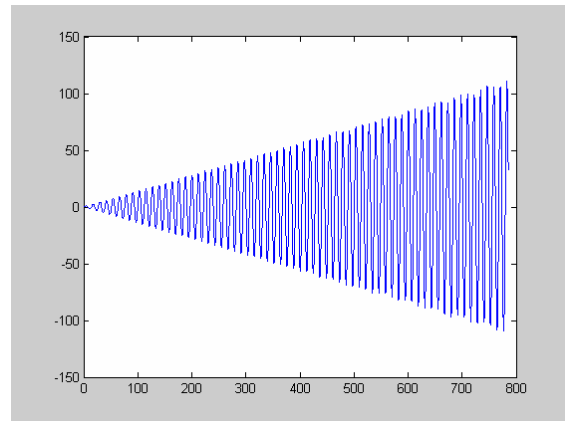


Рис. 7, б. Зависимость смещения второго шарика линейной цепочки от времени ($\gamma = 0.5176$).

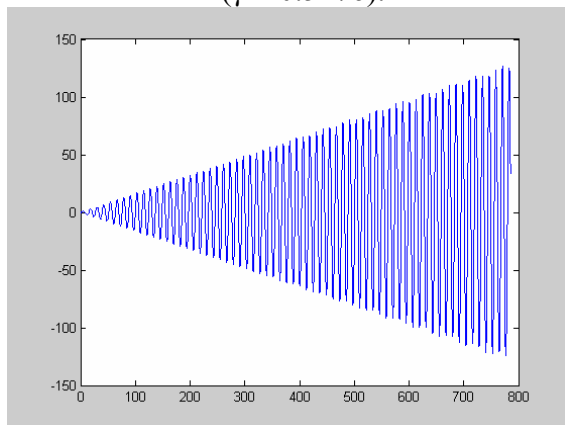


Рис. 7, в. Зависимость смещения третьего шарика линейной цепочки от времени ($\gamma = 0.5176$).

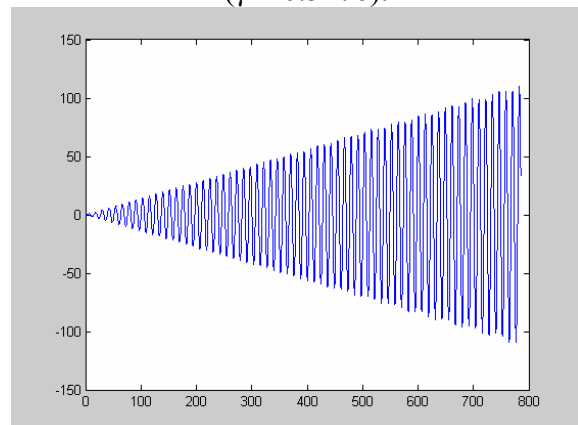


Рис. 7г. Зависимость смещения четвертого шарика линейной цепочки от времени ($\gamma = 0.5176$).

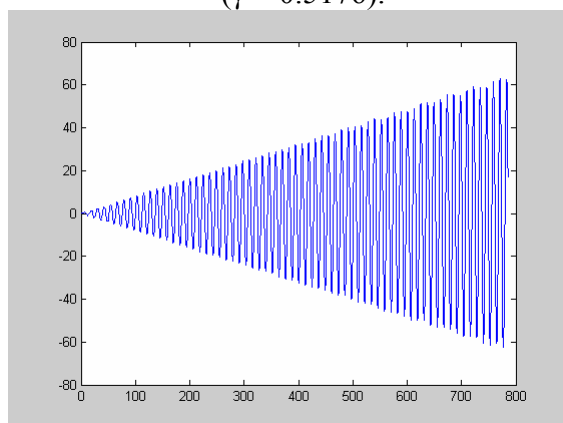


Рис. 7, д. Зависимость смещения пятого шарика линейной цепочки от времени ($\gamma = 0.5176$).

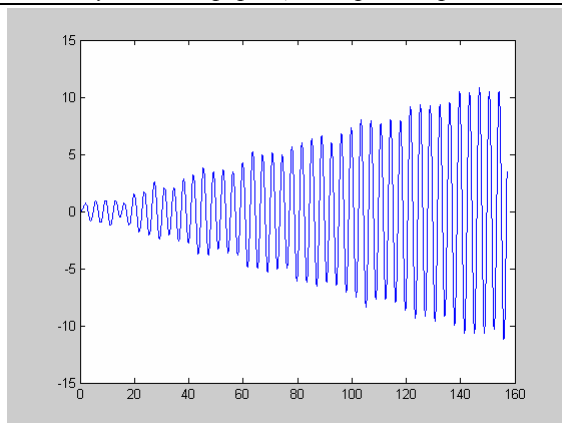


Рис. 8, а. Зависимость смещения первого шарика линейной цепочки от времени ($\gamma = 1.7321$).

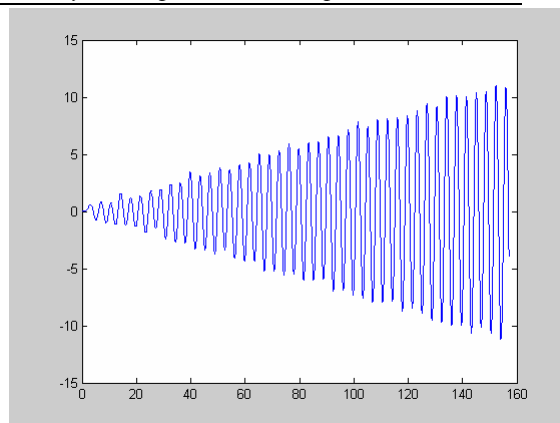


Рис. 8, б. Зависимость смещения второго шарика линейной цепочки от времени ($\gamma = 1.7321$).

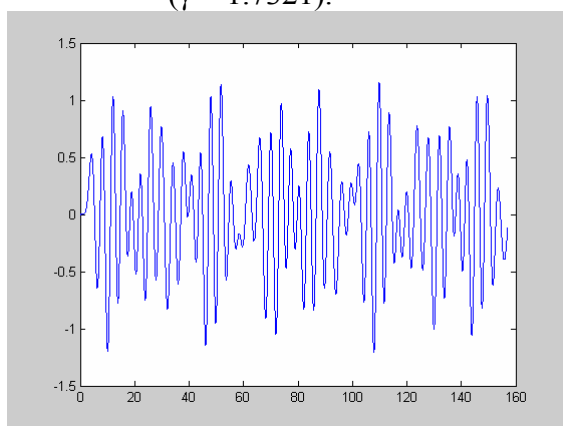


Рис. 8, в. Зависимость смещения третьего шарика линейной цепочки от времени ($\gamma = 1.7321$).

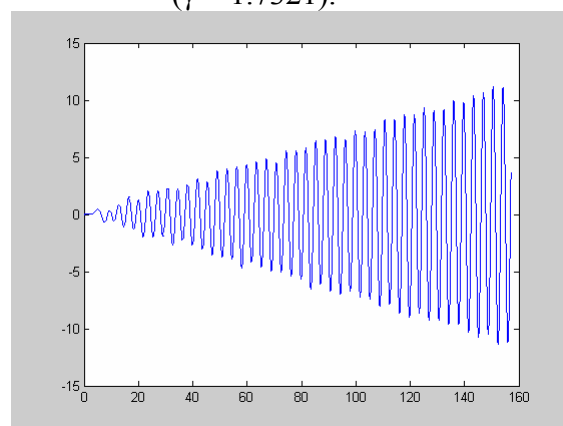


Рис. 8, г. Зависимость смещения четвертого шарика линейной цепочки от времени ($\gamma = 1.7321$).

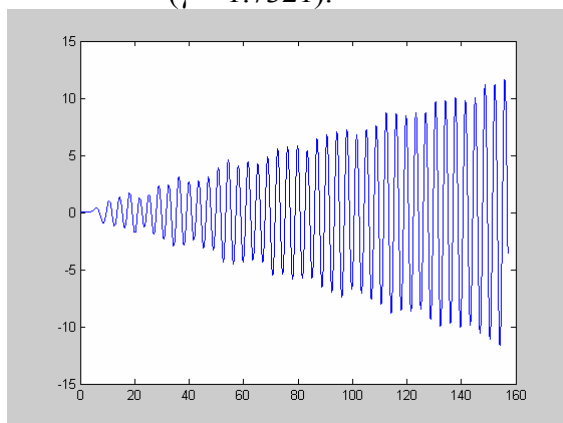


Рис. 8, д. Зависимость смещения пятого шарика линейной цепочки от времени ($\gamma = 1.7321$).

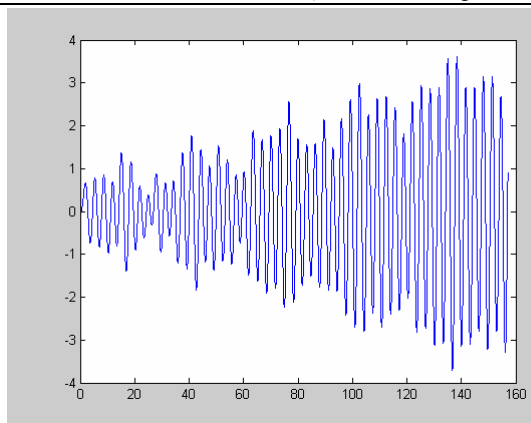


Рис. 9, а. Зависимость смещения первого шарика линейной цепочки от времени ($\gamma = 1.9319$).

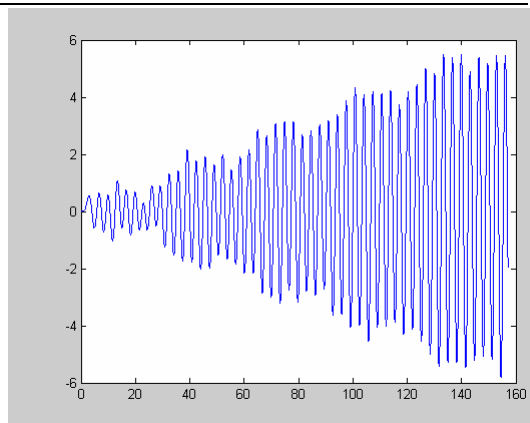


Рис. 9, б. Зависимость смещения второго шарика линейной цепочки от времени ($\gamma = 1.9319$).

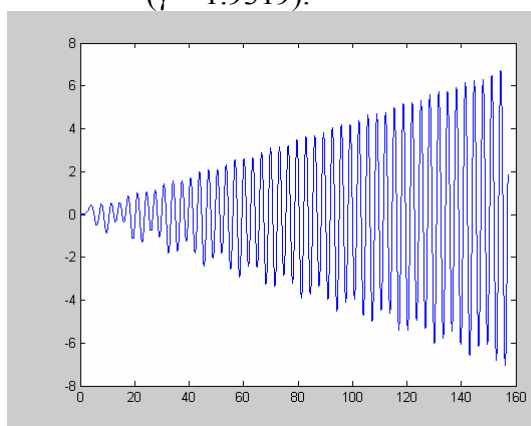


Рис. 9, в. Зависимость смещения третьего шарика линейной цепочки от времени ($\gamma = 1.9319$).

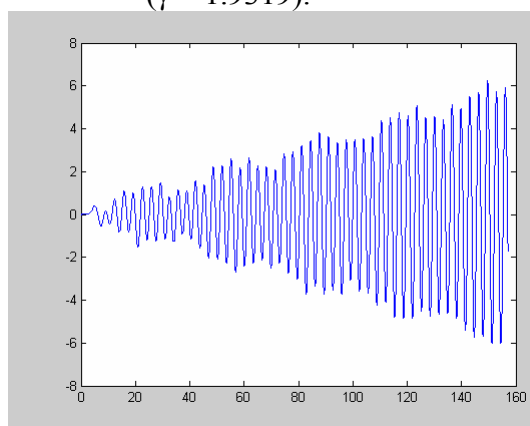


Рис. 9, г. Зависимость смещения четвертого шарика линейной цепочки от времени ($\gamma = 1.9319$).

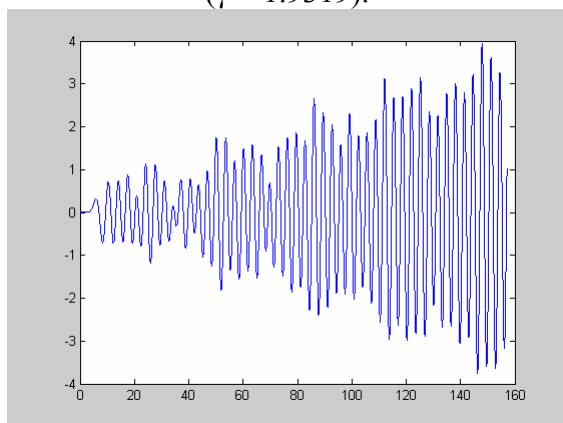


Рис. 9, д. Зависимость смещения пятого шарика линейной цепочки от времени ($\gamma = 1.9319$).

Проведенный нами анализ поведения системы при различных значениях N позволил сделать следующие общие выводы:

1. В общем случае при резонансном воздействии на колебательную систему закон изменения амплитуды колебаний отличается от линейного (рис. 9). Однако можно оценить скорость роста амплитуды, используя ме-

тод наименьших квадратов. (Для этих целей из вектора $x_i(t_i)$ производится выборка значений, соответствующим локальным максимумам, затем по методу наименьших квадратов найти коэффициенты аппроксимирующей прямой $kx+b$. Угол наклона данной прямой характеризует среднюю скорость возрастания амплитуды при резонансе и прямо пропорционален соответствующему элементу матрицы \mathbf{R} .)

2. Появление нулевых членов в матрице \mathbf{R} возможно только при нечетном значении N . (Другими словами при четном N все шарики резонируют на каждой из собственных частот.)

3. При нечетном значении N существуют такие собственные частоты, на которых каждый n -й шарик не резонирует. Значение n может меняться от 2 до $N/2 - 1$. (Например, из матрицы \mathbf{R} для $N = 5$ видно, что при $\gamma = \omega_\alpha$ и $\gamma = \omega_3$ резонируют все шарики, кроме среднего (то есть $n = N/2 - 1 = 3$), а при $\gamma = \omega_2$, $n = 2$, то не резонируют шарики с четными номерами.)

4. При четных значениях N элементы матрицы \mathbf{R} является симметричной (шарики колеблются в фазе), либо антисимметричной (шарики колеблются в противофазе, причем скорость возрастания амплитуды у них одинакова).

5. При нечетных значениях N элементы матрицы \mathbf{R} , симметричные относительно столбца, номер которого равен номеру центрального шарика, одинаковы (шарики колеблются в фазе), либо отличаются знаками (шарики колеблются в противофазе, причем скорость возрастания амплитуды у них одинакова).

Заключение

Проведено исследование особенностей резонансных явлений в системе связанных осцилляторов, выявлены некоторые важные закономерности исследуемой системы. Полученные результаты могут оказаться полезными в радиофизике и электротехнике (например, при расчете электрических фильтров), в теории упругости, физике твердого тела. Разработанная в MATLAB методика анализа данной системы, может быть использована как в методических целях (в качестве компьютерной поддержки при изучении курсов теории колебаний, физики, твердого тела, радиофизики), так и при решении практических задач.

Литература

1. Гетманова Е. Г., Костарев Д. Б. Резонансные явления в системе связанных осцилляторов // Электромагнитные волны и электронные системы.— 2001.— Т.6.— № 5.
2. Коткин Г. Л., Сербо В. Г. Сборник задач по классической механике.— М.: Наука, 1977.

3. Ландау Л. Д., Лифшиц Е. М. Механика.— М.: Наука, 2001.
4. Кандидов В. П., Капцов Г. Н., Харламов А. А. Решение и анализ задач линейной теории колебаний.— М: Изд-во МГУ, 1976.

УДК 691.42

МОДЕЛИРОВАНИЕ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ ТЕПЛОНОСИТЕЛЯ В КАМЕРНОЙ СУШИЛКЕ

*Прокопенко М. Н., Окунева Г. Л., Прасол Д. А.
Белгородский государственный технологический университет
им. В. Г. Шухова, Белгород,
e-mail: tk-apit@yandex.ru*

Процессы тепловой обработки в технологии строительной керамики имеют чрезвычайно важное значение как для качества готовой продукции, так и для экономики производства. Одной из важнейших операций в технологии теплоизоляционных материалов является сушка. Она занимает до 90 % продолжительности всего производственного цикла и на нее расходуется не менее половины всех производственных затрат. Одной из основных задач в области промышленности строительных материалов является повышение качества изделий и увеличение объема выпускаемой продукции за счет усовершенствования технологических процессов и режимов, повышения производительности труда, интенсификации работы существующего оборудования и внедрения комплексной автоматизации технологических процессов.

Решение описанных задач напрямую связано с уровнем детализации и глубиной математического описания технологических процессов сушки керамических изделий, что невозможно без применения в расчетах специализированного программного обеспечения. В работе приводится разработка математической модели движения и распределения теплоносителя в камерной сушилке периодического действия средствами MATLAB, являющаяся основой модели термовлажностной обработки керамических изделий. Сушильным агентом является нагретый воздух. Объект моделирования — камерная сушилка периодического действия.

При построении математической модели используются некоторые ограничения и допущения:

- Решается плоская задача.
- Влажность поступающего воздуха не учитывается и принимается постоянной.
- Рассматривается стационарный процесс для скорости истечения газа.
- Взаимодействием частиц и их влиянием на поле скоростей пренебрегаем.
- Рассматриваем несжимаемый газ, то есть изменение плотности отсутствует ($\rho = const$).
- Имеет место только конвективный теплообмен.

- Влияние градиента избыточного давления пара пренебрежимо мало.
- Камерная сушилка обладает идеальными теплотехническими характеристиками, т. е. отсутствуют потери тепла на нагрев элементов сушилки и изменение влагосодержания теплоносителя за счет конструктивных особенностей сушилки.
- Физические параметры и коэффициенты материала и сушильного агента в процессе сушки принимаются постоянными.

Построение модели движения и распределения нагретого воздуха внутри сушилки основывается на использовании полевых моделей воздухообмена и распределения субстанций во внутреннем пространстве камеры. Основой таких моделей могут служить следующие уравнения.

Уравнение неразрывности:

$$\nabla \cdot V = 0, \quad (1)$$

где V — вектор скорости.

Уравнение движения:

$$\rho \frac{DV}{Dt} = -\nabla \cdot p + \mu \cdot \nabla^2 \cdot V, \quad (2)$$

где V — вектор скорости, ρ — плотность среды, p — статическое давление, μ — динамический коэффициент вязкости.

Уравнение энергии:

$$\rho \cdot c_v \frac{DT}{Dt} = k \cdot \nabla^2 \cdot T + \Phi, \quad (3)$$

где V — вектор скорости, ρ — плотность среды, T — температура, Φ — диссипативный член.

Уравнения (1) и (2) записаны относительно так называемых примитивных переменных p и V . В одном из самых распространенных методов решения уравнений Навье-Стокса для несжимаемой жидкости примитивные переменные заменяются на функцию тока ψ и завихренность ω . Подход с использованием завихренности и функции тока в качестве независимых переменных является одним из самых распространенных методов решения двумерных уравнений Навье-Стокса для несжимаемой жидкости. В нем делают замену переменных, переходя от компонент скорости к функции тока ψ и завихренности ω . В декартовых двумерных координатах функция тока и завихренность определяются как:

$$\frac{\partial \psi}{\partial y} = u, \frac{\partial \psi}{\partial x} = -v, \quad (4)$$

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \quad (5)$$

Используя новые независимые переменные, получаются два новых уравнения.

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{\mu}{\rho} \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right). \quad (6)$$

Это параболическое уравнение с частными производными называется уравнением переноса завихренности. Для его решения можно применять численные методы решения нелинейного уравнения Бюргера.

Дополнительное уравнение для независимых переменных ψ и ω :

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega. \quad (7)$$

Это эллиптическое уравнение с частными производными является уравнением Пуассона.

В результате такой замены переменных возможно разделить смешанную эллиптически-параболическую систему уравнений Навье–Стокса для несжимаемой жидкости на одно параболическое уравнение (уравнение переноса завихренности) и одно эллиптическое уравнение (уравнение Пуассона). Обычно эти уравнения решают методом установления по времени.

Полученные уравнения решаются методом сеток с среде системы научных и инженерных расчетов MATLAB, с учетом конструктивных особенностей сушильной камеры, начальных и граничных условий. Поперечный разрез сушилки высотой $H = 3$ м и длиной $D = 15.6$ м. Подача воздуха осуществляется через два отверстия в потолке сушилки и удаляется воздух через отверстие, расположенное в полу сушилки. Длины проемов $H_1=H_2=H_3=0.6$ м. Скорость потока на входе через первое отверстие U_1 , через второе U_2 , а на выходе U_3 . Внутри сушильной камеры должен существовать баланс воздуха, то есть $Q_{\text{ВХ}} = Q_{\text{ВЫХ}}$.

$$\begin{aligned} Q &= U \cdot S, \\ Q_1 &= U_1 \cdot H_1, \\ Q_2 &= U_2 \cdot H_2, \\ Q_{\text{ВХ}} &= Q_1 + Q_2, \\ Q_{\text{ВЫХ}} &= Q_3 = U_3 \cdot H_3, \\ U_1 \cdot H_1 + U_2 \cdot H_2 &= U_3 \cdot H_3, \\ U_{\text{ВЫХ}} &= U_3 = \frac{U_1 \cdot H_1 + U_2 \cdot H_2}{H_3}. \end{aligned} \quad (8)$$

Задаются краевые условия для скорости и функции тока. $U_y=0$ на всех стенках, $U_x=0$ на стенках, а в проемах $U_x=U_{\text{ВХ}}$ и $U_x=U_{\text{ВЫХ}}$ соответственно. Функция тока ϕ непрерывна и постоянна:

$$\begin{aligned} U_x &= \frac{\partial \phi}{\partial y}, \\ \int \frac{\partial \phi}{\partial y} dy &= \int U_{\text{ВХ}} dy \Rightarrow \phi = U_{\text{ВХ}} y. \end{aligned} \quad (9)$$

Задается число узлов сетки по осям с учетом того, что на проемы должно приходиться по 2 и более узлов. По оси OX : $N = 104$ точки, шаг $h_x = D/N = 0.15$; по оси OY : $M = 20$ точек, шаг $h_y = H/M = 0.15$.

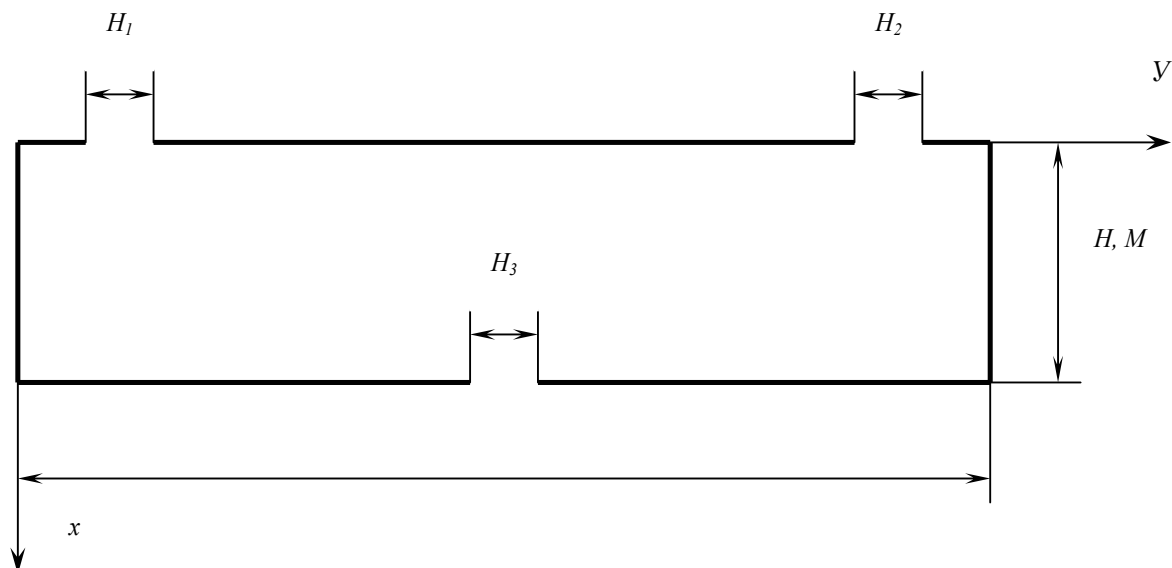


Рис. 1. Размеры элементов сушильной камеры и ориентация осей.

Сеточные функции:

$$\begin{aligned} U(x, y) &\sim U(x_i; y_j) = U_{ij}, \\ U_x(x, y) &\sim Ux(x_i; y_j) = Ux_{ij}, \\ U_y(x, y) &\sim Uy(x_i; y_j) = Uy_{ij}, \\ \varphi(x, y) &\sim F(x_i; y_j) = F_{ij}. \end{aligned} \quad (10)$$

Сеточные аналоги для уравнений:

$$\begin{aligned} \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} &= 0, \\ F_{ij} &= \frac{h_y (F_{i+1,j} + F_{i-1,j}) + h_x (F_{i,j+1} + F_{i,j-1})}{2(h_x^2 + h_y^2)}, \end{aligned} \quad (11)$$

для внутренних точек сетки $i \in [1; N-1], j \in [1; M-1]$.

На границах условия в сеточных функциях запишутся так:

- 1) левая стенка: $j = 0, 0 \leq i \leq M, Ux_{ij} = 0, Uy_{ij} = 0, F_{ij} = U_1 N$;
- 2) правая стенка: $j = N, 0 \leq i \leq M, Ux_{ij} = 0, Uy_{ij} = 0, F_{ij} = U_2 N$;

3) потолок: вне проема: $i = 0, Ux_{ij} = 0, Uy_{ij} = 0, F_{ij} = 0$, в 1-ом проеме:

$i = 0, Ux_{ij} = U_1, Uy_{ij} = 0, F_{ij} = U_1 y$, во 2-ом проеме:

$i = 0, Ux_{ij} = U_2, Uy_{ij} = 0, F_{ij} = U_2 y$.

4) пол: вне проема: $i = M, Ux_{ij} = 0, Uy_{ij} = 0, F_{ij} = U_1 N$ или

$i = M, Ux_{ij} = 0, Uy_{ij} = 0, F_{ij} = U_2 N$, в проеме:

$i = 0, Ux_{ij} = U_3, Uy_{ij} = 0, F_{ij} = U_3 y$.

Начальные условия для скорости внутри сетки $Ux_{ij} = 0, Uy_{ij} = 0$; для функции тока: $F_{ij} = 0$ или $F_{ij} = (F_{0,j} + F_{M,j}) / M$.

Признак окончания пересчета значений функции тока:
 $|F_{ij}^n - F_{ij}^{n+1}| < \xi$.

Текст программы для решения дифференциального уравнения и нахождения значений функции F при поставленных начальных и граничных условиях:

```

l = 0;
R0 = E + 1;
while R0 > E
    R0 = 0;
    l = l + 1;
    for i = 2:20
        for j = 2:52
            Q = (hy*hy*(F(i+1, j) + F(i-1, j)) + hx*hx*(F(i, j+1) + F(i, j-1)))/(2*(hx*hx+hy*hy));
            R = abs(Q - F(i, j));
            if R > R0
                R0 = R;
            end
            F(i, j) = Q;
        end
    end
end

```

```

l = 0;
R0 = E + 1;
while R0 > E
    R0 = 0;
    l = l + 1;
    for i = 2:20
        for j = 54:104
            Q = (hy*hy*(F(i+1, j) + F(i-1, j)) + hx*hx*(F(i, j+1) + F(i, j-1)))/(2*(hx*hx+hy*hy));
            R = abs(Q - F(i, j));
            if R > R0
                R0 = R;
            end
            F(i, j) = Q;
        end
    end
end

```


end
end

В результате решения находится поле скоростей теплоносителя и строятся линии тока — линии наиболее интенсивного движения и распространения нагретого воздуха.

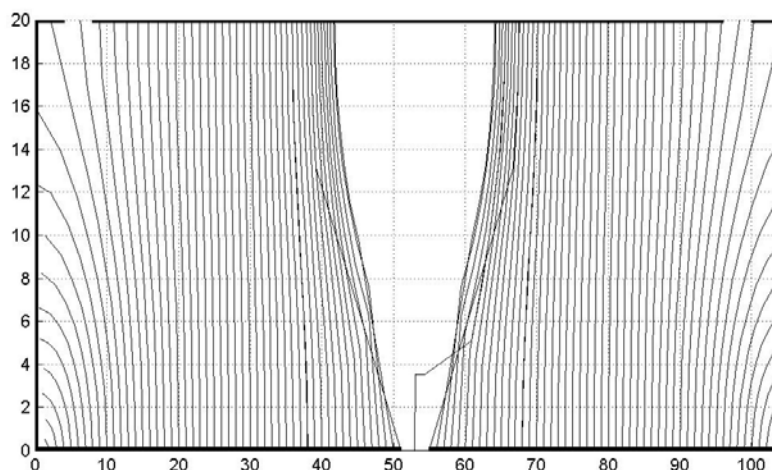


Рис. 2. Линии тока при скорости входного потока 1 м/с.

Таким образом, можно получить поле скоростей и построить линии тока для любых начальных и граничных условий, что даст возможность определить наиболее равномерное распределение теплоносителя в камерной сушилке.

Литература

1. Андерсон Д., Таннехилл Д., Плетчер Р. Вычислительная гидромеханика и теплообмен: В 2-х т. Т.2: Пер. с англ.— М.: Мир, 1990.— С.728–392 с.
2. Мэтьюз Джон Г., Финк, Куртис Д. Численные методы, использование MATLAB. 3-е издание; пер. с англ.— М.: Изд. дом «Вильямс», 2001.— 720с.
3. Потемкин В. Г. Система инженерных и научных расчетов MATLAB 5.X. В 2-х т. Т.1,2.— М.: Диалог-МИФИ, 2000.— 364 с.; 303 с.
4. Роуч П. Вычислительная гидродинамика.— М.: Наука, 1980.— 616 с.

УДК 519.86

ВЫЧИСЛЕНИЕ КОРРЕЛЯЦИОННОЙ РАЗМЕРНОСТИ ВРЕМЕННОГО РЯДА КОТИРОВОК ВАЛЮТ

Спирина А. Н.

*Белорусский государственный университет информатики
и радиоэлектроники, Минск, Белоруссия,
e-mail: spirina_anna@list.ru*

Введение

В работе предлагается подход к анализу поведения финансовых временных рядов, в частности, котировок валют, основанный на применении теории хаоса. Данный подход получает все большее распространение, основная причина которого — несостоятельность традиционной модели анализа рынков (гипотезы эффективного рынка) [1, 2].

Известно, что поведение системы может быть либо случайным, либо коррелированным, либо хаотическим [3]. Хаотическое поведение характеризуется тем, что выглядит как случайное, но таковым не является, оно существенно зависит от начальных условий и, как правило, ассоциировано со странным аттрактором.

Такое поведение часто встречается не только в природе [4], но и на рынках капитала [1–3]. Это свойство важно для прогнозирования поведения временного ряда. Известно, что если поведение системы является хаотическим, то, при определенных условиях, возможно краткосрочное прогнозирование ее поведения.

Один из подходов к определению свойств такого поведения — изучение аттрактора системы. В данной работе этот подход применяется к исследованию ряда котировок валют. Цель исследования — определить, является ли ряд хаотическим, какова корреляционная размерность и размерность вложения аттрактора.

Вычисление корреляционной размерности реальных данных методом Грасбергера и Прокаччи

Алгоритм Грасбергера и Прокаччи вычисления корреляционной размерности описан в различных источниках [3–5]. Он применяется в случаях, когда известны лишь экспериментальные данные, знание математической модели не обязательно. При использовании данного алгоритма необходимо оперировать большим объемом однотипных данных, а также иметь возможность просматривать эти данные на различных графиках. Система MATLAB была выбрана для реализации алгоритма Грасбергера и

Прокачки, т. к. она обладает всеми необходимыми для этого алгоритма возможностями.

Для анализа возьмем дневные котировки доллара США к японской йене за период с 1971г по 1979 включительно (сайт Федеральной Резервной Системы США [6]). Исключая нерабочие дни, имеется 2 241 котировка, минимальное значение — 177,05 , максимальное — 358,44. График котировок представлен на рисунке 1.

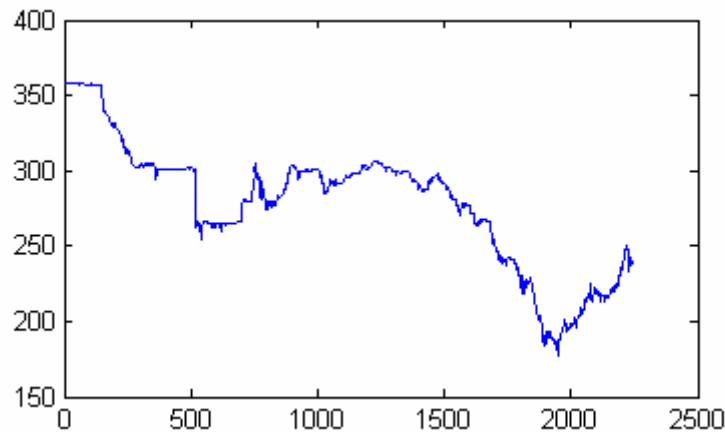


Рис. 1 Исходные данные: дневные котировки USD/JPY.

Вычислим корреляционные суммы по формулам:

$$C_m(R) = 1 / N_{pairs} \sum_{j=m}^N \sum_{k < j} \theta(R - |s_j - s_k|),$$

где s_j — вектор, полученный путем погружения исходного ряда в m -мерное пространство; m — размерность вложения, изменяется от 1 до 30; R — радиус, берутся 1000 значений в интервале от 0.01 до 180; $N_{pairs} = (N - (m - 1))(N - (m - 1) - 1) / 2$ — количество пар точек на восстановленном аттракторе, $N = 2241$ — количество данных во временном ряду, θ — функция Хэвисайда.

Погружение временного ряда в пространство большей размерности применяется благодаря теореме Такенса, которая утверждает, что если $N \rightarrow \infty$, и если m достаточно велико, то множество точек, полученных после погружения (восстановленный аттрактор), топологически эквивалентно аттрактору системы, породившей временной ряд, и имеет ту же корреляционную размерность.

Если аттрактор обладает фрактальной структурой, то для достаточно маленькой величины радиуса R должно выполняться соотношение

$$C_m(R) \propto R^{D_2}.$$

Из этой формулы следует

$$\text{Log}(C_m(R)) \propto D_2 \text{Log}(R).$$

Это значит, что график зависимости $C_m(R)$ от R , изображенный в двойной логарифмической шкале, должен быть прямой линией. Угол наклона этой прямой линии и является величиной D_2 . Если величина D_2 не зависит от m , то она является его корреляционной размерностью.

На рис. 2 представлены графики вычисленных корреляционных сумм исходного ряда относительно радиуса на двойной логарифмической шкале.

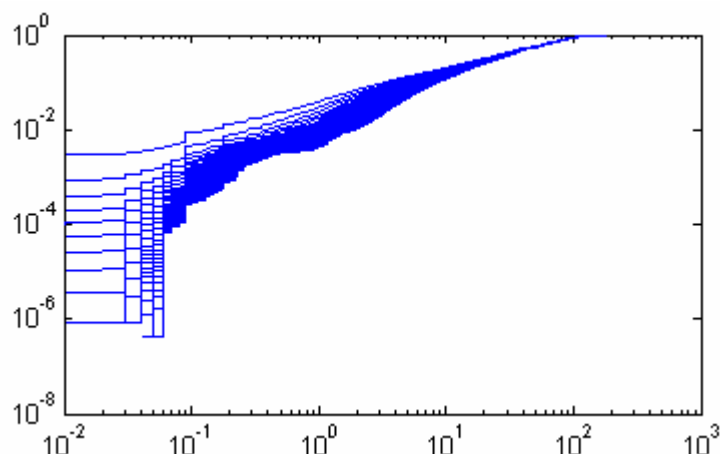


Рис. 2. Корреляционные суммы.

Верхний график соответствует размерности вложения 1, нижний — размерности 30.

Из рис. 2 видно, что графики не являются прямыми линиями. Причина несоответствия теории в том, что мы вычисления производились для конечного количества данных. Самоподобие нарушается, когда радиус стремится к нулю, а также когда он соизмерим с размером аттрактора. Еще одна причина отклонения от прямой линии при приближении радиуса к нулю — наличие в исходных данных шума [4].

Для определения D_2 необходимо найти интервал изменения радиуса, в котором графики линейны и имеют одинаковый наклон. С этой целью построены графики мгновенных наклонов корреляционных сумм. Они представлены на рис. 3.

При увеличении фрагмента этого графика видно, что отклонения от прямых линий становятся незначительными, начиная со значения на графике 0.2, т. е. с величины радиуса

$$10^{0.2} = 1.585.$$

Действительно, при увеличении графиков корреляционных сумм, видно, что, начиная с такой величины радиуса, исчезают колебания графиков (см. рис. 4).

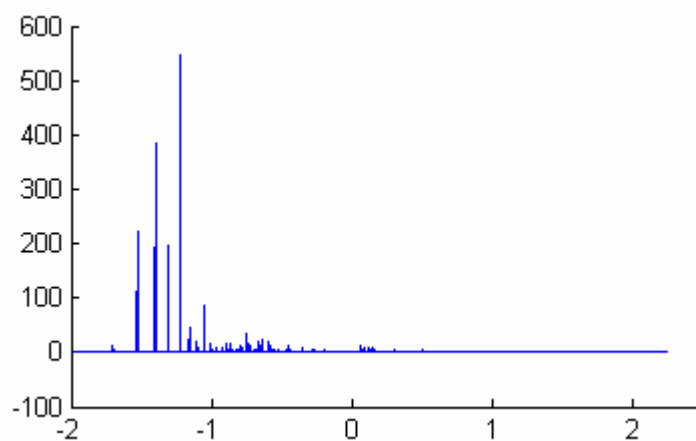


Рис. 3, а. Мгновенные наклоны корреляционных сумм.

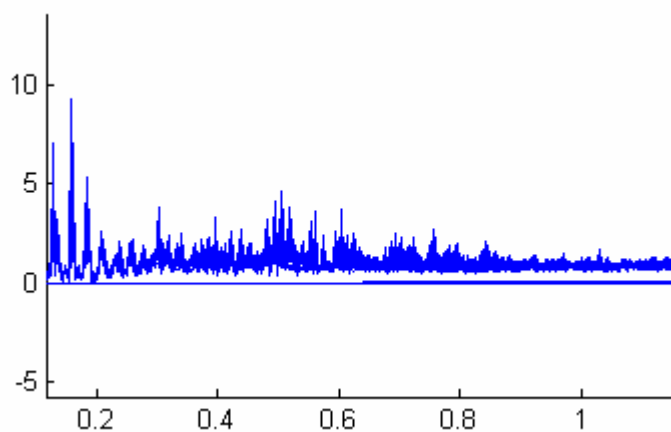


Рис. 3, б. Фрагмент в увеличении.

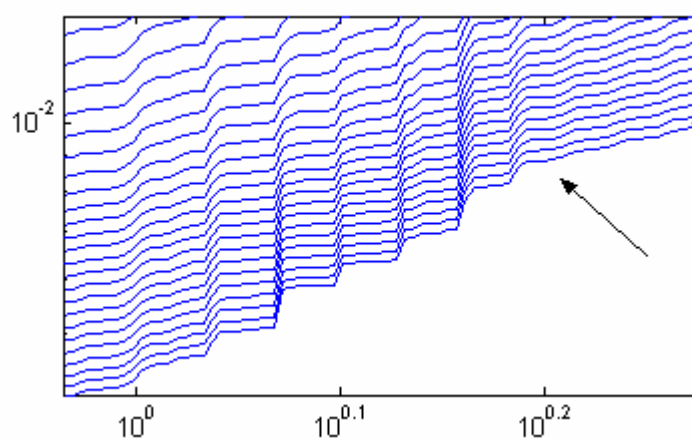


Рис. 4. Фрагмент графика корреляционных сумм.

Найдем участок, на котором эти графики не только линейны, но и имеют тенденцию к совпадению. Для этого опять же построим графики

наклонов, но не мгновенных (дабы избежать незначительных колебаний), а на участках охватывающих 10 значений радиуса (см. рис. 5).

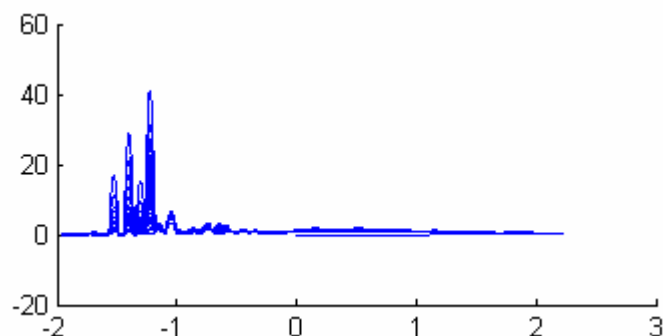


Рис. 5, а. Наклоны корреляционных сумм, охватывающие 10 значений.

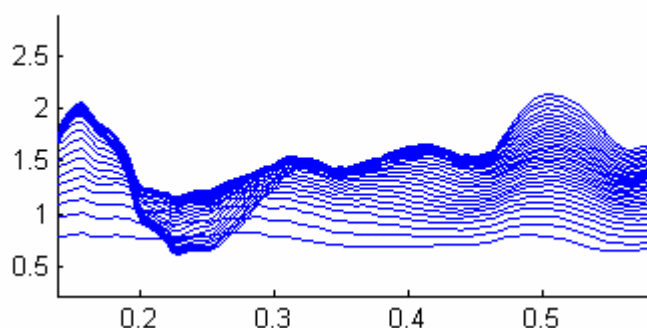


Рис. 5, б. Фрагмент в увеличении, непосредственно прилегающий к значению 0.2.

Из рис. 5, б видно, что, начиная с величины 0.2 (что соответствует радиусу 1.5849) и до величины 0.48 (радиус 3.02), с возрастанием размерности вложения графики имеют тенденцию к совпадению. Построим зависимость углов наклона линейных участков корреляционных сумм от размерности вложения для радиуса, находящегося в указанных пределах (рис. 6).

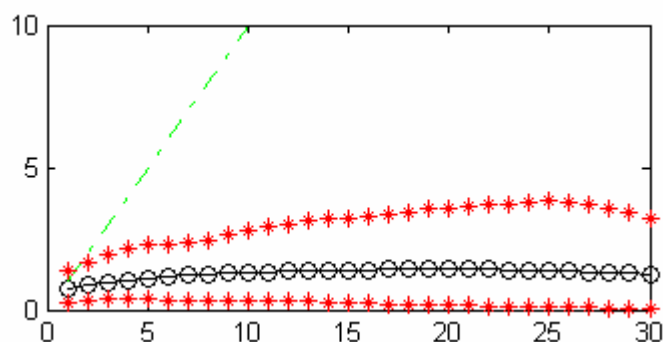


Рис. 6, а. Зависимость наклонов от размерности вложения.

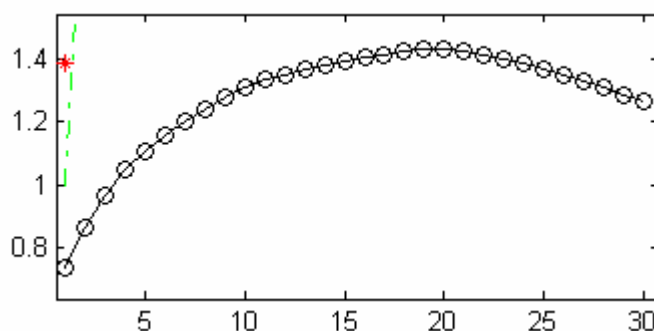


Рис. 6, б. Фрагмент в увеличении.

Звездочками отмечены минимальный и максимальный наклоны исходных графиков, они характеризуют степень несоответствия графика и прямой линии.

Если кривая наклонов близка к диагонали, то она соответствует случайному поведению. Если кривая достигает максимума и стабилизируется, то она соответствует хаотическому поведению [3]. Из рисунка 6б видно, что кривая наклонов достигает своего максимума при размерности вложения 19, этот максимум равен 1.4298.

Отсюда можно сделать следующие выводы. Исходный временной ряд является хаотическим. Корреляционная размерность его аттрактора составляет 1.4298, она достигается при размерности вложения 19. Максимальная разница между ошибками составляет 3.7086. Эти величины получены для радиуса, находящегося в интервале [1.5849 3.02], величина интервала составляет 0.79% от возможного изменения радиуса.

Сравнение результатов вычисления корреляционной размерности в зависимости от различных способов предобработки исходных данных

При вычислении корреляционной размерности в качестве исходных данных обычно берут не сами котировки, а котировки, предобработанные каким либо способом [2,3]. Исследуем влияние способов предобработки на численные характеристики аттрактора указанного выше временного ряда. В таблице 1 представлены результаты вычислений как для исходного ряда, так и для рядов, полученных из исходного различными способами предобработки.

Из табл. 1 видно, что предобработка данных может оказывать влияние на величину корреляционной размерности и размерности вложения.

Таблица 1.

Характеристики данных в зависимости от способа предобработки.

Данные	Корреляционная размерность	Размерность вложения	Макс. разница между мин. и макс. ошибками	Величина интервала	Интервал в % относительно всех данных.
Котировки	1.4298	19	3.7086	[1.5849 3.02]	0.79%
Детрендированные котировки	1.9999	17	3.8794	[0.8610 3.1623]	2.41%
Логарифмы котировок	1.409	19	4.5001	[0.01 0.0056]	0.63%
Детрендированные логарифмы котировок	1.9859	14	5.4072	[0.0025 0.0091]	1.55%
Детрендированные логарифмы котировок (2)	1.995	15	4.1602	[0.0038 0.0091]	1.25%

Заключение

Рассмотренный в работе временной ряд дневных котировок доллара США к японской йене за период с 1971г. по 1979г. является хаотическим. Его корреляционная размерность равна 1.4298, она достигается при размерности вложения 19. Поведение этого ряда можно прогнозировать, для получения последующего значения необходимо использовать 19 предыдущих значений. Это количество переменных, 19, можно уменьшить до 2 (ближайшее целое, большее корреляционной размерности), используя различные методы формирования пространства признаков меньшей размерности [7].

Описанная методика применения алгоритма Грасбергера и Прокаччи может быть применена для нахождения временных рядов, характеризующихся хаотическим поведением.

Предварительная обработка данных может оказывать влияние на величину корреляционной размерности и размерности вложения.

Литература

1. Петерс Э. Хаос и порядок на рынках капитала. Новый аналитический взгляд на цены, циклы и изменчивость рынка: Пер. с англ.— М.:Мир, 2000.
2. Дмитриева Л. А., Куперин Ю. А., Сорока И. В. Методы теории сложных систем в экономике и финансах // Докл. Всероссийской науч.-метод. конф. «Междисциплинарность в науке и образовании».— 2001. (<http://is2001.icape.ru>).

3. Trading on the edge: neural, genetic and fuzzy systems for chaotic financial markets. Edited by Guido J. Deboeck.— Wiley Finance editions, 1994.
4. *Шустер Г.* Детерминированный хаос.— М.:Мир, 1988.
5. *Hegger R., Kantz H., Schreiber Th.* Practical implementation of nonlinear time series methods: The TISEAN package // CHAOS.— 1999.— № 9.— P. 413–435.
6. <http://www.federalreserve.gov/>
7. *Ежов А., Шумский С.* Нейрокомпьютинг и его применение в экономике и бизнесе. Курс лекций для студентов финансово-экономического колледжа МИФИ. (<http://soft.neurok.ru>).

УДК 681.511.22

ОТЫСКАНИЕ ПЕРИОДИЧЕСКИХ РЕШЕНИЙ ОДНОЙ ДИСКРЕТНОЙ СИСТЕМЫ С ГИСТЕРЕЗИСОМ

Степанов А. В.

Санкт-Петербургский государственный университет, Санкт-Петербург,
e-mail: stepan-alex@yandex.ru

Рассмотрим следующую нелинейную систему разностных уравнений:

$$x_{k+1} = Mx_k + p_1 \cos[\pi k / N] + p_2 \sin[\pi k / N] + qu_k, \quad (1)$$

здесь $k \geq k_0$, $p_{1,2}, q \in E^n$, $\|p_1\| + \|p_2\| \neq 0$, $\|q\| \neq 0$, M — вещественная матрица размерности $n \times n$, модули собственных чисел матрицы M отличны от единицы, $N > 1$ — натуральное число, управляющее воздействие вычисляется по формуле:

$$u_k = \begin{cases} m_1, & \gamma^T x_k \leq l_2, \\ m_2, & \gamma^T x_k \geq l_1, \end{cases}$$

$\gamma \in E^n$, $\|\gamma\| \neq 0$, $l_1 < l_2$, $m_1 < m_2$.

Дискретные нелинейные системы возникают во многих задачах управления [1, 2]. Системы вида (1) можно рассматривать как результат дискретизации непрерывных нелинейных систем управления с периодическим внешним воздействием [3]. Использование математических пакетов при изучении таких систем оказывает существенную помощь исследователю, так как применение качественных методов исследования часто затруднено, в связи с необходимостью проводить громоздкие выкладки. Рассмотрим вопрос о существовании периодических решений системы (1).

Решение системы (1) будем считать «склеенным» из кусков решений линейных систем вида

$$x_{k+1} = Mx_k + p_1 \cos[\pi k / N] + p_2 \sin[\pi k / N] + qm, \quad m = m_{1,2}.$$

Точки переключения управляющего воздействия далее будем называть просто точками переключения. Обозначим

$$P = (E - 2 \cos[\pi / N]M + M^2)^{-1},$$

$$\tilde{p}_1 = Mp_1 - \cos[\pi / N]p_1 + \sin[\pi / N]p_2,$$

$$\tilde{p}_2 = Mp_2 - \sin[\pi / N]p_1 - \cos[\pi / N]p_2.$$

Последняя система имеет решение вида:

$$x[k, k_0, x_0, m] = M^{k-k_0}(x_0 - w[k_0, m]) + w[k, m], \quad (2)$$

где

$$\begin{aligned} w[k, m] &= -P[(\cos[\pi k / N]M - \cos[\pi(k-1) / N]E)p_1 + \\ &+ (\sin[\pi k / N]M - \sin[\pi(k-1) / N]E)p_2] - (M - E)^{-1}qm = \\ &= -P(\tilde{p}_1 \cos[\pi k / N] + \tilde{p}_2 \sin[\pi k / N]) - (M - E)^{-1}qm, \end{aligned}$$

причем $x[k_0, k_0, x_0, m] = x_0$.

Теорема 1. Для того, чтобы система (1) имела $2N$ -периодическое решение с двумя точками переключения $x_{1,2}$, необходимо и достаточно, чтобы для некоторых двух целых чисел $0 \leq k_0 < 2N$, $1 \leq K < 2N$ были выполнены условия:

$$\left\{ \begin{array}{l} \gamma^T x_1 < l_1, \\ \gamma^T x_2 > l_2, \\ \gamma^T x[k, k_0, x_1, m_1] \leq l_2, \quad k = \overline{k_0 + 1, k_0 + K - 1}, \\ \gamma^T x[k, k_0 + K, x_2, m_2] \geq l_1, \quad k = \overline{k_0 + K + 1, k_0 + 2N - 1}, \end{array} \right. \quad (3)$$

где

$$\left\{ \begin{array}{l} x_1 = x_1[k_0, K] = (M^{2N} - E)^{-1}(M^{2N-K} - E)(M - E)^{-1}q(m_1 - m_2) + w[k_0, m_1], \\ x_2 = x_2[k_0, K] = (M^{2N} - E)^{-1}(M^K - E)(M - E)^{-1}q(m_2 - m_1) + w[k_0 + K, m_2]. \end{array} \right.$$

Доказательство. Покажем, что равенства

$$\left\{ \begin{array}{l} x_2 = x[k_0 + K, k_0, x_1, m_1], \\ x_1 = x[k_0 + 2N, k_0 + K, x_2, m_2] \end{array} \right.$$

имеют место тогда и только тогда, когда $x_{1,2}$ вычисляются по формулам, приведенным в формулировке, тогда сформулированное утверждение будет очевидным. Вводя обозначения

$$U = \begin{pmatrix} E & -M^{2N-K} \\ -M^K & E \end{pmatrix}, \quad v = \begin{pmatrix} -M^{2N-K}w[k_0 + K, m_2] + w[k_0 + 2N, m_2] \\ -M^K w[k_0, m_1] + w[k_0 + K, m_1] \end{pmatrix},$$

перепишем последнюю систему в следующем виде: $U \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = v$. Т. к. моду-

ли собственных чисел матрицы M отличны от единицы, то $\det U \neq 0$. Следовательно,

$$\left\{ \begin{array}{l} x_2 = x[k_0 + K, k_0, x_1, m_1], \\ x_1 = x[k_0 + 2N, k_0 + K, x_2, m_2] \end{array} \right. \Leftrightarrow \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = U^{-1}v.$$

Замечая, что $w[k_0 + 2N, m] \equiv w[k_0, m]$, имеем:

$$U^{-1}v = \begin{pmatrix} (M^{2N} - E)^{-1}(M^{2N-K} - E)(M - E)^{-1}q(m_1 - m_2) + w[k_0, m_1] \\ (M^{2N} - E)^{-1}(M^K - E)(M - E)^{-1}q(m_2 - m_1) + w[k_0 + K, m_2] \end{pmatrix}.$$

Теорема доказана.

Замечание 1. В формулировке теоремы не указано, как найти значения k_0, K . Для того, чтобы проверить, имеет ли система (1) $2N$ -периодическое решение с двумя точками переключения, достаточно проверить условия (3) перебором всевозможных пар (k_0, K) , $0 \leq k_0 < 2N$, $1 \leq K < 2N$.

Замечание 2. Если модули собственных чисел матрицы M меньше единицы, и для некоторого числа $\varepsilon > 0$ выполнены условия:

$$\left\{ \begin{array}{l} \gamma^T x_1 < l_1 - \varepsilon, \\ \gamma^T x_2 > l_2 + \varepsilon, \\ \gamma^T x[k, k_0, x_1, m_1] < l_2 - \varepsilon, \quad k = \overline{k_0 + 1, k_0 + K - 1}, \\ \gamma^T x[k, k_0 + K, x_2, m_2] > l_1 + \varepsilon, \quad k = \overline{k_0 + K + 1, k_0 + 2N - 1}, \end{array} \right. \quad (4)$$

то система (1) имеет асимптотически устойчивое $2N$ -периодическое решение с точками переключения $x_{1,2}$.

Пример. Пусть

$$M = \begin{pmatrix} -0.7 & 0 \\ 0 & 0.9 \end{pmatrix}, \quad p_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad p_2 = \begin{pmatrix} 0 \\ 1.5 \end{pmatrix}, \quad q = \begin{pmatrix} -1 \\ 0.5 \end{pmatrix}, \quad \gamma = \begin{pmatrix} -8 \\ -1 \end{pmatrix},$$

$$N = 4, \quad m_1 = -1, \quad m_2 = 0.6, \quad l_1 = -0.2, \quad l_2 = 0.1.$$

Тогда для каждой из пар $k_0 = 0, K = 4$, $k_0 = 7, K = 4$, выполнены условия (3), и система (1) имеет периодическое решение. Графики дискретных траекторий этих решений приведены на рис. 1.

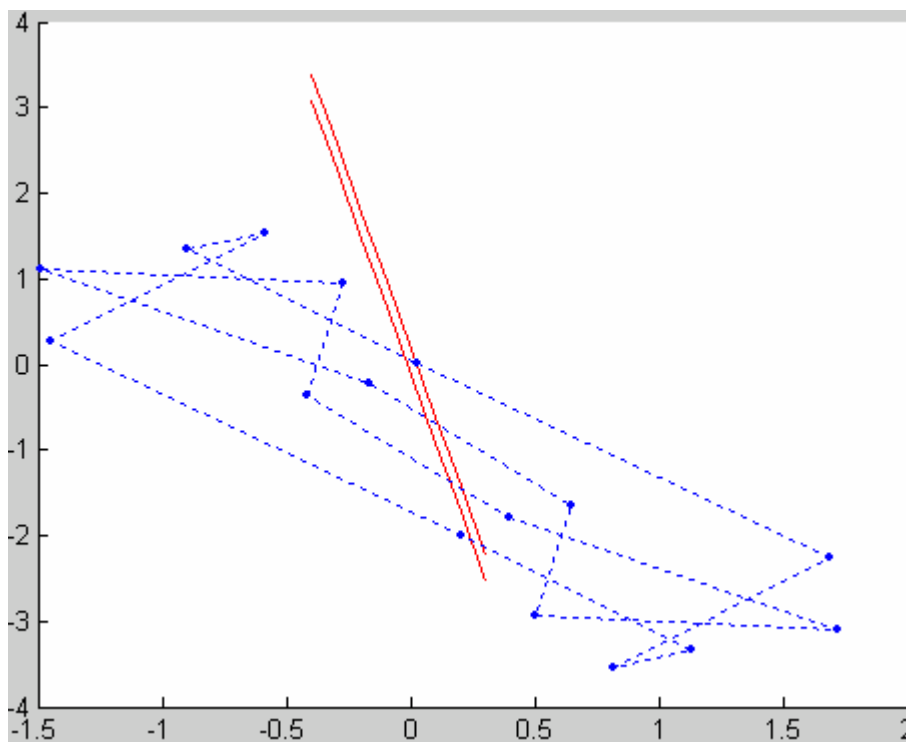


Рис. 1.

В папке «example 1» в разделе «Разработки» находится MATLAB-приложение, осуществляющее поиск $2N$ -периодических решений системы (1) в случае $n = 2$ проверкой условия (3) для всевозможных пар (k_0, K) , $0 \leq k_0 < 2N$, $1 \leq K < 2N$.

Получим достаточные условия существования $2N$ -периодического решения системы (1) в одном частном случае.

Введем обозначения:

$$\begin{aligned} A_1 &= \|\tilde{p}_1\| \|((1 - 2\cos[\pi/N])E + M)P\tilde{p}_1\|, \\ A_2 &= \|\tilde{p}_2\| \|((1 - 2\cos[\pi/N])E + M)P\tilde{p}_2\|, \\ A_3 &= \frac{1 - \|M\|^N}{(1 - \|M\|) \|M^N + E\|}, \\ \alpha_1 &= -\gamma^T P\tilde{p}_1, \quad \alpha_2 = -\gamma^T P\tilde{p}_2, \\ \varphi &= \begin{cases} \arctg(\alpha_2 / \alpha_1) + \pi, & \alpha_1 < 0, \\ \arctg(\alpha_2 / \alpha_1), & \alpha_1 > 0, \\ 3\pi/2, & \alpha_1 = 0, \alpha_2 < 0, \\ \pi/2, & \alpha_1 = 0, \alpha_2 > 0, \end{cases} \\ k_0 &= \lceil N/\pi(\varphi + \pi/2) \rceil + 1, \\ K_1 &= 2(A_1 + A_2), \\ K_2 &= (2 - 2\cos[\pi/N])^{-2} \max \left\{ \begin{aligned} &(A_3 + 1)^2 \cos^{-2}[\pi k_0 / N - \varphi], \\ &A_3^2 \cos^{-2}[\pi(k_0 + N - 1) / N - \varphi] \end{aligned} \right\}. \end{aligned}$$

Теорема 2. Пусть модули собственных чисел матрицы M меньше единицы, пусть $m_1 = -1$, $m_2 = 1$, $\|\gamma\| = 1$. Пусть для некоторых вещественных чисел $\varepsilon > 0$, $\delta > 2\varepsilon$ выполнены неравенства:

$$\begin{aligned} \|M - E\| &\leq \delta, \quad \|q\| \leq \delta, \quad l_1 \in (-\delta + \varepsilon, -\varepsilon), \quad l_2 \in (\varepsilon, \delta - \varepsilon), \\ (\gamma^T \tilde{p}_1)^2 + (\gamma^T \tilde{p}_2)^2 &> K_1 \delta + K_2 \delta^2, \end{aligned}$$

тогда система (1) имеет асимптотически устойчивое $2N$ -периодическое решение с точками переключения

$$\begin{cases} x_1 = -2(M^N + E)^{-1}(M - E)^{-1}q + w[k_0, -1], \\ x_2 = -x_1. \end{cases}$$

Доказательство. Положив $K = N$ в формулах из предыдущей теоремы, получим выражения для $x_{1,2}$, указанные в формулировке. Далее,

$$\begin{aligned} x[k, k_0, x_1, -1] &= (M^N - 2M^{k-k_0} + E)(M^N + E)^{-1}(M - E)^{-1}q - \\ &\quad - P(\tilde{p}_1 \cos[\pi k / N] + \tilde{p}_2 \sin[\pi k / N]). \end{aligned}$$

Следовательно,

$$\gamma^T x[k, k_0, x_1, -1] = \phi_1[k] + \phi_2[k],$$

$$\phi_1[k] = \alpha_1 \cos[\pi k / N] + \alpha_2 \sin[\pi k / N] = \sqrt{\alpha_1^2 + \alpha_2^2} \cos[\pi k / N - \varphi],$$

$$\phi_2[k] = \gamma^T (M^N - 2M^{k-k_0} + E)(M^N + E)^{-1}(M - E)^{-1}q.$$

Для функции $\phi_2[k]$ справедлива оценка: $|\phi_2[k]| \leq A_3 \delta$.

Рассмотрим функцию $\phi_1[k]$. Обозначим

$$P_1 = 2(1 - \cos[\pi / N])E, \quad P_2 = (M - E)(E - 2\cos[\pi / N]E + M),$$

тогда

$$\begin{aligned} P &= (P_1 + P_2)^{-1} = (E + P_1^{-1}P_2)^{-1}P_1^{-1} = P_1^{-1} + ((E + P_1^{-1}P_2)^{-1} - E)P_1^{-1} = \\ &= P_1^{-1} + (E - E - P_1^{-1}P_2)(E + P_1^{-1}P_2)^{-1}P_1^{-1} = P_1^{-1} - P_1^{-1}P_2P_1^{-1} = P_1^{-1} - P_2P_1^{-1}P_1^{-1}, \\ \alpha_1^2 &= (\gamma^T P \tilde{p}_1)^2 = (\gamma^T (P_1 + P_2)^{-1} \tilde{p}_1)^2 = (\gamma^T P_1^{-1} \tilde{p}_1 - \gamma^T P_2 P_1^{-1} P \tilde{p}_1)^2 \geq \\ &\geq (\gamma^T P_1^{-1} \tilde{p}_1)^2 - 2|\gamma^T P_1^{-1} \tilde{p}_1| |\gamma^T P_2 P_1^{-1} P \tilde{p}_1| \geq \\ &\geq (\gamma^T P_1^{-1} \tilde{p}_1)^2 - 2\delta \|P_1^{-1} \tilde{p}_1\| \|((1 - 2\cos[\pi / N])E + M)P_1^{-1} P \tilde{p}_1\| = \\ &= (2 - 2\cos[\pi / N])^{-2} ((\gamma^T \tilde{p}_1)^2 - 2\delta A_1). \end{aligned}$$

Аналогично,

$$\alpha_2^2 \geq (2 - 2\cos[\pi / N])^{-2} ((\gamma^T \tilde{p}_2)^2 - 2\delta A_2),$$

и

$$\alpha_1^2 + \alpha_2^2 \geq (2 - 2\cos[\pi / N])^{-2} ((\gamma^T \tilde{p}_1)^2 + (\gamma^T \tilde{p}_2)^2 - 2\delta(A_1 + A_2)).$$

Далее,

$$\begin{aligned} &(\gamma^T \tilde{p}_1)^2 + (\gamma^T \tilde{p}_2)^2 > K_1 \delta + K_2 \delta^2, \Rightarrow \\ &(\gamma^T \tilde{p}_1)^2 + (\gamma^T \tilde{p}_2)^2 > 2\delta(A_1 + A_2) + \delta^2(2 - 2\cos[\pi / N])^2 \times \\ &\times \max\{(1 + A_3)^2 \cos^{-2}[\pi k_0 / N - \varphi], A_3^2 \cos^{-2}[\pi(k_0 + N - 1) / N - \varphi]\}, \Rightarrow \\ &\alpha_1^2 + \alpha_2^2 \geq (2 - 2\cos[\pi / N])^{-2} ((\gamma^T \tilde{p}_1)^2 + (\gamma^T \tilde{p}_2)^2 - 2\delta(A_1 + A_2)) > \\ &> \delta^2 \max\{(1 + A_3)^2 \cos^{-2}[\pi k_0 / N - \varphi], A_3^2 \cos^{-2}[\pi(k_0 + N - 1) / N - \varphi]\}, \Rightarrow \\ &\begin{cases} \sqrt{\alpha_1^2 + \alpha_2^2} > -(1 + A_3)\delta \cos^{-1}[\pi k_0 / N - \varphi], \\ \sqrt{\alpha_1^2 + \alpha_2^2} > -A_3\delta \cos^{-1}[\pi(k_0 + N - 1) / N - \varphi], \end{cases} \Rightarrow \\ &\begin{cases} \phi_1[k_0] < -\delta - A_3\delta, \\ \phi_1[k_0 + N - 1] < -A_3\delta, \end{cases} \Rightarrow \\ &\begin{cases} \phi_1[k_0] < -\delta - |\phi_2[k_0]|, \\ \phi_1[k_0 + N - 1] < -|\phi_2[k_0 + N - 1]|, \end{cases} \Rightarrow \end{aligned}$$

$$\begin{aligned}
 & \left\{ \begin{array}{l} \phi_1[k_0] + \phi_2[k_0] < -\delta, \\ \phi_1[k_0 + N - 1] + \phi_2[k_0 + N - 1] < 0, \end{array} \right. \Rightarrow \\
 & \left\{ \begin{array}{l} \gamma^T x_1 < -\delta, \\ \gamma^T x[k_0 + N - 1, k_0, x_1, -1] < 0, \end{array} \right. \Rightarrow \\
 & \left\{ \begin{array}{l} \gamma^T x_1 < -\delta, \\ \gamma^T x_2 = -\gamma^T x_1 > -\delta, \\ \gamma^T x[k, k_0, x_1, -1] < 0, \quad k_0 + 1 \leq k \leq k_0 + N - 1, \\ \gamma^T x[k + N, k_0 + N, x_2, 1] = -\gamma^T x[k, k_0, x_1, -1] > 0, \quad k_0 + 1 \leq k \leq k_0 + N - 1, \end{array} \right. \Rightarrow \\
 & \left\{ \begin{array}{l} \gamma^T x_1 < l_1 - \varepsilon, \\ \gamma^T x_2 > l_2 + \varepsilon, \\ \gamma^T x[k, k_0, x_1, -1] < l_2 - \varepsilon, \quad k = \overline{k_0 + 1, k_0 + K - 1}, \\ \gamma^T x[k, k_0 + K, x_2, 1] > l_1 + \varepsilon, \quad k = \overline{k_0 + K + 1, k_0 + 2N - 1}. \end{array} \right.
 \end{aligned}$$

Следовательно, условие (4) выполнено. Теорема доказана.

Замечание 1. Условия $\|M - E\| \leq \delta$, $\|q\| \leq \delta$ будут выполнены, если система (1) получена в результате дискретизации с некоторым достаточно малым шагом некоторой непрерывной системы управления.

Замечание 2. В теореме 2 рассматриваются условия, при выполнении которых управляющее воздействие оказывает существенно меньшее влияние на характер периодического решения системы (1), чем периодическое воздействие, которое находится в правой части этой системы. В силу этого система (1) имеет $2N$ -периодическое решение с двумя точками переключения, которое фактически представляет собой возмущенное $2N$ -периодическое решение линейной системы

$$x_{k+1} = Mx_k + p_1 \cos[\pi k / N] + p_2 \sin[\pi k / N].$$

Пример. Пусть

$$M = \begin{pmatrix} 0.9885 & -0.0083 \\ -0.0083 & 0.9885 \end{pmatrix}, \quad p_1 = \begin{pmatrix} 0.175 \\ 0.1 \end{pmatrix}, \quad p_2 = \begin{pmatrix} -0.1 \\ 0.1 \end{pmatrix}, \quad q = \begin{pmatrix} 0.0099 \\ 0.0198 \end{pmatrix}, \quad \gamma = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix},$$

$$N = 3, \quad l_1 = -0.02, \quad l_2 = 0.01.$$

Можно проверить, что выполнены все условия теоремы 2 (вычисления проводились в среде MATLAB, см. папку «example 2»), и система (1) имеет 6-периодическое решение, график дискретной траектории которого приведен на рис. 2.

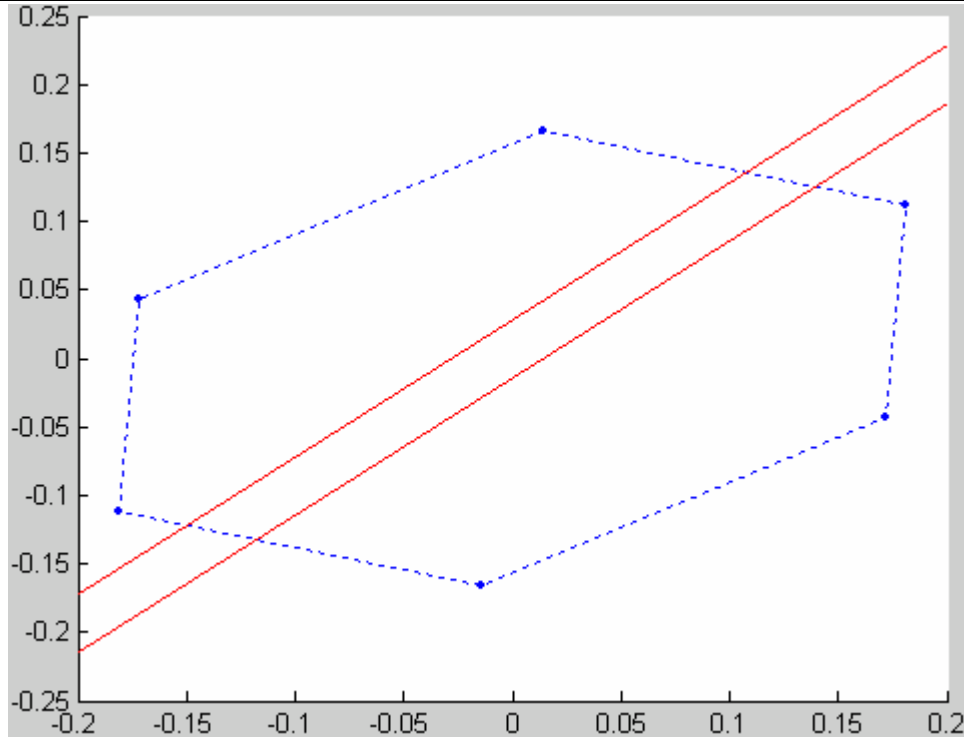


Рис. 2.

Рассмотрим теперь случай, когда управляющее воздействие оказывает существенное влияние на динамику системы (1). Рассмотрим сначала автономную систему:

$$x_{k+1} = Mx_k + qu_k. \quad (5)$$

Если $\det(M - E) \neq 0$, то вектор-функция

$$x[k, k_0, x_0, m] = x[k - k_0, x_0, m] = M^{k-k_0}(x_0 + (M - E)^{-1}qm) - (M - E)^{-1}qm$$

является решением системы (5), и $x[0, x_0, m] = x_0$. Наложим на параметры системы (5) дополнительные ограничения. Будем считать, что матрица M приведена к жордановой форме, и модули ее собственных чисел меньше единицы. Пусть

$$m_1 = -1, \quad m_2 = 1, \quad l_1 = -l, \quad l_2 = l, \quad l > 0, \quad \|\gamma\| = 1,$$

$$M = \text{diag}(M_1, M_2), \quad M_1 = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m), \quad 0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m < 1, \quad m \leq n,$$

$$\gamma = (\gamma_1, \gamma_2, \dots, \gamma_m, 0, \dots, 0), \quad p = (p_1, p_2, \dots, p_m, 0, \dots, 0)^T, \quad (6)$$

$$\gamma_i q_i \leq 0, \quad i = \overline{1, m}, \quad \gamma^T q < 0,$$

$$\gamma^T (M - E)^{-1} q = -\sum_{i=1}^m \frac{\gamma_i p_i}{1 - \lambda_i} > C \cdot l, \quad C > 1.$$

Заметим, что условия, аналогичные (6), рассматривались в работе [4].

Введем обозначения:

$$x_a[k] = (M^k + E)^{-1}(M^k - E)(M - E)^{-1}q,$$

$$\phi_1[k] = \gamma^T x_a[k] = \sum_{i=1}^m \frac{\gamma_i q_i}{1 - \lambda_i} \cdot \frac{1 - \lambda_i^k}{1 + \lambda_i^k},$$

$$\phi_2[k] = \gamma^T x[k, x_a[T], -1].$$

Лемма. Пусть выполнены условия (6), пусть для некоторого вещественного $\delta > 0$ существует $T \in N$ такое, что:

$$\phi_1[T] < -l - \delta, \quad (7)$$

$$\gamma^T M^{-1}(x_a[T] - q) > -l + \delta, \quad (8)$$

тогда система (5) имеет асимптотически устойчивое $2T$ -периодическое решение с точками переключения $\pm x_a[T]$.

Доказательство. Легко проверить, что

$$x[T, x_a[T], -1] = -x_a[T], \quad x[T, -x_a[T], 1] = x_a[T].$$

В силу условий (6) функция $\phi_1[k]$ монотонно убывает с ростом k , и неравенство (7) выполнено для некоторых значений T, δ . Следовательно,

$$\gamma^T x_a[T] < l - \delta, \quad \gamma^T (-x_a[T]) > -l + \delta.$$

Функция $\phi_2[k]$ монотонно возрастает с ростом значения k :

$$\begin{aligned} \Delta\phi_2[k] &= \phi_2[k+1] - \phi_2[k] = \gamma^T M^k (M - E)(x_a[T] - (M - E)^{-1}q) = \\ &= \gamma^T M^k (M^k + E)^{-1} (M^k - E)q - \gamma^T M^k q = -2 \sum_{i=1}^m \frac{\lambda_i^k}{1 + \lambda_i^k} \cdot \gamma_i q_i > 0. \end{aligned}$$

В силу условия (8),

$$\phi_2[T-1] = \gamma^T x[T-1, x_a[T], -1] = -\gamma^T M^{-1}(x_a[T] - q) < l - \delta,$$

откуда следует:

$$\gamma^T x[k, x_a[T], -1] = \phi_2[k] \leq \phi_2[T-1] < l - \delta, \quad k = \overline{1, T-2}.$$

Далее,

$$\gamma^T x[k, -x_a[T], 1] = -\phi_2[k] > -l + \delta, \quad k = \overline{1, T-1}.$$

Следовательно, система (5) имеет асимптотически устойчивое $2T$ -периодическое решение, и множество его точек имеет непустую δ -окрестность, пересечение которой с поверхностями переключения $\gamma^T x = l_{1,2}$ пусто. Доказательство закончено.

Вернемся к системе (1). Пусть выполнены условия (6)–(8), т. е. система (5) имеет асимптотически устойчивое $2T$ -периодическое решение с двумя точками переключения, выберем $N > T$, пусть выполнено условие:

$$\|P\tilde{p}_1\| + \|P\tilde{p}_2\| < \frac{1 - \|M\|}{1 + \|M\|} K\delta, \quad K < 1, \quad (9)$$

тогда решение системы (1), начинающееся в $K\delta$ -окрестности указанного $2T$ -периодического решения системы (5), не покидает этой окрестности с ростом k . Действительно, пусть $\|\tilde{x}_0 - x_a[T]\| \leq K\delta$, обозначим $\tilde{x}[k, k_0, \tilde{x}_0, -1]$ решение системы (1), $x[k, k_0, x_a[T], -1]$ — решение системы (5). Тогда

$$\begin{aligned}\tilde{x}[k, k_0, \tilde{x}_0, -1] &= x[k, k_0, x_a[T], -1] + M^{k-k_0}(\tilde{x}_0 - x_a[T]) - M^{k-k_0}\delta w[k_0] + \delta w[k], \\ \delta w[k] &= -P(\tilde{p}_1 \cos[\pi k / N] + \tilde{p}_2 \sin[\pi k / N]), \\ |\gamma^T(\tilde{x}[k, k_0, \tilde{x}_0, m] - x[k, k_0, x_0, m])| &= |\gamma^T(M^{k-k_0}(\tilde{x}_0 - x_0) - M^{k-k_0}\delta w[k_0] + \delta w[k])| \leq \\ &\leq \|M\| \|\tilde{x}_0 - x_0\| + (1 + \|M\|)(\|P\tilde{p}_1\| + \|P\tilde{p}_2\|) < K\delta, \quad k - k_0 \leq T.\end{aligned}$$

Аналогичные рассуждения можно провести для $x_0 = -x_a[T]$.

Естественным образом возникает вопрос о существовании периодического решения системы (1) в указанной окрестности. Будем искать решение, период которого равен $L = \text{НОК}(2N, 2T)$.

В силу сделанного предположения (9), L -периодическое решение, если оно существует, имеет $\nu = L/T$ точек переключения x_1, \dots, x_ν . Предположим, что в некоторый момент времени k_0 , $0 \leq k_0 \leq L$, мы находимся одной из точек переключения $x_1 = x_1(k_0)$, $\gamma^T x_1(k_0) < -l$. Имеем:

$$\begin{aligned}x_2 &= M^T(x_1 - w[k_0, -1]) + w[k_0 + T, -1], \\ x_3 &= M^{2T}(x_1 - w[k_0, -1]) + M^T(w[k_0 + T, -1] - w[k_0 + T, 1]) + w[k_0 + 2T, -1], \\ &\dots\end{aligned}$$

учитывая, что $w[k, -1] - w[k, 1] = 2(M - E)^{-1}q$, можем выписать следующую формулу для точек переключения:

$$x_{i+1} = M^{iT}(x_1 - w[k_0, -1]) + 2M^T \sum_{j=1}^{i-1} (-1)^{j-1} M^{(j-1)T} (M - E)^{-1}q + w[k_0 + iT, (-1)^i].$$

Для $i = \nu$ имеем:

$$x_{\nu+1} = M^L(x_1 - w[k_0, -1]) + 2M^T \sum_{j=1}^{\nu-1} (-1)^{j-1} M^{(j-1)T} (M - E)^{-1}q + w[k_0, 1] \quad (10)$$

(воспользовались равенством: $w[k_0 + L, (-1)^\nu] = w[k_0, 1]$). Т. к. функция $w[k_0, m]$, $m = \pm 1$, является L -периодической по аргументу k_0 , то тем самым для данного фиксированного k_0 имеем непрерывное отображение замыкания $K\delta$ -окрестности точки $x_a[T]$ в себя. Следовательно, в силу теоремы Брауэра о существовании неподвижной точки, отображение, задаваемое формулой (10), имеет неподвижную точку, т. е. для данного k_0 существует периодическое решение системы (1), одной из точек переключения которого будет точка

$$x_1[k_0] = (M^L - E)^{-1} \left(M^L w[k_0, -1] - w[k_0, 1] - 2M^T \sum_{j=1}^{\nu-1} (-1)^{(j-1)} M^{(j-1)T} (M - E)^{-1}q \right).$$

Таким образом, можем сформулировать окончательный результат:

Теорема 3. Пусть выполнены условия (6)–(9), тогда система (1) для любого $k_0 \geq 0$ имеет асимптотически устойчивое L -периодическое решение с L/T точками переключения, $L = \text{НОК}(2N, 2T)$.

Пример. Пусть

$$M = \begin{pmatrix} 0.4 & 0 \\ 0 & 0.5 \end{pmatrix}, \quad p_1 = \begin{pmatrix} -0.02 \\ 0.01 \end{pmatrix}, \quad p_2 = \begin{pmatrix} -0.1 \\ 0.03 \end{pmatrix}, \quad q = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad \gamma = \begin{pmatrix} -1/\sqrt{65} \\ -8/\sqrt{65} \end{pmatrix},$$

$$N=3, \quad l=1, \quad k_0=0.$$

Все условия теоремы 3 выполнены (см. «example 3»), и система (1) имеет 6-периодическое решение, график дискретной траектории которого приведен на рис. 3.

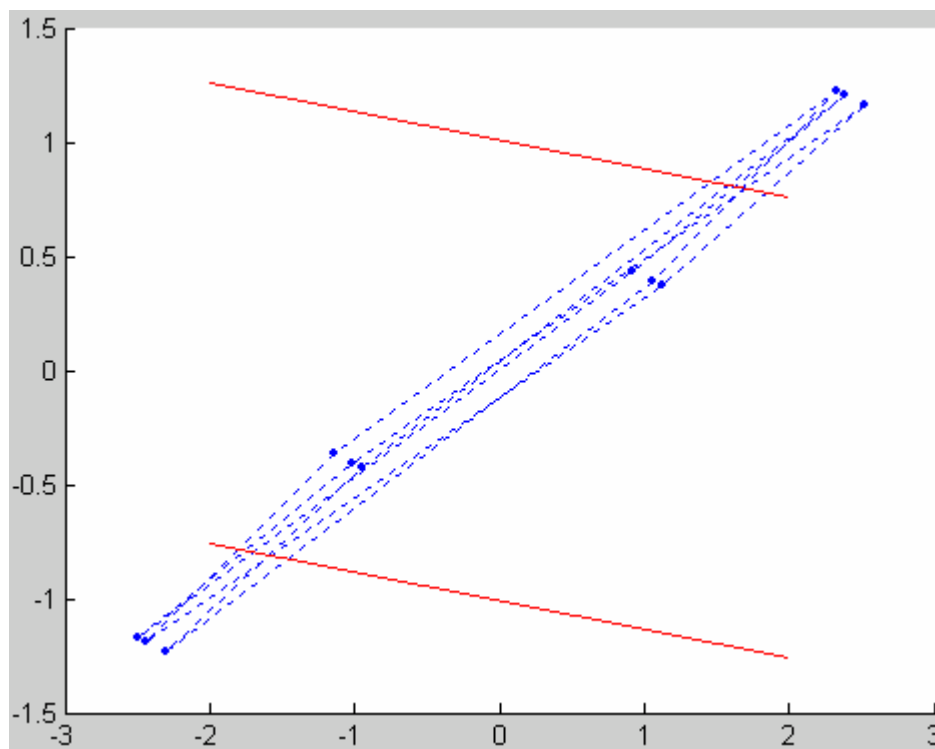


Рис. 3.

Литература

1. Цыпкин Я. З., Попков Ю. С. Теория нелинейных импульсных систем.— М.: Наука, 1973.— 414 с.
2. Косякин А.А., Шамриков Б.М. Колебания в цифровых автоматических системах.— М.: Наука, 1983.— 334 с.
3. Камачкин А. М, Шамберов В. Н. Отыскание периодических решений в нелинейных динамических системах.— СПб.: изд-во СПбГУ, 2002.— 88 с.
4. Зубов С. В, Зубов Н. В. Математические методы стабилизации динамических систем.— СПб.: Изд-во СПбГУ, 1996.— 288 с.

УДК 683:519

ИДЕНТИФИКАЦИЯ КРИТЕРИАЛЬНОЙ ФУНКЦИИ НА БАЗЕ САМООРГАНИЗАЦИИ МОДЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ СИМВОЛЬНОЙ МАТЕМАТИКИ В СРЕДЕ MATLAB

*Тимченко А. А., Снитюк В. Е., Оксамытная Л. П., Ланских Е. П.
Черкасский государственный технологический университет, Черкассы,
e-mail:svit@majar.com*

Введение

Современные интеграционные тенденции, включающие гуманизацию человеческой деятельности, стремление к построению открытого общества, глобализацию и взаимозависимость экономик разных стран создали предпосылки для увеличения начальной энтропии процессов проектирования сложных технических систем. Известны два подхода к проектированию, наиболее распространенные в мире: первый базируется на объектно-ориентированной парадигме [1], в которой главное внимание уделяется оптимизации представления объектов и элементов проектирования, связей между ними с анализом различного рода отношений; при использовании второго подхода (парадигма системного проектирования) во главу угла поставлена оптимизация процесса проектирования [2], посредством его декомпозиции, определения формальных элементарных составляющих и программирования этапов жизненного цикла создаваемой системы. Концепция системного проектирования была разработана в Киевском Институте кибернетики. Наряду с вышеуказанными аспектами будем учитывать указанные в ней принципы [2]:

1. Задача проектирования сложных систем изначально недоопределена, что следует из неизвестности многих начальных данных, ограничений и возможных решающих процедур.
2. Задача проектирования логически противоречива. В ней на начальных этапах решения используется информация, которая может быть получена более поздних этапах.
3. Из-за многокритериальности задачи проектирования сложных систем возникает проблема определения оптимального критерия эффективности предполагаемого решения.

Уменьшение неопределенности задачи системного проектирования может быть достигнуто путем проведения экспериментов и применения методов прогнозирования. Логическая противоречивость устраняется использованием итерационных процедур. Интегральное использование таких

предложений позволит решить и проблему определения оптимального критерия для оценки результата. Его численным выражением служит критериальная функция. Идентификация такой зависимости позволит моделировать различные варианты проектных структур и рассчитывать эффективность возможных способов распределения ресурсов, а также определять степень влияния отдельных факторов на достижение глобальной цели проектирования. Необходимость решения таких задач определяет актуальность проведенного исследования.

Постановка задачи

Глобальной целью проектирования Z является получение проекта в базисе системных свойств, системных ресурсов и структур жизненного цикла [1]. Она декомпозируется на подцели низшего уровня Z_1, Z_2, \dots, Z_n . Уровень достижения каждой из них определяется частными критериальными функциями $\varphi(Z_1), \varphi(Z_2), \dots, \varphi(Z_n)$, которые, в свою очередь, являются функциями входных факторов X_1, X_2, \dots, X_k , т. е.:

$$\varphi(Z_i) = f_i(X_1, X_2, \dots, X_k), \quad i = \overline{1, n}. \quad (1)$$

Функции (1), а также интегральную критериальную функцию

$$F(Z) = F(\varphi(Z_1), \varphi(Z_2), \dots, \varphi(Z_n)) \quad (2)$$

чаще всего представляют в виде аддитивной или мультипликативной зависимости [3,4]. Аддитивная зависимость является линейной и, как следствие, не дает представления о реальных взаимозависимостях факторов для сложной системы. Мультипликативная зависимость сводится к аддитивной и дополнительно требует приведения факторов к единой шкале измерения (нормирования). Преодолеть такие недостатки можно, используя в качестве модели (1) некоторой нелинейной зависимости. Если ее спецификация заранее неизвестна, то, учитывая возможность разложения непрерывной функции в ряд Тейлора и дальнейшего приведения подобных слагаемых (в силу наличия многих переменных), получим (1) как полином Колмогорова–Габора:

$$\varphi(Z_i) = a_0 + \sum_{i=1}^k a_i X_i + \sum_{i < j} a_{ij} X_i X_j + \sum_{i < j < p} a_{ijk} X_i X_j X_p + \dots \quad (3)$$

Необходимо осуществить структурную и параметрическую идентификацию (3), что, в свою очередь, позволит решить задачу оптимизации и найти оптимальные значения факторов и критериальной функции.

Метод решения, особенности его реализации в MATLAB

Процедура решения задачи известна. Она реализована в методе группового учета аргументов (МГУА) [5], автором которого является ака-

демик А. Г. Ивахненко. МГУА имеет многочисленные алгоритмические реализации и в известных программно-алгоритмических системах. Главным его преимуществом есть возможность применения для «коротких» выборок, когда другие методы по объективным причинам не работают.

Алгоритм МГУА содержит такие этапы:

1. Определение базовой функции $y_k = f(x_i, x_j)$, количество слагаемых которой не превышает мощности множества статистических данных (данных экспериментов).
2. Расчет по методу наименьших квадратов параметров всех функций y_k .
3. По критерию внешнего дополнения определение наилучших моделей и удаление остальных.
4. Итерационное уточнение функций с использованием композиции $z_k = f(y_i, y_j)$ (шаги 2-4) до тех пор, пока среднее значение критерия внешнего дополнения не начнет ухудшаться.
5. Выбор функции с оптимальным значением критерия внешнего дополнения и ее восстановление (до переменных $x_i, i = \overline{1, k}$).

Реализация такого алгоритма имеет прямой и обратный ход. И каждый из них характеризуется трудностями практического осуществления. Так, при прямом ходе необходимо упорядочивать по возрастанию среднеквадратическую ошибку на точках контрольной последовательности для всех итераций. Такой шаг алгоритма, соответственно, требует вместе с перестановками значений ошибок переставлять и индексы (i, j) независимых переменных, присутствующих в модели (3), а также ее коэффициенты. Необходимость хранения индексов и коэффициентов моделей на всех итерационных шагах требует использования трехмерных массивов. В среде MATLAB эта процедура значительно упрощена по сравнению с известными программными средами.

Для обратного хода особую трудность представляет получение результирующей модели как суперпозиции моделей, полученных на каждом итерационном шаге. Именно тогда и используются функции символьной математики MATLAB.

На рис. 1 изображена схема итерационного процесса. Ось OY соответствует номеру итерации, OX — порядковому номеру модели, OZ — имеет три координаты a_0, a_1, a_3 (если опорная функция имеет вид:

$$y_k = a_0 + a_1 x_i + a_2 x_i x_j. \quad (4)$$

Структурная схема алгоритма изображена на рис. 2.

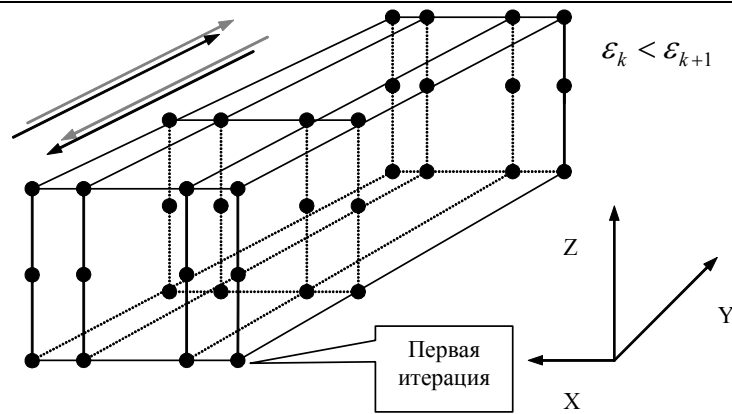


Рис. 1. Итерационный процесс.

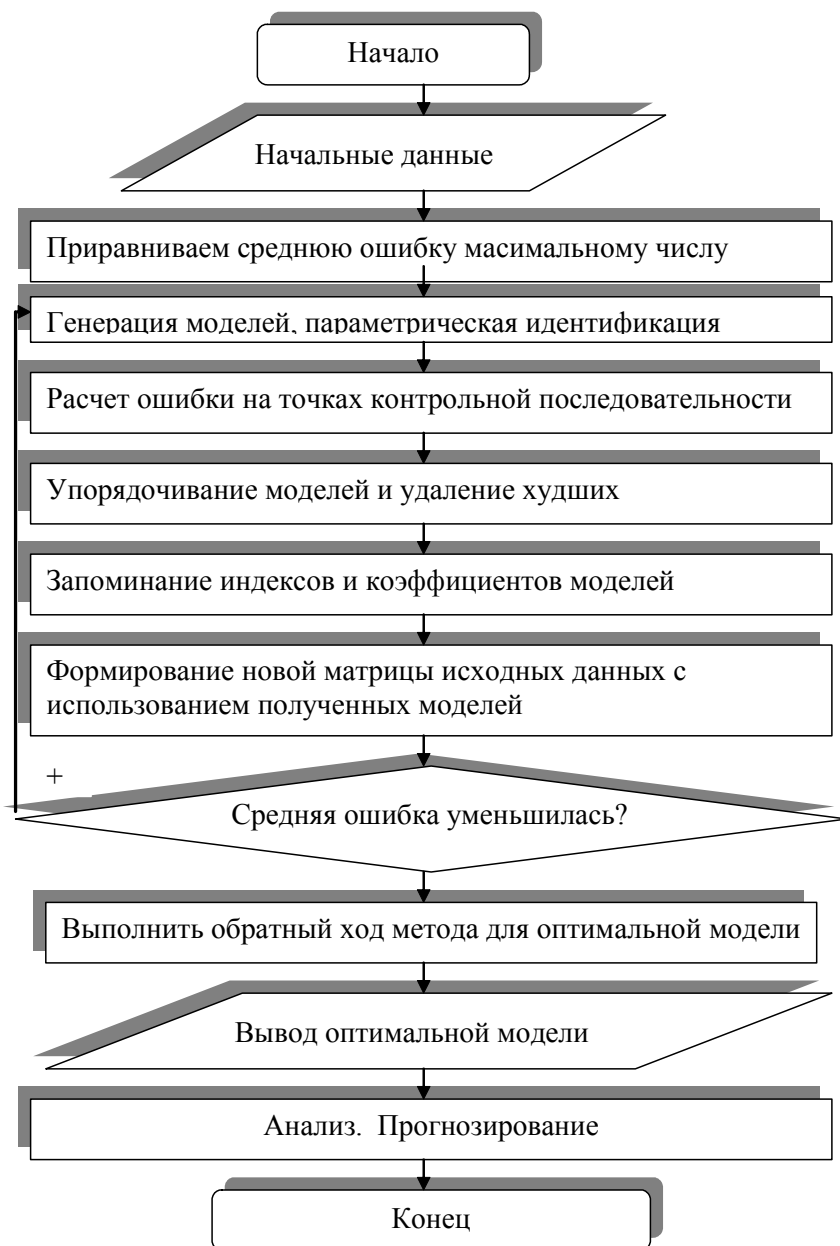


Рис. 2. Структурная схема алгоритма.

Результаты эксперимента

Для проверки правильности метода и его реализации (см. приложение) был проведен эксперимент. В качестве исходной информации использованы данные табл.1. Выходная характеристика вычисляется по формуле:

$$Y = 1 + X_1 X_2 + X_3^2 + X_4^2 - X_1 X_2 X_3 X_4.$$

Первые девять точек отнесем к обучающей последовательности, последнюю — к контрольной. Средняя ошибка на первом шаге итерации равна 4,3452; на втором - 0,155; на третьем - 2,2454. Таким образом, оптимальная модель по критерию минимума среднеквадратической ошибки на контрольной последовательности получена на второй итерации. Поскольку модели упорядочиваются по увеличению ошибки, то такая модель является первой.

Таблица 1				
Исходные данные				
X1	X2	X3	X4	Y
1	9	1	7	-3
2	8	8	2	-171
3	7	3	3	-149
4	6	5	5	-525
5	2	4	4	-117
6	3	2	1	-12
9	4	7	8	-1866
8	5	8	7	-2086
7	1	2	6	-36
4	3	8	3	-202

На первом шаге получена такая наилучшая модель:

$$Y = 67,164 + 48,344 \cdot X_1 - 32,818 \cdot X_1 X_3.$$

На втором шаге:

$$Y = -634,4 + 33,426 \cdot X_1 + 25,919 X_1 X_2 + (-1,4654 + 0,11338 \cdot X_1 + 0,087917 \cdot X_1 X_2) \cdot (270,97 + 26,966 \cdot X_3 - 42,625 \cdot X_3 X_4).$$

Сравнить исходные значения выходной характеристики, значения, рассчитанные на первом ряду селекции и на втором можно по графикам, изображенным на рис. 3.

Очевидно, что модель, полученная на втором ряду селекции значительно лучше описывает начальные данные.

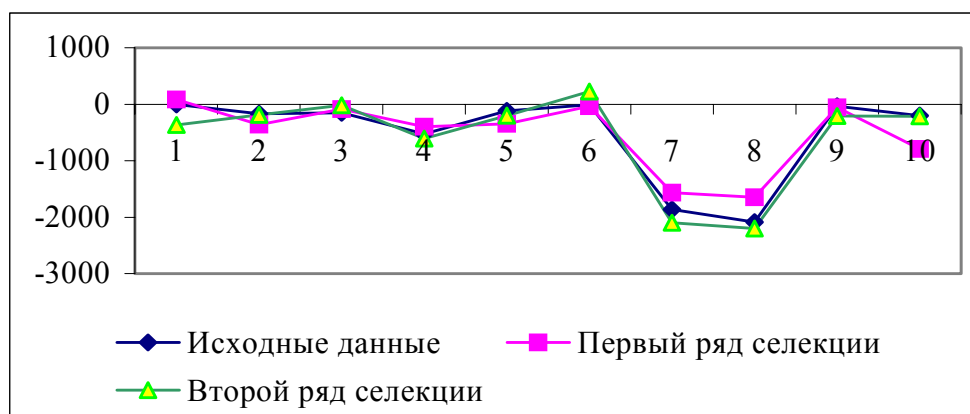


Рис. 3. График значений выходной характеристики.

Заключение

Использование символьной математики для вычисления оптимальной зависимости между входными факторами и выходной характеристикой значительно упрощает программно-алгоритмическую реализацию метода самоорганизации моделей и увеличивает скорость вычислений. Следующим этапом должно стать использование других внешних критериев определения оптимальной модели.

Получение аналитической зависимости позволяет осуществлять ее оптимизацию средствами MATLAB. Необходимо заметить, что функция многих переменных может иметь несколько локальных оптимумов, а предлагаемые методы не страхуют от попадания в них. Поэтому было бы желательно реализовать в среде MATLAB методы эволюционной оптимизации: генетические алгоритмы, эволюционное моделирование, генетическое программирование и эволюционное программирование, тем более, что в других программных пакетах они уже нашли свое применение.

Литература

1. Гради Буч. Объектно-ориентированное проектирование с примерами применения.— М.: Конкорд, 1992.— 519 с.
2. Жук К. Д., Тимченко А. А., Доленко Т. И. Исследование структур и моделирование логико-динамических систем.— К.: Наукова думка, 1975.— 199 с.
3. Молчанов А. А. Моделирование и проектирование сложных систем.— К.: Выща школа, 1988.— 359 с.
4. Перегудов Ф. И., Тарасенко Ф. П. Введение в системный анализ.— М.: Высшая школа, 1989.— 354 с.
5. Ивахненко А. Г. Долгосрочное прогнозирование и управление сложными системами.— К.: Техника, 1975.— 312 с.

Приложение

Пример m-файла реализации алгоритма самоорганизации моделей с использованием символьной математики

% Функция, с помощью которой упорядочиваются в возрастающем порядке ошибки

% на контрольных точках для всех опорных функций, одновременно

% осуществляются перестановки соответствующих индексов.

```
function [s,index,no] = perest(g,no1,n)
```

```
for i=1:n
```

```
    no(i)=i;
```

```
end
```

```
for i=1:n
```

```
    for j=2:n
```

```
        if g(j-1)>g(j)
```

```
            c=g(j-1);
```

```
            g(j-1)=g(j);
```

```
            g(j)=c;
```

```
            d=no(j-1);
```

```
            no(j-1)=no(j);
```

```
            no(j)=d;
```

```
            d=no1(j-1,1);
```

```
            no1(j-1,1)=no1(j,1);
```

```
            no1(j,1)=d;
```

```
            d=no1(j-1,2);
```

```
            no1(j-1,2)=no1(j,2);
```

```
            no1(j,2)=d;
```

```
        end
```

```
    end
```

```
    s=g;
```

```
    index=no1;
```

```
end
```

% Функция, осуществляющая один шаг МГУА по критерию минимума

% среднеквадратической ошибки на контрольных точках

```
function [srednee_eps,bb,inde,qq1] = minsqv(pp,qq,y)
```

```
% pp — матрица, по которой осуществляется идентификация опорных функций
```

```
% qq — контрольная точка
```

```
% y — вектор значений выходных характеристик
```

```
% srednee_eps — среднее значение ошибки для всех опорных функций
```

```
% bb — матрица коэффициентов опорных функций
```

```
% qq1 — новая матрица, используемая на следующем шаге для идентификации
```

```
% inde — матрица индексов переменных
```

```
lo=1;
```

```
for i=1:4
```

```
    for j=i+1:4
```

```
        for k=1:9
```

```
            z(k)=pp(k,i)*pp(k,j);
```

```

xx(k,1)=1;
xx(k,2)=pp(k,i);
xx(k,3)=z(k);
end
% коэффициенты регрессии
a=inv(xx'*xx)*xx'*y;
% среднее квадратическое отклонение для каждой модели
eps(lo)=(-202-(a(1)+a(2)*qq(i)+a(3)*qq(i)*qq(j)))^2/202^2;
% коэффициенты уравнения
for ii=1:3
aa(ii,lo)=a(ii);
noi(lo,1)=i;
noi(lo,2)=j;
end
lo=lo+1;
end
end
% ee — упорядоченный массив ошибок, n1 - их начальные номера
[ee,inde,n1]=perest(eps,noi,lo-1)
% средняя ошибка
srednee_eps=sum(eps)/(lo-1)
% коэффициенты уравнений регрессии
for j=1:3
for i=1:4
bb(j,i)=aa(j,n1(i));
end
end
% формирование новой матрицы
for j=1:4
for k=1:9
qq1(k,j)=bb(1,j)+bb(2,j)*pp(k,inde(j,1))+bb(3,j)*pp(k,inde(j,1))*pp(k,inde(j,2));
end
end

% учебная последовательность
p=[1 7 3 7;
2 6 2 3;
3 5 4 2;
4 4 5 1;
5 3 6 4;
6 2 5 5;
7 1 4 3;
8 8 3 2;
9 7 12 2];
pp=p;
% контрольная точка
qq=[4; 6; 1; 2];
% выходная характеристика
y=[41; 12; -2; -11; -24; -17; -20; 44; -94];
ny=1;

```

```
s1=1000000000000000;
sr_eps=1000000;
while s1>sr_eps
s1=sr_eps
b1=0;
[sr_eps,bb,inde,qq1] = minsqv(pp,qq,y)
for i=1:4
perem_memory(i,1,ny)=inde(i,1);
perem_memory(i,2,ny)=inde(i,2);
end
perem_memory
for j=1:3
for i=1:4
koef(j,i,ny)=bb(j,i);
end
end
ny=ny+1
pp=qq1;
% b1=bb;
end
% уравнения в символьном виде
syms x1 x2 x3 x4
digits(5)
sym(koef(1,1,1),'d')
y1=sym(koef(1,1,1),'d')+sym(koef(2,1,1),'d')*x1+sym(koef(3,1,1),'d')*x1*x3
y2=sym(koef(1,2,1),'d')+sym(koef(2,2,1),'d')*x1+sym(koef(3,2,1),'d')*x1*x2
y3=sym(koef(1,3,1),'d')+sym(koef(2,3,1),'d')*x1+sym(koef(3,3,1),'d')*x1*x4
y4=sym(koef(1,4,1),'d')+sym(koef(2,4,1),'d')*x3+sym(koef(3,4,1),'d')*x3*x4

z1=sym(koef(1,1,2),'d')+sym(koef(2,1,2),'d')*y2+sym(koef(3,1,2),'d')*y2*y4
z2=sym(koef(1,2,2),'d')+sym(koef(2,2,2),'d')*y2+sym(koef(3,2,2),'d')*y2*y3
z3=sym(koef(1,3,2),'d')+sym(koef(2,3,2),'d')*y3+sym(koef(3,3,2),'d')*y3*y4
z4=sym(koef(1,4,2),'d')+sym(koef(2,4,2),'d')*y1+sym(koef(3,4,2),'d')*y1*y3
```

УДК 681.51.033

ПАКЕТ ПРОГРАММ СТРУКТУРНОЙ И ПАРАМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ ЛИНЕЙНЫХ ОДНОМЕРНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

*Фатыев В. А.,**Тульский государственный университет, Тула,**e-mail: info@tsu.tula.ru**Понятский В. М.,**ФГУП Конструкторское бюро приборостроения, Тула,**e-mail: granat@home.tula.net**Каргин А. В.**Тульский государственный университет, Тула,**e-mail: viant@schekino.tula.net*

Введение

На рис. 1 изображена структурная схема одномерной динамической системы типа «черный ящик» в непрерывном времени. Данная система имеет одну выходную (управляемую) переменную и одну входную (управляющую) переменную. На рис. 1 обозначены: измеряемое значение входной переменной $u(t)$, не измеряемое значение выходной переменной $y(t)$, измеряемое значение зашумленной выходной переменной $z(t)$, и, действующий на выход системы, не измеряемый шум $e(t)$. Рассмотрим далее только линейные динамические системы [3].

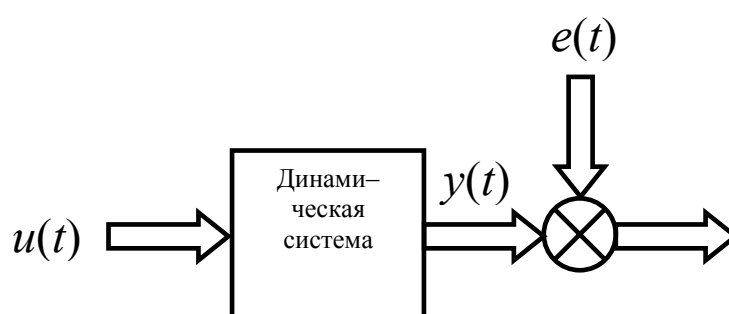


Рис. 1. Структурная схема одномерной динамической системы.

Идентификация линейных одномерных динамических систем предусматривает четыре основных этапа: определение порядка модели, оценивание значений неизвестных коэффициентов, статистический анализ результатов идентификации и переход к требуемому виду модели.

Для решения данной задачи авторами разработан пакет прикладных программ, функционирующих в среде MS Window 9x/2000/NT/XP.

1. Структура пакета программ

Методы, реализующие все этапы идентификации одномерных динамических систем, реализованы в виде библиотек функций. Библиотеки выполнены в виде динамически подключаемых библиотек (.DLL). Структура пакета программ показана на рис. 2.

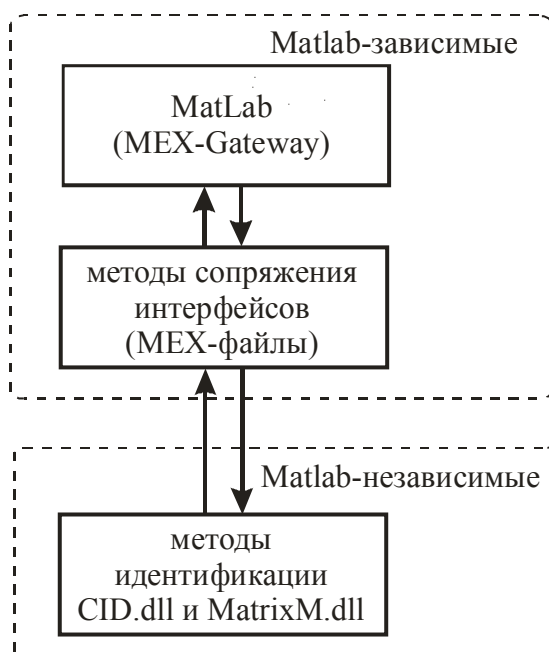


Рис. 2. Структура пакета программ идентификации.

Пакет программ можно разделить на две части: одна включает в себя библиотеки функций, не требующие наличия установленных и выполняемых файлов из дистрибутива MATLAB, а другая часть программ фактически состоит из интерфейсных модулей, которые позволяют вызывать из MATLAB методы из MATLAB-независимых библиотек первой части.

Такая структура пакета программ позволяет с одной стороны получить большую гибкость при разработке модулей, которые не зависят от MATLAB, что расширяет область их применения, а с другой стороны — получить возможность вызова данных методов из MATLAB, что открывает пользователю доступ к разнообразным функциям MATLAB.

В пакет программ входят функции [1–2]:

- определения порядка модели разностного уравнения (структурной идентификации) методами функции потерь и матрицы моментов [1,4,9,10];
- оценивания значений неизвестных коэффициентов модели разностного уравнения известного порядка (параметрической идентификации) методом наименьших квадратов и их модификациями: методом инструментальной переменной, расширенными матрич-

- ными методами; как в режиме реального времени, так и в режиме отложенного времени [1–6, 9–10];
- оценки адекватности построенной модели и значимости оценок ее коэффициентов [1–2];
 - преобразования математических моделей [1, 2, 7–8] (см. рис. 3).

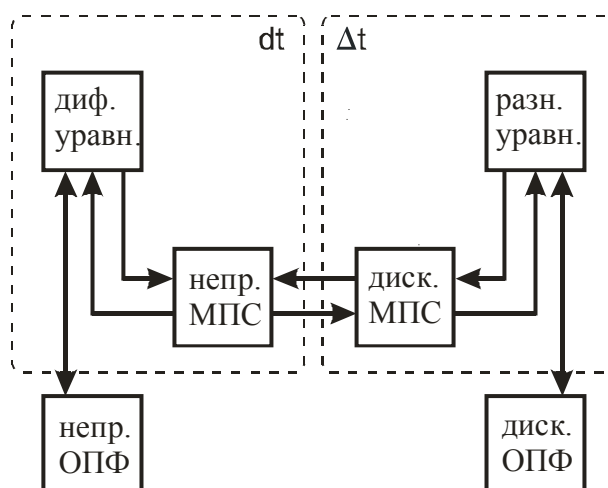


Рис. 3. Взаимные преобразования математических моделей:

диф.уравн. — дифференциальное уравнение; разн. уравн. — разностное уравнение; непр. МПС — модель пространства состояний в непрерывном времени; дискр. МПС — модель пространства состояний в дискретном времени; непр. ОПФ — непрерывная операторная передаточная функция; диск. ОПФ — дискретная операторная передаточная функция.

2. Математические модели линейных одномерных стационарных динамических систем и методы их построения

В данной главе дан краткий обзор типов математических моделей линейных стационарных одномерных динамических систем, построение которых реализовано в пакете программ. Приведено описание каждого метода структурной и параметрической идентификации. Сжато изложены способы взаимного преобразования математических моделей и некоторые методы статистического анализа результатов идентификации. Более подробно данный материал изложен в работах [1–4, 7–10].

2.1 Структура математических моделей одномерных стационарных динамических систем

2.1.1 Дифференциальное уравнение

Дифференциальное уравнение стационарной линейной одномерной динамической системы при отсутствии внутреннего и внешнего шума имеет следующий вид [1–4]:

$$\tilde{\mathbf{a}}^T \tilde{\mathbf{y}}(t) = \tilde{\mathbf{b}}^T \tilde{\mathbf{u}}(t), \quad (1)$$

где $\tilde{\mathbf{a}}^T = (a_0, a_1, \dots, a_n)$; $\tilde{\mathbf{b}}^T = (b_0, b_1, \dots, b_n)$; $\tilde{\mathbf{y}}^T(t) = \left(y(t), \frac{dy(t)}{dt}, \dots, \frac{d^n y(t)}{d^n t} \right)$;

$\tilde{\mathbf{u}}^T(t) = \left(u(t), \frac{du(t)}{dt}, \dots, \frac{d^n u(t)}{d^n t} \right)$, $a_0, a_1, \dots, a_n, b_0, b_1, \dots, b_n$ — коэффициенты модели; n — порядок модели; $y(t)$ — значение выходной переменной; $u(t)$ — значение входной переменной.

При воздействии внешнего шума выражение (1) следует дополнить следующим выражением:

$$z(t) = y(t) + \varepsilon(t), \quad (2)$$

где $z(t)$ — значение зашумленной выходной переменной в момент времени t ; $y(t)$ — значение выходной переменной в момент времени t из модели (1); $\varepsilon(t)$ — значение внешнего шума в момент времени t .

2.1.2. Разностное уравнение

Разностное уравнение стационарной линейной одномерной динамической системы при отсутствии внутреннего и внешнего шума имеет вид [1-4]:

$$\tilde{\mathbf{a}}^T \tilde{\mathbf{y}}(k) = \tilde{\mathbf{b}}^T \tilde{\mathbf{u}}(k), \quad (3)$$

где $\tilde{\mathbf{a}}^T = (a_0, a_1, \dots, a_n)$; $\tilde{\mathbf{b}}^T = (b_0, b_1, \dots, b_n)$; $\tilde{\mathbf{y}}^T(k) = (y(k), y(k-1), \dots, y(k-n))$; $\tilde{\mathbf{u}}^T(k) = (u(k), u(k-1), \dots, u(k-n))$, $a_0, a_1, \dots, a_n, b_0, b_1, \dots, b_n$ — коэффициенты модели; n — порядок модели; $y(k)$ — значение выходной переменной в момент времени $k\Delta t$; Δt — интервал дискретизации системы по времени; $u(k)$ — значение входной переменной в момент времени $k\Delta t$.

При воздействии внешнего шума выражение (3) следует дополнить следующим выражением:

$$z(k) = y(k) + \varepsilon(k), \quad (4)$$

где $z(k)$ — значение зашумленной выходной переменной в момент времени $k\Delta t$; $y(k)$ — значение выходной переменной в момент времени $k\Delta t$ из модели (3); $\varepsilon(k)$ — значение внешнего шума в момент времени $k\Delta t$.

Составим модель (3),(4) для N последовательных значений входного сигнала:

$$\mathbf{y} = \mathbf{F}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \quad (5)$$

где $\mathbf{y}^T = (y(n), y(n+1), \dots, y(N))$ – вектор значений выходной переменной

системы; $\mathbf{F} = \begin{pmatrix} u(n) & \dots & u(0) & y(n-1) & \dots & y(0) \\ u(n+1) & \dots & u(1) & y(n) & \dots & y(1) \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ u(N) & \dots & u(N-n) & y(N-1) & \dots & y(N-n) \end{pmatrix}$ – матрица

наблюдения; $\boldsymbol{\theta}^T = (d_0 : d_1 : \dots : d_n : -c_1 : \dots : -c_n)$, $d_i = a_0^{-1}b_i, i = \overline{0, n}, c_j = a_0^{-1}a_j, j = \overline{1, n}$ – вектор параметров модели; $\boldsymbol{\varepsilon}^T = (\varepsilon(n), \varepsilon(n+1), \dots, \varepsilon(N))$ – вектор значений внешнего шума (нормального случайного процесса с нулевым математическим ожиданием).

2.1.3. Модель пространства состояний в непрерывной форме

Модель пространства состояний в непрерывной форме для линейной одномерной динамической системы при отсутствии внутреннего и внешнего шума имеет вид [1-4]:

$$\begin{cases} \frac{d\tilde{\mathbf{x}}(t)}{dt} = \mathbf{T}\tilde{\mathbf{x}}(t) + \mathbf{g}u(t), \\ y(t) = \mathbf{h}^T\tilde{\mathbf{x}}(t) + ku(t), \end{cases} \quad (6)$$

где $\tilde{\mathbf{x}}(t)$ – вектор значений переменных состояния модели в момент времени t , размерностью $n \times 1$; \mathbf{T} – матрица динамики, размерностью $n \times n$; \mathbf{g} – вектор управления, размерностью $n \times 1$; $u(t)$ – значение входной переменной в момент времени t ; $y(t)$ – значение выходной переменной в момент времени t ; \mathbf{h} – вектор наблюдения, размерностью $n \times 1$; k – коэффициент входа.

2.1.4. Модель пространства состояний в дискретной форме

Модель пространства состояний в дискретной форме для линейной одномерной динамической системы при отсутствии внутреннего и внешнего шума имеет вид [1-4]:

$$\begin{cases} \tilde{\mathbf{x}}(k+1) = \mathbf{A}\tilde{\mathbf{x}}(k) + \mathbf{b}u(k), \\ y(k) = \mathbf{c}^T\tilde{\mathbf{x}}(k) + du(k), \end{cases} \quad (7)$$

где $\tilde{\mathbf{x}}(k+1)$ – вектор значений переменных состояния модели в момент времени $(k+1)\Delta t$, размерностью $n \times 1$; n – порядок модели; \mathbf{A} – матрица динамики, размерностью $n \times n$; \mathbf{b} – вектор управления, размерностью $n \times 1$; $u(k)$ – значение входной переменной в момент времени $k\Delta t$; $y(k)$ – значение выходной переменной в момент времени $k\Delta t$; \mathbf{c} – вектор наблюдения, размерностью $n \times 1$; d – коэффициент входа.

2.1.5. Непрерывная передаточная функция

Модель непрерывной операторной передаточной функции для линейной одномерной динамической системы при нулевых начальных условиях имеет вид [1]:

$$w(s) = \frac{y(s)}{u(s)} = \frac{\tilde{k}_0 + \tilde{k}_1 s + \dots + \tilde{k}_n s^n}{\tilde{r}_0 + \tilde{r}_1 s + \dots + \tilde{r}_n s^n} = \frac{\tilde{\mathbf{k}}^T \mathbf{s}}{\tilde{\mathbf{r}}^T \mathbf{s}}, \quad (8)$$

где s — оператор дифференцирования Лапласа; $y(s) = L\{y(t)\}$ — непрерывное преобразование Лапласа выходной переменной $y(t)$; $u(s) = L\{u(t)\}$ — непрерывное преобразование Лапласа входной переменной $u(t)$; n — порядок модели; $\tilde{\mathbf{k}}^T = (\tilde{k}_0, \tilde{k}_1, \dots, \tilde{k}_n)$, $\tilde{\mathbf{r}}^T = (\tilde{r}_0, \tilde{r}_1, \dots, \tilde{r}_n)$ — вектора коэффициентов модели; $\mathbf{s}^T = (1, s, \dots, s^n)$.

2.1.6. Дискретная передаточная функция

Модель дискретной операторной передаточной функции для линейной одномерной динамической системы при нулевых начальных условиях имеет вид:

$$w(z) = \frac{y(z)}{u(z)} = \frac{k_0 + k_1 z^{-1} + \dots + k_n z^{-n}}{r_0 + r_1 z^{-1} + \dots + r_n z^{-n}} = \frac{\mathbf{k}^T \mathbf{z}}{\mathbf{r}^T \mathbf{z}}, \quad (9)$$

где z — оператор Z-преобразования; $y(z) = Z\{y(t)\}$ — Z-преобразование выходной переменной $y(t)$; $u(z) = Z\{u(t)\}$ — Z-преобразование входной переменной $u(t)$; n — порядок модели; $\mathbf{k}^T = (k_0, k_1, \dots, k_n)$, $\mathbf{r}^T = (r_0, r_1, \dots, r_n)$ — вектора коэффициентов модели; $\mathbf{z}^T = (1, z^{-1}, \dots, z^{-n})$.

2.2. Определение порядка разностного уравнения

2.2.1 Структурная идентификация по функции потерь

Рассмотрим метод структурной идентификации, использующий в качестве критерия функцию потерь v из [1, 2, 4]:

$$v(n) = \frac{1}{N-n} \left[\hat{\mathbf{e}}^T(n) \hat{\mathbf{e}}(n) \right], \quad (11)$$

где n — порядок разностного уравнения (3)–(4); N — дискретное время наблюдения; $\hat{\mathbf{e}}(n) = \hat{\mathbf{y}}(n) - \mathbf{y}(n)$ — невязки модели; $\hat{\mathbf{y}}^T(n) = (\hat{y}(0), \hat{y}(\Delta t), \dots, \hat{y}(N\Delta t))$, $\mathbf{y}^T(n) = (y(0), y(\Delta t), \dots, y(N\Delta t))$ — соответ-

ственно значения предсказанных по модели (3)–(4) значений выхода системы и экспериментальные значения выхода системы.

Функция потерь (11) численно характеризует величину отклонения предсказанного по модели значения выхода системы от его экспериментального значения. При увеличении порядка системы значение функции потерь будет убывать (рис. 4), но, при воздействии внешнего шума, никогда не достигнет нуля (явление «сверхсогласия» [4]).

За оценку истинного порядка системы принимается такое значение порядка, при увеличении которого функция потерь убывает незначительно.

Функция потерь для стационарной линейной одномерной динамической системы, на примере модели разностного уравнения (3) при $n=3$, приведена на рис. 4.

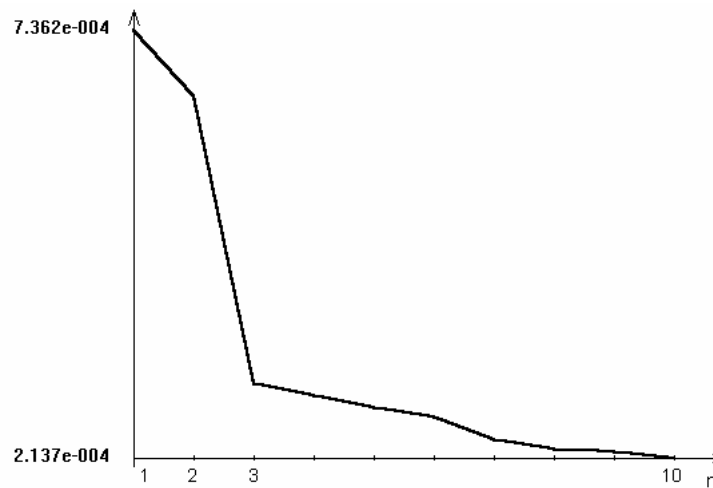


Рис. 4. Функция потерь.

2.2.2. Структурная идентификация по матрице моментов

Матрица моментов (корреляций) имеет следующий вид [1,2,4]:

$$\mathbf{Q}(u, y, \hat{n}) = \mathbf{F}(\hat{n})^T \mathbf{F}(\hat{n}), \quad (12)$$

где $\mathbf{Q}(u, y, \hat{n})$ — матрица моментов для \hat{n} -го порядка системы; $\mathbf{F}(\hat{n})$ — матрица наблюдений (5), для порядка системы n , равного \hat{n} .

В случае, свободном от шумов матрица $\mathbf{Q}(u, y, \hat{n})$ вырождается при $\hat{n} > n$. Если входной сигнал постоянно возбуждающий, то $\mathbf{Q}(u, y, \hat{n})$ имеет ранг равный $r=2n+1$. Однако использование ранга для определения порядка часто затруднено наличием в системе шумов, тогда матрица $\mathbf{Q}(u, y, \hat{n})$ не вырождается при достижении истинного порядка системы, и ее строки не становятся линейно зависимыми.

В качестве теста определения порядка используется относительный определитель матрицы $\mathbf{Q}(u, y, \hat{n})$:

$$\text{rel det } \mathbf{Q}(u, y, \hat{n}) = \frac{\det \mathbf{Q}(u, y, \hat{n})}{\Re}, \quad (13)$$

где \Re — наибольший по модулю элемент матрицы $\mathbf{Q}(u, y, \hat{n})$.

2.3. Определение оценок коэффициентов разностного уравнения

2.3.1. Метод наименьших квадратов

Рассмотрим модель разностного уравнения (5). Введем следующий квадратичный критерий:

$$v = \frac{1}{N-n} [\mathbf{y} - \mathbf{F}\boldsymbol{\theta}]^T [\mathbf{y} - \mathbf{F}\boldsymbol{\theta}]. \quad (14)$$

Найдем такие оценки значений параметров модели (5), которые минимизируют значение критерия (14) по методу наименьших квадратов (МНК) [1–4]:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} v = [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{y} \quad (15)$$

с ковариационной матрицей

$$\text{cov}(\hat{\boldsymbol{\theta}}) = [\mathbf{F}^T \mathbf{F}]^{-1} \sigma_e^2, \quad (16)$$

где σ_e^2 — дисперсия нормального случайного процесса $\varepsilon(k)$ из (5).

2.3.2. Обобщенный метод наименьших квадратов

Обобщенный метод наименьших квадратов (ОМНК) использует для устранения смещенности оценок параметров разностного уравнения (5) операцию фильтрации [1,2,4].

Приведем алгоритм получения ОМНК-оценок параметров разностного уравнения (5) (алгоритм А) [1,2].

Исходные данные алгоритма А: Δt — интервал дискретизации системы по времени; N — дискретное время наблюдения системы; $u(k), y(k), k = \overline{0, N}$ — реализации входного и выходного сигналов; r — порядок фильтра; n — порядок разностного уравнения (5); d — малая величина, $MaxIter$ — предельное число итераций.

Алгоритм А.

А.1. Задать $\Delta t \leftarrow 1$, N , $u(k), y(k), k = \overline{0, N}$, r , n , d , $MaxIter$.

А.2. Используя $u(k), y(k), k = \overline{0, N}$, производим оценку вектора параметров модели $\hat{\boldsymbol{\theta}}$ по (15), которая является смещенной.

А.3. Получаем вектор оценок невязок $\hat{\varepsilon}$ из соотношения $\hat{\varepsilon} = \mathbf{y} - \mathbf{F}\hat{\boldsymbol{\theta}}$.

А.4. Получаем оценку параметров фильтра $\hat{\mathbf{d}}$ по выражению:

$$\hat{\mathbf{d}} = -[\mathbf{E}^T \mathbf{E}]^{-1} \mathbf{E}^T \hat{\varepsilon}, \quad (17)$$

где

$$\hat{\varepsilon}^T = (\hat{\varepsilon}(n+1), \hat{\varepsilon}(n+2), \dots, \hat{\varepsilon}(N)); \mathbf{E} = \begin{pmatrix} \hat{\varepsilon}(n) & \hat{\varepsilon}(n-1) & \dots & \hat{\varepsilon}(n-r+1) \\ \hat{\varepsilon}(n+1) & \hat{\varepsilon}(n) & \dots & \hat{\varepsilon}(n-r+2) \\ \vdots & \vdots & \dots & \vdots \\ \hat{\varepsilon}(N-1) & \hat{\varepsilon}(N-2) & \dots & \hat{\varepsilon}(N-r) \end{pmatrix}.$$

А.5. Используя (17), фильтруем $u(k), y(k), k = \overline{0, N}$ в соответствии с выражениями:

$$\begin{cases} u_{\Phi}(k) = u(k) + \hat{d}_1 u(k-1) + \dots + \hat{d}_r u(k-r), \\ y_{\Phi}(k) = y(k) + \hat{d}_1 y(k-1) + \dots + \hat{d}_r y(k-r), \\ u(l) = y(l) = 0, l < 0, k = \overline{0, N}. \end{cases}$$

А.6. Производим оценку параметров модели (5):

$$\hat{\boldsymbol{\theta}}_{\Phi} = [\mathbf{F}_{\Phi}^T(\mathbf{u}_{\Phi}, \mathbf{y}_{\Phi}) \mathbf{F}_{\Phi}(\mathbf{u}_{\Phi}, \mathbf{y}_{\Phi})]^{-1} \mathbf{F}_{\Phi}^T(\mathbf{u}_{\Phi}, \mathbf{y}_{\Phi}) \mathbf{y}_{\Phi},$$

где $\mathbf{F}_{\Phi}^T(\mathbf{u}_{\Phi}, \mathbf{y}_{\Phi})$ — матрица наблюдения \mathbf{F} модели (5), построенная по отфильтрованным значениям входной $\mathbf{u}_{\Phi}^T = (u_{\Phi}(0), u_{\Phi}(1), \dots, u_{\Phi}(N))$ и выходной $\mathbf{y}_{\Phi}^T = (y_{\Phi}(0), y_{\Phi}(1), \dots, y_{\Phi}(N))$ переменных.

А.7. $\mathbf{u} \leftarrow \mathbf{u}_{\Phi}, \mathbf{y} \leftarrow \mathbf{y}_{\Phi}, \hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}}_{\Phi}$.

А.8. Вычисляем значение критерия ν из (14).

А.9. Повторяем шаги 3–8, пока наблюдается улучшение оценок, т. е. пока уменьшается значение ν , и выполняется условие $\nu > d$ или пока не повторим шаги 3–8 *MaxIter* раз.

А.10. Получили $\hat{\boldsymbol{\theta}}$ — ОМНК-оценку параметров разностного уравнения (5).

А.11. Конец.

2.3.3. Рекуррентный метод наименьших квадратов

Рекуррентный метод наименьших квадратов (рМНК) аналогичен по свойствам получаемых оценок МНК, однако, он может быть использован для идентификации линейных динамических систем в реальном масштабе времени. Приведем выражения для модели (5) [1-4]:

Введем вектор последних наблюдений в момент времени $k\Delta t$ — нижняя строка матрицы \mathbf{F} из (5):

$$\mathbf{f}_k^T = (u(k), \dots, u(k-n), y(k-1), \dots, y(k-n)), \quad (18)$$

МНК-оценки на k -м шаге процедуры оценивания находятся по выражению:

$$\begin{cases} \hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} - \mathbf{P}_{k-1} \mathbf{f}_k \left[1 + \mathbf{f}_k^T \mathbf{P}_{k-1} \mathbf{f}_k \right]^{-1} \left(\mathbf{f}_k^T \hat{\boldsymbol{\theta}}_{k-1} - y(k) \right), \\ \mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1} \mathbf{f}_k \left[1 + \mathbf{f}_k^T \mathbf{P}_{k-1} \mathbf{f}_k \right]^{-1} \mathbf{f}_k^T \mathbf{P}_{k-1}. \end{cases} \quad (19)$$

Отметим, что начальные значения матрицы \mathbf{F}_{N_0} и начальные оценки параметров $\hat{\boldsymbol{\theta}}_{N_0}$ можно получать по результатам первых N_0 опытов:

$$\begin{cases} \mathbf{P}_{N_0} = \left[\mathbf{F}_{N_0}^T \mathbf{F}_{N_0} \right]^{-1}, \\ \hat{\boldsymbol{\theta}}_{N_0} = \mathbf{P}_{N_0} \mathbf{F}_{N_0}^T \mathbf{y}_{N_0}, \end{cases} \quad (20)$$

где $N_0 \geq 3n+1$; \mathbf{F}_{N_0} и \mathbf{y}_{N_0} — матрица и вектор, получаемые соответственно из матрицы \mathbf{F} и вектора \mathbf{y} выражения (5) при $N \leftarrow N_0$.

2.3.4. Метод инструментальной переменной

Рассмотрим метод инструментальной переменной (МИП). Введем квадратичный критерий v , аналогичный (14) [1,2,4]:

$$v = \frac{1}{N-n} \left[\mathbf{y} - \mathbf{F} \hat{\boldsymbol{\theta}} \right]^T \mathbf{Z} \mathbf{Z}^T \left[\mathbf{y} - \mathbf{F} \hat{\boldsymbol{\theta}} \right], \quad (21)$$

где \mathbf{Z} — так называемая матрица инструментальной переменной, удовлетворяющая определенным условиям [1,2,4], например,

$$\mathbf{Z} = \begin{pmatrix} u(n) & \dots & u(0) & x(n-1) & \dots & x(0) \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ u(N) & \dots & u(N-n) & x(N-1) & \dots & x(N-n) \end{pmatrix}, \quad x(k) — \text{выход некоторо-}$$

го фиксированного фильтра [4]. Таким фильтром, например, может являться модель (5), в которой в качестве векторов параметров $\tilde{\mathbf{a}}$ и $\tilde{\mathbf{b}}$ выступают их МНК-оценки, полученные по выражению (15) с учетом (5), на вход подается входной сигнал $u(k)$, а с выхода снимается значение инструментальной переменной $x(k)$.

Соответствующее выражение для оценок параметров модели (5) запишем в виде:

$$\hat{\boldsymbol{\theta}} = \left[\mathbf{Z}^T \mathbf{F} \right]^{-1} \mathbf{Z}^T \mathbf{y}. \quad (22)$$

2.3.5. Рекуррентный метод инструментальной переменной

Рекуррентный метод инструментальной переменной (рМИП) аналогичен по свойствам получаемых оценок МИП, однако он может быть ис-

пользован для идентификации динамической системы в реальном масштабе времени.

Введем вектор-строку \mathbf{z}_k^T — последнюю строку матрицы \mathbf{Z} из (21).

Тогда выражение (22) примет вид [1,2,4]:

$$\begin{cases} \hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \mathbf{P}_{k-1} \mathbf{z}_k \left(1 + \mathbf{f}_k^T \mathbf{P}_{k-1} \mathbf{z}_k\right)^{-1} \mathcal{E}(k), \\ \mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1} \mathbf{z}_k \left(1 + \mathbf{f}_k^T \mathbf{P}_{k-1} \mathbf{z}_k\right)^{-1} \mathbf{f}_k^T \mathbf{P}_{k-1}, \\ \mathcal{E}(k) = y(k) - \mathbf{f}_k^T \hat{\boldsymbol{\theta}}_{k-1}. \end{cases} \quad (23)$$

Отметим, что начальные значения матрицы \mathbf{P}_{N_0} и начальные оценки параметров $\hat{\boldsymbol{\theta}}_{N_0}$ можно получать по результатам первых $N_0 = (3n+2)$ опытов:

$$\begin{cases} \mathbf{P}_{N_0} = \left(\mathbf{Z}_{N_0}^T \mathbf{F}_{N_0}\right)^{-1}, \\ \hat{\boldsymbol{\theta}}_{N_0} = \mathbf{P}_{N_0} \mathbf{Z}_{N_0}^T \mathbf{y}_{N_0}, \end{cases} \quad (24)$$

где $\mathbf{y}_{N_0}^T = (y(n), \dots, y(N_0))$ — вектор значений выходного сигнала;

$\mathbf{F}_{N_0} = \begin{pmatrix} u(n) & \dots & u(0) & y(n-1) & \dots & y(0) \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ u(N_0) & \dots & u(N_0-n) & y(N_0-1) & \dots & y(N_0-n) \end{pmatrix}$ — матрица первых

наблюдений; $\mathbf{Z}_{N_0} = \begin{pmatrix} u(n) & \dots & u(0) & x(n-1) & \dots & x(0) \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ u(N_0) & \dots & u(N_0-n) & x(N_0-1) & \dots & x(N_0-n) \end{pmatrix}$ — мат-

рица инструментальной переменной; $x(k)$ — выход некоторого фиксированного фильтра [4]. Таким фильтром, например, может являться модель (5), в которой в качестве векторов параметров $\tilde{\mathbf{a}}$ и $\tilde{\mathbf{b}}$ выступают их МНК-оценки, полученные по выражению (15); на вход подается входной сигнал $u(k)$, а с выхода снимается значение инструментальной переменной $x(k)$.

2.3.6. Рекуррентный расширенный матричный метод

Рекуррентный расширенный матричный метод (rPMM) — другая модификация МНК, в которой для получения несмещенных оценок применяется операция расширения модели [1,2,4].

Расширим модель невязок следующим образом:

$$\hat{\boldsymbol{\varepsilon}}(k) = \mathbf{c}^T \boldsymbol{\xi}(k-s) - \mathbf{d}^T \hat{\boldsymbol{\varepsilon}}(k-r) + \xi(k), \quad (25)$$

где $\mathbf{c}^T = (c_1, \dots, c_s)$, $\mathbf{d}^T = (d_1, \dots, d_r)$ — коэффициенты модели шума; s, r — порядки модели; $\hat{\boldsymbol{\varepsilon}}^T(k-r) = (\hat{\varepsilon}(k-1), \dots, \hat{\varepsilon}(k-r))$, $\hat{\varepsilon}(k)$ — ошибки уравнения (невязки) модели (5); $\hat{\boldsymbol{\xi}}^T(k-s) = (\hat{\xi}(k-1), \dots, \hat{\xi}(k-s))$, $\hat{\xi}(k)$ — «белый шум».

Подставив (25) в (5) и перейдя к оценкам, получим:

$$\hat{\mathbf{z}}(k) = y(k) - \mathbf{h}_k^T \hat{\boldsymbol{\theta}}, \quad \hat{\xi}(k) = \hat{\varepsilon}(k) - \mathbf{v}_k^T \hat{\boldsymbol{\lambda}}, \quad (26)$$

где $\mathbf{h}_k^T = (u(k), \dots, u(k-n), y(k-1), \dots, y(k-n))$ — строка матрицы \mathbf{F} из (5); $\mathbf{v}_k^T = (\hat{\xi}(k-1), \dots, \hat{\xi}(k-s), \hat{\varepsilon}(k-1), \dots, \hat{\varepsilon}(k-r))$; $\hat{\boldsymbol{\lambda}}^T = (\hat{\mathbf{c}}^T; -\hat{\mathbf{d}}^T)$.

На основании (26) соответствующие рекуррентные соотношения, запишем аналогично (19), следующим образом:

$$\begin{cases} \hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} - \mathbf{P}_{k-1} \tilde{\mathbf{f}}_k \left[1 + \tilde{\mathbf{f}}_k^T \mathbf{P}_{k-1} \tilde{\mathbf{f}}_k \right]^{-1} \left(\tilde{\mathbf{f}}_k^T \hat{\boldsymbol{\theta}}_{k-1} - y(k) \right), \\ \mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1} \tilde{\mathbf{f}}_k \left[1 + \tilde{\mathbf{f}}_k^T \mathbf{P}_{k-1} \tilde{\mathbf{f}}_k \right]^{-1} \tilde{\mathbf{f}}_k^T \mathbf{P}_{k-1}. \end{cases} \quad (27)$$

где $\hat{\boldsymbol{\theta}}_k^T = (\hat{\boldsymbol{\theta}}^T; \hat{\boldsymbol{\lambda}}^T)$, $\tilde{\mathbf{f}}_k^T = (\mathbf{h}_k^T; \mathbf{v}_k^T)$.

Отметим, что для получения начальных значений матрицы \mathbf{P}_{N_0} и $\hat{\boldsymbol{\theta}}_{N_0}$, следует по результатам первых N_0 опытов получить $\hat{\boldsymbol{\theta}}_{нач1}$ и $\mathbf{P}_{нач1}$ — МНК-оценки параметров модели (5) по (15), а затем перейти к искомым величинам:

$$\hat{\boldsymbol{\theta}}_{N_0} = \left(\hat{\boldsymbol{\theta}}_{нач1}^T, \vec{\mathbf{0}}^T \right)^T, \quad \hat{\mathbf{P}}_{N_0} = \begin{pmatrix} \hat{\mathbf{P}}_{нач1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad (28)$$

где $\vec{\mathbf{0}}, \mathbf{0}$ — нулевые вектор и матрица соответствующей размерности, \mathbf{I} — единичная матрица соответствующей размерности. $\xi_{N_0} = \varepsilon_{N_0} = 0$.

2.3.7. Рекуррентный метод инструментальной переменной, комбинированный с расширенным матричным методом

Простой МИП не предусматривает возможность оценивания параметров шума, что ухудшает функционирование метода при условии действия на объект выходной аддитивной помехи. Одним из способов устранения этого недостатка является модификация МИП — метод инструментальной переменной, комбинированный с расширенным матричным методом.

По аналогии с МИП и РММ, введем вектор инструментальной переменной:

$$\tilde{\mathbf{z}}_k^T = (u(N), \dots, u(N-n), x(N-1), \dots, x(N-n),$$

$$\hat{\xi}(N-1), \dots, \hat{\xi}(N-s), \hat{\varepsilon}(N-1), \dots, \hat{\varepsilon}(N-r)), \quad (29)$$

где $x(k)$ — выход некоторого фиксированного фильтра [1, 2, 4], см. (24).

Рекуррентные соотношения (27) примут вид:

$$\begin{cases} \hat{\theta}_k = \hat{\theta}_{k-1} + \mathbf{P}_{k-1} \tilde{\mathbf{z}}_k \left(1 + \mathbf{f}_k^T \mathbf{P}_{k-1} \tilde{\mathbf{z}}_k\right)^{-1} \hat{\varepsilon}(k), \\ \mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1} \tilde{\mathbf{z}}_k \left(1 + \mathbf{f}_k^T \mathbf{P}_{k-1} \tilde{\mathbf{z}}_k\right)^{-1} \mathbf{f}_k^T \mathbf{P}_{k-1}, \\ \hat{\varepsilon}(k) = y(k) - \mathbf{f}_k^T \hat{\theta}_{k-1}. \end{cases} \quad (30)$$

Начальные значения матриц \mathbf{P}_{N_0} и $\hat{\theta}_{N_0}$ получаем по выражению (28).

2.3. Статистический анализ результатов идентификации

Статистический анализ результатов идентификации одномерных линейных стационарных динамических систем включает получение значений критериев качества [1–4] и проверку гипотез значимости полученных оценок коэффициентов и адекватности построенной модели экспериментальным данным.

2.3.1. Проверка адекватности

Проверка гипотезы адекватности выбранной модели осуществляется по критерию Фишера (F -критерий):

$$F_{\text{расч}} = \frac{S_{\text{ад}}}{S_y}, \quad (31)$$

где $F_{\text{расч}}$ — расчетное значение критерия Фишера для выходной переменной системы; $S_{\text{ад}} = \frac{1}{N-m} \sum_{i=1}^N (y(i) - y_{\text{мод}}(i))^2$ — дисперсия адекватности;

N — число дискретных значений выходов модели; m — число значимых параметров модели; $y(i)$ — реальное значение выхода объекта в момент времени $i\Delta t$; $y_{\text{мод}}(i)$ — значение выхода объекта в момент времени $i\Delta t$, рас-

считанное по модели; $S_y = \frac{1}{N-m} \sum_{i=1}^N (y(i) - \bar{y})^2$ — оценка дисперсии значе-

ний реального выходного сигнала; $\bar{y} = \frac{1}{N} \sum_{i=1}^N y(i)$ — выборочное среднее выходного сигнала.

Модель считается адекватной экспериментальным данным, если выполняется неравенство:

$F_{\text{расч}} < F_{\text{табл}}$, где $F_{\text{табл}}$ — табличное значение критерия Фишера при выбранном уровне значимости и двух степенях свободы: числителя $\nu_1 = N - m$ и знаменателя $\nu_2 = N - 1$.

2.3.2. Проверка значимости коэффициентов

Проверка значимости коэффициентов (не равенства их нулю) осуществляется по статистическому критерию Стьюдента (t -критерию):

$$t_{j \text{ расч}} = \frac{|\hat{a}_j|}{S_{a_j}}, \quad (32)$$

где $t_{j \text{ расч}}$ — расчетное значение критерия Стьюдента для оценки параметра выбранной модели $\hat{a}_j, j = \overline{1, m}, m$ — число оцениваемых коэффициентов модели; S_{a_j} — оценка дисперсии параметра a_j , определяемая, например, по матрице дисперсии-ковариации (16).

Гипотезу о значимости оценки следует отвергнуть, если выполняется условие:

$$t_{j \text{ расч}} < t_{\text{табл}},$$

где $t_{\text{табл}}$ — табличное значение критерия Стьюдента при выбранном уровне значимости и числе степеней свободы $\nu = N - m$; m — число оцениваемых коэффициентов модели.

Незначимость оценки \hat{a}_j означает, что соответствующий коэффициент может быть исключен из рассматриваемой модели, так как гипотеза о том, что $a_j = 0$ не отвергается.

2.4. Взаимные преобразования моделей

2.4.1. Преобразование разностного уравнения в дискретную модель пространства состояний

Вид преобразования разностного уравнения, заданного моделью (3), в дискретную модель пространства состояний (7) осуществляется по выражениям [1–3]:

$$\mathbf{A} = \begin{pmatrix} -a_1 a_0^{-1} & 1 & 0 & \cdots & 0 \\ -a_2 a_0^{-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} a_0^{-1} & 0 & 0 & \cdots & 1 \\ -a_n a_0^{-1} & 0 & 0 & \cdots & 0 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_1 - a_1 a_0^{-1} b_0 \\ b_2 - a_2 a_0^{-1} b_0 \\ \vdots \\ b_{n-1} - a_{n-1} a_0^{-1} b_0 \\ b_n - a_n a_0^{-1} b_0 \end{pmatrix}, \mathbf{c}^T = (a_0^{-1}, \mathbf{0}), d = a_0^{-1} b_0, \quad (33)$$

где $\mathbf{0}$ — нулевая вектор-строка, размерностью $1 \times (n-1)$.

Отметим, что данное преобразование выполнимо только при ненулевом значении коэффициента a_0 модели (3).

2.4.2. Преобразование дифференциального уравнения в непрерывную модель пространства состояний

Преобразование дифференциального уравнения, заданного моделью (1), в непрерывную модель пространства состояний (6) осуществляется по выражениям [1-3]:

$$\mathbf{T} = \begin{pmatrix} -a_{n-1}a_n^{-1} & 1 & 0 & \cdots & 0 \\ -a_{n-2}a_n^{-1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_1a_n^{-1} & 0 & 0 & \cdots & 1 \\ -a_0a_n^{-1} & 0 & 0 & \cdots & 0 \end{pmatrix}, \mathbf{g} = \begin{pmatrix} b_{n-1} - a_{n-1}a_n^{-1}b_n \\ b_{n-2} - a_{n-2}a_n^{-1}b_n \\ \vdots \\ b_1 - a_1a_n^{-1}b_n \\ b_0 - a_0a_n^{-1}b_n \end{pmatrix}, \mathbf{h}^T = (a_n^{-1}, \mathbf{0}), k = a_n^{-1}b_n, \quad (34)$$

где $\mathbf{0}$ — нулевая вектор-строка, размерностью $1 \times (n-1)$.

Отметим, что данное преобразование выполнимо только при ненулевом значении коэффициента a_n модели.

2.4.3. Преобразование дискретной модели пространства состояний в разностное уравнение

Преобразование дискретной модели пространства состояний (7) в разностное уравнение (3) осуществляется по выражениям [1-3, 8]:

$$\mathbf{a}^{*T} \tilde{\mathbf{y}}(k) = \mathbf{b}^{*T} \tilde{\mathbf{u}}(k), \quad (35)$$

где $\mathbf{a}^* = \begin{pmatrix} 1 \\ (-\mathbf{c}^T \mathbf{A}^n \mathbf{P}^{-1})^T \uparrow \end{pmatrix}$ и $\mathbf{b}^* = \begin{pmatrix} d \\ \mathbf{Pb} - (\mathbf{c}^T \mathbf{A}^n \mathbf{P}^{-1} \mathbf{Q})^T \uparrow \end{pmatrix}$ совпадают соответственно с векторами $\tilde{\mathbf{a}}$ и $\tilde{\mathbf{b}}$ разностного уравнения (5),

$$\tilde{\mathbf{y}}(k) = \begin{pmatrix} y(k+n) \\ \hat{\mathbf{y}}(k) \uparrow \end{pmatrix}, \tilde{\mathbf{u}}(k) = \begin{pmatrix} u(k+n) \\ \hat{\mathbf{u}}(k) \uparrow \end{pmatrix}, \quad \hat{\mathbf{y}}(k) = \begin{pmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+n-1) \end{pmatrix}; \mathbf{P} = \begin{pmatrix} \mathbf{c}^T \\ \mathbf{c}^T \mathbf{A} \\ \vdots \\ \mathbf{c}^T \mathbf{A}^{n-1} \end{pmatrix};$$

$$\mathbf{Q} = \begin{pmatrix} d & 0 & \cdots & 0 \\ \mathbf{c}^T \mathbf{b} & d & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{c}^T \mathbf{A}^{n-2} \mathbf{b} & \mathbf{c}^T \mathbf{A}^{n-3} \mathbf{b} & \cdots & d \end{pmatrix}; \hat{\mathbf{u}}^T(k) = (u(k), u(k+1), \dots, u(k+n-1)), \uparrow -$$

матричный оператор, который введем следующим образом [8].

Пусть, \mathbf{M} — матрица размерностью $l \times n$, где l — число строк, n — число столбцов, тогда:

$$\forall \mathbf{M} = \{m_{i,j}, j = \overline{1, n}, i = \overline{1, l}\}, \exists \tilde{\mathbf{M}} = \mathbf{M} \Downarrow = \{\tilde{m}_{i,j}, j = \overline{1, n}, i = \overline{1, l}\},$$

$$\tilde{m}_{i,j} = m_{k,j}, j = \overline{1, n}, i = \overline{1, l}, k = m - i - 1.$$

Отметим далее, что преобразование (35) возможно только при невырожденной матрице \mathbf{P} . Это означает, что данное преобразование возможно, если и только если линейная одномерная динамическая система (7) полностью наблюдаема [3].

2.4.4. Преобразование непрерывной модели пространства состояний в дифференциальное уравнение

Преобразование непрерывной модели пространства состояний (6) в дифференциальное уравнение (1) осуществляется по выражениям [1-3,8]:

$$\tilde{\mathbf{a}}^{*T} \tilde{\mathbf{y}}(t) = \tilde{\mathbf{b}}^{*T} \tilde{\mathbf{u}}(t), \quad (36)$$

где $\tilde{\mathbf{a}}^{*T} = (-\mathbf{h}^T \mathbf{T}^n \mathbf{P}^{-1}, 1)$; $\tilde{\mathbf{b}}^{*T} = ([\mathbf{P} \mathbf{g}] \Downarrow^T - \mathbf{h}^T \mathbf{T}^n \mathbf{P}^{-1} \mathbf{Q}, k)$ совпадают соответственно с векторами $\tilde{\mathbf{a}}$ и $\tilde{\mathbf{b}}$ дифференциального уравнения (1),

$$\tilde{\mathbf{y}}(k) = \begin{pmatrix} \tilde{\mathbf{y}}(t) \\ \frac{d^n y(t)}{d^n t} \end{pmatrix}, \tilde{\mathbf{u}}(k) = \begin{pmatrix} \tilde{\mathbf{u}}(t) \\ \frac{d^n u(t)}{d^n t} \end{pmatrix}, \tilde{\mathbf{y}}(t) = \begin{pmatrix} y(t) \\ \frac{dy(t)}{dt} \\ \vdots \\ \frac{d^{n-1} y(t)}{d^{n-1} t} \end{pmatrix}; \mathbf{P} = \begin{pmatrix} \mathbf{h}^T \\ \mathbf{h}^T \mathbf{T} \\ \vdots \\ \mathbf{h}^T \mathbf{T}^{n-1} \end{pmatrix};$$

$$\mathbf{Q} = \begin{pmatrix} k & 0 & \dots & 0 \\ \mathbf{h}^T \mathbf{g} & k & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{h}^T \mathbf{T}^{n-2} \mathbf{g} & \mathbf{h}^T \mathbf{T}^{n-3} \mathbf{g} & \dots & k \end{pmatrix}; \tilde{\mathbf{u}}^T(t) = \left(u(t), \frac{du(t)}{dt}, \dots, \frac{d^{n-1} u(t)}{d^{n-1} t} \right).$$

Преобразование (36) возможно только при невырожденной матрице \mathbf{P} . Это означает, что данное преобразование возможно, если и только если линейная одномерная динамическая система (6) полностью наблюдаема [3].

2.4.5. Преобразование непрерывной модели пространства состояний в дискретную модель пространства состояний

Преобразование непрерывной модели пространства состояний (6) в дискретную модель пространства состояний (7) осуществим в соответствии с [3]:

$$\mathbf{A} = e^{\mathbf{T}\Delta t} = \sum_{k=0}^{\infty} \frac{\mathbf{T}^k \Delta t^k}{k!}, \mathbf{b} = \Delta t \sum_{k=0}^{\infty} \frac{\mathbf{T}^k \Delta t^k}{(k+1)!} \mathbf{g}, \mathbf{c}^T = \mathbf{h}^T, d = k. \quad (37)$$

2.4.6. Преобразование дискретной модели пространства состояний в непрерывную модель пространства состояний

Преобразование дискретной модели пространства состояний (7) в непрерывную модель пространства состояний (6) осуществим в соответствии с [3]:

$$\begin{cases} \mathbf{T} = \frac{1}{\Delta t} \ln \mathbf{A} = \frac{1}{\Delta t} \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(\mathbf{A} - \mathbf{I})^k}{k}, \\ \mathbf{h}^T = \mathbf{c}^T, k = d, \\ \mathbf{g} = \frac{1}{\Delta t} \sum_{k=0}^{\infty} (-1)^k \frac{(\mathbf{A} - \mathbf{I})^k}{k+1} \mathbf{b}, \end{cases} \quad (38)$$

где \mathbf{I} — единичная матрица размерности $n \times n$.

Недостатком выражения (38) является его медленная сходимость, по сравнению с (37), поэтому предпочтительнее использовать рекуррентную формулу Ньютона

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{I} + \mathbf{A}e^{-\mathbf{X}_i}, \quad (39)$$

где $\mathbf{X}_{i+1}, \mathbf{X}_i$ — значение $\ln \mathbf{A}$ на $(i+1)$ -ом и i -ом шаге соответственно; \mathbf{I} —

единичная матрица размерности $n \times n$; $e^{-\mathbf{X}_i} = \sum_{k=0}^{\infty} \frac{[-\mathbf{X}_i]^k}{k!}$;

$\mathbf{X}_0 = \mathbf{0}; \mathbf{X}_1 = \mathbf{A} - \mathbf{I}, \mathbf{0}$ — нулевая матрица размерности $n \times n$.

2.4.7. Преобразование разностного уравнения в дискретную передаточную функцию

При выполнении ряда условий [1], в том числе при нулевых начальных условиях, дискретная передаточная функция (9) примет вид:

$$w(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_n z^{-n}}{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}},$$

коэффициенты которой численно равны соответствующим коэффициентам модели (3).

2.4.8. Преобразование дифференциального уравнения в непрерывную передаточную функцию

При выполнении ряда условий [1], в том числе при нулевых начальных условиях, непрерывная передаточная функция (8) примет вид

$$w(s) = \frac{b_0 + b_1s + \dots + b_ns^n}{a_0 + a_1s + \dots + a_ns^n},$$

коэффициенты которой численно равны соответствующим коэффициентам из (1).

3. Идентификация линейных нестационарных динамических систем

Сущность первого рассмотренного в работе подхода заключается в том, что выбирается временной интервал ΔT , на котором принимается предположение о том, что идентифицируемый объект не изменяет свои динамические свойства, и осуществляется структурная и параметрическая идентификация объекта по стационарной модели (5). Полученные оценки принимаются в качестве оценок порядка нестационарного разностного уравнения, соответствующих данному временному интервалу.

Затем осуществляется переход к следующему временному интервалу, и вычисляются оценки порядка стационарной на данном интервале модели. Процесс повторяют на всем времени наблюдения объекта.

Другой подход реализуют рекуррентные методы идентификации, которые позволяют находить в реальном времени оценки коэффициентов стационарной модели в конце временного интервала. При надлежащем выборе ширины временного окна такие оценки будут близки к значениям коэффициентов нестационарной модели в конце данного временного интервала.

4. Маскирование коэффициентов моделей динамических систем

Если известно, что все компоненты исследуемой системы функционируют в непрерывном времени, то коэффициент b_0 разностного уравнения (3) следует принять равным нулю [3]. Такое решение повлечет за собой изменения в алгоритмах и методах, рассмотренных выше. Опишем кратко такие изменения:

а) уменьшится на единицу размерность векторов оцениваемых параметров модели, включавших коэффициент b_0 — данный коэффициент будет исключен из них;

б) изменится размерность матриц F из (14)–(16), (20)–(22), (24) и Z из (21), (22), (24) и векторов z_k (22) и f_k (18), (23): крайний левый столбец (элемент) будет удален;

в) изменится размерность матриц \mathbf{P} из (14)–(16), (20)–(22), (24): столбец и строка, соответствующие исключаемому коэффициенту будут удалены;

г) уменьшится на единицу число опытов, необходимых для начала процедур рекуррентного оценивания параметров разностного уравнения.

Обобщив данные изменения, введем понятие маскирования как процедуры исключения из модели произвольных незначимых коэффициентов до начала и в процессе процедуры идентификации. Это позволит учитывать априорную незначимость ряда коэффициентов модели.

Опишем, какие изменения произойдут при исключении из модели (5) произвольного коэффициента:

а) уменьшится на единицу размерность векторов оцениваемых параметров θ модели (5), включавших исключаемый коэффициент — данный коэффициент будет исключен из них;

б) изменится размерность матриц \mathbf{F} из (14)–(16), (20)–(22), (24) и \mathbf{Z} из (21), (22), (24) и векторов \mathbf{z}_k (22) и \mathbf{f}_k (18), (23): столбец (элемент), соответствующий исключаемому коэффициенту будет удален;

в) изменится размерность матриц \mathbf{P} из (14)–(16), (20)–(22), (24): столбец и строка, соответствующие исключаемому коэффициенту будут удалены;

г) уменьшится на единицу число опытов, необходимых для начала процедур рекуррентного оценивания параметров разностного уравнения (3).

Аналогичную процедуру нужно выполнить для маскирования коэффициентов расширенных моделей (3), (17) и (3), (25).

Для разностного уравнения (3) введем матрицу-маску **mask** размерности $2 \times n$. Первый ее столбец содержит значения порядков левой g и правой p частей разностного уравнения (3) (в частном случае они могут быть равны: $g=p=n$). Значение каждого другого элемента матрицы **mask**, равное нулю показывает, что соответствующий коэффициент разностного уравнения (3) будет исключен из модели до начала и в процессе процедуры идентификации. Значение того же элемента матрицы **mask**, равное 1 показывает, что будут находиться оценки соответствующего коэффициента разностного уравнения.

В табл. 1 показана структура матрицы маски **mask**, дополненной для описания маскирования расширенных моделей, используемых в ряде рассмотренных методов параметрической идентификации.

Таблица 1.
Матрица-маска коэффициентов модели.

Индекс элемента	Область допустимых значений	Описание
(1,1)	целое, $0 \leq p \leq g$	g — порядок левой части разностного уравнения (3) (знаменателя передаточной функции).
(2,1)- ($g+2,1$)	только 1 или 0	Маска коэффициентов левой части разностного уравнения: если коэффициент данной маски равен 1, то соответствующий ему коэффициент левой части разностного уравнения оценивается, иначе (равен 0) — соответствующий коэффициент не оценивается и признается априори равным нулю. Элемент (2,1) соответствует коэффициенту α_0 модели (3), элемент (3,1) — коэффициенту α_1 и т. п. до коэффициента α_g . Коэффициент α_0 всегда принимается априори равным 1, не зависимо от значения элемента (2,1).
(1,2)	целое, $0 \leq p \leq g$	p — порядок правой части разностного уравнения (3) (числителя передаточной функции системы).
(2,2)- ($p+2,2$)	только 1 или 0	Маска коэффициентов правой части разностного уравнения: если коэффициент данной маски равен 1, то соответствующий ему коэффициент правой части разностного уравнения оценивается, иначе (равен 0) — соответствующий коэффициент не оценивается и признается априори равным нулю. Элемент (2,2) соответствует коэффициенту β_0 модели (3), элемент (3,2) — коэффициенту β_1 и т. п. до коэффициента β_p .
(1,3)	целое, $r \geq 0$	r — порядок знаменателя передаточной функции модели шума см. (17) из [1]
(2,3)- ($r+2,3$)	только 1 или 0	Маска коэффициентов знаменателя передаточной функции модели шума: если коэффициент данной маски равен 1, то соответствующий ему коэффициент модели шума используется при фильтрации во внутреннем цикле ОМНК, иначе (равен 0) — соответствующий коэффициент не используется (признается априори равным нулю). Элемент (2,3) соответствует коэффициенту d_0 модели (17), элемент (3,3) — коэффициенту d_1 и т. п. до коэффициента d_r . Коэффициент d_0 всегда принимается априори равным 1, не зависимо от значения элемента (2,3).
(1,4)	целое, $0 \leq s \leq r_s$	r_s — порядок знаменателя передаточной функции модели шума (25)
(2,4)- ($s+2,4$)	только 1 или 0	Маска коэффициентов знаменателя передаточной функции модели шума: если коэффициент данной маски равен 1, то соответствующий ему коэффициент модели шума оценивается, иначе (равен 0) — соответствующий коэффициент не оценивается и признается априори равным нулю. Элемент (2,4) соответствует коэффициенту d_0 модели (25), элемент (3,4) — коэффициенту d_1 и т. п. до коэффициента d_s . Коэффициент d_0 всегда принимается априори равным 1, не зависимо от значения элемента (2,4).

Индекс элемента	Область допустимых значений	Описание
(1,5)	целое, $0 \leq s \leq r_s$	s — порядок числителя передаточной функции модели шума (25)
(2,5)- ($s+2,5$)	только 1 или 0	Маска коэффициентов числителя передаточной функции модели шума: если коэффициент данной маски равен 1, то соответствующий ему коэффициент модели шума оценивается, иначе (равен 0) — соответствующий коэффициент не оценивается и признается априори равным нулю. Элемент (2,5) соответствует коэффициенту c_0 модели (25), элемент (3,5) — коэффициенту c_1 и т. п. до коэффициента c_s .

Примечание: значение не указанных в табл.1 элементов матрицы-маски игнорируется (может быть любым).

5. Описание функций пакета программ

5.1. Определение порядка разностного уравнения

В пакете программ реализованы две функции для определения порядка разностного уравнения (3): StructMetodsV1 — по функции потерь (11) и StructMetodsMoments — по матрице моментов (12)–(13).

Также реализована функция StructIdentify, которая оценивает порядок разностного уравнения по интервалам стационарности нестационарной динамической системы [1,2] и использует для этого одну из функций StructMetodsV1 или StructMetodsMoments.

5.1.1. StructMetodsV1

Варианты вызова:

[V1, returnflag, order] = StructMetodsV1(U, Y, MaxG, showFlag, outType)

[V1, returnflag] = StructMetodsV1(U, Y, MaxG, showFlag)

[V1, returnflag] = StructMetodsV1(U, Y, MaxG)

[V1] = StructMetodsV1(U, Y, MaxG)

Описание:

Осуществляет структурную идентификацию по функции потерь (11) для разностного уравнения (3).

Исходные данные:

U — вектор значений входного сигнала \vec{u} , размерностью $1 \times N$:

$$\vec{u}^T = (u(0), u(\Delta T), \dots, u((N-1)\Delta T)),$$

где N — число измерений.

Y — вектор значений выходного сигнала \vec{y} , размерностью $1 \times N$:

$$\vec{y}^T = (y(0), y(\Delta T), \dots, y((N-1)\Delta T)).$$

MaxG — скаляр — максимальный рассматриваемый порядок системы.

showFlag — скаляр, если showFlag ≠ 0, то отображает результат идентификации в виде графика, если showFlag=0 (или при вызове по умолчанию — варианты 3 и 4) — не отображает и не выводит сообщений на экран.

outType — скаляр — тип выводимых значений: если outType=0 или не указан, то выводятся значения функции потерь, если outType=1, то выводятся десятичные логарифмы функции потерь.

Выходные данные:

V1 — если outType не указан при вызове или outType=0, то $V1 = \vec{V}_1^T$ — вектор-строка значений функции потерь (11) для различных порядков системы, размерностью MaxG x 1:

$$\vec{V}_1^T = (V_1(1), V_1(2), \dots, V_1(MaxG)).$$

При варианте вызова 1 при outType=1: $V1 = \log_{10} \vec{V}_1^T$.

При варианте вызова 1 order ← значение выбранной оценки порядка системы.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag ← 0.

Реализация: MEX-файл: StructMethodsV1.dll; библиотеки CID.dll, MatrixM.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[0.340;0.344;0.350;0.351;0.351;0.354;0.362;0.363;0.368;0.366];
```

```
Y=[0.432;0.410;0.371;0.357;0.345;0.383;0.424;0.399;0.354;0.325];
```

```
[V1, fl] = StructMethodsV1(U,Y,3)
```

```
V1 = 0.00066002 0.00026373 2.7964e-005
```

```
fl = 1
```

5.1.2 StructMethodsMoments

Варианты вызова:

```
[rdQ, returnflag, order] = StructMethodsMoments(U, Y, MaxG, showFlag, outType)
```

```
[rdQ, returnflag] = StructMethodsMoments(U, Y, MaxG, showFlag)
```

```
[rdQ, returnflag] = StructMethodsMoments(U, Y, MaxG)
```

```
[rdQ] = StructMethodsMoments(U, Y, MaxG)
```

Описание:

Осуществляет структурную идентификацию по матрице моментов (12)–(13) для разностного уравнения (3).

Исходные данные:

U — вектор значений входного сигнала \vec{u} , размерностью $1 \times N$.

Y — вектор значений выходного сигнала \vec{y} , размерностью $1 \times N$.

MaxG — скаляр — максимальный рассматриваемый порядок системы.

showFlag — скаляр, если showFlag $\neq 0$, то отображает результат идентификации в виде графика, если showFlag=0 (или при вызове по умолчанию — варианты 3 и 4) — не отображает и не выводит сообщений на экран.

outType — скаляр — тип выводимых значений: если outType=0 или не указан, то выводятся значения функции потерь, если outType=1, то выводятся десятичные логарифмы функции потерь.

Выходные данные:

rdQ — если outType не указан при вызове или outType=0, то $\text{rdQ} = \vec{d}_q^T$ — вектор-строка относительных определителей (13) системы, размерностью MaxG \times 1:

$$\vec{d}_q^T = (\text{rel det } Q(\vec{u}, \vec{y}, 1), \text{rel det } Q(\vec{u}, \vec{y}, 2), \dots, \text{rel det } Q(\vec{u}, \vec{y}, \text{MaxG})),$$

При варианте вызова 1 при outType=1: $\text{rdQ} = |\log_{10} \vec{d}_q^T|$.

При варианте вызова 1 order \leftarrow значение выбранной оценки порядка системы.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag \leftarrow 0.

Реализация: MEX-файл: StructMetodsMoments.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[0.340;0.344;0.350;0.351;0.351;0.354;0.362;0.363;0.368;0.366];
Y=[0.432;0.410;0.371;0.357;0.345;0.383;0.424;0.399;0.354;0.325];
[reldetQ, fl] = StructMetodsMoments(U,Y,4)
reldetQ = 0.0085789 1.2796e-009 9.4988e-018 -3.2156e-047
fl = 1
```

5.1.3. StructIdentify

Варианты вызова:

```
[vals, Orders, returnflag]=StructIdentify (U, Y, MaxG, logFlag, wTime, shift, MetNum)
[vals, Orders]=StructIdentify (U, Y, MaxG, logFlag, wTime, shift, MetNum)
```

Описание:

Осуществляет структурную идентификацию одним из методов StructMetodsMoments или StructMetodsV1 по временным окнам для стационарного разностного уравнения (3).

Исходные данные:

U — вектор значений входного сигнала \vec{u} , размерностью $1 \times N$.

Y — вектор значений выходного сигнала \vec{y} , размерностью $1 \times N$.

MaxG — скаляр — максимальный рассматриваемый порядок системы.

logFlag — скаляр. Если logFlag = 0, то в качестве результатов идентификации выводятся значения определителей матрицы моментов (или функции потерь), а если logFlag = 1 — выводятся значения десятичных логарифмов определителей матрицы моментов (или значений функции потерь).

wTime — ширина временного окна в тактах (1 такт = Δt с),

shift — величина приращения при сдвиге границ временного окна, в тактах,

MetNum — код используемого метода структурной идентификации, возможные значения:

1 — по функции потерь (см. StructMethodsV1);

2 — по матрице моментов (см. StructMethodsMoments).

Выходные данные:

vals — матрица настройки значений функции потерь (определителей матрицы моментов) или их логарифмы.

Orders — вектор настройки выбранных на каждом окне значений порядков модели.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag \leftarrow 0.

Реализация: MEX-файл: StructIdentify.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
```

```
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
```

```
% StructMethodsV1:
```

```
[vals, Orders, fl]= StructIdentify (U, Y, 4, 1, 400, 40, 1);
```

```
fl=1
```

```
% StructMethodsMoments:
```

```
[vals, Orders, fl]= StructIdentify (U, Y, 4, 1, 400, 40, 2);
```

```
fl=1
```

5.2. Определение оценок коэффициентов разностного уравнения

В пакете программ реализованы семь функций для определения оценок коэффициентов разностного уравнения: MetodLSQ — МНК (15)–(16), MetodGLSQ — ОМНК (алгоритм А), MetodLSQOnline — рМНК (19)–(20), MetodIV — МИП (21)–(22), MetodIVOnline — рМИП (23)–(24), MetodEMMOnline — рРММ (26)–(28) и MetodIVEMMOnline — комбинированный рРММ и рМИП (30).

Также реализована функция ParamIdentify, которая оценивает коэффициенты модели разностного уравнения по интервалам стационарности

нестационарной динамической системы [1,2] и использует для этого одну из вышеперечисленных функций.

5.2.1. MetodLSQ

Варианты вызова:

`[alfa, beta, e, estC, Cov, returnflag] = MetodLSQ(U, Y, mask, showFlag)`

`[alfa, beta, e, estC, Cov, returnflag] = MetodLSQ(U, Y, mask)`

`[alfa, beta, e, estC, Cov] = MetodLSQ(U, Y, mask)`

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) методом наименьших квадратов (15–16).

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы `mask` равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Исходные данные:

`U` — вектор значений входного сигнала u , размерностью $1 \times N$.

`Y` — вектор значений выходного сигнала y , размерностью $1 \times N$.

`mask` — порядки разностного уравнения и матрица-маска параметров (первые две строки матрицы из табл. 1), в т.ч.:

`g = mask.(1,2)` — порядок знаменателя передаточной функции системы,

`p = mask.(1,2)` — порядок числителя передаточной функции системы, $0 \leq p \leq g$.

`showFlag` — скаляр, если `showFlag` $\neq 0$, то отображает в отдельных окнах результаты идентификации, иначе (или при вызове по умолчанию — варианты 2 и 3) — не отображает и не выводит сообщений на экран.

Выходные данные:

`alfa` — вектор α — оценок параметров модели (3), размерностью $1 \times (g+1)$,

`beta` — вектор β — оценок параметров модели (3), размерностью $1 \times (g+1)$, если $p < g$, то в конец добавлены нули,

`e` — вектор невязок e из (5), расширенный до размерности $1 \times N$ добавлением в начало вектора e из (5) нулей,

`estC` — вектор полученных оценок $\hat{\theta}$ коэффициентов модели (5), полученный по (15) из [1], размерность может быть меньше, чем $g+p+1$, в зависимости от маски `mask`.

`Cov` — матрица дисперсии-ковариации оценок $\text{cov}(\hat{\theta})$ (16), в которую включены оценки дисперсии и ковариации только тех коэффициентов, которые составляют вектор `estC`.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag ← 0.

Реализация: MEX-файл: MetodLSQ.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[0.340;0.344;0.350;0.351;0.351;0.354;0.362;0.363;0.368;0.366];
```

```
Y=[0.432;0.410;0.371;0.357;0.345;0.383;0.424;0.399;0.354;0.325];
```

```
mask=[1 1 1;1 0 1];
```

```
[alfa,beta,e,estC,Cov,fl]=MetodLSQ(U,Y,mask,0)
```

```
alfa =
```

```
1.0000
```

```
-0.6782
```

```
beta =
```

```
0
```

```
0.3170
```

```
e =
```

```
0
```

```
0.0092
```

```
-0.0161
```

```
-0.0056
```

```
-0.0084
```

```
0.0378
```

```
0.0520
```

```
-0.0033
```

```
-0.0317
```

```
-0.0317
```

```
estC =
```

```
0.3170
```

```
0.6782
```

```
Cov =
```

```
0.0769 -0.0700
```

```
-0.0700 0.0642
```

```
fl = 1
```

5.2.2. MetodGLSQ

Варианты вызова:

```
[alfa, beta, e, estC, Cov, resPars, returnflag, SetsPar] = MetodGLSQ(U, Y, mask, MaxIter, eps, showFlag)
```

```
[alfa, beta, e, estC, Cov, resPars, returnflag] = MetodGLSQ(U, Y, mask, MaxIter, eps)
```

```
[alfa, beta, e, estC, Cov, resPars] = MetodGLSQ(U, Y, mask, MaxIter, eps)
```

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) обобщенным методом наименьших квадратов (по алгоритму А).

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы mask равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Цикл ОМНК-оценивания заканчивается, когда значение каждого элемента вектора $|\hat{\theta}_i - \hat{\theta}_{i-1}|$ становится меньше eps или выполняется условие $i > \text{MaxIter}$.

Исходные данные:

U — вектор значений входного сигнала u , размерностью $1 \times N$.

Y — вектор значений выходного сигнала y , размерностью $1 \times N$.

mask — порядки разностного уравнения и матрица-маска параметров (первые три строки матрицы из табл. 1), в т.ч.:

$g = \text{mask}(1,2)$ — порядок знаменателя передаточной функции системы,

$p = \text{mask}(1,2)$ — порядок числителя передаточной функции системы,

$r = \text{mask}(1,3)$ — порядок знаменателя передаточной функции модели шума см. (3.10) из [1], $0 \leq p \leq g, r > 0$.

MaxIter — наибольшее число выполнения внутреннего цикла фильтрации-оценивания: $i > \text{MaxIter}$, где i — текущее количество итераций в цикле — второе условие завершения оценивания-фильтрации. $\text{MaxIter} > 2$.

eps — пороговая величина критерия завершения вычислений $|\hat{\theta}_i - \hat{\theta}_{i-1}| < \text{eps}$.

showFlag — скаляр, если $\text{showFlag} \neq 0$, то отображает в отдельных окнах результаты идентификации, иначе (или при вызове по умолчанию — варианты 2 и 3) — не отображает и не выводит сообщений на экран.

Выходные данные:

alfa — вектор α — оценок параметров модели (3), размерностью $1 \times (g+1)$,

beta — вектор β — оценок параметров модели (3), размерностью $1 \times (g+1)$, если $p < g$, то в конец добавлены нули,

e — вектор невязок e из (5), расширенный до размерности $1 \times N$ добавлением в начало вектора e из (5) нулей,

estC — вектор полученных оценок $\hat{\theta}$ коэффициентов модели (5), полученный по алгоритму А, размерность может быть меньше, чем $g+p+1$, в зависимости от маски mask .

Cov — матрица дисперсии-ковариации оценок $\text{cov}(\hat{\theta})$ (16), в которую включены оценки дисперсии и ковариации только тех коэффициентов, которые составляют вектор estC .

resPars — матрица выходных параметров процедуры ОМНК оценивания:

$\text{resPars.}(1,1) \leftarrow 1$ — в случае успешного завершения по условию $|\hat{\theta}_i - \hat{\theta}_{i-1}| < \text{eps}$,

$\text{resPars.}(1,1) \leftarrow 0$ — если завершение процедуры оценивания произошло по условию $i > \text{MaxIter}$,

$\text{resPars.}(2,1) \leftarrow i$ — число произведенных итераций в цикле.

$\text{resPars.}(1,2) - \text{resPars.}(x,2)$ — элементы последней вектор-строки $|\hat{\theta}_i - \hat{\theta}_{i-1}|^T$, x — размерность вектора $\hat{\theta}$.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и $\text{returnflag} \leftarrow 0$.

Реализация: MEX-файл: MetodGLSQ.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[0.340;0.344;0.350;0.351;0.351;0.354;0.362;0.363;0.368;0.366];
Y=[0.432;0.410;0.371;0.357;0.345;0.383;0.424;0.399;0.354;0.325];
mask=[1 1 1;1 0 1;1 0 1];
[alfa,beta,e,estC,Cov,EndPars,fl]=MetodGLSQ(U,Y,mask,100,1e-6,0)
```

$\text{alfa} =$

1.0000

-2.2930

$\text{beta} =$

0

-1.3729

$e =$

0

-0.0070

0.0857

-0.0435

0.0063

0.0552

0.0495

-0.0374

-0.0577

-0.0211

$\text{estC} =$

-1.3729

2.2930

$\text{Cov} =$

0.2561 -0.2345

-0.2345 0.2153

$\text{EndPars} =$

1.0000 67.0000

-8.4771e-007 7.8944e-007

$\text{fl} = 1$

5.2.3. MetodLSQOnline

Варианты вызова:

[alfa, beta, e, estC, P, Cov, returnflag, SetPars]=MetodLSQOnline(U, Y, mask, showFlag, initC, initP)

[alfa, beta, e, estC, P, Cov, returnflag] = MetodLSQOnline (U, Y, mask, showFlag, initC, initP)

[alfa, beta, e, estC, P, Cov] = MetodLSQOnline (U, Y, mask, showFlag)

[alfa, beta, e, estC, P, Cov] = MetodLSQOnline (U, Y, mask)

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) рекуррентным методом наименьших квадратов (18)–(20).

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы mask равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Исходные данные:

U — вектор значений входного сигнала u , размерностью $1 \times N$.

Y — вектор значений выходного сигнала y , размерностью $1 \times N$.

mask — порядки разностного уравнения и матрица-маска параметров (первые две строки матрицы из табл. 1), в т. ч.:

$g = \text{mask}(1,2)$ — порядок знаменателя передаточной функции системы,

$p = \text{mask}(1,2)$ — порядок числителя передаточной функции системы, $0 \leq p \leq g$.

Если initC — пустая матрица (или не указана в параметрах вызова — варианты 3 и 4), то $\hat{\theta}_{нач}$ по формуле (19), иначе $\hat{\theta}_{нач} = \text{initC}$;

Если initP — пустая матрица (или не указана в параметрах вызова — варианты 2-4), то $P_{нач}$ по формуле (19), иначе $P_{нач} = \text{initP}$.

showFlag — скаляр, если showFlag $\neq 0$, то отображает в отдельных окнах результаты идентификации, иначе (или при вызове по умолчанию — варианты 2 и 3) — не отображает и не выводит сообщений на экран.

Выходные данные:

alfa — вектор α — оценок параметров модели (3), размерностью $1 \times (g+1)$,

beta — вектор β — оценок параметров модели (3), размерностью $1 \times (g+1)$, если $p < g$, то в конец добавлены нули,

e — вектор невязок e из (5), расширенный до размерности $1 \times N$ добавлением в начало вектора e из (5) нулей,

estC — вектор полученных оценок $\hat{\theta}$ коэффициентов модели (5), полученный по алгоритму А, размерность может быть меньше, чем $g+p+1$, в зависимости от маски mask.

P — матрица P_N (19),

Cov — матрица $\text{cov}(\hat{\theta}_N) = \mathbf{P}_N \frac{\mathbf{e}^T \mathbf{e}}{N/2}$, где размерность \mathbf{e} равна $N/2$ ($N/2$

последних значений невязок),

SetPars — матрица настройки параметров — элементов вектора $\hat{\theta}_i, i = \overline{0, N}$.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и $\text{returnflag} \leftarrow 0$.

Реализация: MEX-файл: *MetodLSQOnline.dll*; библиотеки *CID.dll*, *Matrix.dll* и *MATLABLnk.dll*.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
mask=[1 1 1 ;1 0 1];
[alfa,beta,e,estC,P,Cov,fl]= MetodLSQOnline(U,Y,mask);
alfa =
    1.0000
   -0.7102
beta = 0
    0.2797
e = [0;0;0;-0.0117;-0.0003;-0.0063;-0.0096;...];%matrix:10000x1
estC =
    0.2797
    0.7102
P = 3.0292e-004  -1.2795e-006
   -1.2795e-006   1.9025e-003
Cov =
    2.0106e-008  -8.4927e-011
   -8.4927e-011   1.2628e-007
fl = 1
```

5.2.4. MetodIV

Варианты вызова:

```
[alfa, beta, e, estC, returnflag] = MetodIV(U, Y, mask, MaxIter, eps, showFlag)
[alfa, beta, e, estC, returnflag] = MetodIV(U, Y, mask, initType, initF, showFlag)
[alfa, beta, e, estC, returnflag] = MetodIV(U, Y, mask, initType, initF)
[alfa, beta, e, estC] = MetodIV(U, Y, mask, MaxIter, eps)
```

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) методом инструментальной переменной (21)-(22).

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы *mask* равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Для задания значений инструментальной переменной использует оценки параметров модели или ее непосредственные значения.

Исходные данные:

U — вектор значений входного сигнала u , размерностью $1 \times N$.

Y — вектор значений выходного сигнала y , размерностью $1 \times N$.

$mask$ — порядки разностного уравнения и матрица-маска параметров (первые две строки матрицы из табл. 1 или три строки вызове MetodGLSQ), в т. ч.:

$g = mask.(1,2)$ — порядок знаменателя передаточной функции системы,

$p = mask.(1,2)$ — порядок числителя передаточной функции системы,

$r = mask.(1,3)$ — порядок знаменателя передаточной функции модели шума (17) (вся третья строка только при вызове ОМНК, см. MetodGLSQ, если при вызове указаны скаляры MaxIter и eps), $0 \leq p \leq g, r > 0$.

Если initType-скаляр и initF-матрица(вектор) не указаны, а указаны скаляры MaxIter и eps, то получает оценки параметров фильтра предварительным вызовом метода ОМНК (MetodGLSQ) со структурой модели, какая прописана в 1-3 строках mask.

Иначе если указан initType=1, то initF — матрица параметров линейного динамического фильтра, выход которого используется в качестве инструментальной переменной, структура которой приведена в табл. 2.

Таблица 2.
Матрица описания разностного уравнения.

Индекс элемента	Область допустимых значений	Описание
(1,1)	$g \geq 0, g \in \mathbb{Z}$	g — порядок левой части разностного уравнения (3) (знаменателя передаточной функции). Рассматривается измененная модель (3): порядки левой части g и правой p могут быть различными.
(2,1)- (g+2,1)	\mathbb{R}	Значения коэффициентов левой части разностного уравнения. Элемент (2,1) равен коэффициенту α_0 модели (3), элемент (3,1) — коэффициенту α_1 и т. п. до коэффициента α_g .
(1,2)	$0 \leq p \leq g; p, g \in \mathbb{Z}$	p — порядок правой части разностного уравнения модели (3) (числителя передаточной функции системы)
(2,2)- (p+2,2)	\mathbb{R}	Значения коэффициентов правой части разностного уравнения. Элемент (2,2) равен коэффициенту β_0 модели (3), элемент (3,2) — коэффициенту β_1 и т. п. до коэффициента β_p .

Если initType=1, то initF — вектор значений инструментальной переменной, размерностью $1 \times N$.

showFlag — скаляр, если showFlag $\neq 0$, то отображает в отдельных окнах результаты идентификации, иначе (или при вызове по умолчанию — варианты 3 и 4) — не отображает и не выводит сообщений на экран.

Выходные данные:

alfa — вектор α — оценок параметров модели (3), размерностью $1 \times (g+1)$,

beta — вектор β — оценок параметров модели (3), размерностью $1 \times (g+1)$, если $p < g$, то в конец добавлены нули,

e — вектор невязок e из (5), расширенный до размерности $1 \times N$ добавлением в начало вектора e из (5) нулей,

estC — вектор полученных оценок $\hat{\theta}$ коэффициентов модели (5), полученный по (22), размерность может быть меньше, чем $g+p+1$, в зависимости от маски mask.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: МЕХ-файл: MetodIV.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
mask=[1 1 1;1 0 1;1 1 1];
[alfa, beta, e, estC, fl]= MetodIV(U, Y, mask, 40, 1e-6)
alfa =
    1.0000
   -0.7102
beta =    0
        0.2797
e = [0;0.0004;-0.0095;-0.0088;0.0059;...] %matrix: 10000x1
estC =
    0.2797
    0.7102
fl = 1
```

5.2.5. MetodIVOnline

Варианты вызова:

```
[alfa, beta, e, estC, Cov, P, returnflag, SetPars] = MetodIVOnline (U, Y, mask, showFlag, initType, initF, initC, initP)
```

```
[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVOnline (U, Y, mask, showFlag, initType, initF, initC, initP)
```

```
[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVOnline (U, Y, mask, showFlag, MaxIter, eps, initC, initP)
```

```
[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVOnline (U, Y, mask, showFlag, initType, initF)
```

```
[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVOnline (U, Y, mask, showFlag, MaxIter, eps)
```

```
[alfa, beta, e, estC, Cov, P] = MetodIVOnline (U, Y, mask, showFlag, MaxIter, eps)
```

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) рекуррентным методом инструментальной переменной (23)–(24).

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы mask равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Исходные данные:

U — вектор значений входного сигнала u , размерностью $1 \times N$.

Y — вектор значений выходного сигнала y , размерностью $1 \times N$.

mask — порядки разностного уравнения и матрица-маска параметров (первые две строки матрицы из табл. 1 или три строки указаны скаляры MaxIter и eps), в т.ч.:

$g = \text{mask}(1,2)$ — порядок знаменателя передаточной функции системы,

$p = \text{mask}(1,2)$ — порядок числителя передаточной функции системы,

$r = \text{mask}(1,3)$ — порядок знаменателя передаточной функции модели шума (17) (вся третья строка только при вызове OMHK , см. MetodGLSQ , если при вызове указаны скаляры MaxIter и eps), $0 \leq p \leq g, r > 0$.

Для задания значений инструментальной переменной использует оценки параметров модели:

Если initType -скаляр и initF -матрица (вектор) не указаны, а указаны скаляры MaxIter и eps , то получает оценки параметров фильтра предварительным вызовом метода OMHK (MetodGLSQ) со структурой модели, какая прописана в 1-3 строках mask .

Иначе если указан $\text{initType}=1$, то initF -матрица параметров линейного динамического фильтра, выход которого используется в качестве инструментальной переменной, структуру которой см. в табл. 2.

Если $\text{initType}=1$, то initF — вектор значений инструментальной переменной, размерностью $1 \times N$.

Начальные оценки процедуры параметрического оценивания:

Если initC и initP — не указаны в параметрах вызова (варианты 4-6), то $\hat{\theta}_{\text{нач}}$ и $P_{\text{нач}}$ по формуле (24), иначе $\hat{\theta}_{\text{нач}} = \text{initC}$; $P_{\text{нач}} = \text{initP}$.

Если указана матрица SetPars , то составляет матрицу настройки параметров — элементов вектора $\hat{\theta}_N$ и выводит ее в SetPars , иначе — не составляет и не выводит.

showFlag — скаляр, если $\text{showFlag} \neq 0$, то отображает в отдельных окнах результаты идентификации, иначе — не отображает и не выводит сообщений на экран.

Выходные данные:

alfa — вектор α — оценок параметров модели (3), размерностью $1 \times (g+1)$,

beta — вектор β — оценок параметров модели (3), размерностью $1 \times (g+1)$, если $p < g$, то в конец добавлены нули,

e — вектор невязок e из (5), расширенный до размерности $1 \times N$ добавлением в начало вектора e из (5) нулей,

estC — вектор полученных оценок $\hat{\theta}$ коэффициентов модели (5), полученный по (23), размерность может быть меньше, чем $g+p+1$, в зависимости от маски mask.

P — матрица P_N (23),

Cov — матрица $\text{cov}(\hat{\theta}_N) = P_N \frac{e^T e}{N/2}$, где размерность e равна $N/2$ ($N/2$ последних значений невязок),

SetPars — матрица настройки параметров — элементов вектора $\hat{\theta}_i, i = \overline{0, N}$.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: MEX-файл: MetodIVOnline.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
mask=[1 1 1 ;1 0 1;1 1 1];
[alfa, beta, e, estC, Cov, P, fl] = MetodIVOnline (U, Y, mask, 0, 40, 1e-6)
alfa =
    1.0000
   -0.7102
beta =     0
    0.2797
e = [0;0;0;-0.0117;-0.0002; -0.0063;-0.0097;...];%matrix: 10000x1
estC =
    0.2797
    0.7102
Cov =1.0e-006 *
    0.0403  -0.0002
   -0.1002   0.3575
P = 0.00030343  -1.8108e-006
   -0.00075452   0.0026924
fl = 1
```

5.2.6. MetodEMMOnline

Варианты вызова:

```
[alfa, beta, e, estC, Cov,P,returnflag, SetPars] = MetodEMMOnline (U,Y, mask, show-Flag, initC, initP, initKsi, initEps)
```

```
[alfa, beta, e, estC, Cov,P,returnflag] = MetodEMMOnline (U,Y, mask, showFlag, initC,
```

initP, initKsi, initEps)

[alfa, beta, e, estC, Cov,P,returnflag,SetPars] = MetodEMMOnline (U,Y, mask, showFlag)

[alfa, beta, e, estC, Cov,P,returnflag] = MetodEMMOnline (U,Y, mask, showFlag)

[alfa, beta, e, estC, Cov,P]=MetodEMMOnline (U,Y, mask)

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) рекуррентным расширенным матричным методом (25)–(28).

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы **mask** равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Исходные данные:

U — вектор значений входного сигнала **u**, размерностью $1 \times N$.

Y — вектор значений выходного сигнала **y**, размерностью $1 \times N$.

mask — порядки разностного уравнения и матрица-маска параметров (первые две строки и 4–5 строки матрицы из табл. 1), в т. ч.:

$g = \text{mask}(1,2)$ — порядок знаменателя передаточной функции системы,

$p = \text{mask}(1,2)$ — порядок числителя передаточной функции системы,

$r_s = \text{mask}(1,4)$ — порядок знаменателя передаточной функции модели шума (25),

$s = \text{mask}(1,5)$ — порядок числителя передаточной функции модели шума (25), $0 \leq p \leq g, 0 \leq s \leq r_s$.

Принимает следующие начальные параметры процедуры рекуррентного оценивания:

Если не указаны матрицы **initC**, **initP**, **initKsi**, **initEps**, то \mathbf{P}_{N_0} и $\hat{\boldsymbol{\theta}}_{N_0}$ определяются по (28).

Если **initC**, **initP**, **initKsi**, **initEps** указаны при вызове, то: $\hat{\boldsymbol{\theta}}_{N_0} = \text{initC}$, $\mathbf{P}_{N_0} = \text{initP}$, $N_{\text{нач}} = \max\{g, r_s\} + 1$, и элементы вектора \mathbf{v}_k из (26)–(27):

$$\boldsymbol{\xi}_{\text{нач}} = (\xi(0), \dots, \xi(N_{\text{нач}} - 1))^T = \text{initKsi},$$

$$\boldsymbol{\varepsilon}_{\text{нач}} = (\varepsilon(0), \dots, \varepsilon(N_{\text{нач}} - 1))^T = \text{initEps}.$$

Если указана матрица **SetPars**, то составляет матрицу настройки параметров — элементов вектора $\hat{\boldsymbol{\theta}}_k$ и выводит ее в **SetPars**, иначе — не составляет и не выводит.

showFlag — скаляр, если **showFlag** $\neq 0$, то отображает в отдельных окнах результаты идентификации, иначе — не отображает и не выводит сообщений на экран.

Выходные данные:

alfa — вектор $\boldsymbol{\alpha}$ — оценок параметров модели (3), размерностью $1 \times (g+1)$,

beta — вектор β — оценок параметров модели (3), размерностью $1 \times (g+1)$, если $p < g$, то в конец добавлены нули,

e — вектор невязок e из (5), расширенный до размерности $1 \times N$ добавлением в начало вектора e из (5) нулей,

estC — вектор полученных оценок $\hat{\theta}_k$ коэффициентов модели (5), полученный по (27), размерность может быть меньше, чем $g+p+1$, в зависимости от маски mask.

Cov — матрица $\text{cov}(\hat{\theta}_N) = P_N \frac{e^T e}{N/2}$, где размерность e равна $N/2$ ($N/2$ последних значений невязок),

P — матрица P_N (27),

SetPars — матрица настройки параметров — элементов вектора $\hat{\theta}_i, i = 0, N$.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и $\text{returnflag} \leftarrow 0$.

Реализация: MEX-файл: MetodEMMOnline.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
mask=[1 1 1 ;1 0 1;0 0 0;1 1 1;1 0 1];
[alfa, beta, e, estC, Cov, P, fl] = MetodEMMOnline (U,Y,mask,1)
alfa = 1.0000
      -0.7107
beta =      0
      0.2794
e = [0;0;0;0.1029;-0.0727;0.0402;-0.0884;...]; %matrix: 10000x1
estC =
      0.2794
      0.7107
      0.5191
      0.1159
Cov = 1.0e-004 *
      0.0004 -0.0000 -0.0003 -0.0001
      -0.0000  0.0025  0.0006 -0.0016
      -0.0003  0.0006  0.5202  0.1154
      -0.0001 -0.0016  0.1154  0.5865
P = 0.0003 -0.0000 -0.0003 -0.0001
      -0.0000  0.0019  0.0004 -0.0012
      -0.0003  0.0004  0.3916  0.0869
      -0.0001 -0.0012  0.0869  0.4415
fl = 1
```


5.2.7. MetodIVEMMOnline

Варианты вызова:

[alfa, beta, e, estC, Cov, P, returnflag, SetPars] = MetodIVEMMOnline (U, Y, mask, showFlag, initType, initF, initC, initP, initKsi, initEps)

[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVEMMOnline (U, Y, mask, showFlag, initType, initF, initC, initP, initKsi, initEps)

[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVEMMOnline (U, Y, mask, showFlag, MaxIter, eps, initC, initP, initKsi, initEps)

[alfa, beta, e, estC, Cov, P, returnflag, SetPars] = MetodIVEMMOnline (U, Y, mask, showFlag, MaxIter, eps, initC, initP, initKsi, initEps)

[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVEMMOnline (U, Y, mask, showFlag, initType, initF)

[alfa, beta, e, estC, Cov, P, returnflag] = MetodIVEMMOnline (U, Y, mask, showFlag, MaxIter, eps)

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) рекуррентным методом инструментальной переменной, комбинированным с рекуррентным расширенным матричным методом (29)–(30).

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы mask равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Для задания значений инструментальной переменной использует оценки параметров модели или ее непосредственные значения.

Исходные данные:

U — вектор значений входного сигнала u , размерностью $1 \times N$.

Y — вектор значений выходного сигнала y , размерностью $1 \times N$.

mask — порядки разностного уравнения и матрица-маска параметров (все строки матрицы из табл. 1), в т. ч.:

$g = \text{mask}(1,2)$ — порядок знаменателя передаточной функции системы,

$p = \text{mask}(1,2)$ — порядок числителя передаточной функции системы,

$r = \text{mask}(1,3)$ — порядок знаменателя передаточной функции модели шума (17) (вся третья строка только при вызове ОМНК, см. MetodGLSQ, если при вызове указаны скаляры MaxIter и eps), $r > 0$.

$r_s = \text{mask}(1,4)$ — порядок знаменателя передаточной функции модели шума (25),

$s = \text{mask}(1,5)$ — порядок числителя передаточной функции модели шума (25), $0 \leq p \leq g, 0 \leq s \leq r_s$.

Для задания значений инструментальной переменной использует оценки параметров модели:

Если initType-скаляр и initF-матрица(вектор) не указаны, а указаны скаляры MaxIter и eps, то получает оценки параметров фильтра предвари-

тельным вызовом метода ОМНК (MetodGLSQ) со структурой модели, какая прописана в 1-3 строках mask.

Иначе если указан initType=1, то initF—матрица параметров линейного динамического фильтра, выход которого используется в качестве инструментальной переменной, структуру которой см. в табл. 2.

Если initType=1, то initF — вектор значений инструментальной переменной, размерностью $1 \times N$.

Принимает следующие начальные параметры процедуры рекуррентного оценивания:

Если не указаны матрицы initC, initP, initKsi, initEps, то \mathbf{P}_{N_0} и $\hat{\boldsymbol{\theta}}_{N_0}$ определяются по (28).

Если initC, initP, initKsi, initEps указаны при вызове, то: $\hat{\boldsymbol{\theta}}_{N_0} = \text{initC}$, $\mathbf{P}_{N_0} = \text{initP}$, $N_{нач} = \max\{g, r_s\} + 1$, и элементы вектора \mathbf{v}_k из (26)-(27), (29):

$$\boldsymbol{\xi}_{нач} = (\xi(0), \dots, \xi(N_{нач} - 1))^T = \text{initKsi},$$

$$\boldsymbol{\varepsilon}_{нач} = (\varepsilon(0), \dots, \varepsilon(N_{нач} - 1))^T = \text{initEps}.$$

Если указана матрица SetPars, то составляет матрицу настройки параметров — элементов вектора $\hat{\boldsymbol{\theta}}_k$ и выводит ее в SetPars, иначе — не составляет и не выводит.

showFlag — скаляр, если showFlag $\neq 0$, то отображает в отдельных окнах результаты идентификации, иначе — не отображает и не выводит сообщений на экран.

Выходные данные:

alfa — вектор $\boldsymbol{\alpha}$ — оценок параметров модели (3), размерностью $1 \times (g+1)$,

beta — вектор $\boldsymbol{\beta}$ — оценок параметров модели (3), размерностью $1 \times (g+1)$, если $p < g$, то в конец добавлены нули,

e — вектор невязок \mathbf{e} из (5), расширенный до размерности $1 \times N$ добавлением в начало вектора \mathbf{e} из (5) нулей,

estC — вектор полученных оценок $\hat{\boldsymbol{\theta}}_k$ коэффициентов модели (5), полученный по (30), размерность может быть меньше, чем $g+p+1$, в зависимости от маски mask.

Cov — матрица $\text{cov}(\hat{\boldsymbol{\theta}}_N) = \mathbf{P}_N \frac{\mathbf{e}^T \mathbf{e}}{N/2}$, где размерность \mathbf{e} равна $N/2$ ($N/2$ последних значений невязок),

P — матрица \mathbf{P}_N (30),

SetPars — матрица настройки параметров — элементов вектора $\hat{\boldsymbol{\theta}}_i, i = 0, N$.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag ← 0.

Реализация: MEX-файл: MetodIVEMMOnline.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
mask=[1 1 1 ;1 0 1;1 1 1;1 1 1;1 0 1];
[alfa, beta, e, estC, Cov, P, fl] = MetodIVEMMOnline(U,Y,mask,0, 40, 1e-6)
alfa =
    1.0000
   -0.7117
beta =    0
    0.2794
e = [0;0;0;0.1029; -0.0799;0.0475;-0.0635;...]; %matrix: 10000x1
estC =
    0.2794
    0.7117
    0.5144
    0.1108
Cov = 1.0e-004 *
    0.0004 -0.0000 -0.0003 -0.0001
   -0.0010  0.0036  0.0014  0.0007
   -0.0009  0.0021  0.5155  0.1118
    0.0001 -0.0009  0.1111  0.5757
P = 0.0003 -0.0000 -0.0002 -0.0001
   -0.0008  0.0027  0.0010  0.0005
   -0.0007  0.0016  0.3876  0.0841
    0.0001 -0.0007  0.0835  0.4328
fl = 1
```

5.2.8. ParamIdentify

Варианты вызова:

[SetsPars, e, SetsCov, returnflag]= ParamIdentify(U, Y, mask, wTime, shift, MetNum, {параметры метода})

[SetsPars, e, SetsCov] = ParamIdentify(U, Y, mask, wTime, shift, MetNum, {параметры метода})

Описание:

Осуществляет параметрическую идентификацию разностного уравнения (3) одним из вышеперечисленных методов по временным окнам.

Находит оценки только тех коэффициентов модели, для которых значение соответствующих элементов матрицы mask равно 1, другие параметры системы принимаются равными 0 с нулевой дисперсией.

Для задания значений инструментальной переменной использует оценки параметров модели или ее непосредственные значения.

Исходные данные:

U — вектор значений входного сигнала u , размерностью $1 \times N$.

Y — вектор значений выходного сигнала y , размерностью $1 \times N$.

mask — порядки разностного уравнения и матрица-маска параметров (см. описание каждого метода и табл. 1).

параметры идентификации нестационарностей:

wTime — ширина временного окна в тактах,

shift — величина приращения при сдвиге границ временного окна, в тактах,

MetNum — код используемого метода параметрической идентификации, возможные значения:

1 — МНК (см. MetodLSQ);

2 — ОМНК (см. MetodGLSQ);

3 — МИП (см. MetodIV);

4 — pМНК (см. MetodLSQOnline);

5 — pМИП (см. MetodIVOnline);

6 — pPMM (см. MetodEMMOnline);

7 — pPMM+pМИП (см. MetodIVEMMOnline)

{параметры метода} — набор параметров выбранного метода идентификации, идущие в строке вызова выбранного метода после матрицы mask (showFlag не задается при вызове).

Выходные данные:

SetsPars — вектор настройки дискретных параметров модели,

e — вектор невязок, размерностью $1 \times N$, невязки составлены по принципу «скользящего окна».

Если MetNum < 3 (т. е. при МНК и ОМНК), то SetsCov — матрица настройки дисперсии параметров дискретной модели разностного уравнения (только значимых по маске mask параметров), иначе SetsCov — скаляр, равный -1.

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag ← 0.

Реализация: MEX-файл: ParamIdentify.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
```

```
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
```

```
mask=[1 1 1;1 0 1;1 1 1;1 1 1;1 0 1];
```

```
%МНК:
```

```
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,1);
```

```
fl = 1
```

```
%ОМНК:
```

```
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,2,40, 1e-6);
```

```
fl = 1
```

```

%МИП:
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,3,40, 1e-6);
fl = 1
%pМНК:
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,4);
fl = 1
%pМИП:
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,5,40, 1e-6);
fl = 1
% pPMM:
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,6);
fl = 1
% pPMM+pМИП:
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,7,40, 1e-6);
fl = 1

```

5.3. Статистический анализ результатов идентификации

В пакете программ реализована одна функция статистического анализа результатов идентификации по временным окнам — GetParamsValidity, дополняющая функцию ParamIdentify.

Варианты вызова GetParamsValidity:

```

[Student, V1, Fisher, returnflag]= GetParamsValidity (SetsPars, Y, e, SetsCov, mask,
wTime, shift, t, F)
[Student, V1, Fisher]= GetParamsValidity (SetsPars, Y, e, SetsCov, mask, wTime, shift,
t, F)

```

Описание:

Осуществляет расчет критериев качества параметрической идентификации, произведенной функцией ParamIdentify, по временным окнам.

Исходные данные:

SetsPars — вектор настройки дискретных параметров модели (см. ParamIdentify),

Y — вектор значений выходного сигнала, размерностью $1 \times N$,

e — вектор невязок, размерностью $1 \times N$ (см. ParamIdentify),

SetCov — матрица настройки ковариационной матрицы оценок модели (см. ParamIdentify),

mask — порядки разностного уравнения и матрица-маска параметров (первые две строки матрицы из табл. 1), в т. ч.:

$g = \text{mask}(1,1)$ — порядок знаменателя передаточной функции системы,

$p = \text{mask}(1,2)$ — порядок числителя передаточной функции системы, $0 \leq p \leq g$,

wTime — ширина временного окна в тактах,

shift — величина приращения при сдвиге границ временного окна, в тактах,

t — табличное значение критерия Стьюдента для заданного уровня значимости и длины выборки — см. (32);

F — табличное значение критерия Фишера для заданного уровня значимости и длины выборки — см. (31).

Выходные данные:

Если SetsCov — скаляр, то Student — скаляр, равный -1, иначе Student — матрица настройки критерия Стьюдента для каждого параметра модели по (32) (включает столбец значений уровня значимости),

V1 — матрица настройки эpsilon-критерия (функции потерь) дискретных параметров модели,

Fisher — матрица настройки критерия Фишера по (31) (включает столбец значений уровня значимости).

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag ← 0.

Реализация: MEX-файл: GetParamsValidity.dll; библиотеки CID.dll, MatrixM.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
U=[ -0.548;0.610;0.266;-0.536;0.450;-0.702;...]; %matrix: 10000x1
Y=[ 0;-0.1529;0.0526;0.1029;-0.0709;0.0641;...]; %matrix: 10000x1
mask=[1 1 1;1 0 1];
%МНК:
[SetsPars,e,SetsCov,fl] = ParamIdentify (U,Y,mask,400,400,1);
fl=1
[Student, V1, Fisher, fl] = GetParamsValidity (SetsPars, Y, e, SetsCov, mask, 400, 400, 1.5, 1.2);
fl=1
```

5.4. Взаимные преобразования математических моделей

В пакете программ реализованы 6 функций взаимного преобразования моделей линейных стационарных одномерных динамических систем (см. рис. 3):

disEqu2dis — преобразование разностного уравнения в дискретную модель пространства состояний по (33),

dif2unint — преобразование дифференциального уравнения в непрерывную модель пространства состояний по (34),

dis2disEqu — преобразование дискретной модели пространства состояний в разностное уравнение по (35),

unint2dif — преобразование непрерывной модели пространства состояний в дифференциальное уравнение по (36),

unint2discrete — преобразование непрерывной модели пространства состояний в дискретную модель пространства состояний по (37),

discrete2uint — преобразование дискретной модели пространства состояний в непрерывную модель пространства состояний по (38)-(39).

5.4.1. *disEqu2dis*

Варианты вызова:

[A, b, c, d, returnflag] = disEqu2dis(alfa, beta)

[A, b, c, d] = disEqu2dis(alfa, beta)

Описание:

Осуществляет преобразование разностного уравнения (3) в дискретную модель пространства состояний (7) по выражению (33).

Исходные данные:

alfa — вектор α , размерностью $1 \times (n+1)$, модели (3),

beta — вектор β , размерностью $1 \times (n+1)$, модели (3).

Выходные данные:

A — матрица A, размерностью $n \times n$ из (7),

b — вектор \mathbf{b} , размерностью $1 \times n$, из (7),

c — вектор \mathbf{c} , размерностью $1 \times n$, из (7),

d — коэффициент d из (7), (матрица 1×1).

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: MEX-файл: disEqu2dis.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

alfa = [1.0000; -2.0075; 1.0075];

beta = [0; 0.0126; -0.0125];

[A, b, c, d, fl] = disEqu2dis(alfa, beta)

A = 0 1.0000
 -1.0075 2.0075

b = 0

1

c = -0.0125

0.0126

d = 0; fl = 1

5.4.2. *dif2uint*

Варианты вызова:

[T, g, h, k, returnflag] = dif2uint(alfaDif, betaDif)

[T, g, h, k] = dif2uint(alfaDif, betaDif)

Описание:

Осуществляет преобразование дифференциального уравнения (1) в непрерывную модель пространства состояний (6) по выражению (34).

Исходные данные:

alfaDif — вектор $\tilde{\mathbf{b}}$, размерностью $1 \times (n+1)$, модели (1),

betaDif — вектор $\tilde{\mathbf{v}}$, размерностью $1 \times (n+1)$, модели (1).

Выходные данные:

T — матрица T, размерностью $n \times n$ из (6),

g — вектор g, размерностью $1 \times n$, из (6),

h — вектор h, размерностью $1 \times n$, из (6),

k — коэффициент k из (6), (матрица 1×1).

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: MEX-файл: dif2unint.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
alfaDif = [1; 2; 0.5];
betaDif = [1; 0.4; 0];
[T,g,h,k,fl]=dif2unint(alfaDif,betaDif)
T= 0 1
    -2 -4
g= 0
    2
h= 1.0000
    0.4000
k = 0; fl= 1
```

5.4.3. dis2disEqu

Варианты вызова:

[alfa, beta, returnflag] = dis2disEqu(A, b, c, d)

[alfa, beta] = dis2disEqu(A, b, c, d)

Описание:

Осуществляет преобразование дискретной модели (7) пространства состояний в разностное уравнение (3) по выражению (35).

Исходные данные:

A — матрица A, размерностью $n \times n$ из (7),

b — вектор b, размерностью $1 \times n$, из (7),

c — вектор c, размерностью $1 \times n$, из (7),

d — коэффициент d из (7), (матрица 1×1).

Выходные данные:

alfa — вектор $\tilde{\mathbf{b}}$, размерностью $1 \times (n+1)$, модели (3),

beta — вектор $\tilde{\mathbf{v}}$, размерностью $1 \times (n+1)$, модели (3).

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: MEX-файл: dis2disEqu.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
A = [ 1.0025  0.0075; 0.0025  1.0050];
```

```
b = [0.0025; 0.0050];
```

```
c = [1; 2];
```

```
d = 0;
```

```
[alfa, beta, fl] = dis2disEqu(A, b, c, d)
```

```
alfa = 1.0000
```

```
      -2.0075
```

```
      1.0075
```

```
beta = 0
```

```
      0.0126
```

```
     -0.0125
```

```
fl = 1
```

5.4.4. *unint2dif*

Варианты вызова:

```
[alfaDif, betaDif, returnflag] = unint2dif(T, g, h, k)
```

```
[alfaDif, betaDif] = unint2dif(T, g, h, k)
```

Описание:

Осуществляет преобразование непрерывной модели (6) пространства состояний в дифференциальное уравнение (1) по выражению (36).

Исходные данные:

T — матрица T, размерностью $n \times n$ из (6),

g — вектор g, размерностью $1 \times n$, из (6),

h — вектор h, размерностью $1 \times n$, из (6),

k — коэффициент k из (6), (матрица 1×1).

Выходные данные:

alfaDif — вектор α , размерностью $1 \times (n+1)$, модели (1),

betaDif — вектор $\tilde{\beta}$, размерностью $1 \times (n+1)$, модели (1).

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: MEX-файл: unint2dif.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
T = [ 0 1; -2 -4];
```

```
g = [ 0; 2];
```

```
h = [ 1.0000; 0.4000];
```

```
k = 0;
```

```
[alfaDif, betaDif, fl] = unint2dif(T, g, h, k)
```

```
alfaDif =
```

```
      1.0000
```

```
      2.0000
```

```
0.5000
betaDif =
1.0000
0.4000
0
fl = 1
```

5.4.5. *unint2discrete*

Варианты вызова:

```
[A, b, c, d, returnflag] = unint2discrete (T, g, h, k, dt)
[A, b, c, d] = unint2discrete (T, g, h, k, dt)
```

Описание:

Осуществляет преобразование непрерывной модели пространства состояний (6) в дискретную модель пространства состояний (7) по выражению (37).

Исходные данные:

T — матрица T, размерностью $n \times n$ из (6),
g — вектор g, размерностью $1 \times n$, из (6),
h — вектор h, размерностью $1 \times n$, из (6),
k — коэффициент k из (6), (матрица 1×1),
dt — скаляр — интервал дискретизации системы по времени (в секундах).

Выходные данные:

A — матрица A, размерностью $n \times n$ из (7),
b — вектор b, размерностью $1 \times n$, из (7),
c — вектор c, размерностью $1 \times n$, из (7),
d — коэффициент d из (7), (матрица 1×1).
returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: MEX-файл: unint2discrete.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

```
T=[1 3; 1 2]; g=[1;2]; h=[1; 2]; k=0;
[A, b, c, d , fl] = unint2discrete (T, g, h, k, 0.0025)
A = 1.0025  0.0075
    0.0025  1.0050
b = 0.0025
    0.0050
c = 1
    2
d = 0
fl = 1
```

5.4.6. discrete2uint

Варианты вызова:

[T, g, h, k, returnflag] = discrete2uint (A, b, c, d, dt)

[T, g, h, k] = discrete2uint (A, b, c, d, dt)

Описание:

Осуществляет преобразование дискретной модели пространства состояний (7) в непрерывную модель пространства состояний (6) по выражениям (38)-(39).

Исходные данные:

A — матрица A, размерностью $n \times n$ из (7),

b — вектор b, размерностью $1 \times n$, из (7),

c — вектор c, размерностью $1 \times n$, из (7),

d — коэффициент d из (7), (матрица 1×1).

dt — скаляр — интервал дискретизации системы по времени (в секундах).

Выходные данные:

T — матрица T, размерностью $n \times n$ из (6),

g — вектор g, размерностью $1 \times n$, из (6),

h — вектор h, размерностью $1 \times n$, из (6),

k — коэффициент k из (6), (матрица 1×1),

returnflag — флаг успешного завершения: в случае успешного завершения равен 1, иначе — функция выводит сообщение об ошибке и returnflag $\leftarrow 0$.

Реализация: MEX-файл: discrete2uint.dll; библиотеки CID.dll, Matrix.dll и MATLABLnk.dll.

Пример вызова из MATLAB:

A = [0 1.0000; -1.0075 2.0075];

b = [0; 1];

c = [-0.0125; 0.0126];

d = 0;

[T,g,h,k,fl]=discrete2uint(A,b,c,d,0.0025)

T =

-398.5075 398.5075

-401.4963 401.4963

g =

-256

256

h =

-0.0125

0.0126

k = 0

fl = 1

Литература

1. Фатуев В. А., Каргин А. В., Понятский В. М. Структурно-параметрическая идентификация динамических систем: Учеб. пособие.— Тула: Изд-во ТулГУ, 2003.— 156 с.
2. Фатуев В. А., Юдаев А. В., Понятский В. М., Каргин А. В., Оберман М. С. Структурно-параметрическая идентификация многомерных нестационарных динамических систем // Тр. III международной конф. «Идентификация систем и задачи управления» (SICPRO'04).— М.: ИПУ РАН, 2004.— С.159–186.
3. Стрейц В. Метод пространства состояний в теории дискретных линейных систем управления.— М.: Наука, 1985.— 296 с.
4. Толчеев В. О., Ягодкина Т. В. Методы идентификации линейных одномерных динамических систем.— М.: МЭИ, 1997.— 108 с.
5. Фатуев В. А. Идентификация объектов управления: Учеб. пособие.— Тула: ТулПИ, 1977.— 74 с.
6. Фатуев В. А. Построение оптимальных моделей динамики по экспериментальным данным: Учеб. пособие.— Тула: Тул. гос. тех. ун-т, 1993.— 104 с.
7. Фатуев В. А., Каргин А. В. К вопросу о связи коэффициентов канонической формы модели пространства состояний и ординат импульсных переходных функций каналов линейной многомерной динамической системы // Известия Тульского государственного университета. Серия «Вычислительная техника. Автоматика. Управление».— 2001.— Вып. 7.— С.20–26.
8. Фатуев В. А., Каргин А. В. К вопросу о связи коэффициентов математических моделей стационарных одномерных линейных динамических систем // Известия Тульского государственного университета. Серия «Вычислительная техника. Автоматика. Управление».— 2002.— Вып.8.— С.3–18.
9. Фатуев В. А., Каргин А. В. Структурно-параметрическая идентификация нестационарных линейных одномерных динамических систем // Известия Тульского государственного университета. Серия «Вычислительная техника. Автоматика. Управление».— 2002.— Вып.8.— С.18–22.
10. Фатуев В. А., Каргин А. В., Понятский В. М. Структурно-параметрическая идентификация нестационарных динамических объектов // Известия Тульского государственного университета. Серия «Проблемы проектирования и производства систем и комплексов».— 2003.— Вып.6.— С.349–353.

УДК 517.9

ПОСТРОЕНИЕ СИММЕТРИЧЕСКИХ БИФУРКАЦИОННЫХ МНОГООБРАЗИЙ АНАЛИТИЧЕСКИХ ДИФФЕРЕНЦИАЛЬНЫХ СИСТЕМ

*Халин А. Л., Халина Н. А.**Московский государственный университет им. М. В. Ломоносова, Москва,
e-mail: allkhalin@email.ru*

1. Введение

Одной из основных математических моделей динамических систем была и, по всей видимости, будет \mathbf{R} -аналитическая дифференциальная система. Описание такой модели требует использование различных математических объектов. Исследование особенностей или бифуркаций дифференциальных систем является основной задачей анализа дифференциальной системы и играет существенную роль для многих областей математики, связанных как с дифференциальными, так и с дискретными математическими структурами [1–8].

Постановка задачи изучения таких систем включает два необходимых элемента, понятие «полноты» дифференциальной системы или, что эквивалентно, ее полной интегрируемости, а также понятие структурной устойчивости. Последнее характеризует качественные свойства систем, их разбиение на классы эквивалентности над стационарным многообразием оператора дифференцирования, которое можно задать локальной системой параметров дифференциальной системы.

Бифуркация есть структурная неустойчивость дифференциальной системы. Одним из видов бифуркации является симметрическая бифуркация, описываемая симметрическими функциями собственных значений производной оператора системы. В последние годы разработаны эффективные методы вычисления для построения критических поверхностей симметрических бифуркаций дифференциальных систем с размерностью выше двух с использованием алгебраических преобразований, численного интегрирования и дифференциальных методов поиска решений нелинейных уравнений [1–8].

Рассматривается класс аналитических дифференциальных систем, представляемых автономной системой обыкновенных дифференциальных уравнений с \mathbf{R} -аналитической правой частью. Пусть даны \mathbf{R} -аналитическое многообразие P конечной размерности $n+m$, \mathbf{R} -аналитическое многообразие параметров M размерности n и заданная в каждой точке многообразия P автономная система дифференциальных уравнений, зависящая локально от векторного переменного x и векторного параметра a :

$$\frac{dx}{dt} = f(x, a), \quad (1)$$

где $a = (a_1, \dots, a_n)$ — набор локальных параметров в некоторой окрестности U \mathbf{R} -аналитического многообразия M , $y = (x_1, \dots, x_m, a_1, \dots, a_n)$ — набор локальных координат в некоторой окрестности W многообразия P , $f(x, a)$ — регулярная \mathbf{R} -аналитическая векторная функция в окрестности W многообразия P . Для многообразий P и M существует регулярное \mathbf{R} -аналитическое отображение $p: P \rightarrow M$, задающее локально тривиальное расслоение $\eta = (P, p, M)$, то есть для окрестности U существует \mathbf{R} -аналитический диффеоморфизм окрестности $W = p^{-1}(U)$ произвольной точки многообразия P на прямое произведение областей $U \times V$, где V — окрестность с локальными координатами $x = (x_1, \dots, x_m)$.

Система уравнений (1) по теореме о существовании решения системы обыкновенных дифференциальных уравнений определяет в некоторой окрестности любой точки слоя расслоения η систему локальных однопараметрических групп локальных преобразований $\{G_i\}$, $i = 1, \dots, n$, индуцирующую соответствующую систему аналитических поливекторных полей [7].

Расслоение со структурной группой G определяет ассоциированное расслоение, где в качестве векторного G -пространства выступает слой касательного расслоения слоя расслоения η .

Задание системы интегрируемых распределений системой (1) определяет класс локально изоморфных G -расслоений над многообразием M и соответствующий ему класс локально гомотопных G -слоев G -расслоения многообразия P [1, 9].

Для сечения $s: M \rightarrow P$ расслоения η выполняется условие $p \circ s = id$. Сечения тривиального расслоения находятся в естественном биективном соответствии с локальными непрерывными решениями системы (1) [10]. Таким образом, локальная разрешимость системы есть существование локального сечения, и кольцо сечений является координатным кольцом касательного расслоения.

Пространство дифференциальных форм есть сопряженное пространство к слою касательного расслоения. Дуальное расслоение $\chi = (P^*, \tau, TP^*)$ к касательному расслоению TP многообразия P определим как аналитическое расслоение координатного кольца аналитических функций P^* многообразия P над базой, координатным кольцом функций касательного расслоения, которое есть кольцо дифференциальных форм.

Для расслоения $\tilde{\eta}$ всегда существует локальное сечение, так как любая \mathbf{R} -аналитическая функция, заданная на аналитическом многообра-

зии, локально интегрируема. Система (1) задает векторный базис базы дуального расслоения.

2. Полная дифференциальная система

Векторное пространство поливекторных полей и сопряженное ему пространство дифференциальных форм для системы (1) не является полным во всех точках расслоения $\eta = (P, p, M)$. Для глобальной разрешимости системы на всем расслоении η необходимо определить решение системы для любой точки многообразия P , то есть пополнить пространство \mathbf{R} -аналитических дифференциальных форм, задаваемых системой (1). Его надо пополнить дифференциальными уравнениями для параметров, которые задают собой касательные нулевого сечения расслоения η .

В окрестности любой неособой точки дифференциальной системы (1) или, что эквивалентно, в окрестности любой неособой точки расслоения $\chi = (P^*, \tau, TP^*)$ существует n локальных аналитических первых функционалов, которые есть локальные сечения расслоения χ .

Особые точки дифференциальной системы или особые точки пространства аналитических функционалов определяются условием равенства нулю его градиента и соответствуют нулевым сечениям расслоения χ , чему способствует теорема Сарда [11] и аналитичность множества особых точек. Множество особых точек системы (1) определяется однородной системой дифференциальных уравнений для дифференциальной системы (1). Пространство функционалов неполное вследствие неполноты пространства дифференциальных форм над многообразием P .

Для аналитичности на компактной окрестности любой точки касательного расслоения многообразия P и полноты дифференциальной системы аналитический функционал должен быть продолжен на весь слой касательного расслоения.

По теореме Лиувилля аналитическая функция должна быть константой на любом линейном подмногообразии слоя касательного расслоения к P и на любом максимальном интегральном подмногообразии многообразия P [12]. Набор собственных значений $\{\lambda_i\}$, $i = 1, \dots, n$, системы (1) образует набор инвариантов на касательном расслоении относительно действия групп преобразований $\{G_i\}$, $i = 1, \dots, n$ системы (1).

Считаем дифференциальную систему (1) полной, если существует поверхностно-односвязное накрытие расслоения η . Это означает, что существует конечный предел среднего потока векторного поля системы (1) для любой точки многообразия P , равного отношению потока векторного поля через замкнутую поверхность, ограничивающую некоторую компакт-

ную окрестность точки к величине объема этой окрестности, который является дивергенцией поля в точке.

Данное утверждение вытекает из теоремы Стокса для любой компактной окрестности любой точки касательного расслоения TP , в частности, для любой точки многообразия M можно определить аналитический функционал, что дает возможность продолжить аналитически дуальное расслоение χ на все многообразие P , то есть на многообразие M [13].

Интегрируемость системы эквивалентна существованию n ограниченных \mathbf{R} -аналитических функционалов на пространстве аналитических дифференциальных форм [14-17]. Данное пространство однозначно определяет систему первых интегралов, обобщенных \mathbf{R} -аналитических решений дифференциальной системы (1). Топология равномерной сходимости пространства обобщенных \mathbf{R} -аналитических функций есть слабая топология на пополненном пространстве дифференциальных форм.

3. Бифуркация дифференциальной системы

Дифференциальная система инвариантна относительно \mathbf{R} -аналитических преобразований, а пространство ее обобщенных решений определено с точностью до константы. На многообразии P можно ввести локально выпуклую топологию, топологию векторного пространства дифференциальных форм, то есть кокасательного расслоения многообразия P .

Для многообразия P существует нормальная окрестность нулевого сечения расслоения η или нуля пространства векторных полей системы (1) как нормальная окрестность конечномерного векторного пространства, что эквивалентно существованию не обращающегося в нуль сечения нормального расслоения многообразия M .

Обобщенные элементарные симметрические функции собственных значений задают фундаментальную систему нормальных окрестностей нуля \mathbf{R} -аналитического пространства сечений расслоения касательного расслоения η : $d\eta = (TP, dp, TM)$, где dp — дифференциал аналитического отображения p [8, 15]. Они образуют набор n полунорм на пространстве дифференциальных форм и полностью определяют на нем топологию. Набор этих полунорм остается инвариантным относительно изоморфизма аналитического многообразия P с точностью до инвариантного множителя неравного нулю. В силу \mathbf{R} -аналитичности рассматриваемой системы функций фундаментальные решения можно продолжить по непрерывности на все многообразие P .

Множество критических точек или точек бифуркации расслоения P определяется как множество многообразия M , для которых у нулевого сечения расслоения $\eta = (P, p, M)$ не существует нормальной окрестности,

что эквивалентно существованию обращающегося в нуль сечения нормального расслоения подмногообразия M .

Бифуркационное многообразие коразмерности один задается одной из элементарных симметрических функций. Это является следствием того, что функции определяют нормальную окрестность нулевого сечения касательного расслоения многообразия P , а также в силу их алгебраической независимости над любым полем [15].

Любая симметрическая функция собственных значений оператора системы (1) задает бифуркационное многообразие коразмерности один на аналитическом многообразии параметров системы. Система аналитических уравнений для определения бифуркационного многообразия может быть получена объединением уравнений, определяющих стационарные точки, и уравнений, определяемых симметрическими формами относительно собственных значений производной оператора системы (1):

$$\begin{aligned} f(y, a) &= 0; \\ F(y, a) &= 0. \end{aligned} \quad (2)$$

Для аналитической правой части уравнений (1) симметрическая бифуркационная функция может быть представлена единственным способом как многочлен от элементарных симметрических функций собственных значений [18–19]. Так как правая часть n -мерного векторного дифференциального уравнения (1) представляет собой \mathbf{R} -аналитическую вектор-функцию, то по теореме Вейерштрасса на любом локально компактном многообразии ее можно аппроксимировать n полиномами p_i от n переменных. Результирующая система уравнений

$$dy/dt = h(y, a) \quad (3)$$

будет аппроксимировать исходную систему уравнений. Компоненты векторной функции $p(y, a)$ в правой части системы являются полиномами $h_i(y, a)$, $i=1, 2, \dots, n$, положительной степени.

Таким же образом система, определяющая бифуркацию, сводится к полиномиальной системе уравнений:

$$\begin{aligned} h(y, a) &= 0; \\ H(y, a) &= 0, \end{aligned} \quad (4)$$

где $H(y, a)$ — полином, аппроксимирующий бифуркационную функцию $F(y, a)$.

Проведенные рассуждения справедливы в общем случае, когда бифуркация определяется симметрической полиномиальной функцией от собственных значений матрицы Якоби системы (1). При полиномиальной зависимости векторной функции системы дифференциальных уравнений (3) от переменных и параметров дифференциальной системы бифуркационное многообразие (4), определяемое симметрической полиномиальной функцией, является алгебраическим многообразием на многообразии M .

4. Примеры симметрических бифуркаций

Рассмотрим в качестве примера критические многообразия симметрических бифуркаций системы Лоренца, задаваемой системой уравнений:

$$\begin{aligned} dy_1 / dt &= a_1 (y_1 - y_3); \\ dy_2 / dt &= -y_1 y_3 - a_2 y_1 - y_2; \\ dy_3 / dt &= y_1 y_2 + a_3 y_3. \end{aligned} \quad (5)$$

Идея преобразования полиномиальной системы уравнений (4) состоит в последовательном исключении переменных до получения уравнения, содержащего только параметры системы уравнений (1). Процесс исключения может сочетать операции вычисления результатов и факторизации относительно исключаемых переменных, оставляя промежуточные выражения алгебраическими, таким образом, он позволяет избежать оперирования с громоздкими символьными выражениями и оптимально учитывать вычислительные ограничения. Результат является целочисленным полиномом от коэффициентов полиномов [6, 19–20]. Результатом алгебраических операций будет уравнение, содержащее только параметры системы дифференциальных уравнений, определяющее бифуркационную поверхность в пространстве параметров [21].

При полиномиальной зависимости векторной функции системы дифференциальных уравнений (1) от переменных и параметров системы критическая поверхность, соответствующая симметрической полиномиальной функции бифуркации, является алгебраическим многообразием в пространстве параметров.

Вычисления были проведены для дифференциальной системы Лоренца (5) в математическом пакете MATLAB [22] для полного набора первых интегралов системы (1), симметрических относительно собственных значений $\{\lambda_1, \lambda_2, \lambda_3\}$ якобиана системы J . Бифуркационные многообразия коразмерности один определяются элементарными симметрическими функциями собственных значений вследствие алгебраической их независимости над любым полем [20]:

$$\begin{aligned} S1 &= \text{trace } J = \lambda_1 + \lambda_2 + \lambda_3; \\ S2 &= \lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_1 \lambda_3; \\ S3 &= \det J = \lambda_1 \lambda_2 \lambda_3. \end{aligned} \quad (6)$$

На рис. 1, 2, 3 показаны критические многообразия инвариантов $S1, S2, S3$ системы Лоренца в пространстве фазовых переменных (y_1, y_2, y_3) для трех значений параметра $a_1 = -5, 0, 5$. Картина фазового портрета системы качественно меняется при прохождении хотя бы одной из стационарных точек через приводимые на рисунках поверхности. Параметрическое пространство разбивается на открытые подобласти поверхностями нулевого

уровня инвариантов $S1$, $S2$, $S3$, границы которых являются рассматриваемыми бифуркационными многообразиями.

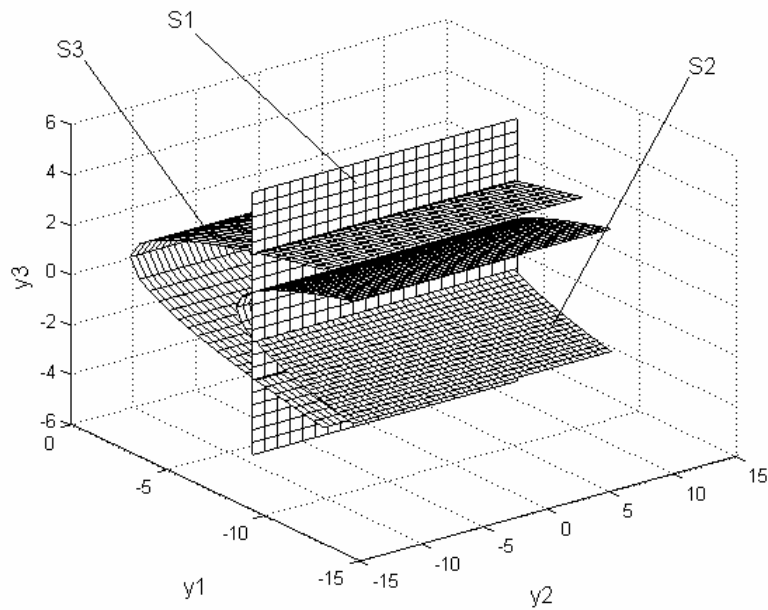


Рис. 1. Бифуркационные многообразия инвариантов $S1$, $S2$, $S3$ системы Лоренца в пространстве фазовых переменных (y_1, y_2, y_3) для значения параметра $a_1 = -5$.

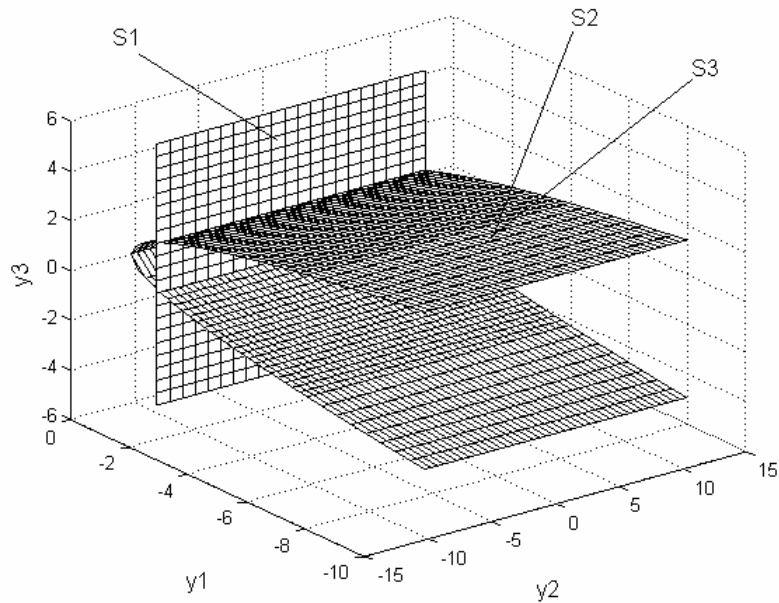


Рис. 2. Бифуркационные многообразия инвариантов $S1$, $S2$, $S3$ системы Лоренца в пространстве фазовых переменных (y_1, y_2, y_3) для значения параметра $a_1 = 0$.

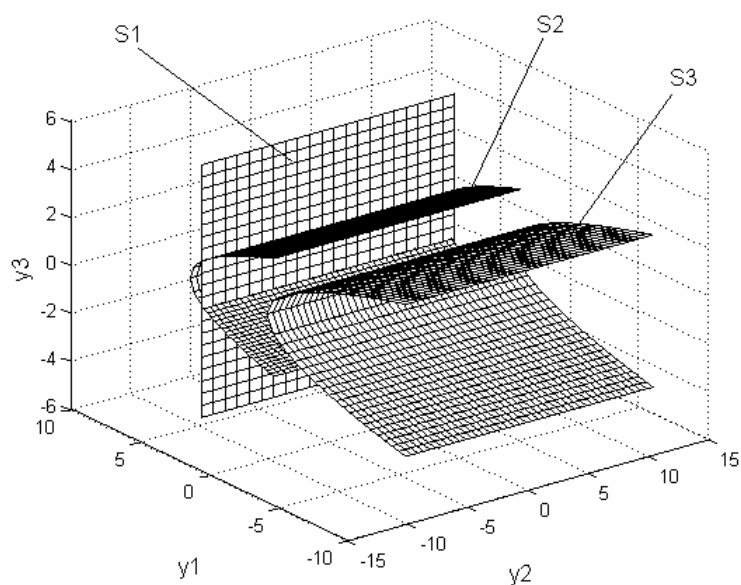


Рис. 3. Бифуркационные многообразия инвариантов $S1$, $S2$, $S3$ системы Лоренца в пространстве фазовых переменных (y_1, y_2, y_3) для значения параметра $a_1 = 5$.

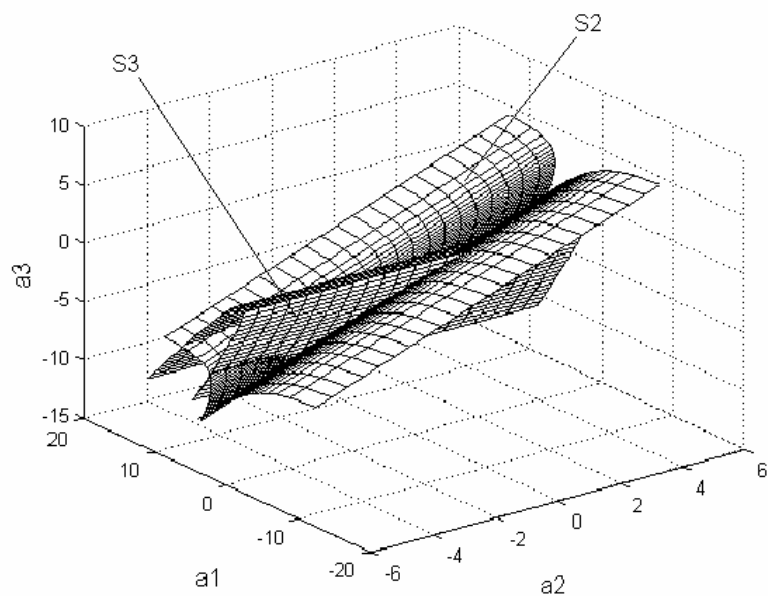


Рис. 4. Бифуркационные многообразия инвариантов $S2$, $S3$ системы Лоренца в пространстве параметрических переменных (a_1, a_2, a_3) .

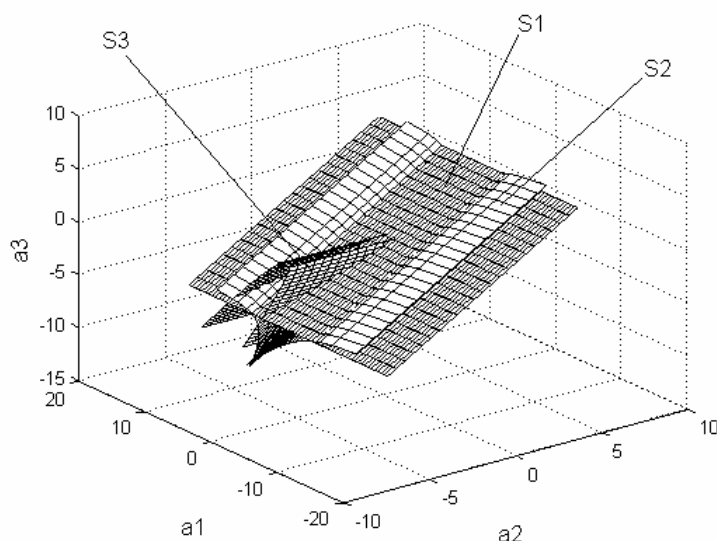


Рис. 5. Бифуркационные многообразия инвариантов $S1$, $S2$, $S3$ системы Лоренца в пространстве фазовых переменных (a_1, a_2, a_3) .

Литература

1. Халин А. Л. Симметрические бифуркации в системах обыкновенных дифференциальных уравнений // Труды Средневолжского математического общества.— 2002.— Т.3–4.— №1.— С.255–256.
2. Kuznesov Y. A. Elements of Applied Bifurcation Theory. // Applied Mathematical Sciences.— New York: Springer — Verlag, 1995.— 515 p.
3. Khalin A. L. Numerical Construction of an Andronov-Hopf Bifurcation Surface. // The 5th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2001).— Saint-Petersburg, Russia, July 4–6, 2001.— P.987–990.
4. Братусь А. С., Халин А. Л. Метод поиска точек бифуркации Андронова–Хопфа // ЖВМиМФ.— 2002.— Т.42.— №3.— С.336–350.
5. Bratus' A. S., Khalin A. L. A Method for Detection of Hopf Bifurcation Points // Computational Mathematics and Mathematical Physics.— V.42.— N.3.— 2002.— P.321–334.
6. Халин А. Л. Символьное вычисление бифуркационных поверхностей систем дифференциальных уравнений с полиномиальной правой частью.— Саранск: Средневолжское матем. общество, 2000 (препринт №26).
7. Халин А. Л. Симметрические бифуркационные многообразия динамических систем // Труды Всероссийской научной конференции «Проектирование научных и инженерных приложений в среде MATLAB».— М.: ИПУ РАН, 2002.— С.528–538.
8. Халин А. Л. Симметрические бифуркации систем обыкновенных дифференциальных уравнений на аналитических многообразиях // Труды II Международной конференции «Идентификация систем и задачи управ-

- ления» SICPRO'03 (Москва, 29–31 января 2003).— М.: ИПУ РАН, 2003.— С.2233–2240.
9. Мищенко А. С. Векторные расслоения и их применения.— М.: Наука, 1984.— 208 с.
10. Постников М. М. Дифференциальная геометрия.— М.: Наука, 1988.— 496 с.
11. Дубровин Б. А., Новиков С. П., Фоменко А. Т. Современная геометрия.— М.: Наука, 1979.— 760 с.
12. Владимиров В. С. Методы теории функций многих комплексных переменных.— М.: Наука, 1964.— 412 с.
13. Ефимов А. В. Математический анализ (специальные разделы). Т.1.— М.: Высшая школа, 1980.— 279 с.
14. Khalin A. L. A Transversality Topology of the Generalized Solutions in Analytical Differential Systems // The Fourth International Conference «Tools for Mathematical Modelling». Abstracts.— Saint-Petersburg. Russia, June 23–28, 2003.— P.89.
15. Халин А. Л. Структурная устойчивость и нормализация аналитических дифференциальных систем // Труды III Международной конференции «Идентификация систем и задачи управления» SICPRO'04. (Москва 28–30 января 2004).— М.: ИПУ РАН, 2004.— С.1791–1796.
16. Гельфанд И. М., Шилов Г. Е. Обобщенные функции и действия над ними.— М.: Добросовест, 2000.— 412 с.
17. Колмогоров А. Н., Фомин С. В. Элементы теорий функции и функционального анализа.— М.: Наука, 1989.— 624 с.
18. Хорн Р., Джонсон И. Матричный анализ.— М.: Мир, 1989.— 655 с.
19. Кокс Д., Литл Дж., О'Ши Д. Идеалы, многообразия и алгоритмы.— М.: Мир, 2000.— 687 с.
20. Ван дер Варден Б. Л. Алгебра.— М.: Наука, 1979.— 624 с.
21. Дэвенпорт Дж., Сирэ И., Турнье Э. Компьютерная алгебра.— М.: 1991, 352 с.
22. Потемкин В. Г. Система MATLAB. Справочное пособие.— М.: Диалог-МИФИ, 1997.— 350 с.

УДК 519.6

ИСПОЛЬЗОВАНИЕ ПАКЕТА MATLAB/SIMULINK, ДЛЯ МОДЕЛИРОВАНИЯ И ИССЛЕДОВАНИЯ ГИРОСТАБИЛИЗАТОРА С АКТИВНЫМ ДИНАМИЧЕСКИМ ГАСИТЕЛЕМ КОЛЕБАНИЙ

Черников С. А., Самер аль-Салек

*Московский государственный технический университет им. Н. Э. Баумана, Москва,
e-mail: SAMER-SALEK@mail.ru*

Рассматриваются основные результаты исследования активным динамическим гасителем колебаний (АДГК) для гиростабилизатора (ГС).

Динамическим гасителем или динамическим гасителем колебаний (ДГК) называют устройство, в котором возникает сила инерции, уменьшающая уровень колебаний защищаемой конструкции.

Большинство работ [1–7], посвященных теме применения ДГК в гиросистеме, ограничены применением пассивного гасителя.

В докладе рассматривается активный динамический гаситель колебаний АДГК, предлагается кинематическая схема, описываются уравнения движения одноосного гиростабилизатора с активным динамическим гасителем колебаний (АДГК); а также структурная схема АДГК с динамической настройкой, где жесткость гасителя пропорциональна квадрату частоты.

Учитывая ошибку измерения частоты, была выполнена расчетная модель в пакете MATLAB-SIMULINK для исследования гиростабилизатора с динамическим гасителем колебаний (АДГК) с динамической настройкой жесткости.

Основные результаты исследования, используя пакет MATLAB-SIMULINK:

1. В области рабочих частот АДГК можно улучшить динамическую точность гиросистемы.
2. Наиболее эффективна гиросистема с АДГК в области резонансных частот (коэффициент гашения может достигнуть $\eta=2000$).
3. Требования к точности измерения частоты для получения высокой эффективности: погрешность измерения частоты должна быть не более 1%.
4. Можно расширить частотные области работы гасителя при использовании АДГК.
5. Можно уменьшить момент инерции активного гасителя в 10 раз относительно пассивного гасителя, что является весьма актуальным для гироскопических систем стабилизации.

Литература

1. Назаров Б. И., Черников С. А. и др. Командно-измерительные приборы.— М.: ВНИИВМ, 1987.— 639 с.
2. Черников С. А. Инерционное демпфирование как метод уменьшения динамического дрейфа ГС // Изв. вузов. Приборостроение.— 1971.— Т.14.— №6.— С.87–91.
3. Черников С. А. Инерционное демпфирование систем гироскопической стабилизации // Изв. вузов. Приборостроение.— 1969.— №8.— С.66–70.
4. Черников С. А. Динамическая стабилизация недемпфированных упругих гиросистем // Изв. АН СССР. МТТ.— 1971.— №3.— С.11–19.
5. Новожилов И. В. Силовой гироскопический стабилизатор с динамическим поглотителем колебаний // Изв. АН СССР. Механика и машиностроение.— 1962.— №4.— С.112–114.
6. Черников С. А. Инерционное демпфирование как метод уменьшения динамического дрейфа ГС // Изв. вузов. Приборостроение.— 1971.— Т.14.— №6.— С.87–91.
7. Черников С. А., Хенди Б. Улучшение динамических характеристик инерционно демпфируемых гироскопических систем введением люфта в диссипативную связь // Вестник МГТУ. Сер. Приборостроение.— 1999.— №1.— С.69–75.

УДК 517.94

АЛГОРИТМЫ ПРИБЛИЖЕННОГО ИССЛЕДОВАНИЯ ЗАДАЧИ О БИФУРКАЦИИ АНДРОНОВА-ХОПФА

*Юмагулов М. Г., Нуров И. Д. Шарафутдинов И. В.
Сибайский институт Башкирского государственного университета, Сибай,
Институт математики АН РТ, Душанбе,
e-mail: yum_mg@mail.ru, nid1@mail.ru*

1. Введение

Математическое моделирование часто требует детального изучения динамических систем, содержащих параметры. Одними из наиболее интересных явлений при изучении таких систем представляются различные бифуркации, означающие качественную перестройку функционирования системы. Обнаружение бифуркации является одним из важных этапов исследования динамической системы. Интерес представляют все аспекты — нахождение критических (бифуркационных) значений параметров, построение (аналитическое или приближенное) рождающихся решений, анализ их устойчивости, исследование типа бифуркации (суб- или суперкритичность) и т. д.

Одним из основных типов бифуркационного поведения динамической системы является возникновение автоколебаний из стационарного состояния системы — бифуркация Андронова-Хопфа. Этот тип бифуркации связан с переходом через мнимую ось пары комплексных собственных значений линеаризованной системы. Исследованию задачи о бифуркации Андронова-Хопфа посвящено огромное число работ (см., например, [1-3] и имеющуюся там библиографию). Аналитическое решение этой задачи возможно, как правило, только для специально конструируемых уравнений. Поэтому указанная задача нуждается в алгоритмической и программной поддержке.

Алгоритмы и программы для задачи о бифуркации Андронова-Хопфа создавались многими исследователями (см., например, [2] и имеющуюся там библиографию). Работы в этом направлении стимулировались как запросами практики (задачи гидродинамики, биологии и др.), так и общетеоретическими потребностями — необходимостью иметь универсальные программные продукты для проведения вычислительных экспериментов, решения модельных задач и др.

В настоящей работе предлагается схема приближенного исследования задачи о бифуркации Андронова-Хопфа, основанная на идеях метода функционализации параметра [4]. Предлагаемая схема позволяет провести детальное численное исследование задачи, получить асимптотические

формулы для решений, провести анализ их устойчивости и др. Схема допускает эффективную реализацию в системе MATLAB. В работе приведены некоторые результаты, полученные на основе соответствующих программ.

2. Бифуркация Андронова-Хопфа.

Приведем основные положения схемы приближенного исследования задачи о бифуркации Андронова-Хопфа. При этом ограничимся рассмотрением системы уравнений

$$\frac{dx}{dt} = A(\lambda)x + a(x; \lambda), \quad x \in R^N, N \geq 2, \lambda \in R^1. \quad (1)$$

Здесь матрица $A(\lambda)$ и вектор-функция $a(x; \lambda)$ зависят от скалярного параметра λ , причем

$$\lim_{|x| \rightarrow 0} \max_{|\lambda - \lambda_0|} \frac{|a(x; \lambda)|}{|x|} = 0.$$

Основным является предположение:

П1). Матрица $A(\lambda_0)$ имеет простые собственные значения $\pm \omega_0 i$, $\omega_0 > 0$.

Так как собственные значения $\pm \omega_0 i$ матрицы $A_0 = A(\lambda_0)$ и транспонированной к ней матрицы A_0^* простые, то найдутся такие пары линейно независимых векторов $e, g \in R^N$ и $e^*, g^* \in R^N$, что

$$A_0 e = -\omega_0 g, A_0 g = \omega_0 e, A_0^* e^* = \omega_0 g^*, A_0^* g^* = -\omega_0 e^*; \quad (2)$$

$$|e| = |g| = 1, (e, e^*) = (g, g^*) = 1, (e, g^*) = (g, e^*) = 0. \quad (3)$$

Пусть, наряду с **П1)** выполнено также предположение:

П2). Имеет место соотношение (в нем $A' = A'(\lambda_0)$):

$$(A' e, e^*) + (A' g, g^*) \neq 0. \quad (4)$$

Пару (λ_0, T_0) называют (см. [2–4]) точкой бифуркации Андронова-Хопфа для системы (1), если каждому $\varepsilon > 0$ соответствует такое $\lambda = \lambda_\varepsilon \in (\lambda_0 - \varepsilon, \lambda_0 + \varepsilon)$, при котором система (1) имеет ненулевое T_ε -периодическое решение $x_\varepsilon(t)$, где $T_\varepsilon \in (T_0 - \varepsilon, T_0 + \varepsilon)$ и $|x_\varepsilon(t)| \leq \varepsilon, -\infty < t < \infty$.

Положим $T_0 = 2\pi/\omega_0$. Пусть выполнены предположения **П1)–П2)**. Тогда [4] пара (λ_0, T_0) будет точкой бифуркации Андронова-Хопфа для системы (1). При этом рождающиеся периодические решения $x(t; \lambda)$ существуют при малых $|\lambda - \lambda_0|$ в точности в одном из трех случаев: (1) $\lambda > \lambda_0$; (2) $\lambda < \lambda_0$; (3) $\lambda = \lambda_0$, причем в первых двух случаях каждому λ отвечает не более одного цикла малой амплитуды.

Ниже через $C=C[0,1]$ обозначается пространство непрерывных на отрезке $0 \leq t \leq 1$ вектор-функций $x(t)$, $x(0)=x(1)$, с равномерной метрикой.

Определим функции

$$e(t) = e \cos 2\pi t - g \sin 2\pi t, \quad g(t) = g \cos 2\pi t + e \sin 2\pi t \quad (5)$$

и функционалы

$$\alpha[x(t)] = (x_c, g^*) + (x_s, e^*), \quad \beta[x(t)] = (x_c, e^*) - (x_s, g^*), \quad (6)$$

где e, g, e^*, g^* — векторы из (2) и (3), а векторы x_c и x_s — это отвечающие $\cos 2\pi t$ и $\sin 2\pi t$ коэффициенты Фурье функции $x(t) \in C[0,1]$.

3. Итерационная процедура

Приведем основные положения предлагаемой итерационной процедуры построения бифурцирующих решений системы (1).

Определим зависящие от малого параметра $q > 0$ функционалы:

$$\lambda_q[y(t)] = \lambda_0 - 1 + \frac{1}{2q}\beta[y(t)], \quad T_q[y(t)] = T_0 + \frac{1}{2q}\alpha[y(t)], \quad (7)$$

где $\alpha(x)$ и $\beta(x)$ — функционалы (6). Тогда $\lambda_q[qe(t)] = \lambda_0$ и $T_q[qe(t)] = T_0$. Основным является операторное уравнение:

$$G_q[y(t)] + W_q[y(t)] = 0, \quad (8)$$

где

$$G_q[y(t)] = y(1) + T_q[y(t)] \int_0^t A[\lambda_q(y(t))]y(s) ds - y(t), \quad (9)$$

$$W_q[y(t)] = T_q[y(t)] \int_0^t a[y(s), \lambda_q(y(t))] ds. \quad (10)$$

Равенство $y(t) \equiv x(tT_q)$ осуществляет взаимно-однозначное соответствие между решениями $y(t)$ уравнения (8) и T_q -периодическими решениями $x(t)$ системы (1) при $\lambda = \lambda_q$. Здесь $\lambda_q = \lambda_q[y(t)]$ и $T_q = T_q[y(t)]$.

Для каждого малого $q > 0$ оператор (9) действует в $C[0,1]$ и дифференцируем по Фреше в окрестности функции $y = qe(t)$, при этом:

(а) производная Фреше $G_q(y)$ при $y = qe(t)$ не зависит от q и представима в виде

$$G_q[qe(t)]h(t) = Bh(t) - h(t) + \frac{1}{2T_0}\alpha[h(t)][e(t) - e] + \frac{T_0}{4\pi}\beta[h(t)]A'[g(t) - g], \quad (11)$$

где $\alpha(x)$ и $\beta(x)$ — функционалы (6) и

$$Bh(t) = h(1) + T_0 \int_0^t A(\lambda_0)h(s) ds;$$

(b) оператор $G_q[qe(t)]: C[0,1] \rightarrow C[0,1]$ непрерывно обратим тогда и только тогда, когда выполнено соотношение (4).

Положим $\Gamma = (G_q[qe(t)])^{-1}: C[0,1] \rightarrow C[0,1]$ и определим оператор

$$U_q y(t) = y(t) - \Gamma[G_q(y(t)) + W_q(y(t))].$$

Следующее утверждение является основным в предлагаемой схеме приближенного исследования задачи о бифуркации Андронова-Хопфа для системы (1).

Теорема 1. Для каждого достаточно малого $q > 0$ последовательные приближения

$$y_{q,n+1}(t) = U_q y_{q,n}(t), \quad n = 0, 1, 2, \dots \quad (12)$$

(при $y_{q,0}(t) = qe(t)$) равномерно сходятся к некоторой функции 1 при $n \rightarrow \infty$.

Тот факт, что пара (λ_0, T_0) является точкой бифуркации Андронова-Хопфа для системы (1), обосновывается в следующем утверждении.

Теорема 2. Функции $y_q(t)$ удовлетворяют соотношениям $\lambda_q = \lambda_q[y_q(t)] \rightarrow \lambda_0$, $T_q = T_q[y_q(t)] \rightarrow T_0$ и $y_q(t) \rightarrow 0$ при $q \rightarrow 0$. При $\lambda = \lambda_q$ система (1) имеет T_q -периодическое решение $x_q(t) \equiv y_q(t/T_q)$.

4. Алгоритм численного исследования бифуркации.

Теоремы 1 и 2 позволяют провести численное исследование бифуркации Андронова-Хопфа для системы (1) в виде следующего алгоритма:

1. строятся удовлетворяющие соотношениям (2) и (3) векторы $e, g, e^*, g^* \in R^N$, а также функция (5);
2. составляется подпрограмма-функция для вычисления функционалов (7) и операторов (9) и (10);
3. составляется подпрограмма-функция для вычисления обратного оператора к оператору (11);
4. составляется программа вычисления итераций по схеме (12).

Указанный алгоритм реализован авторами в виде MATLAB-программы. Вычисления по этой программе иллюстрируются в приводимом ниже примере.

5. Дополнение

Предложенная выше схема приближенного исследования бифуркации позволяет не только численно строить бифурцирующие решения. Она развита и дополнена с целью решения, в частности, следующих задач.

1. Получение асимптотических формул для значений параметра λ_q , периода T_q и периодических решений $x_q(t) \equiv y_q(t/T_q)$. Эти формулы были получены в [5] в виде равенств: $\lambda_q = \lambda_0 + \lambda_1 q^2 + o(q^2)$, $T_q = T_0 + T_1 q^2 + o(q^2)$ и $y_q(t) = qe(t) + q^2 e_1(t) + o(q^2)$, где для λ_1 , T_1 и $e_1(t)$ были получены явные формулы.
2. Определение типа бифуркации (суб- или суперкритичность) (см. [6]).
3. Анализ устойчивости решений. На основе асимптотических формул разработан (см. [5]) алгоритм вычисления характеристического показателя Δ , знак которого определяет свойства устойчивости.
4. Исследование систем, содержащих запаздывания той или иной природы (см. [7]), а также систем, содержащих сложные нелинейности, например, гистерезисного типа.

6. Пример: система Лоренца

Рассмотрим систему Э. Лоренца [1–3]

$$\begin{cases} \dot{x} = -\sigma x + \sigma y, \\ \dot{y} = -xz + \lambda x - y, \\ \dot{z} = xy - bz, \end{cases} \quad (13)$$

описывающую движения в подогреваемом снизу слое жидкости или газа. Здесь σ — число Прандтля, величина b отражает геометрию области, λ — число Рэлея, принимаемое за бифуркационный параметр.

Система (13) при $\lambda > 1$ имеет два состояния равновесия $x = y = \pm\sqrt{b(\lambda - 1)}$, $z = \lambda - 1$. Линеаризуя правую часть системы (13) в окрестности точки $x = y = \sqrt{b(\lambda - 1)}$, $z = \lambda - 1$, приходим к системе вида (1), в которой

$$A(\lambda) = \begin{bmatrix} -\sigma & \sigma & 0 \\ 1 & -1 & -k \\ k & -k & -b \end{bmatrix}, \quad a(x; \lambda) = \begin{bmatrix} 0 \\ -x_1 x_3 \\ x_1 x_2 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix};$$

здесь $k = \sqrt{b(\lambda - 1)}$. При $\sigma > b + 1$ число $\lambda_0 = \sigma(\sigma + b + 3)/(\sigma - b - 1)$ является точкой бифуркации Хопфа.

Исследования бифуркации в системе (2) проводилось с использованием разработанной MATLAB-программы. Для физически важных значе-

ний параметров $\sigma = 10$ и $b = 8/3$ бифуркационное значение равно $\lambda_0 = 470/19$, а матрица $A_0 = A(\lambda_0)$ имеет вид

$$A_0 = \begin{bmatrix} -10 & 10 & 0 \\ 1 & -1 & -7,956 \\ 7,956 & 7,956 & -2,667 \end{bmatrix}$$

и имеет пару чисто мнимых собственных значений $\pm \omega_0 i$, где $\omega_0 = 9,6245$. Тогда $T_0 = 2\pi/\omega_0 = 0,6528$. Вычислим собственные векторы матриц A_0 и A_0^* в соответствии с равенствами (2) и (3):

$$e = \begin{bmatrix} 0,1454 \\ -0,3899 \\ 0,9093 \end{bmatrix}, \quad g = \begin{bmatrix} 0,5562 \\ 0,6961 \\ 0,4541 \end{bmatrix}, \quad e^* = \begin{bmatrix} 0,1364 \\ -0,6347 \\ 0,8058 \end{bmatrix}, \quad g^* = \begin{bmatrix} 0,4584 \\ 0,8738 \\ 0,3014 \end{bmatrix}.$$

Приведем окончательные результаты. Для асимптотических формул получим $\lambda_1 = -0,0423$ и $T_1 = 0,0015$. Следовательно, рассматриваемая бифуркация является субкритической. Используя значения λ_1 и T_1 для анализа устойчивости получим $\Delta = 0,0395$, т. е. бифурцирующие решения неустойчивы. Это подтверждает результаты прямых численных расчетов Э. Лоренца, а также вычислений по бифуркационным формулам [2].

Литература

1. Гукенхеймер Дж., Холмс Ф. Нелинейные колебания, динамические системы и бифуркации векторных полей.— М.-Ижевск: Институт компьютерных исследований, 2002.— 560 с.
2. Хэссард Б., Казаринов Н., Вэн И. Теория и приложения бифуркации рождения цикла.— М.: Мир, 1985.— 280 с.
3. Kuznetsov Yu. A. Elements of Applied Bifurcation Theory.— Applied Mathematical Sciences (V.112). Springer-Verlag, New-York etc., 1995.— 504 p.
4. Козякин В. С., Красносельский М. А. Метод функционализации параметра в задаче о точках бифуркации // Доклады АН СССР.— 1980.— Т.254.— №5.— С.1061–1064.
5. Красносельский М. А., Кузнецов Н. А., Юмагулов М. Г. Операторный метод анализа устойчивости циклов при бифуркации Хопфа // Автоматика и телемеханика.— 1996.— № 12.— С.24–30.
6. Кузнецов Н. А., Матвеев Н. И., Юмагулов М. Г. Признаки суб- и суперкритичности в задаче о бифуркации Хопфа и задачи односторонней бифуркации // Автоматика и телемеханика.— 1998.— №12.— С.51–59.
7. Юмагулов М. Г. Метод функционализации параметра в итерационных процедурах исследования бифуркации Хопфа для уравнений с последствием // Доклады РАН.— 1993.— Т.331.— №1.— С.24–27.