

01 НЕ МИФИ

004

П64

МОСКОВСКИЙ
ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ЭНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ

В. Г. Потемкин

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ

(Алгоритмы исследования
систем управления)

МОСКВА 1980

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ СССР

МОСКОВСКИЙ
ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ

В. Г. Потемкин

АВТОМАТИЗИРОВАННОЕ
ПРОЕКТИРОВАНИЕ

(алгоритмы исследования
систем управления)

Утверждено
редсоветом института
в качестве учебного пособия

МОСКВА 1980

Потемкин В. Г. Автоматизированное проектирование (алгоритмы исследования систем управления). Учебное пособие. - М. : Изд. МИФИ, 1980, 88с.

В данном учебном пособии изложены методы и алгоритмы исследования динамических характеристик при проектировании систем управления. С позиций единого подхода, основанного на спектральном разложении матриц динамической системы, рассмотрено решение некоторых задач расчета систем. Даны сведения об области применения и структуре алгоритмов, приведены соответствующие процедуры на языке ПЛ/1. Описан пакет прикладных программ для решения задач исследования систем управления, работа с которым может быть организована как в пакетном, так и диалоговом режимах на ЭВМ единой серии.

Учебное пособие предназначено для студентов в целях приобретения практических навыков расчета динамических систем при изучении курсов теории управления с применением ЭВМ, организации практических и лабораторных занятий. Оно может быть полезно и преподавателям вузов соответствующих специальностей.



ПРЕДИСЛОВИЕ

Развитие современной теории управления связано с широким внедрением в практику проектирования систем управления средств вычислительной техники.

Разработка алгоритмов, одинаково пригодных как для проектирования непрерывных, так и дискретных динамических систем, является сложной задачей. Существуют подходы, основанные на частотных методах [18], разложениях в ряд матричной экспоненты [15, 19], решении характеристических уравнений [18, 19]. В данном пособии, которое является логическим продолжением ранее опубликованного учебного пособия [9], развивается подход, основанный на спектральном разложении матриц системы. Это позволяет с единых алгоритмических позиций решить всю совокупность проблем, встречающихся при проектировании систем управления, а именно задачи линеаризации, построения дискретных моделей, соединения динамических систем, вычисления коэффициентов, а также полюсов и нулей передаточных функций; вычисления и построения частотных характеристик, корневых годографов и переходных процессов; оценки качества управляемых процессов; анализа автоколебаний.

В пособии описано 14 модулей, входящих в состав пакета программ исследования динамических систем. Перечень задач, которые могут быть решены с помощью модулей пакета, не исчерпывается теми, которые приведены в пособии. Из-за ограниченности объема не описаны алгоритмы решения задач анализа автоколебаний, оценки управляемости, наблюдаемости, цифровой реализации непрерывных законов управления, которые могут быть решены на базе описанных модулей. Наиболее удобным способом решения этих задач является режим диалога проектировщика с ЭВМ с помощью алфавитно-цифровых или графических терминалов. Проблема организации диалога подробно описана в работе [10].

Автор приносит благодарность сотрудникам кафедры за обсуждение материала и оформление рукописи пособия.

1. ЛИНЕАРИЗАЦИЯ ДИНАМИЧЕСКИХ СИСТЕМ

При математическом описании динамики систем управления [9] наиболее полным является описание посредством обыкновенных нелинейных дифференциальных уравнений в форме Коши, определенных на интервале $(t_0, t]$

$$\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{u}), \quad \vec{x}_0 = \vec{x}(t_0), \quad (1)$$

где \vec{x} - n -мерный вектор состояния системы;

\vec{u} - m -мерный вектор управления;

\vec{x}_0 - вектор начальных условий.

В настоящее время решение задач анализа и синтеза может быть выполнено только для линейных моделей динамики объекта управления. В связи с этим рассмотрим методы и вычислительные алгоритмы линеаризации уравнений типа (1) с применением ЭВМ.

Исследуем для простоты процедуру линеаризации уравнения (1) относительно положения равновесия \vec{x}_e при кусочно-постоянном управлении. Можно сформулировать следующую задачу линеаризации для системы нелинейных уравнений (1): построить линеаризованную модель динамической системы вида

$$\delta \dot{\vec{x}} = A \delta \vec{x} + B \delta \vec{u} + \vec{h}(\delta \vec{x}, \delta \vec{u}), \quad (2)$$

где вектор \vec{h} имеет компоненты второго и более высоких порядков малости по сравнению с нормами вариаций вектора состояния $\|\delta \vec{x}\|$ и вектора управления $\|\delta \vec{u}\|$ относительно равновесных значений. Поставленную задачу линеаризации можно рассматривать как задачу аппроксимации нелинейной векторной функции $\delta \vec{f}(\delta \vec{x}, \delta \vec{u}) = \vec{f}(\vec{x}, \vec{u}) - \vec{f}(\vec{x}_e, \vec{u}_e)$ линейной векторной функцией $A \delta \vec{x} + B \delta \vec{u}$. Выбор матриц A и B подчинен условию минимизации среднеквадратической ошибки следующего вида:

$$J = \min_{A, B} \left\| \delta \vec{f} - [A \mid B] \begin{bmatrix} \delta \vec{x} \\ \delta \vec{u} \end{bmatrix} \right\|_2. \quad (3)$$

Сформулированная задача может быть решена методом наименьших квадратов [2, 11]. Образует последовательность измерений, пронумерованных от 1 до N , необходимых для оценки матриц A и B ,

$$\begin{bmatrix} \delta \vec{f}_1^T \\ \delta \vec{f}_2^T \\ \vdots \\ \delta \vec{f}_N^T \end{bmatrix}_{N \times n} = \begin{bmatrix} \delta \vec{x}_1^T & \delta \vec{u}_1^T \\ \delta \vec{x}_2^T & \delta \vec{u}_2^T \\ \vdots & \vdots \\ \delta \vec{x}_N^T & \delta \vec{u}_N^T \end{bmatrix}_{N \times (n+m)} \begin{bmatrix} A_{n \times n}^T \\ \vdots \\ B_{m \times n}^T \end{bmatrix}_{(n+m) \times n} \quad (4a)$$

или в компактной матричной форме

$$F = XM. \quad (4б)$$

Оценка $n^2 + mn$ неизвестных параметров, объединенных в матрицу M , имеет вид [2]:

$$\hat{M} = X^\# F = (X^T X)^{-1} X^T F, \quad (5)$$

где $X^\#$ — матрица, псевдообратная матрице измерений $X_{N \times (n+m)}$; \hat{M} — оценка матрицы M , зависящая от числа измерений N , причем $\min N = n + m$.

Соотношение (5) определяет алгоритм для решения поставленной задачи оценки параметров линейной модели (2).

Схема алгоритма представлена на рис. 1 и включает следующие блоки: моделирование исходных нелинейных уравнений или снятие характеристик с реальной системы (блок 1); формирование матриц F и X уравнения (4) (блок 2); вычисление матрицы псевдообратной матрице измерений $X^\#$ (блок 3); формирование оценки \hat{M} (блок 4); вывод на печать матриц \hat{A} и \hat{B} (блок 5); оценка ошибки аппроксимации J (блок 6).

Алгоритмы моделирования рассмотрены в работах [8, 16, 17].

Рассмотрим алгоритм *MINFIT* [13] псевдообращения прямоугольной матрицы измерений X . В рассматриваемом случае число строк равно или превышает число столбцов матрицы. Поскольку ранг матрицы X неизвестен, воспользуемся алгоритмом псевдообращения, основанном на спектральном разложении прямоугольной матрицы X по сингулярным числам.

Пусть X — действительная матрица размером $m \times n$ ($m \geq n$).

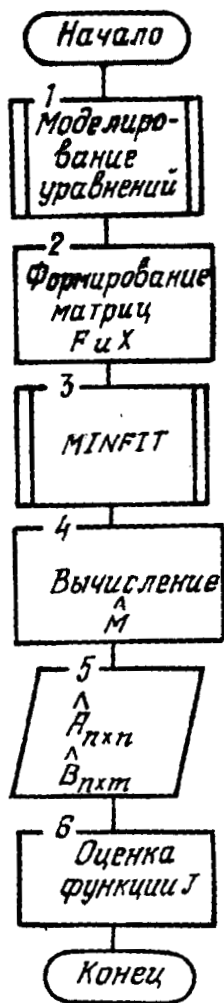


Рис. 1. Схема алгоритма решения задачи линеаризации

Известно [14], что эту матрицу можно представить в виде

$$X = U \Sigma V^T, \quad (6)$$

где $U^T U = V^T V = V V^T = I_{n \times n}$; $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$.

Матрица U составлена из n ортонормированных собственных векторов, соответствующих n наибольшим собственным значениям матрицы XX^T , а матрица V - из ортонормированных собственных векторов матрицы $X^T X$. Диагональные элементы σ_i ($i=1, \dots, n$) матрицы Σ - неотрицательные значения квадратных корней от собственных значений матрицы $X^T X$; они называются сингулярными числами. Допустим, что

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$. Если ранг матрица X равен r , то значения $\sigma_{r+1} = \sigma_{r+2} = \dots = \sigma_n = 0$. Разложение прямоугольной матрицы X вида (6) называется спектральным разложением, или разложением по сингулярным числам.

В процедуре *MINFIT* представленной ниже, сначала матрица X будет приведена к двухдиагональной форме с помощью преобразований Хаусхолдера, а затем использован *QR*-алгоритм для вычисления сингулярных чисел двухдиагональной матрицы.

Данная процедура является результатом переработки процедуры *min fit* [13] на языке АЛГОЛ на язык ПЛ/1. Она осуществляет спектральное разложение исходной матрицы по сингулярным числам и может быть применена для определения матрицы X , минимизирующей нормы $\|XM - F\|_E$ и $\|M\|_E$. Матрица M формируется следующим образом:

$$M = V \text{diag}(\sigma_i^\#) U^T F, \quad (7)$$

где

$$\sigma_i^\# = \begin{cases} 1/\sigma_i, & \text{если } \sigma_i > 0; \\ 0, & \text{если } \sigma_i = 0. \end{cases}$$

С помощью алгоритма вычисляются сингулярные числа σ_i матрицы V и $C = U^T F$.

Применение процедуры *MINFIT*. Для оценки параметров линеаризованной системы (2) можно использовать процедуру *MINFIT* с подстановкой фактических параметров в соответствии с уравнением (4б). Обращение к процедуре имеет вид:

$$\text{CALL MINFIT}(NN, NM, N, EPS, TOL, XF, Q). \quad (8)$$

После вычисления с использованием процедуры *MINFIT* могут быть сформированы блочная матрица $C_{NM \times NM} = [\sigma_1^1 \sigma_2^1 \dots \sigma_N^1]$ и матрица $V_{NM \times NM}$. Согласно соотношениям (5) и (7)

оценку неизвестных параметров матрицы $M_{NM \times N}$ проводят следующим образом:

$$\begin{bmatrix} \lambda^T \\ A_{N \times N} \\ \hline \hat{B}_{M \times N}^T \end{bmatrix} = M_{NM \times N} = V_{NM \times NM} \times [q_1 \vec{c}_1 : q_2 \vec{c}_2 : \dots : q_N \vec{c}_N]_{NM \times N} \quad (9)$$

Таким образом, линеаризованная модель вида (2) для исходной нелинейной системы (1) может быть записана так:

$$\dot{\vec{x}} = A \vec{x} + B \vec{u}, \quad \vec{x}_0 = \vec{x}(t_0). \quad (10)$$

Входные параметры процедуры MINFIT :

NN - число строк матрицы X ;

NM - число столбцов матрицы X ;

N - число столбцов матрицы F ;

EPS - константа, используемая для проверки сходимости, она не должна быть меньше машинной точности $macheps$, удовлетворяющей условию $1 + macheps > 1$. Для вычислительных машин серии ЕС ЭВМ $macheps = 2^{-56} (\sim 10^{-16})$;

TOL - константа, зависящая от типа вычислительной машины, равная $eta / macheps$, где eta - наименьшее положительное число, представимое в машине. Для вычислительных машин серии ЕС ЭВМ $eta = 2^{256} (10^{-24})$;

$XF(NN, NM + N)$ - массив для размещения матриц $X_{NM \times NM}$ и $F_{NM \times N}$.

Выходные параметры процедуры MINFIT:

$XF(NN, NM+N) = \begin{bmatrix} V_{NM \times NM} & \vdots & C_{NM \times N} \\ \hline Z_{(NN-NM) \times (NM+N)} \end{bmatrix}$ - массив, в котором размещены матрица собственных векторов $V_{NM \times NM}$ и матрица $C_{NM \times N}$. Массив Z полезной информации не содержит;

$Q(N)$ - массив сингулярных чисел.

Программа процедуры *MINFIT* :

```
MINFIT:
PROC(M,N,P,EPS,TOL,AB,Q);
DCL (M,N,P) BIN FIXED,(EPS,TOL,AB(+,+),Q(+)) DEC(12); DCL I,J,K,L,
L1,N1,NP,(C,F,G,H,S,X,Y,Z,E(N)) DEC(12); G,X=0; NP=N+P;
DO I=1 TO N;
  E(I)=G; S=0; L=I+1;
  DO J=I TO M;
    S=S+AB(J,I)+AB(J,I);
  END;
  IF S<TOL THEN G=0; ELSE
  DO;
    F=AB(I,I); IF F<0 THEN G=SQRT(S); ELSE G=-SQRT(S);
    H=F+G-S; AB(I,I)=F-G;
    DO J=L TO NP;
      S=0;
      DO K=I TO M;
        S=S+AB(K,I)+AB(K,J);
      END;
      F=S/H;
      DO K=I TO M;
        AB(K,J)=AB(K,J)+F+AB(K,I);
      END;
    END;
  END;
  Q(I)=G; S=0; IF I<=M THEN
  DO J=L TO N;
    S=S+AB(I,J)+AB(I,J);
  END;
  IF S<TOL THEN G=0; ELSE
  DO;
    F=AB(I,I+1); IF F<0 THEN G=SQRT(S); ELSE G=-SQRT(S);
    H=F+G-S; AB(I,I+1)=F-G;
    DO J=L TO N;
      E(J)=AB(I,J)/H;
    END;
  DO J=L TO M;
```

```

      S=0;
      DO K=L TO N;
        S=S+AB(J,K)*AB(I,K);
      END;
      DO K=L TO N;
        AB(J,K)=AB(J,K)+S*E(K);
      END;
    END;
  END;
  Y=ABS(Q(I))+ABS(E(I)); IF Y>X THEN X=Y;
END;
DO I=N TO 1 BY -1;
  IF G=0 THEN
  DO;
    H=AB(I,I+1)*G;
    DO J=L TO N;
      AB(J,I)=AB(J,I)/H;
    END;
    DO J=L TO N;
      S=0;
      DO K=L TO N;
        S=S+AB(I,K)*AB(K,J);
      END;
      DO K=L TO N;
        AB(K,J)=AB(K,J)+S*AB(K,I);
      END;
    END;
    DO J=L TO N;
      AB(I,J),AB(J,I)=0;
    END;
    AB(I,I)=1; G=E(I); L=I;
  END;
  EPS=EPS+X; N1=N+1;
  DO I=M+1 TO N;
    DO J=N1 TO NP;
      AB(I,J)=0;

```

```

END;
END;
DO K=N TO 1 BY -1;
TESTFS:
  DO L=K TO 1 BY -1;
    IF ABS(E(L))<=EPS THEN GOTO TESTFC; IF ABS(Q(L-1))<=EPS THEN
    GOTO CANCEL;
  END;
CANCEL:
  C=0; S=1; L1=L-1;
  DO I=L TO K;
    F=S+E(I); E(I)=C+E(I); IF ABS(F)<=EPS THEN GOTO TESTFC;
    G=Q(I); Q(I)=SQRT(F*F+G*G); H=Q(I); C=G/H; S=-F/H;
    DO J=N1 TO NP;
      Y=AB(L1,J); Z=AB(I,J); AB(L1,J)=C*Y+S*Z; AB(I,J)=-S*Y+C*Z;
    END;
  END;
TESTFC:
  Z=Q(K); IF L=K THEN GOTO CONVER; X=Q(L); Y=Q(K-1);
  G=E(K-1); H=E(K); F=((Y-Z)*(Y+Z)+(G-H)*(G+H))/(2*H*Y);
  G=SQRT(F*F+1); IF F<0 THEN F=((X-Z)*(X+Z)+H*(Y/(F-G)-E))/X;
  ELSE F=((X-Z)*(X+Z)+H*(Y/(F+G)-H))/X; C,S=1;
  DO I=L+1 TO K;
    G=E(I); Y=Q(I); H=S*G; G=C*G; E(I-1),Z=SQRT(F*F+H*H);
    C=F/Z; S=H/Z; F=X*C+G*S; G=-X*S+G*C; H=Y*S; Y=Y*C;
    DO J=1 TO N;
      X=AB(J,I-1); Z=AB(J,I); AB(J,I-1)=X*C+Z*S; AB(J,I)=-X*S+Z*C;
    END;
    Q(I-1),Z=SQRT(F*F+H*H); C=F/Z; S=H/Z; P=C*G+S*Y; X=-B*G+C*Y;
    DO J=N1 TO NP;
      Y=AB(I-1,J); Z=AB(I,J); AB(I-1,J)=C*Y+S*Z; AB(I,J)=-S*Y+C*Z;
    END;

```

2. ПОСТРОЕНИЕ ДИСКРЕТНЫХ МОДЕЛЕЙ СИСТЕМ УПРАВЛЕНИЯ

При расчете систем управления, в состав которых входят вычислительные устройства, цифровые регуляторы, датчики с цифровым выходом и т.п., возникает необходимость построения дискретных моделей неизменяемой части системы, а также устройств управления и коррекции.

Допустим, что рассматриваемые динамические системы являются линеаризованными и описываются уравнением (10). Изложим общий подход к построению дискретных моделей.

Аналитическое решение системы уравнений вида (10) известно [10] и определяется так:

$$\vec{x}(t) = e^{A(t-t_0)} \vec{x}(t_0) + \int_{t_0}^t e^{A(t-\tau)} B \vec{u}(\tau) d\tau. \quad (11)$$

Зафиксируем моменты времени $t = kT$ и $t_0 = (k-1)T$, где T - такт дискретности, обуславливающий выборку из процессов $\vec{x}(t)$ и $\vec{u}(t)$. Тогда состояние системы \vec{x}_k в момент времени kT связано с предшествующим состоянием \vec{x}_{k-1} в момент $(k-1)T$ следующим соотношением:

$$\vec{x}_k = \Phi \vec{x}_{k-1} + \int_0^T e^{A(T-\tau)} B \vec{u}(\tau) d\tau. \quad (12)$$

Нетрудно видеть, что состояние \vec{x}_k зависит от изменения входа \vec{u} на интервале дискретности. Различные предположения о форме сигнала $\vec{u}(\tau)$ на этом интервале приводят к разным типам дискретных моделей.

Простейшее предположение о постоянстве входа \vec{u} внутри интервала дискретности $\vec{u}(\tau) = \vec{u}_{k-1}$ приводит к следующей дискретной модели:

$$\vec{x}_k = \Phi \vec{x}_{k-1} + F \vec{u}_{k-1}, \quad (13)$$

где $\Phi = e^{AT}$ - переходная матрица системы (10);

$F = (\Phi - I)A^{-1}B$ - матрица коэффициентов передачи по входам дискретной модели.

```

END;
E(L)=0; E(K)=F; Q(K)=X; GOTO TESTFS;
CONVER:
DO; IF Z<0 THEN
Q(K)=-Z;
DO J=1 TO N;
AB(J,K)=-AB(J,K);
END;
END;
END MINFIT;
    
```

Предположение о линейном изменении входа $\bar{u}(\tau)$ на интервале дискретности по закону $\bar{u}(\tau) = \bar{u}_{k-1} + \frac{\bar{u}_{k-1} - \bar{u}_{k-2}}{T} \tau$ приводит к дискретной модели

$$\bar{x}_k = \Phi \bar{x}_{k-1} + F \bar{u}_{k-1} + G \frac{\bar{u}_{k-1} - \bar{u}_{k-2}}{T}, \quad (14)$$

где $\Phi = e^{AT}$ — переходная матрица системы (10);

$$F = (\Phi - I) A^{-1} B;$$

$$G = (F - IT) A^{-1} B.$$

Можно использовать аппроксимации и более высоких порядков. Очевидно, что при увеличении порядка аппроксимирующей функции возможно некоторое снижение требований к величине периода квантования для получения сопоставимых показателей качества исходной непрерывной и построенной дискретной моделей.

На практике при построении дискретных моделей неизменяемой части системы чаще всего используют модель вида (13). При цифровой реализации корректирующих устройств и устройств управления могут применяться дискретные модели более высоких порядков.

Для упрощения последующего изложения положим, что в системе (10) вход является скалярным. При исследовании линейных моделей это не ограничивает общности рассуждений. В этом случае дискретная модель, описываемая соотношением (12), принимает вид:

$$\bar{x}_k = \Phi(T) \bar{x}_{k-1} + \int_0^T \Phi(T-\tau) \bar{b} u(\tau) d\tau. \quad (15)$$

Аппроксимируем вход $u(\tau)$ степенной функцией порядка ℓ :

$$u(\tau) = \sum_{j=0}^{\ell} \mu_j \tau^j = [1 \tau \tau^2 \dots \tau^{\ell}] \begin{bmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_{\ell} \end{bmatrix}. \quad (16)$$

Оценить коэффициенты этой функции можно по N измерениям входа в моменты $u(k-1), \dots, u(k-\ell-1), \dots, u(k-N)$, стремясь через эти точки провести кривую ℓ -го порядка наилучшего среднеквадратичного приближения. Для этого воспользуемся методом наименьших квадратов в следующей форме. Измерения входа в N точках связаны с вектором оцениваемых параметров следующим соотношением:

$$\begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-N) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & (-1)^{\ell} \\ \dots & \dots & \dots & \dots \\ 1 & -(N-1) & \dots & [-(N-1)]^{\ell} \end{bmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 T \\ \vdots \\ \mu_{\ell} T^{\ell} \end{bmatrix}, \quad (17)$$

или в векторно-матричной форме записи:

$$\bar{u}_{N \times 1} = S_{N \times (\ell+1)} \hat{\mu}_{(\ell+1) \times 1}. \quad (18)$$

Оценка вектора $\hat{\mu}$ задается процедурой псевдообращения матрицы S , так что

$$\hat{\mu} = S^{\#} \bar{u} = (S^T S)^{-1} S^T \bar{u}. \quad (19)$$

Подстановка (16) в (15) позволяет получить дискретную модель в виде:

$$\begin{aligned} \bar{x}_k &= \Phi \bar{x}_{k-1} + [H_0 \bar{b}; H_1 \bar{b}/T; \dots; H_{\ell} \bar{b}/T^{\ell}] S^{\#} \bar{u} = \\ &= \Phi \bar{x}_{k-1} + \bar{f}_0 u_{k-1} + \bar{f}_1 u_{k-2} + \dots + \bar{f}_{N-1} u_{k-N}, \end{aligned} \quad (20)$$

где $H_j = \int_0^T \Phi(T-\tau) \tau^j d\tau, j=0, 1, \dots, \ell$.

Для вычисления матрицы H_j можно применить рекуррентное соотношение:

$$\left. \begin{aligned} H_j &= (j H_{j-1} - I T^j) A^{-1}; \\ H_0 &= (e^{AT} - I) A^{-1}. \end{aligned} \right\} \quad (21)$$

Схема алгоритма расчета дискретных моделей представлена на рис. 2. В этом алгоритме предусмотрено использование процедур *COMEIG*, *MFUNC*, *DISCR*. Процедура *COMEIG* [13] осуществляет спектральное разложение исходной матрицы системы. Процедура *MFUNC* позволяет вычислить некоторые функции от матриц, используемые в теории управления. Процедура *DISCR* реализует расчет дискретной модели в форме соотношения (20).

Процедура *DISCR*, используя спектральное представление матрицы исходной непрерывной системы, вычисляет массивы Φ , F следующей структуры:

$$\Phi(N+NN-1, N+NN-1) = \left[\begin{array}{c|ccc|c} e^{AT} & \vec{f}_1 & \vec{f}_2 & \dots & \vec{f}_{NN-1} \\ \hline & & 0 & & \\ \hline 0 & & & I_{(NN-2) \times (NN-2)} & 0 \end{array} \right]; \quad (22)$$

$$F(N+NN-1) = [\vec{f}_0; 1; 0]^T.$$

Массив, состоящий из векторов \vec{f}_i , $i=1, 2, \dots, NN-1$, и вектор \vec{f}_0 определяются из следующего массива размером $N \times NN$:

$$[\vec{f}_0; \vec{f}_1; \vec{f}_2; \dots] = [\vec{f}_0; \vec{f}_1; \vec{f}_2; \dots; \vec{f}_{NN-1}] = [H_0 \times B; H_1 \times B/T; \dots; H_L \times B/T^{**L}] \times S\#, \quad (23)$$

где $S\# = (INV(ST * S)) * ST$;

ST - матрица, транспонированная S ;

H_0, H_1, \dots, H_L - матрицы, вычисляемые в соответствии с формулой (21).

Схема алгоритма *DISCR* приведена на рис. 3.

Применение процедуры *DISCR*. Обращение к процедуре *DISCR* имеет вид:

$$CALL DISCR (N, A, B, T, NN, L, \Phi, F). \quad (24)$$

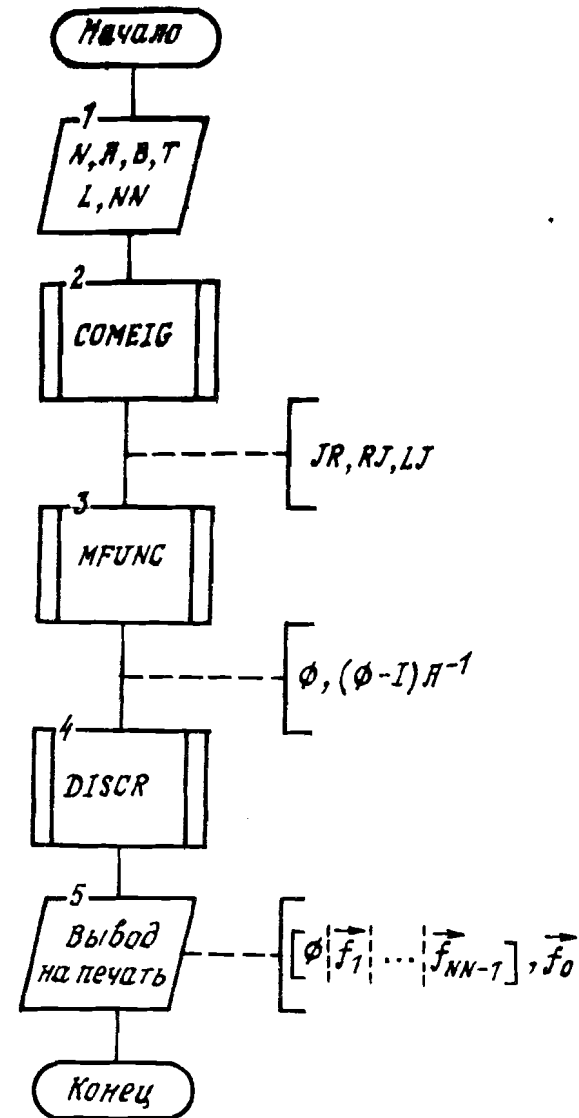


Рис. 2. Схема алгоритма расчета дискретной модели

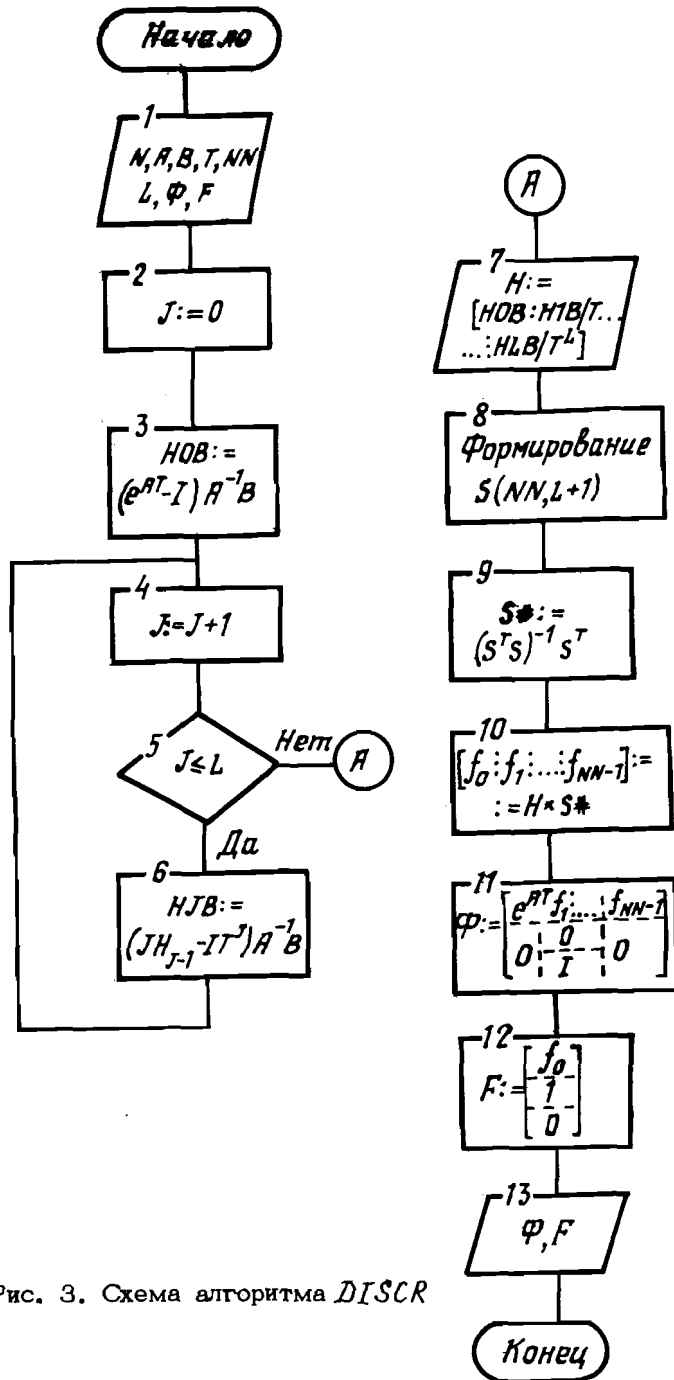


Рис. 3. Схема алгоритма DISCR

Эту процедуру можно использовать для вычисления дискретных моделей неизменяемой части системы и исследования различных способов аппроксимации входного сигнала при цифровой реализации непрерывных корректирующих фильтров.

Входные параметры процедуры DISCR :

- N - порядок исходной непрерывной системы;
- A - матрица исходной непрерывной системы;
- B - вектор передачи управления непрерывной системы с одним входом^{*};
- T - такт дискретности;
- NN - число измерений входного сигнала, используемых для построения аппроксимирующей степенной функции наилучшего среднеквадратического приближения входного сигнала;
- L - порядок степенной функции;
- Φ - матрица размером $(N+NN-1) \times (N+NN-1)$ в верхнем левом углу которой расположена матрица e^{AT} размером $N \times N$; остальные элементы нулевые.

Выходные параметры процедуры DISCR :

- Φ - матрица дискретной модели размером $(N+NN-1) \times (N+NN-1)$ вида (22);
- F - вектор передачи управления размера $N+NN-1$ вида (22).

* Процедура вычисляет дискретную модель для системы с одним входом. В случае многомерной системы с k входами необходимо k -кратное обращение к процедуре DISCR.

Программа процедуры *DISCR* :

```
DISCR:
PROC(N, AA, H, TAKT, M, L, A, B);
DCL I, J, K, L, M, N, ((A, AA)(*), (B, H)(*), D(N, N), S(M, L+1), (ST, T)(L+1, M),
(STS, TS)(L+1, L+1), (C, G)(N, N), E(N, L+1), R(N, M), TAK, TAKT) DEC(12);
CALL FAD(AA, N, D); C, G=0; E=0; KOT=L+1; B=0; S=1;
DO J=2 TO L+1;
  S(1, J)=0;
  DO I=2 TO M;
    S(I, J)=(1-I)**(J-1);
  END;
END;
CALL MATR(S, M, KOT, ST); CALL MATR(ST, S, KOT, M, KOT, STS); CALL FAD(
STS, KOT, TS); CALL MATR(TS, ST, KOT, KOT, M, T);
DO I=1 TO N;
  DO J=1 TO N;
    G(I, J)=A(I, J);
  END;
END;
DO I=1 TO N;
  G(I, I)=G(I, I)-1;
END;
CALL MATR(G, D, N, N, N, C);
DO K=1 TO L+1;
  DO I=1 TO N;
    DO J=1 TO N;
      E(I, K)=E(I, K)+C(I, J)*H(J)/TAKT**(K-1);
    END;
  END;
  TAK=-TAK**K;
  DO I=1 TO N;
    DO J=1 TO N;
      C(I, J)=K*C(I, J);
    END;
  END;
  DO I=1 TO N;
    C(I, I)=C(I, I)+TAK;
  END;
G=C; CALL MATR(C, D, N, N, N, C);

END;
CALL MATR(E, T, A, KOT, M, R); B=0;
DO I=1 TO N;
  B(I)=R(I, 1);
END;
IF M>1 THEN
DO;
  B(N+1)=1;
  DO I=1 TO N;
    B(I)=B(I)+B(N+1);
  END;
END;
```



```

END;
CALL MATR(E,T,N,KOT,M,R); B=0;
DO I=1 TO N;
  B(I)=R(I,1);
END;
IF M>1 THEN
DO;
  B(N+1)=1;
  DO I=1 TO N+M-1;
    DO J=1 TO N+M-1;
      IF I>N : J>N THEN A(I,J)=0;
    END;
  END;
  DO I=1 TO N;
    DO J=N+1 TO N+M-1;
      A(I,J)=R(I,J-N+1);
    END;
  END;
  IF M>2 THEN
  DO I=N+1 TO N+M-2;
    A(I+1,I)=1;
  END;
END;
MATR:
PROC(A,N,M,AT);
  DCL ((A,AT)(+,*)) DEC(15),N,M,I,J;
  DO I=1 TO M;
    DO J=1 TO N;
      A(I,J)=A(J,I);
    END;
  END MATR;
MATB:
PROC(A,B,N,K,M,C);
  DCL N,M,K-1,J,KJ1,((A,B,C)(+,*)) DEC(15);
  DO I=1 TO N;
    DO J=1 TO M;
      C(I,J)=0;
      DO KJ1=1 TO K;
        C(I,J)=C(I,J)+A(I,KJ1)+B(KJ1,J);
      END;
    END;
  END;

```

```

END MATR;
FAD: PROC(A,N,B);
  DCL N,I,J,K,P DEC(15),((A,B)(*,*),A1(N,N)) DEC(15);
  DO I=1 TO N;
  DO J=1 TO N;
  A1(I,J)=A(I,J); IF I=J THEN B(I,J)=1; ELSE B(I,J)=0;
  END;
  DO K=1 TO N;
  DO I=1 TO N;
  IF K=1 THEN GOTO M1; CALL MATR(A,B,N,N,M,A1);
  P=0;
  DO I=1 TO N;
  DO J=1 TO N;
  P=P+A1(I,I);
  END;
  M2: P=P/K; IF P=0 THEN PUT SKIP LIST('СЛОВА=0'); IF K=N THEN GOTO
  M3;
  DO I=1 TO N;
  DO J=1 TO N;
  B(I,J)=B(I,J)/P;
  END;
  M3: GOTO M3;
  DO I=1 TO N;
  DO J=1 TO N;
  IF I=J THEN B(I,J)=A1(I,J)+P; ELSE B(I,J)=A1(I,J);
  END;
  END FAD;
END DISCR;

```

3. СПЕКТРАЛЬНОЕ РАЗЛОЖЕНИЕ ПРОИЗВОЛЬНЫХ ДЕЙСТВИТЕЛЬНЫХ И КОМПЛЕКСНЫХ МАТРИЦ. ВЫЧИСЛЕНИЕ ФУНКЦИЙ ОТ МАТРИЦ

Описание динамических систем в векторно-матричной форме позволяет разрабатывать алгоритмы расчета таких систем с единой позиции, используя результаты спектральной теории матриц [3, 10]. При этом в дополнение к обычно используемым характеристикам (передаточные функции, частотные характеристики, корневые годографы) могут быть добавлены характеристики управляемости, наблюдаемости, чувствительности. Теоретической предпосылкой для разработки алгоритмов вычисления этих характеристик является возможность представления решения линеаризованной системы (10) на основе спектрального разложения исходной матрицы. В дальнейшем без ограничения общности исследования линеаризованных систем можно рассматривать системы с одним входом и одним выходом, описываемые следующими уравнениями:

для непрерывных систем (аналог уравнений (10))

$$\left. \begin{aligned} \dot{\vec{x}} &= A\vec{x} + \vec{b}u, \quad \vec{x}_0; \\ y &= \vec{c}^T \vec{x}, \end{aligned} \right\} \quad (25)$$

где A - матрица непрерывной системы;
 \vec{b} - вектор передачи управления;
 \vec{c}^T - вектор формирования выхода;
 \vec{x}_0 - вектор начальных условий;

для дискретных систем (аналог уравнений (13))

$$\left. \begin{aligned} \vec{x}_k &= \Phi \vec{x}_{k-1} + \vec{f}u, \quad \vec{x}_0; \\ y_k &= \vec{c}^T \vec{x}_k, \end{aligned} \right\} \quad (26)$$

где Φ - переходная матрица;
 \vec{f} - вектор передачи управления;
 \vec{c}^T - вектор формирования выхода;
 \vec{x}_0 - вектор начальных условий.

Решение уравнения (25) с учетом разложения движения системы по элементарным составляющим представляется в виде [9]:

$$y(t) = \sum_{i=1}^n \vec{c}^T \vec{r}_i \vec{\ell}_i^H \vec{x}_0 e^{\lambda_i t} + \sum_{i=1}^n \vec{c}^T \vec{r}_i \vec{\ell}_i^H \vec{b} \int_0^t e^{\lambda_i(t-\tau)} u(\tau) d\tau, \quad (27)$$

где $\{e^{\lambda_i t}, i=1, 2, \dots, n\}$ - элементарные составляющие движения;

$\{\lambda_i, i=1, 2, \dots, n\}$ - собственные значения матрицы A;

$\{\vec{r}_i, i=1, 2, \dots, n\}$ - система правых собственных векторов матрицы A;

$\{\vec{\ell}_i, i=1, 2, \dots, n\}$ - система левых собственных векторов матрицы A.

Представление вида (27) связано с приведением исходной матрицы A к диагональной форме

$$L^H A R = \Lambda, \quad (28)$$

где $\Lambda = \text{diag}(\lambda_i)$;

$$R = [\vec{r}_1 | \vec{r}_2 | \dots | \vec{r}_n];$$

$$L^H = [\vec{\ell}_1 | \vec{\ell}_2 | \dots | \vec{\ell}_n]^H;$$

или в координатной форме

$$A = \sum_{i=1}^n \lambda_i \vec{r}_i \vec{\ell}_i^H. \quad (29)$$

Последнее соотношение представляет собой спектральное разложение матрицы A. Оно показывает, что система собственных значений и векторов полностью определяет исходную матрицу. Та же система собственных векторов позволяет вычислить и любую аналитическую функцию от матрицы [2, 3]. В задачах теории управления используют следующие матричные функции:

$$\Phi = e^{A\tau} = \sum_{i=1}^n e^{\lambda_i \tau} \vec{r}_i \vec{\ell}_i^H, \quad (30a)$$

$$\vec{f} = (e^{A\tau} - I) A^{-1} \vec{b} = \sum_{i=1}^n \frac{e^{\lambda_i \tau} - 1}{\lambda_i} \vec{r}_i \vec{\ell}_i^H \vec{b}; \quad (30b)$$

$$\psi = \frac{2}{T} t h \frac{AT}{2} = \sum_{i=1}^n \frac{2}{T} t h \frac{\lambda_i T}{2} \vec{r}_i \vec{\ell}_i^H; \quad (30в)$$

$$\vec{\psi} = \frac{2}{T} t h \frac{AT}{2} A^{-1} \vec{b} = \sum_{i=1}^n \frac{t h \lambda_i T / 2}{\lambda_i T / 2} \vec{r}_i \vec{\ell}_i^H \vec{b}. \quad (30г)$$

Возможность использования соотношений (29, 30 а - 30г) связана со спектральным разложением исходной матрицы A, или, что то же самое, с решением полной проблемы собственных значений для матрицы A [13]. Выше рассмотрен случай действительной матрицы A с произвольным спектром собственных значений. Однако в некоторых задачах теории управления возникает потребность использовать комплексные матрицы, например, при анализе методом корневого годографа автоколебаний в нелинейных системах.

Поэтому ниже рассмотрены алгоритмы спектрального разложения произвольной комплексной матрицы.

Эффективные вычислительные процедуры решения полной проблемы собственных значений для произвольных комплексных матриц были построены сравнительно недавно [13]. Анализ различных методов показывает, что имеются две группы алгоритмов.

Первая группа исходит из метода Якоби и связана с его обобщениями на случай комплексных матриц посредством преобразования преобразований комплексного вращения и сдвига. Такое преобразование обеспечивает приведение исходной матрицы к нормальной форме, а за счет уменьшения нормы внедиагональных элементов - к диагональной форме. Соответствующий алгоритм реализован в виде процедуры COMEIG.

Вторая группа [13] требует предварительного приведения исходной матрицы к форме Хессенберга, затем используют алгоритмы для вычисления только собственных значений, или совместно собственных значений и векторов матрицы Хессенберга. В последнем случае для определения системы векторов исходной матрицы необходимы обратные преобразования.

Ниже подробно описана процедура *COMEIG* и ее реализация на алгоритмическом языке ПЛ/1.

Пусть A — комплексная квадратная матрица порядка n . В процедуре *COMEIG* формируется матрица $R=R_1 R_2 \dots R_k$ как произведение невырожденных комплексных матриц R_i таким образом, чтобы внедиагональные элементы результирующей матрицы $R^{-1}AR = \text{diag}(\lambda_i) + \Delta\Delta$ оказались произвольно малыми величинами. Вычислительный процесс является итерационным. На каждой итерации формируется очередная матрица преобразования R_i . По окончании итерационного процесса элементами диагональной матрицы являются собственные значения, а столбцы матрицы R определяют систему правых собственных векторов исходной матрицы.

Каждая матрица элементарного преобразования R_i образуется как произведение матриц $H_i S_i$, где матрица H_i определяет комплексное плоское вращение, а матрица S_i является матрицей комплексного сдвига.

Элементы матрицы комплексного вращения определяются следующими соотношениями:

$$\left. \begin{aligned} H: h_{kk} &= h_{mm} = \cos \alpha; \\ h_{km} &= -h_{mk}^* = -e^{j\alpha} \sin \alpha; h_{ij} = \delta_{ij}. \end{aligned} \right\} \quad (31a)$$

Элементы матрицы комплексного сдвига определяются так:

$$\left. \begin{aligned} S: s_{kk} &= s_{mm} = ch y; \\ s_{km} &= s_{mk}^* = -j e^{j\beta} sh y; s_{ij} = \delta_{ij}. \end{aligned} \right\} \quad (31b)$$

Пары индексов (k, m) и соответствующий им элемент преобразования a_{km} выбираются на каждом шаге итерационного процесса циклическим перебором.

Выбор параметров вращения α и β на каждом шаге подчиняется условиям [13]:

$$\left. \begin{aligned} \text{Re}(a_{km} - a_{mk}) \cos \alpha + \text{Im}(a_{km} + a_{mk}) \sin \alpha &= 0; \\ \text{Im}(a_{km} + a_{mk}) \cos \alpha &\geq 0; \\ \text{tg } 2\alpha &= |a_{km} - a_{mk}^*| / \text{Im}(a_{kk} - a_{mm}). \end{aligned} \right\} \quad (32)$$

Параметры комплексного сдвига γ и β выбирают так, чтобы уменьшать на каждом шаге евклидову норму приводимой матрицы, тем самым приближая ее к нормальной форме. Эти параметры должны удовлетворять соотношениям:

$$\left. \begin{aligned} \cos \beta &= -\text{Im}[(AA^H - A^H A)_{km}] / |(A^H A - AA^H)_{km}|; \\ \sin \beta &= \text{Re}[(AA^H - A^H A)_{km}] / |(A^H A - AA^H)_{km}|; \\ \text{th } \gamma &= |(A^H A - AA^H)_{km}| / (G + 2(|a_{kk} - a_{mm}|^2 - |c_{km} e^{-j\beta} + c_{mk} e^{j\beta}|^2)); \\ G &= \sum_{i \neq k, m} (|a_{ki}|^2 + |a_{ik}|^2 + |a_{mi}|^2 + |a_{im}|^2). \end{aligned} \right\} \quad (33)$$

Для сокращения числа умножений каждое очередное преобразование $R_i = H_i S_i$ выполняется одновременно.

Сформированная матрица преобразования $R_i = Q_i + j P_i$ проверяется на выполнение условия:

$$(q_{km}^2 + p_{km}^2)^{1/2} > 10^{-14} \sqrt{(p_{km}^2 + p_{mk}^2)^{1/2}} > 10^{-14}. \quad (34)$$

Если это условие выполнено, то соответствующее преобразование подобия матрицы A также будет выполнено; если условие (34) нарушено, то выбирают переход к новому элементу преобразования.

Затем осуществляют проверку сходимости. Если преобразованная матрица ближе к нормальной в пределах заданной точности, то процесс вычисления заканчивают. Максимальное число итераций в процедуре ограничено и составляет 35.

Схема алгоритма спектрального разложения матрицы приведена на рис. 4.

Применение процедур спектрального разложения (*COMEIG*, *COMEEG*, *COMAIG*, *COVEIG*). В пакете программ исследования динамических систем предусмотрено три разновидности процедуры спектрального разложения. Для вычисления собственных значений и векторов произвольной действительной матрицы предназначена процедура *COMEIG*, обращение к которой имеет вид

$$\text{CALL COMEIG}(N, A, R, L, Y). \quad (35)$$

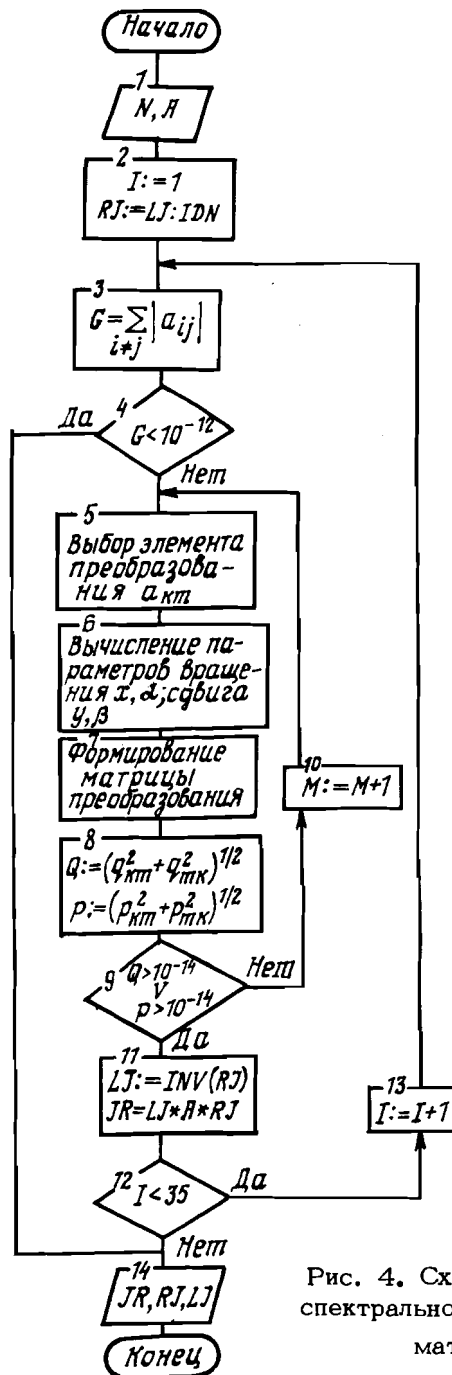


Рис. 4. Схема алгоритма спектрального разложения матрицы

Если процедуру спектрального разложения применяют для последовательности матриц с малыми, но конечными возмущениями значений элементов, то целесообразно использовать ранее вычисленные значения правых и левых собственных векторов. Это уменьшает число итераций при спектральном разложении возмущенной матрицы. Именно с этой целью разработана модификация процедуры (35), допускающая ее итеративное использование. Обращение к такой процедуре имеет вид:

$$\text{CALL COMEEG}(N, A, R, L, Y). \quad (36)$$

Для вычисления собственных значений комплексной матрицы (при этом собственные вектора не вычисляются) следует применять процедуру *COMAIG*, обращение к которой имеет вид:

$$\text{CALL COMAIG}(N, A, Y, S). \quad (37)$$

Для вычисления собственных значений и векторов произвольной комплексной матрицы предназначена процедура *COVEIG*, обращение к которой имеет вид:

$$\text{CALL COVEIG}(N, U, V, Q, P, D, F, Y, S). \quad (38)$$

Здесь приведен текст только одной процедуры *COMAIG*, имеющей наибольшее применение.

Входные параметры процедур спектрального разложения (*COMAIG*, *COMEEG*, *COMAIG*, *COVEIG*):

- N* – порядок исходной матрицы;
- A* – исходная комплексная матрица $U + jV$.

Выходные параметры процедур спектрального разложения (*COMAIG*, *COMEEG*, *COMAIG*, *COVEIG*):

- A* – диагональная форма исходной матрицы $U + jV = \text{diag}(\text{Re } \lambda_i) + j \text{diag}(\text{Im } \lambda_i)$;
- R* – матрица правых собственных векторов $Q + jP$;
- L* – матрица левых собственных векторов $D + jF$; из процедуры выводится матрица $L^H = D^H - jF^H$;
- Y* – вектор, указывающий порядок следования собственных значений. Если $Y(I) = 0$, то соответствующее собственное значение действительно; если $Y(I) = 1$, то собственное значение комплексно;
- S* – массив $S[2, N]$, который содержит собственные значения матрицы *A* в следующем порядке:

$$\begin{aligned} \text{Re } \lambda_1 &= S(1, 1), & \text{Im } \lambda_1 &= S(2, 1); \\ \text{Re } \lambda_2 &= S(1, 2), & \text{Im } \lambda_2 &= S(2, 2); \\ \text{Re } \lambda_n &= S(1, N), & \text{Im } \lambda_n &= S(2, N). \end{aligned}$$

Программа процедуры COMEIG :

```

COMEIG:
30 PROC(N,A,T,TT,Y):
DCL N,Y(*), (A,T,TT)(*,*) FLOAT(12); DCL (EN(N),Z(N,N),U(N,N),UU(N,N))
) FLOAT(12); DCL (I,J,K,M,IT) BIN FIXED, MARK BIT(1); DCL (EPS,TAU,
TEM,TEP,MAX,HJ,HR,HI,G,TE,TEE,BR,BI,ER,EI,DR,DI, ROOT,ROOT1,ROOT2,CA,
SA,CB,SB,CX,SX,SH,CH,ETA,TSE,ND,NC,S,C, COT2X,D,DE,SIG,COTX,COS2A,
SIN2A,TANH,C1R,C2R,S1R,S2R,C1I,C2I, S1I,S2I,ISW,B,EA,ZIM,ZIK,AKI,AIM,
TIM,TIK,UIM,UIK,AKI,AMI,ZKI, ZMI,TTKI,TTMI,UUKI,UUMI) DEC FLOAT(42);
MARK='0'B; EPS=1E-14; T,U,Z,TT,UU=0.0;
DO I=1 TO N;
T(I,I),TT(I,I)=1.0;
END;
BLOCK2:
DO IT=1 TO 35;
IF MARK THEN GO TO DONE; TAU=0.0;
DO K=1 TO N;
TEM=0.0;
DO I=1 TO N;
IF I/=K THEN TEM=ABS(A(I,K))+ABS(Z(I,K))+TEM;
END;
TAU=TAU+TEM; EN(K)=TEM+ABS(A(K,K))+ABS(Z(K,K));
END;
DO K=1 TO N-1;
MAX=EN(K); I=K;
DO J=1+K TO N;
IF EN(J)>MAX THEN
DO;
MAX=EN(J); I=J;
END;
END;
IF I/=K THEN
DO;
EN(I)=EN(K);
DO J=1 TO N;
TEP=A(K,J); A(K,J)=A(I,J); A(I,J)=TEP; TEP=Z(K,J);
Z(K,J)=Z(I,J); Z(I,J)=TEP;
END;
DO J=1 TO N;
TEP=A(J,K); A(J,K)=A(J,I); A(J,I)=TEP; TEP=Z(J,K);
Z(J,K)=Z(J,I); Z(J,I)=TEP; TEP=T(J,K); T(J,K)=TEP;
T(J,I)=TEP; TEP=U(J,K); U(J,K)=U(J,I); U(J,I)=TEP;
TEP=TT(K,J); TT(K,J)=TT(I,J); TT(I,J)=TEP; TEP=UU(K,J);
UU(K,J)=UU(I,J); UU(I,J)=TEP;
END;
END;
IF TAU<100.0*EPS THEN GO TO DONE; MARK='1'B;
DO K=1 TO N-1;
DO M=K+1 TO N;
HJ,HR,HI,G=0.0;
DO I=1 TO N;
IF I/=K & I/=M THEN
DO;
HR=HR+A(K,I)*A(M,I)+Z(K,I)*Z(M,I); HR=HR-A(I,K)*A(I,M)-Z(I,
K)*Z(I,M); HI=HI+Z(K,I)*A(M,I)-A(K,I)*Z(M,I)-A(I,K)*Z(I,
M)+Z(I,K)*A(I,M); TE=A(I,K)*A(I,K)+Z(I,K)*Z(I,K)+A(M,I)*A(
M,I)+Z(M,I)*Z(M,I); TEE=A(I,M)*A(I,M)+Z(I,M)*Z(I,M)+A(K,I)
*A(K,I)+Z(K,I)*Z(K,I); G=G+TE+TEE; HJ=HJ-TE+TEE;
END;
END;
BR=A(K,M)+A(M,K); BI=Z(K,M)+Z(M,K); ER=A(K,M)-A(M,K);
EI=Z(K,M)-Z(M,K); DR=A(K,K)-A(M,M); DI=Z(K,K)-Z(M,M);
TE=BR*BR+EI*EI+DR*DR; TEE=BI*BI+ER*ER+DI*DI; IF TE>=TEE THEN
DO;
ISW=1.0; C=BR; S=EI; D=DR; DE=DI; ROOT2=SQRT(TE);
END;
ELSE
DO;
ISW=-1.0; C=BI; S=-ER; D=DI; DE=DR; ROOT2=SQRT(TEE);
END;
ROOT1=SQRT(S*S+C*C); IF D<0.0 THEN SIG=-1.0; ELSE SIG=1.0;
SA=0.0; IF C>0.0 THEN CA=1.0; ELSE CA=-1.0;
IF ROOT1<EPS THEN
DO;
SX,SA=0.0; CX,CA=1.0; IF ISW<0.0 THEN EA=EI; ELSE EA=ER;
IF ISW>0.0 THEN B=BI; ELSE B=-BR; ND=D*D+DE*DE;
GO TO ENTER1;
END;

```

```

IF ABS(S)>EPS THEN
DO;
CA=C/ROOT1; SA=S/ROOT1;
END;
COT2X=D/ROOT1; IF ABS(COT2X)>1E9 THEN COTX=ABS(COT2X);
ELSE COTX=SQRT(1.0+COT2X+COT2X); COTX=COT2X+SIG*COTX;
IF ABS(COTX)>1E9 THEN SX=ABS(COTX); ELSE SX=SQRT(1.0+COTX*
COTX); SX=SIG/SX; CX=SX*COTX; ETA=(ER*BR+BI*EI)/ROOT1;
TSE=(BR*BI-ER*EI)/ROOT1; TE=SIG*(-ROOT1*DE+TSE*D)/ROOT2;
TEE=(D*DE+ROOT1*TSE)/ROOT2; ND=ROOT2*ROOT2+TEE+TEE;
TEE=HI*CX+SX; COS2A=CA*CA-SA*SA; SIN2A=2.0*CA*SA;
TEM=HR*COS2A+HI*SIN2A; TEP=HI*COS2A-HR*SIN2A; HR=CX*CX+HR-SX*
SX*TEM-CA*TEE; HI=CX*CX+HI+SX*SX*TEP-SA*TEE; B=ISW*TE*CA+ETA*
SA; EA=CA*ETA-ISW*TE*SA;

```

ENTER1:

```

S=HR-SIG*ROOT2+EA; C=HI-SIG*ROOT2+B; ROOT=SQRT(C+C+S*S);
IF ROOT<EPS THEN
DO;
CH,CH=1.0; SB,SH=0.0; GO TO TRANS;
END;
CB=-C/ROOT; SB=S/ROOT; TEE=CB*B-EA*SB; NC=TEE*TEE;
TANH=ROOT/(G+2.0*(NC+ND)); CH=1.0/SQRT(1.0-TANH*TANH);
SH=CH*TANH;

```

TRANS:

```

TEM=SX*SH+(SA*CB-SB*CA); C1R=CX*CH-TEM; C2R=CX*CH+TEM;
C1I,C2I=-SX*SH*(CA*CB+SA*SB); TEP=SX*CH+CA; TEM=CX*SM*SB;
S1R=TEP-TEM; S2R=-TEP-TEM; TEP=SX*CH*SA; TEM=CX*SH*CB;
S1I=TEP+TEM; S2I=TEP-TEM; TEM=SQRT(S1R*S1R+S1I*S1I);
TEP=SQRT(S2R*S2R+S2I*S2I); IF TEM>EPS ! TEP>EPS THEN
DO;
MARK='0'B;
DO I=1 TO N;
AKI=A(K,I); AMI=A(M,I); ZKI=Z(K,I); ZMI=Z(M,I);
A(K,I)=C1R*AKI-C1I*ZKI+S1R*AMI-S1I*ZMI; Z(K,I)=C1R*ZKI+
C1I*AKI+S1R*ZMI+S1I*AMI; A(M,I)=S2R*AKI-S2I*ZKI+C2R*AMI-
C2I*ZMI; Z(M,I)=S2R*ZKI+S2I*AKI+C2R*ZMI+C2I*AMI;
TTKI=TT(K,I); TTMI=TT(M,I); UUKI=UU(K,I); UUMI=UU(M,I);
TT(K,I)=C1R*TTKI-C1I*UUKI+S1R*TTMI-S1I*UUMI; UU(K,I)=C1R*

```

3-2544

```

UUKI+C1I*TTKI+S1R*UUMI+S1I*TTMI; TT(M,I)=S2R*TTKI-S2I*
UUKI+C2R*TTMI-C2I*UUMI; UU(M,I)=S2R*UUKI+S2I*TTKI+C2R*
UUMI+C2I*TTMI;
END;
DO I=1 TO N;
AIK=A(I,K); AIM=A(I,M); ZIK=Z(I,K); ZIM=Z(I,M);
A(I,K)=C2R*AIK-C2I*ZIK-S2R*AIM+S2I*ZIM; Z(I,K)=C2R*ZIK+
C2I*AIK-S2R*ZIM-S2I*AIM; A(I,M)=-S1R*AIK+S1I*ZIK+C1R*AIM-
C1I*ZIM; Z(I,M)=-S1R*ZIK-S1I*AIK+C1R*ZIM+C1I*AIM;
TIK=T(I,K); TIM=T(I,M); UIK=U(I,K); UIM=U(I,M);
T(I,K)=C2R*TIK-C2I*UIK-S2R*TIM+S2I*UIM; U(I,K)=C2R*UIK+
C2I*TIK-S2R*UIM-S2I*TIM; T(I,M)=-S1R*TIK+S1I*UIK+C1R*TIM-
C1I*UIM; U(I,M)=-S1R*UIK-S1I*TIK+C1R*UIM+C1I*TIM;
END;

```

END BLOCK2;

DONE:

```

Y(N)=0;
DO I=1 TO N-1;
IF ABS(1-A(I,I)/A(I+1,I+1))>1E-3 THEN Y(I)=0; ELSE
DO;
Y(I),Y(I+1)=1.0; A(I,I+1)=Z(I+1,I+1); A(I+1,I)=Z(I,I);
DO K=1 TO N;
TT(I+1,K)=TT(I,K)+UU(I,K); TT(I,K)=TT(I,K)-UU(I,K);
T(K,I+1)=T(K,I)-U(K,I); T(K,I)=T(K,I)+U(K,I);
END;
I=I+1;
END;

```

END;

DO J=1 TO N;

```

TE=0.0;
DO I=1 TO N;
TE=TE+T(I,J)*TT(J,I);
END;
DO I=1 TO N;
T(I,J)=T(I,J)/TE;
END;

```

END;

RETURN;

END COMEIG;

3.1. Точность спектрального разложения.
Оценка спектрального числа обусловленности

Точность спектрального разложения существенно зависит от погрешности задания исходной матрицы. В качестве меры точности при вычислении собственных значений используют спектральное число обусловленности [9, 12], которое позволяет следующим образом оценить погрешность вычисления:

$$|\Delta\lambda| \leq k \|\Delta A\|_2, \quad (39)$$

где k - спектральное число обусловленности;
 ΔA - матрица погрешностей, учитывающая точность задания элементов исходной матрицы A .

Для оценки спектрального числа обусловленности k удобно использовать следующее неравенство:

$$\max_i \|\vec{r}_i\|_2 \|\vec{e}_i\|_2 \leq k \leq \sum_{i=1}^n \|\vec{r}_i\|_2 \|\vec{e}_i\|_2. \quad (40)$$

Вычисление верхнего и нижнего пределов этой оценки по результатам выполнения процедуры *COMEIG* не вызывает трудностей.

Соответствующие вычисления оформлены в виде процедуры КН.

Схема алгоритма КН (рис. 5) непосредственно вытекает из структуры неравенства (40).

Применение процедуры КН. Обращение к процедуре КН имеет вид:

$$CALL\ KN(N, RJ, LJ, Y, MIN, MAX). \quad (41)$$

Процедуру КН можно применять только после выполнения процедуры *COMEIG*.

Входные параметры процедуры КН:

N - порядок матрицы;

RJ - матрица правых собственных векторов в жордановой форме, полученная с помощью алгоритма *COMEIG*;

LJ - матрица левых собственных векторов в жордановой форме;

Y - вектор, указывающий порядок следования собственных значений.

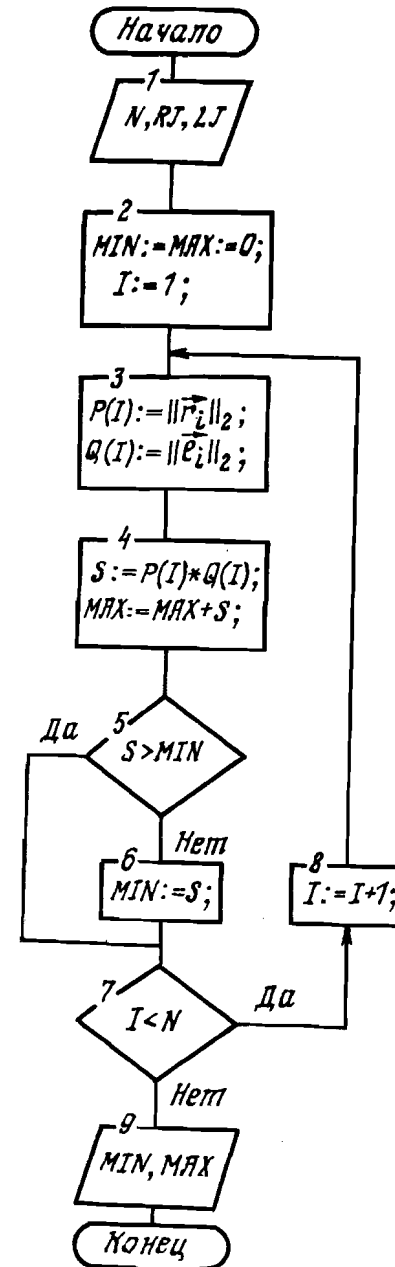


Рис. 5. Схема алгоритма КН

Выходные параметры процедуры КН:

MIN - нижняя граница спектрального числа обусловлен-

ности;

MAX - верхняя граница спектрального числа обусловлен-

ности.

Программа процедуры КН:

```

КН:
PROC(N,R,L,Y,MIN,MAX);
DCL N,I,J; DCL ((R,L) (*,*), MIN, MAX, R1, L1, S) DEC(12), Y(*);
MIN, MAX=0;
DO I=1 TO N;
  R1=L1=0;
  DO J=1 TO N;
    IF Y(I)<0.5 THEN
      DO; R1=R1+R(J,I)**2; L1=L1+L(I,J)**2;
      END;
    ELSE
      DO; R1=R1+((R(J,I)+R(J+1,I))/2)**2+((R(J,I)-R(J+1,I))/2)**2;
      L1=L1+((L(I,J)+L(I,J+1))/2)**2+((L(I,J)-L(I,J+1))/2)**2;
      END;
    S=SQRT(R1*L1); IF MIN<S THEN MIN=S; IF Y(I)>0.5 THEN
      DO; S=S*2; I=I+1;
      END;
    MAX=MAX+S;
  END J;
END I;
END КН;

```

3.2. Вычисление функций от матриц

В задачах исследования динамических систем необходимо вычисление следующих функций от матриц:

$$\Phi = e^{At}, \quad F = (\Phi - I)A^{-1}; \quad (42a)$$

$$\Psi = \frac{2}{T} t h \frac{AT}{2}, \quad B = \frac{2}{T} t h \frac{AT}{2} A^{-1}; \quad (42b)$$

$$\psi = \frac{2}{T} (\Phi + I)^{-1} (\Phi - I), \quad B = \frac{2}{T} (\Phi + I)^{-1}. \quad (42в)$$

Все перечисленные матричные функции являются аналитическими и могут быть вычислены на основе спектрального разложения исходной матрицы согласно следующему соотношению:

$$f(A) = \sum_{i=1}^n f(\lambda_i) \vec{r}_i \vec{p}_i^H, \quad (43)$$

где $f(A)$ - любая из функций (42a - 42в).

Соответствующие вычисления матричных функций оформлены в виде процедуры MFUNC.

Схема алгоритма MFUNC представлена на рис. 6. Все операции, показанные на схеме, выполняются над матрицами.

Применение процедуры MFUNC. Процедура MFUNC позволяет подготовить информацию для вычисления передаточных функций, корневых годографов и частотных характеристик при описании систем как функций от различных комплексных аргументов: для непрерывных систем на плоскости S , дискретных систем на плоскостях Z и \bar{Z} . Здесь переменные Z и \bar{Z} связаны между собой и с переменной S соотношениями:

$$Z = e^{sT}; \quad \bar{Z} = \frac{2}{T} t h \frac{sT}{2}; \quad \bar{S} = \frac{2}{T} \frac{Z-1}{Z+1}. \quad (44)$$

Обращение к процедуре MFUNC имеет вид:

$$CALL MFUNC(N, T, Y, JR, RJ, LJ, C, D, P, Q). \quad (45)$$

Процедуру MFUNC можно использовать только после выполнения процедуры COMEIG.

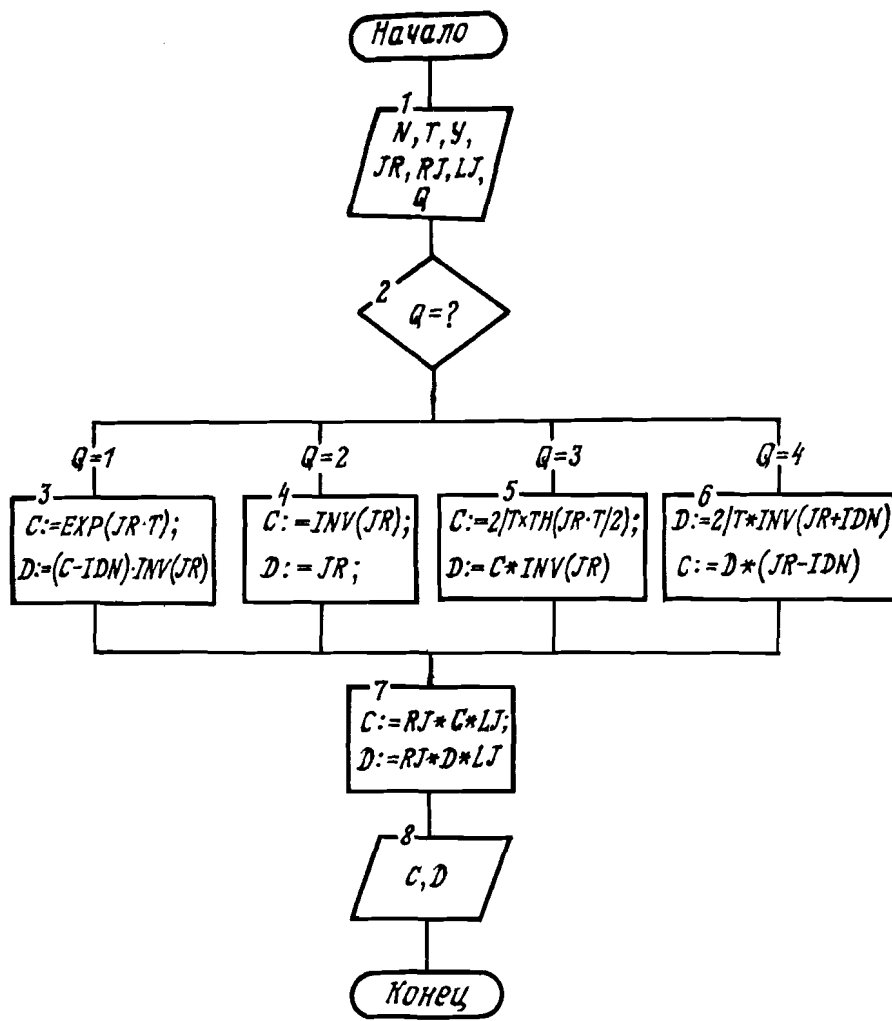


Рис. 6. Схема алгоритма MFUNC

Входные параметры процедуры MFUNC:

- N - порядок матрицы;
- T - такт дискретности;
- Y - вектор, указывающий порядок следования собственных значений;
- JR - матрица системы в жордановой форме;
- RJ, LJ - матрицы правых и левых собственных векторов при приведении к жордановой форме;
- Q - параметр, определяющий вид вычисляемых функций и назначаемый пользователем в соответствии с табл. 1.

Выходные параметры процедуры MFUNC:

- C, D - матрицы, вид которых определяется параметром Q (табл. 1);
- P - массив $P(2, N)$, содержащий собственные значения матрицы C .

Таблица 1
Функции, вычисляемые в процедуре MFUNC

Признак Q	Вычисляемые функции	
	C	D
1	e^{AT}	$(e^{AT} - I) A^{-1}$
2	A^{-1}	A
3	$\frac{2}{T} \operatorname{th} \frac{AT}{2}$	$\frac{2}{T} \operatorname{th} \frac{AT}{2} A^{-1}$
4	$\frac{2}{T} (\Phi + I)^{-1} (\Phi - I)$	$\frac{2}{T} (\Phi + I)^{-1}$

40 Программа процедуры MFUNC:

```

MFUNC:
PROC(N,T,Y,A,R,L,C,D,P,Q);
C,D=0; P=0; DCL(N,I,J,K,Q) FIXED BIN,Y(*),(T,(A,R,L,C,D,P)(*,*),
X,G,H,LAM,OME,SI,CO,REC,IMC,RED,IMD) DEC(12),(M(9),Z(9)) LABEL;
IF Q>=4 THEN Q=4; IF Q<=1 THEN Q=1;
DO I=1 TO N;
LAM=A(I,I); OME=A(I+1,I); H=EXP(A(I,I)*T); IF Y(I)<0.5 THEN
DO;
M(1): IMC,IMD=0.0; GOTO M(Q);
M(2): REC=H; RED=(REC-1.0)/LAM; GOTO OUT;
M(3): REC=1.0/LAM; RED=LAM; GOTO OUT;
M(4): REC=2.0/T*(H-1.0)/(H+1.0); RED=REC/LAM; GOTO OUT;
OUT: RED=2.0/T/(LAM+1.0); REC=RED*(LAM-1.0);
;
END;
ELSE
DO;
G=LAM*LAM+OME*OME; CO=COS(OME*T); SI=SIN(OME*T); GOTO Z(Q);
Z(1): REC=H*CO; IMC=H*SI; RED=((H*CO-1.0)*LAM+H*OME*SI)/G;
IMD=(H*SI+LAM-OME*(H*CO-1.0))/G; GOTO OUT1;
Z(2): REC=LAM/G; IMC=-OME/G; RED=LAM; IMD=OME; GOTO OUT1;
Z(3): X=(H*H+1.0+2.0*H*CO)*T/2.0; REC=(H*H-1.0)/X; IMC=2.0*H*SI/X;
RED=(REC+LAM+IMC*OME)/G; IMD=(IMC*LAM-REC*OME)/G; GOTO OUT1;
Z(4): X=((LAM+1.0)**2+OME*OME)*T/2.0; REC=(LAM*LAM-1.0+OME*OME)/X;
IMC=2.0*OME/X; RED=(LAM+1.0)/X; IMD=-OME/X;
OUT1: ; I=I+1;
END;
DO J=1 TO N;
DO K=1 TO N;
H=L(I,K)*R(J,I); G=0.0; IF Y(I)>0.5 THEN
DO;
H=H+R(J,I-1)*L(I-1,K); G=R(J,I)*L(I-1,K)-R(J,I-1)*L(I,K);
END;
C(J,K)=C(J,K)+REC*H+IMC*G; D(J,K)=D(J,K)+RED*H+IMD*G;
END;
END;
P(1,I)=REC; P(2,I)=IMC; IF Y(I)>0.5 THEN
DO;
P(1,I-1)=REC; P(2,I-1)=-IMC;
END;
END;
END MFUNC;

```

4. СОЕДИНЕНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

Существуют следующие виды соединений динамических систем между собой: последовательное, параллельное и соединение с обратной связью (ОС) [9].

Любая линейная динамическая система может быть описана следующими четверками: непрерывная - $\{A, \vec{b}, \vec{c}^T, d\}$; дискретная - $\{\Phi, \vec{f}, \vec{c}^T, d\}$. Нетрудно видеть, что с точностью до обозначений описание дискретных и непрерывных моделей совпадает; поэтому в дальнейшем, используя четверку для описания непрерывной модели, будем иметь в виду, что выводы справедливы и для дискретной модели с заменой A на Φ и \vec{b} на \vec{f} .

Пусть подсистема S_1 порядка n_1 задана четверкой $\{A^{(1)}, \vec{b}^{(1)}, \vec{c}^{(1)T}, d^{(1)}\}$, а подсистема S_2 порядка n_2 - четверкой $\{A^{(2)}, \vec{b}^{(2)}, \vec{c}^{(2)T}, d^{(2)}\}$. Тогда результирующая система S , являющаяся соединением подсистем S_1 и S_2 , будет описана различными четверками вида $\{A, \vec{b}, \vec{c}^T, d\}$ в зависимости от способа соединения (рис. 7, а - в):

при последовательном соединении (см. рис. 7, а)

$$A = \begin{bmatrix} A^{(1)} & \vec{b}^{(1)} \vec{c}^{(2)T} \\ 0 & A^{(2)} \end{bmatrix}; \quad \vec{b} = \begin{bmatrix} \vec{b}^{(1)} \\ \vec{b}^{(2)} \end{bmatrix}; \quad \vec{c}^T = [\vec{c}^{(1)T} \quad d^{(1)} \vec{c}^{(2)T}]; \quad d = d^{(1)} d^{(2)}; \quad (46a)$$

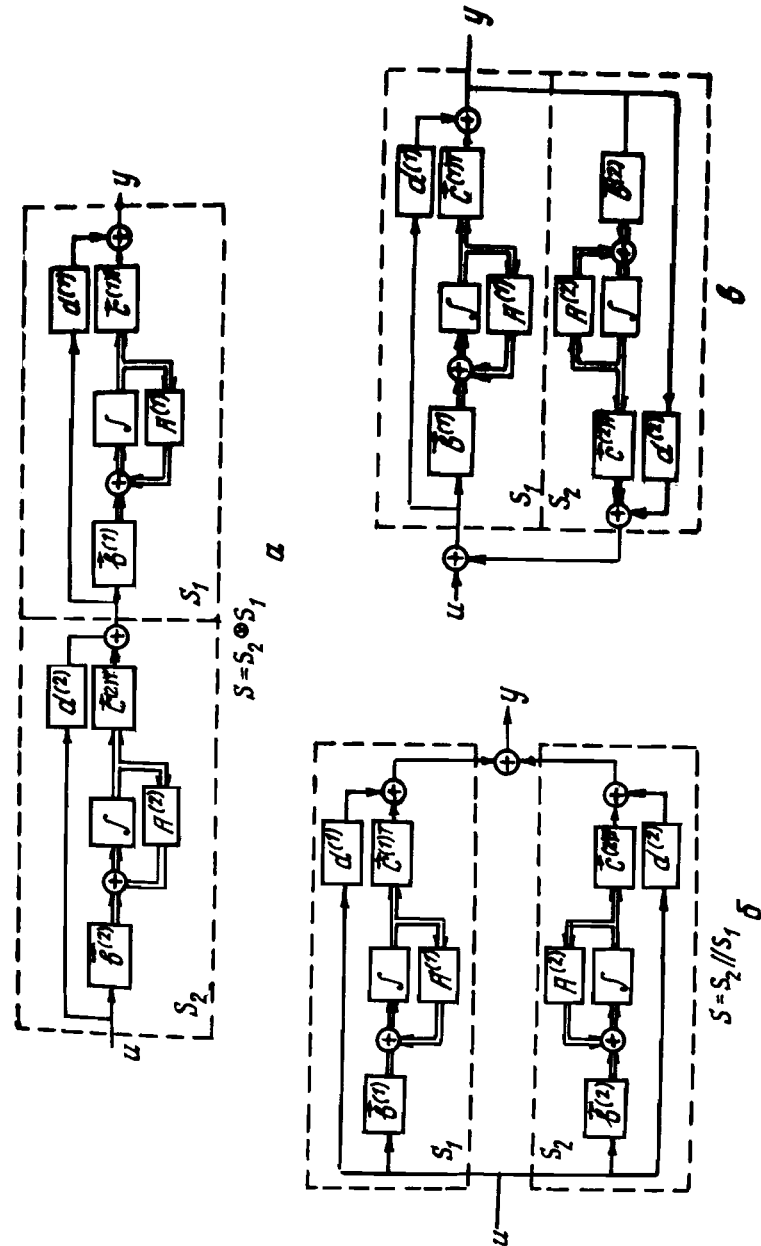


Рис. 7. Способы соединения динамических систем: а - последовательное; б - параллельное; в - с ОС

при параллельном соединении (см. рис. 7,б)

$$A = \left[\begin{array}{c|c} A^{(1)} & 0 \\ \hline 0 & A^{(2)} \end{array} \right]; \quad \vec{b} = \left[\begin{array}{c} \vec{b}^{(1)} \\ \vec{b}^{(2)} \end{array} \right]; \quad (46a)$$

$$\vec{c}^T = [\vec{c}^{(1)T}; \vec{c}^{(2)T}]; \quad d = d^{(1)} + d^{(2)};$$

при соединении подсистем S_1 и S_2 с ОС (см. рис. 7,в):

$$A = \left[\begin{array}{c|c} A^{(1)} \pm q d^{(2)} [\vec{b}^{(1)} \vec{c}^{(1)T}] & \pm q [\vec{b}^{(1)} \vec{c}^{(2)T}] \\ \hline q [\vec{b}^{(2)} \vec{c}^{(2)T}] & A^{(2)} \pm q d^{(1)} [\vec{b}^{(2)} \vec{c}^{(2)T}] \end{array} \right]; \quad \vec{b} = \left[\begin{array}{c} q \vec{b}^{(1)} \\ q d, \vec{b}^{(2)} \end{array} \right]; \quad (46b)$$

$$\vec{c}^T = [q \vec{c}^{(1)T}; \pm q d^{(1)} \vec{c}^{(2)T}]; \quad d = q d^{(1)},$$

где $q = 1/(1 \mp d^{(1)} d^{(2)})$ и верхний знак соответствуют положительной, а нижний - отрицательной ОС.

Вычисление и формирование четверок результирующей системы оформлено в виде процедуры *JOINT*.

Схема алгоритма *JOINT* представлена на рис. 8.

Все операции, указанные на схеме, выполняются над матрицами.

Применение процедуры *JOINT*. Обращение к процедуре *JOINT* имеет вид:

$$\text{CALL JOINT}(N1, A1, B1, C1, D1, N2, A2, B2, C2, D2, \\ AO, BO, CO, DO, P). \quad (47)$$

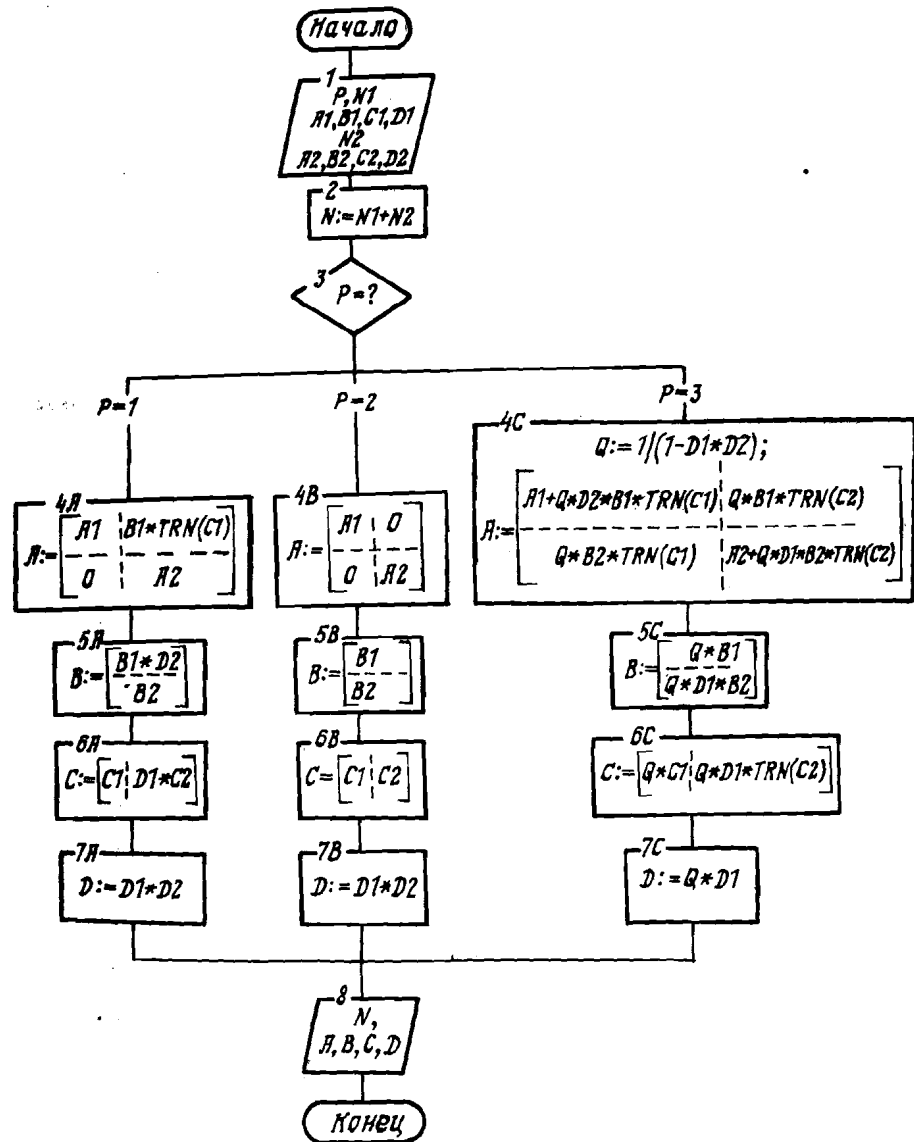


Рис. 8. Схема алгоритма *JOINT*

Входные параметры процедуры JOINT :

N_1 - порядок подсистемы S_1 ;
 $A_1(N_1, N_1), B_1(N_1), C_1(N_1), D_1$ - четверка S_1 ;
 N_2 - порядок подсистемы S_2 ;
 $A_2(N_2, N_2), B_2(N_2), C_2(N_2), D_2$ - четверка S_2 ;

P - признак соединения: последовательное ($P=1$), параллельное ($P=2$), с ОС ($P=3$).

Выходные параметры процедуры JOINT :

$A_0(N, N), B_0(N), C_0(N), D_0$ - четверка результирующей системы S порядка $N=N_1+N_2$.

Программа процедуры JOINT:

```
JOINT:
PROC(A1, B1, C1, D1, N1, A2, B2, C2, D2, N2, A0, B0, C0, D0, N0, P);
DCL A1(*, *), B1(*), C1(*), A2(*, *), B2(*, *), C2(*, *), DCL A0(*, *), B0(*, *), C0(*, *), D0(*, *);
N0=N1+N2; IF P>0 THEN
DO;
D7=D1*D2; IF D7=-1 THEN GO TO M; Q=1/(1+D7);
DO I=1 TO N1;
B0(I)=Q*B1(I);
DO J=1 TO N1;
A0(I, J)=A1(I, J)-Q*D2*B1(I)*C1(J);
END;
DO J=N1+1 TO N0;
A0(I, J)=-Q*B1(I)*C2(J-N1);
END;
END;
DO I=N1+1 TO N0;
B0(I)=Q*D1*B2(I-N1);
DO J=1 TO N1;
A0(I, J)=Q*B2(I-N1)*C1(J);
END;
DO J=N1+1 TO N0;
A0(I, J)=A2(I-N1, J-N1)-Q*D1*B2(I-N1)*C2(J-N1);
END;
IF P>1 THEN
DO;
DO I=1 TO N1;
C0(I)=Q+C1(I);
END;
DO I=N1+1 TO N0;
C0(I)=-Q*D1*C2(I-N1);
END;
DO=Q*D1;
END;
ELSE
DO;
```

```

DO I=1 TO N1;
  CO(I)=P+D2*C1(I);
END;
DO I=N1+1 TO NO;
  CO(I)=Q+C2(I-N1);
END;
DO D=D1+D2;
END;
ELSE
DO;
IF P>-1 THEN
DO;
DO I=1 TO N1;
  BO(I)=B1(I); CO(I)=C1(I);
DO J=1 TO N1;
  AO(I,J)=A1(I,J);
END;
END;
DO I=N1+1 TO NO;
  BO(I)=B2(I-N1); CO(I)=C2(I-N1);
DO J=N1+1 TO NO;
  AO(I,J)=A2(I-N1,J-N1);
END;
END;
DO D=D1+D2;
END;
ELSE
DO;
DO I=1 TO N1;
  BO(I)=B1(I); CO(I)=C1(I)+D2;
DO J=1 TO N1;
  AO(I,J)=A1(I,J);
END;
END;
DO I=N1+1 TO NO;
  BO(I)=B2(I-N1)+D1; CO(I)=C2(I-N1);
DO J=1 TO N1;
  AO(I,J)=B2(I-N1)*C1(J);
END;
DO J=N1+1 TO NO;
  AO(I,J)=A2(I-N1,J-N1);
END;
END;
DO D=D1+D2;
END;
M;
END JOINT;

```

5. ПЕРЕДАТОЧНЫЕ ФУНКЦИИ

При исследовании динамических систем важную роль играет понятие передаточной функции. Передаточная функция для непрерывной модели (10) определяется как отношение преобразований Лапласа выходного и входного сигналов и может быть описана соотношением:

$$W(s) = \vec{c}^T (sI - A)^{-1} \vec{b}. \quad (48)$$

Передаточная функция для дискретной модели определяется следующим образом:

$$W(z) = \vec{c}^T (zI - \Phi)^{-1} \vec{f}, \quad (49)$$

или при замене переменной z на $\bar{s} = \frac{z-1}{T}$,

$$W(\bar{s}) = \vec{c}^T (\bar{s}I - \Psi)^{-1} \vec{\psi} (1 - \bar{s}T/2), \quad (50)$$

где $\Psi = \frac{2}{T}(\Phi + I)^{-1}(\Phi - I)$; $\vec{\psi} = \frac{2}{T}(\Phi + I)^{-1} \vec{f}$.

Выражения (48 - 50) представляют дробно-рациональную функцию вида:

$$W(v) = \frac{\beta_1 v^{n-1} + \beta_2 v^{n-2} + \dots + \beta_n}{v^n + \alpha_1 v^{n-1} + \dots + \alpha_n}, \quad (51)$$

где формальный параметр v принимает значения s , z или \bar{s} . Их трудно использовать для вычисления передаточных функций, поскольку в них входят нечисловые параметры s , z и \bar{s} . Существенно другое: все эти выражения имеют одинаковую структуру, определяемую соответствующей векторно-матричной тройкой (\vec{c}^T, A, \vec{b}) , $(\vec{c}^T, \Phi, \vec{f})$ или $(\vec{c}^T, \Psi, \vec{\psi})$. Это означает, что можно построить алгоритм, одинаково пригодный для вычисления любой из указанных передаточных функций.

Наиболее просто это выполнить, используя спектральное разложение матриц A , Φ или Ψ , с последующим вычислением коэффициентов знаменателя α_i по формулам:

$$\left. \begin{aligned} \alpha_1 &= -(\lambda_1 + \lambda_2 + \dots + \lambda_n); \\ \alpha_2 &= \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \dots + \lambda_n \lambda_{n-1}; \\ \alpha_n &= (-i)^n \lambda_1 \lambda_2 \dots \lambda_n. \end{aligned} \right\} (52)$$

Тогда для вычисления коэффициентов числителя β_i справедливы следующие формулы [9]:

$$\left. \begin{aligned} \beta_1 &= \vec{c}^T \vec{b}; \\ \beta_2 &= \vec{c}^T A \vec{b} + \alpha_1 \vec{c}^T \vec{b}; \\ &\dots \\ \beta_n &= \vec{c}^T A^{n-1} \vec{b} + \alpha_1 \vec{c}^T A^{n-2} \vec{b} + \dots + \alpha_{n-1} \vec{c}^T \vec{b}. \end{aligned} \right\} (53)$$

Схема алгоритма TRFUNC вычисления коэффициентов числителя и знаменателя передаточной функции показана на рис. 9.

Расчет коэффициентов знаменателя выполняют, исходя из вычисленных в процедуре COMEIG собственных значений матрицы. Эта часть алгоритма построена следующим образом. В качестве исходного берется полином нулевого порядка P_0 . Затем проверяется на комплексность первое собственное значение. Если оно действительное, то степень полинома увеличивается на единицу. В случае, если собственное значение комплексное, степень полинома увеличивается на два. Таким образом, рекуррентные соотношения для полиномов $P_{i+1}(s)$ и $P_i(s)$ имеют вид:

$$\left. \begin{aligned} P_{i+1}(s) &= P_i(s)(s - \lambda_i), & \text{если } \text{Im } \lambda_i = 0; \\ P_{i+1}(s) &= P_i(s)[s^2 - 2\text{Re } \lambda_i s + (\text{Re } \lambda_i)^2 + (\text{Im } \lambda_i)^2], & \text{если } \text{Im } \lambda_i \neq 0. \end{aligned} \right\} (54)$$

Вычисление коэффициентов числителя выполняется в соответствии с формулами (53).

Применение процедуры TRFUNC. Обращение к процедуре TRFUNC имеет вид:

CALL TRFUNC (N, P, A, B, C, T, BE, AL).

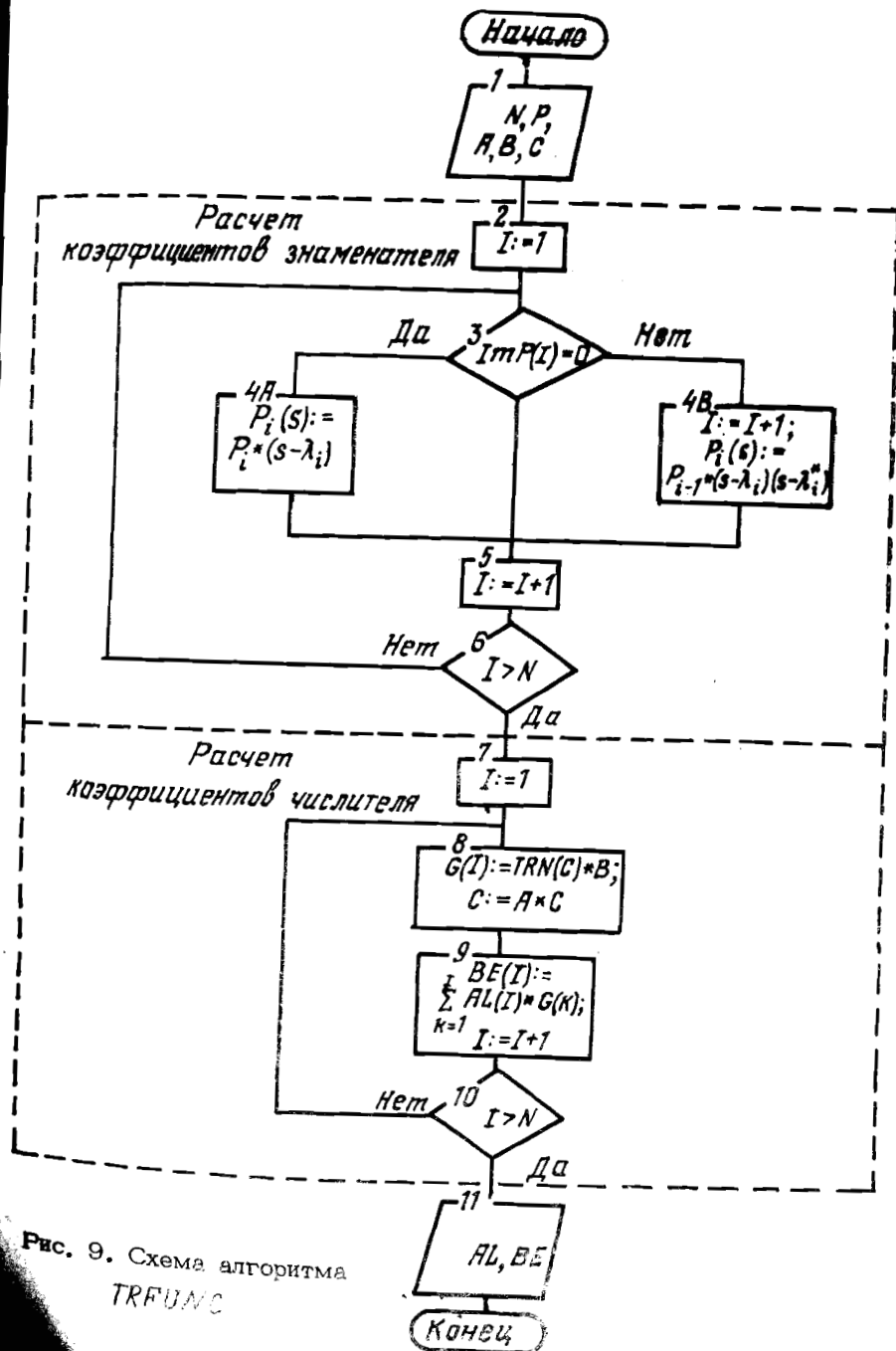


Рис. 9. Схема алгоритма TRFUNC

Для практического анализа систем важное значение имеют нули и полюса передаточной функции. Поэтому вычисление коэффициентов передаточной функции необходимо дополнить вычислением нулей многочлена числителя. При этом следует иметь в виду, что часть из них при старших степенях могут быть нулевыми, т.е. порядок числителя может оказаться меньше $n-1$. Для отыскания нулей многочлена удобно воспользоваться методом [12] и алгоритмом Берстоу [19].

Этот алгоритм позволяет отыскивать корни алгебраического уравнения порядка N .

Входные параметры процедуры TRFUNC :

N - порядок системы;

P - массив собственных значений матрицы, сформированный для матриц Φ и Ψ в процедуре MFUNC.

Для матрицы A массив P необходимо сформировать в управляющей программе самостоятельно или обращением к процедуре MFUNC с параметрами

$A = INV(JORD)$, $Q = 2$, где $JORD$ - матрицы системы в жордановой форме;

A - матрица системы вида A , Φ или Ψ ;

B - вектор передачи управления вида \vec{b} , \vec{f} или \vec{c} ;

CC - вектор формирования выхода.

Выходные параметры процедуры TRFUNC :

BE - коэффициенты числителя $[\beta_1, \beta_2, \dots, \beta_m]$ размером N ;

AL - коэффициенты знаменателя $[\alpha_1, \alpha_2, \dots, \alpha_n]$ размером N .

Программа процедуры TRFUNC :

```

TRFUNC:
PROCEDURE(N,P,F,FE,C,BETA,ALPHA);
DCL N,((P,F))(*,*),((FB,C,BETA,ALPHA))(*), DEC(12); DCL I,J,K,CC,CC1,
PP,Y(N),POL(C:N+4) DEC(12); POL=0; BETA=0; PP=0; POL(2)=1;
DO I=1 TO N;
  C1(I)=C(I);
END;
DO I=1 TO N;
  IF ASS(P(2,I))<1E-9 THEN
    DO J=I+2 BY -1 TO 2;
      POL(J)=POL(J-1)-POL(J)*P(1,I);
    END;
  ELSE
    DO J=I+3 BY -1 TO 2;
      POL(J)=POL(J)*P(1,I)+P(2,I)-POL(J-1)*P(1,I)*2+
        POL(J-2);
    END;
  I=I+1;
END;
DO ALPHA(I)=POL(N-I+2);
DO J=1 TO N;
  PP(I)=PP(I)+C1(J)*FB(J); CC=0.0;
  DO K=1 TO N;
    CC=CC+C(K)*F(K,J);
  END;
  C1(J)=CC;
END;
DO K=1 TO N;
  CC(K)=C1(K); IF K>=N-I+1 THEN BETA(I)=BETA(I)+POL(K+2)*PP(K+I-N);
END;
RETURN;
END TRFUNC;

```

Применение процедуры BERSTOW. Обращение к процедуре BERSTOW имеет вид:

CALL BERSTOW(N,H,U). (56)

Входные параметры процедуры BERSTOW:

N - порядок уравнения;

H - массив размером $1 \times (N+1)$ коэффициентов уравнения

$$A(N)x^N + A(N-1)x^{N-1} + \dots + A(0).$$

Выходные параметры процедуры BERSTOW:

U - массив корней алгебраического уравнения размером $2 \times N$, размещенных следующим образом:

1-й корень: $U(1,1) = \text{Re } \lambda_1$, $U(1,2) = \text{Im } \lambda_1$;

2-й корень: $U(2,1) = \text{Re } \lambda_2$, $U(2,2) = \text{Im } \lambda_2$;

N-й корень: $U(N,1) = \text{Re } \lambda_N$, $U(N,2) = \text{Im } \lambda_N$.

Программа процедуры BERSTOW:

```

BERSTOW:
PROC(N,H,U,V);
I4=-1; I2=I4; N1=N+1; P,O,R=0;
L: IF H(1)≠0 THEN GOTO L50; N1=N1-1; V(N1),U(N1)=0;
DO I=1 TO N1;
H(I)=H(I+1);
END;
GOTO L;
L50: IF N1=1 THEN GOTO L199; IF N1≠2 THEN GOTO L55; R=-H(1)/H(2);
GOTO L150;
L55: IF N1=3 THEN GOTO L60; P=H(2)/H(3); Q=H(1)/H(3); GOTO L160;
L60: IF ABS(H(N1-1)/H(N1))≠ABS(H(2)/H(1)) THEN GOTO L87; I2=-I2;
M=N1/2;
DO I=1 TO M;
N2=N1+1-I; F=H(N2); H(N2)=H(I); H(I)=F;
END;
IF Q=0 THEN P=0; ELSE
DO;
P=P/Q; Q=1/Q;
END;
IF R≠0 THEN R=1/R;
L87: B(N1),C(N1)=H(N1); B(N1+1),C(N1+1)=0; N3=N1-1; E=5.0E-14;
L100:

```

```

DO J=1 TO 100;
DO I1=1 TO N3;
I=N1-I1; B(I)=H(I)+R*B(I+1); C(I)=B(I)+R*C(I+1);
END;
IF C(2)=0 THEN R=R+1; ELSE R=R-B(1)/C(2);
DO I1=1 TO N3;
I=N1-I1; B(I)=H(I)-P*B(I+1)-Q*B(I+2); C(I)=B(I)-P*C(I+1)-Q*C(I+2);
END;
IF H(2)=0 THEN GOTO L123; IF ABS(B(2)/H(1))>E THEN GOTO L125;
GOTO L124;
L123:
IF ABS(B(2)/H(2)) > E THEN GOTO L125;
L124:
IF ABS(B(1)/H(1)) <= E THEN GOTO L160;
L125:
C1=C(2)-B(2); D=C(3)*+2-C1*C(4); IF D=0 THEN
DO;
P=P-2; Q=Q*(Q+1);
END;
ELSE
DO;
P=P+(B(2)*C(3)-B(1)*C(4))/D; Q=Q+(-B(2)*C1+B(1)*C(3))/D;
END;
END;
F=10+E; GOTO L100;
L150:
N1=N1-1; V(N1)=0; IF I2<0 THEN U(N1)=1/R; ELSE U(N1)=R;
DO I=1 TO N1;
H(I)=B(I+1);

```

```

END;
GOTO L50;
L160:
N1=N1-2; IF I2<0 THEN
DO;
Q1=1/Q; P1=P/(Q*2);
END;
ELSE
DO;
Q1=Q; P1=P/2;
END;
F=P1**2-Q1; IF F >=0 THEN GOTO L176; U(N1+1),U(N1)=-P1;
V(N1+1)=SQRT(-F); V(N1)=-V(N1+1); GOTO L195;
L176:
IF P1=0 THEN U(N1+1)=-SQRT(F); ELSE U(N1+1)=-P1/ABS(P1)*(ABS(P1)+SQRT(F)); V(N1),V(N1+1)=0; U(N1)=Q1/U(N1+1);
L195:
DO I=1 TO N1;
H(I)=B(I+2);
END;
GOTO L50;
L199:
; DCL (P1,Q1,C1,D,E,F,R,P,Q,(U,V,H)(*), (B,C)(N+2) ) DEC(12);
END BERSTOW;

```

6. ЧАСТОТНЫЕ ХАРАКТЕРИСТИКИ

В практике проектирования систем управления широко применяются частотные методы исследования. Эти методы эффективны как при анализе непрерывных, так и дискретных систем.

Соотношения (48) и (50) позволяют получить следующие выражения для частотных характеристик непрерывных и дискретных моделей систем управления:

$$W(j\omega) = \vec{c}^T (j\omega I - A)^{-1} \vec{b}; \quad (57)$$

$$W(j\bar{\omega}) = \vec{c}^T (j\bar{\omega} I - \Psi)^{-1} \vec{\varphi} (1 - j\bar{\omega} T/2). \quad (58)$$

Частоты $\bar{\omega}$ и ω связаны соотношением:

$$\bar{\omega} = \frac{2}{T} \operatorname{tg} \frac{\omega T}{2}. \quad (59)$$

Таким образом, при малых значениях частот частотные характеристики непрерывной и дискретной систем приблизительно совпадают.

Соотношения (57) и (58) используют для вычисления частотных характеристик в заданном диапазоне частот. Однако наиболее эффективный способ построения алгоритма связан с использованием их спектрального разложения.

Соответствующие соотношения принимают вид:

$$W(j\omega) = \sum_{i=1}^n \frac{\vec{c}^T \vec{r}_i \vec{p}_i^H \vec{b}}{j\omega - \lambda_i}; \quad (60)$$

$$W(j\bar{\omega}) = \left[\sum_{i=1}^n \frac{\vec{c}^T \vec{r}_i \vec{p}_i^H \vec{b}}{j\bar{\omega} - \frac{2}{T} \operatorname{th} \frac{\lambda_i T}{2}} \quad \frac{\operatorname{th} \frac{\lambda_i T}{2}}{\frac{\lambda_i T}{2}} \right] (1 - j\bar{\omega} T/2). \quad (61)$$

Для того чтобы построить алгоритм вычисления частотных характеристик согласно соотношениям (60), (61), необходимо сначала рассмотреть часть алгоритма, связанную с вычислением частотной характеристики непрерывной системы.

Представим выражение (60) в виде суммы действительной и мнимой частей:

$$W(j\omega) = U(\omega) + jV(\omega) = \sum_{i=1}^n U_i(\omega) + j \sum_{i=1}^n V_i(\omega). \quad (62)$$

Составляющие $U_i(\omega)$ и $V_i(\omega)$ частотной характеристики от каждого из собственных движений системы вычисляются следующим образом. Компоненты, соответствующие действительному собственному значению λ_i , вычисляются по формулам:

$$\left. \begin{aligned} U_i &= -\lambda_i \frac{\vec{c}^T \vec{r}_i \vec{\ell}_i^H \vec{b}}{\omega^2 + \lambda_i^2}; \\ V_i &= -\omega \frac{\vec{c}^T \vec{r}_i \vec{\ell}_i^H \vec{b}}{\omega^2 + \lambda_i^2}, \end{aligned} \right\} \quad (63a)$$

где \vec{r}_i , $\vec{\ell}_i$ — правые и левые собственные векторы, соответствующие собственному значению λ_i ;
 \vec{b} — вектор передачи управления;
 ω — круговая частота.

В случае комплексного собственного значения $\lambda_i = \alpha_i + j\beta_i$ вычисление U_i и V_i производят по формулам:

$$\left. \begin{aligned} U_i &= -\frac{\alpha_i(\operatorname{Re} \vec{r}_i \operatorname{Re} \vec{\ell}_i^H - \operatorname{Im} \vec{r}_i \operatorname{Im} \vec{\ell}_i^H) + (\beta_i - \omega)(\operatorname{Re} \vec{r}_i \operatorname{Im} \vec{\ell}_i^H + \operatorname{Im} \vec{r}_i \operatorname{Re} \vec{\ell}_i^H)}{(\omega - \beta_i)^2 + \alpha_i^2} \vec{b}; \\ V_i &= -\frac{\alpha_i(\operatorname{Re} \vec{r}_i \operatorname{Im} \vec{\ell}_i^H + \operatorname{Im} \vec{r}_i \operatorname{Re} \vec{\ell}_i^H) - (\beta_i - \omega)(\operatorname{Re} \vec{r}_i \operatorname{Re} \vec{\ell}_i^H - \operatorname{Im} \vec{r}_i \operatorname{Im} \vec{\ell}_i^H)}{(\omega - \beta_i)^2 + \alpha_i^2} \vec{b}. \end{aligned} \right\} \quad (63b)$$

Для получения частотных характеристик дискретной системы формулы (63 а, 63 б) должны вычисляться при подстановке вместо λ_i значений $\bar{\lambda}_i = \frac{2}{T} \pm j \frac{\lambda_i T}{2}$. Кроме того, после вычисления составляющих $U_i(\omega)$ и $V_i(\omega)$ они должны быть преобразованы согласно соотношению:

$$\bar{U}_i(\bar{\omega}) + j\bar{V}_i(\bar{\omega}) = [U_i(\bar{\omega}) + V_i(\bar{\omega})](\bar{\lambda}_i / \lambda_i)(1 - j\bar{\omega} T/2). \quad (64)$$

После этого действительную и мнимую части частотной характеристики при заданном значении частоты вычисляют по формулам:

$$\left. \begin{aligned} \operatorname{Re} W(j\bar{\omega}) &= \sum_{i=1}^n \bar{U}_i(\bar{\omega}); \\ \operatorname{Im} W(j\bar{\omega}) &= \sum_{i=1}^n \bar{V}_i(\bar{\omega}). \end{aligned} \right\} \quad (65)$$

Схема алгоритма *FREQLS* вычисления частотной характеристики показана на рис. 10.

На схеме алгоритма отмечены только два цикла. Внешний цикл по переменной *J* предназначен для формирования выходных массивов *U* и *V*, а цикл по *I* для вычисления элементарных составляющих частотных характеристик непрерывной или дискретной систем $U_i(\omega)$, $V_i(\omega)$. При вычислении для случая комплексных собственных значений введены следующие обозначения:

$$\lambda_i = \alpha_i + j\beta_i; \quad \bar{\lambda}_i = \frac{2}{T} t h \frac{\lambda_i T}{2} = \bar{\alpha}_i + j\bar{\beta}_i;$$

$$Q_1 = \operatorname{Re} \vec{r}_i \operatorname{Re} \vec{\ell}_i^H - \operatorname{Im} \vec{r}_i \operatorname{Im} \vec{\ell}_i^H;$$

$$Q_2 = \operatorname{Re} \vec{r}_i \operatorname{Im} \vec{\ell}_i^H + \operatorname{Im} \vec{r}_i \operatorname{Re} \vec{\ell}_i^H.$$

Для вычисления шага изменения частоты использовано соотношение:

$$\Delta\omega = \left(\frac{\omega_0}{\omega_k} \right)^{1/k-1}. \quad (66)$$

Применение процедуры *FREQLS*. Обращение к процедуре *FREQLS* имеет вид:

$$\text{CALL FREQLS}(N, CT, JR, RJ, LJ, B, Y, T, W, RE, IM). \quad (67)$$

Процедура предназначена для вычисления частотной характеристики системы с одним входом и одним выходом. Мно-

гократное обращение к процедуре *FREQLS* позволяет вычислить частотные характеристики по всем требуемым входам и выходам. Таким образом, введенное допущение, не ограничивая общности решаемой задачи, позволяет существенно сократить объемы требуемой оперативной памяти ЭВМ.

При вычислении частотных характеристик можно рекомендовать 11 точек на декаду. Тогда последовательность значений частот на интервале $[1, 10]$ будет следующей:

1; 1,26; 1,58; 1,99; 2,51; 3,16; 3,98; 5,01; 6,31; 7,94; 10,0.

Входные параметры процедуры *FREQLS*:

- N - порядок исследуемой системы;
 - CT - вектор формирования выхода размером $1 \times N$;
 - JR - матрица собственных значений в жордановой канонической форме размером $N \times N$;
 - RJ - матрица правых собственных векторов при приведении к жордановой канонической форме размером $N \times N$;
 - LJ - матрица левых собственных векторов при приведении к жордановой канонической форме размером $N \times N$;
 - B - вектор коэффициентов передачи управления размером $N \times 1$;
 - Y - вектор, указывающий порядок следования собственных значений;
 - T - такт дискретности, при расчете непрерывной системы необходимо положить T , равным нулю;
 - $W(1)$ - нижняя граница диапазона частот при вычислении частотной характеристики;
 - $W(2)$ - верхняя граница диапазона частот при вычислении частотной характеристики;
 - $W(3)$ - число точек частотной характеристики.
- Параметры $W(1)$, $W(2)$, $W(3)$ входят в массив W .

Выходные параметры процедуры *FREQLS*:

- $W(J)$ - текущее значение частоты, $J = 1, 2, \dots, k$;
- RE - значение действительной части частотной характеристики из массива $U(J)$ на частоте ω ;
- IM - значение мнимой части частотной характеристики из массива $V(J)$ на частоте ω .

Программа процедуры *FREQLS*:

FREQLS:

```

PROC(N,CT,R,A,L,B,Y,W,FR);
DCL N,K,(W0,T),(CT,B,W)(*),(A,R,L,FR)(*),DEC(12),Y(*);
DCL(W1,CR,ICR,LB,ILB,IMA,U,V,RK,IK,D)DEC(12),I,J,M; W1=W(1);
T=W(2); K=W(3);
DO M=1 TO K;
  U,V=0;
  DO I=1 TO N;
    IF Y(I)<0.5 THEN
      DO;
        CR,LB=0.0;
        DO J=1 TO N;
          CR=CR+CT(J)*R(J,I); LB=LB+L(I,J)*B(J);
        END;
        D=CR+LB/(W1+W1+A(I,I)+A(I,I)); U=U-A(I,I)*D; V=V-W1*D;
      END;
    ELSE
      DO;
        IMA=A(I+1,I); CR,ICR,LB,ILB=0.0;
        DO J=1 TO N;
          CR=CR+CT(J)*(R(J,I)+R(J,I+1))/2; ICR=ICR+CT(J)*(R(J,I)-R(J,
          I+1))/2; LB=LB+B(J)*(L(I,J)+L(I+1,J))/2; ILB=ILB+B(J)*(L(I+
          1,J)-L(I,J))/2;
        END;
        RK=CR*LB-ICR*ILB; IK=CR*ILB+ICR*LB;
        DO J=1,2;
          D=(W1-IMA)*(W1-IMA)+A(I,I)*A(I,I); U=U-(A(I,I)*RK-
          IK*(IMA-W1))/D; V=V-(A(I,I)*IK+RK*(IMA-W1))/D; IMA=-IMA;
          IK=-IK;
        END;
        I=I+1;
      END;
    END;
  END;
  FR(M,1)=W1; FR(M,2)=U; FR(M,3)=V; W1=W1*T;
END FREQLS;

```


7. ОЦЕНКА КАЧЕСТВА

При исследовании систем управления наряду с прямыми показателями качества (перерегулирование, время регулирования и т.п.), оцениваемыми по виду переходного процесса, применяются обобщенные показатели в форме интегральных оценок переходной динамической составляющей ошибки по регулируемой координате. Для этого используют следующие оценки:

$$\left. \begin{aligned} J_0 &= \int_0^{\infty} e^2(t) dt; \\ J_1 &= \int_0^{\infty} \left\{ e^2(t) + r_1^2 \left(\frac{de}{dt} \right)^2 \right\} dt; \\ &\dots \dots \dots \\ J_n &= \int_0^{\infty} \left\{ e^2(t) + r_1^2 \left(\frac{de}{dt} \right)^2 + r_2^2 \left(\frac{d^2e}{dt^2} \right)^2 + \dots + r_n^2 \left(\frac{d^n e}{dt^n} \right)^2 \right\} dt, \end{aligned} \right\} (68)$$

в которых учитывают не только составляющие самой выходной координаты, но и ее производных. Для аналитического вычисления интегралов в соотношениях (68) целесообразно воспользоваться формулой Парсеваля [3], которая позволяет перейти от анализа во временной к анализу в частотной области. Кроме того, такой подход позволяет выразить оценку J_n по аналогии с оценкой J_0 в форме:

$$\left. \begin{aligned} J_0 &= \frac{1}{2\pi} \int_{-\infty}^{\infty} |E(j\omega)|^2 d\omega; \\ J_1 &= J_0 + r_1^2 \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega^2 |E(j\omega)|^2 d\omega; \\ &\dots \dots \dots \\ J_n &= J_{n-1} + r_n^2 \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega^{2n} |E(j\omega)|^2 d\omega. \end{aligned} \right\} (69)$$

Легко видеть, что вычисление любого из интегралов в соотношениях (69) сводится к вычислению интеграла вида

$$I = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{B(j\omega)B(-j\omega)}{A(j\omega)A(-j\omega)} d\omega, \quad (70)$$

где $B(s=j\omega)$ и $A(s=j\omega)$ — полиномы, имеющие вид:

$$B(s) = b_1 s^{n-1} + \dots + b_{n-1} s + b_n;$$

$$A(s) = a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n.$$

Если полином $A(s)$ не имеет нулей на мнимой оси, то интеграл (70) существует и может быть вычислен по следующим рекуррентным формулам [7]:

$$\left. \begin{aligned} I_k &= I_{k-1} + \beta_k^2 / 2L_k, \quad k=1, 2, \dots, n; \\ I_0 &= 0; \\ I &= I_n, \end{aligned} \right\} (71)$$

где рекурсия связана с понижением порядка полиномов $B(s)$ и $A(s)$. Для полиномов порядка $k \leq n$ использованы обозначения:

$$B_k(s) = b_1^{(k)} s^{k-1} + \dots + b_{k-1}^{(k)} s + b_k^{(k)};$$

$$A_k(s) = a_0^{(k)} s^k + a_1^{(k)} s^{k-1} + \dots + a_{k-1}^{(k)} s + a_k^{(k)}.$$

Таким образом, $B(s) = B_n(s)$, $A(s) = A_n(s)$, а величины L_k и β_k вычисляются по формулам:

$$\left. \begin{aligned} L_k &= a_0^{(k)} / a_1^{(k)}; \\ \beta_k &= b_1^{(k)} / a_1^{(k)}. \end{aligned} \right\} (72)$$

Рекуррентные соотношения для полиномов имеют вид:

$$\left. \begin{aligned} B_{k-1}(s) &= B_k(s) - \beta_k \tilde{A}_k(s); \\ A_{k-1}(s) &= A_k(s) - L_k s \tilde{A}_k(s), \\ &k = n, n-1, \dots, 1, \end{aligned} \right\} (73)$$

где

$$A_k(s) = \bar{A}_k(s) + \tilde{A}_k(s);$$

$$\bar{A}_k(s) = a_0^{(k)} s^n + a_2^{(k)} s^{n-2} + \dots = \frac{1}{2} [A_k(s) + (-1)^n A_k(-s)];$$

$$\tilde{A}_k(s) = a_1^{(k)} s^{n-1} + a_3^{(k)} s^{n-3} + \dots = \frac{1}{2} [A_k(s) - (-1)^n A_k(-s)].$$

Последовательность вычисления коэффициентов $a_i^{(k)}$ представлена в табл. 2. Каждая четная строка в ней получается сдвигом элементов предшествующей строки влево и подстановкой нулей. Четные строки правой части таблицы идентичны строкам ее левой части. Элементы нечетных строк получают-ся вычитанием из элементов предшествующей нечетной строки элементов четной строки с коэффициентами $\alpha_n, \alpha_{n-1}, \dots, \alpha_1$ для левой части таблицы и с коэффициентами $\beta_n, \beta_{n-1}, \dots, \beta_1$ для правой части таблицы.

Таблица для вычисления коэффициентов полиномов пониженного порядка преобразуется в вычислительный алгоритм *ISQERR*, схема которого представлена на рис. 11.

Применение процедуры ISQERR. Обращение к процедуре *ISQERR* имеет вид:

$$\text{CALL ISQERR}(N, A, B, V, IERR). \quad (74)$$

Процедура предназначена для вычисления интегралов вида (70). Определение интегральных оценок (69) выполняется многократным применением процедуры *ISQERR* для вычисления интегралов с подынтегральными выражениями вида $|\omega^k E(j\omega)|^2, k=1, \dots, n-m-1$, где m - порядок полинома числителя $B(j\omega)$, а n - порядок полинома знаменателя $A(j\omega)$.

Входные параметры процедуры ISQERR:

- N - порядок полинома знаменателя передаточной функции;
- $A(N+1)$ - массив коэффициентов знаменателя передаточной функции;
- $B(N)$ - массив коэффициентов числителя передаточной функции.

Нумерация коэффициентов определяется из следующего представления:

$$\frac{B(s)}{A(s)} = \frac{B(1)s^{N-1} + B(2)s^{N-2} + \dots + B(N)}{A(1)s^N + A(2)s^{N-1} + \dots + A(N+1)}, \text{ причем } A(1) > 0.$$

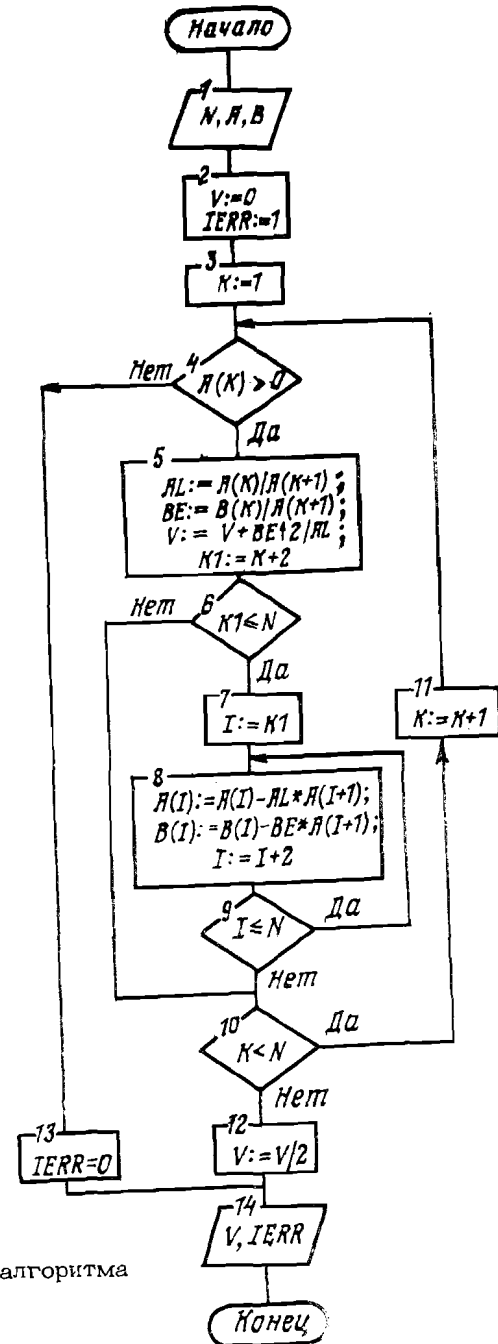


Рис. 11. Схема алгоритма *ISQERR*

Вычисление полиномов

$$A_k(s), B_k(s)$$

Номер строки	Коэффициенты $a_i^{(k)}$	λ_i	Коэффициенты $b_i^{(k)}$	β_i
1	$a_0^{(n)}, a_1^{(n)}, a_2^{(n)}, a_3^{(n)}, \dots$	$\lambda_1^{(n)}$	$b_1^{(n)}, b_2^{(n)}, b_3^{(n)}, \dots$	$\beta_1^{(n)}$
2	$a_1^{(n-1)}, a_2^{(n-1)}, a_3^{(n-1)}, \dots$	$\lambda_2^{(n-1)}$	$a_1^{(n-1)}, a_2^{(n-1)}, a_3^{(n-1)}, \dots$	$\beta_2^{(n-1)}$
3	$a_0^{(n-1)}, a_1^{(n-1)}, a_2^{(n-1)}, \dots$	$\lambda_1^{(n-1)}$	$b_1^{(n-1)}, b_2^{(n-1)}, b_3^{(n-1)}, \dots$	$\beta_1^{(n-1)}$
\vdots	\vdots	\vdots	\vdots	\vdots
$2n-3$	$a_0^{(2)}, a_1^{(2)}, a_2^{(2)}$	$\lambda_0^{(2)}$	$b_1^{(2)}, b_2^{(2)}$	$\beta_1^{(2)}$
$2n-2$	$a_1^{(2)}, 0$	$\lambda_2^{(2)}$	$a_1^{(2)}, 0$	$\beta_2^{(2)}$
$2n-1$	$a_0^{(1)}, a_1^{(1)}$	$\lambda_0^{(1)}$	$b_1^{(1)}$	$\beta_1^{(1)}$
$2n$	$a_1^{(1)}, 0$	$\lambda_1^{(1)}$	$a_1^{(1)}, 0$	$\beta_1^{(1)}$

Выходные параметры процедуры ISQERR:

V - значение интеграла (70);

IERR - переменная, используемая для выдачи информации об устойчивости системы. Если IERR=1, то все нули полинома знаменателя расположены в левой полуплоскости комплексной переменной s; если IERR=0, то значение $\lambda^{(1)}$ на какой-то итерации оказалось отрицательным, и система неустойчива.

Программа процедуры ISQERR:

```

ISQERR:
PROC(N, A, B, V, IERR);
DCL N, IERR; DCL(A(*), B(*), V) FLOAT(12); DCL K, K1; IERR=1;
V=0; IF A(1) <= 0 THEN GOTO NEW1;
NA:
DO K=1 TO N;
IF A(K+1) <= 0 THEN GOTO NEW1; ALFA=A(K)/A(K+1); BETA=B(K)/A(K+1);
V=V+BETA**2/ALFA; K1=K+2; IF K1-N > 0 THEN GOTO TWENTY;
NB:
DO I=K1 TO N BY 2;
A(I)=A(I)-ALFA*A(I+1); B(I)=B(I)-BETA*A(I+1);
END NB;
TWENTY:
END NA;
RETURN;
NEW1:
IERR=0;
END ISQERR;
    
```

8. КОРНЕВОЙ ГОДОГРАФ

Анализ траекторий корней характеристического уравнения замкнутой системы управления в функции одного или нескольких параметров регулятора является одной из основных задач при проектировании систем управления. Траектории корней, или корневые годографы, широко применяются как в классической теории [4], так и современной теории управления [5].

Поставленная задача сводится к определению собственных значений матрицы замкнутой системы в функции некоторого параметра μ . Ее решение включает следующие этапы: задание параметра μ ; формирование матрицы замкнутой системы по характеристикам неизменяемой части и характеристикам регулятора, зависящим от параметра μ , с помощью процедуры *JOINT*; обращение к процедуре *COMEEG*; повторение вышеуказанных этапов для нескольких значений μ ; построение кривых корневого годографа в функции параметра μ .

Наиболее целесообразный способ решения такой задачи — режим диалога с ЭВМ. В этом случае выбор значений параметра μ и принятие решения об отборе точек корневого годографа, выводимого на график, выполняет проектировщик.

Схема диалогового алгоритма расчета корневого годографа представлена на рис. 12.

Исходные данные о неизменяемой части системы $\{A^{(1)}, B^{(1)}, C^{(1)}\}$ хранятся в некотором наборе данных во внешней памяти ЭВМ (блок 1).

В блоке 2 присваивается начальное значение счетчику числа вариантов. Ввод параметра μ выполняется с дисплея (блок 3). В блоке 4 вычисляются характеристики регулятора $(A^{(2)}(\mu), B^{(2)}(\mu), C^{(2)}(\mu), D^{(2)}(\mu))$ при заданном значении μ . В блоке 5 происходит обращение к ранее описанным процедурам *JOINT* (47) и *COMEEG* (36). В блоке 6 вычисленные значения $\lambda_i (i=1, 2, \dots, N)$ выводятся на экран дисплея и в блоке 7 принимается решение о продолжении ($Q=0$) (блок 8) или завершении ($Q=1$) расчета корневого годографа. В блоке 9 выводятся на печать на АЦПУ все $I \times N$ значений вычисленных корней замкнутой системы в функции параметра μ . В блоке 10 осуществляется формирование массива собственных значений для вывода их на печать в виде кривых корневого годографа с помощью процедуры *SPLIT*. В блоке 11 корневой годограф выводится на АЦПУ.

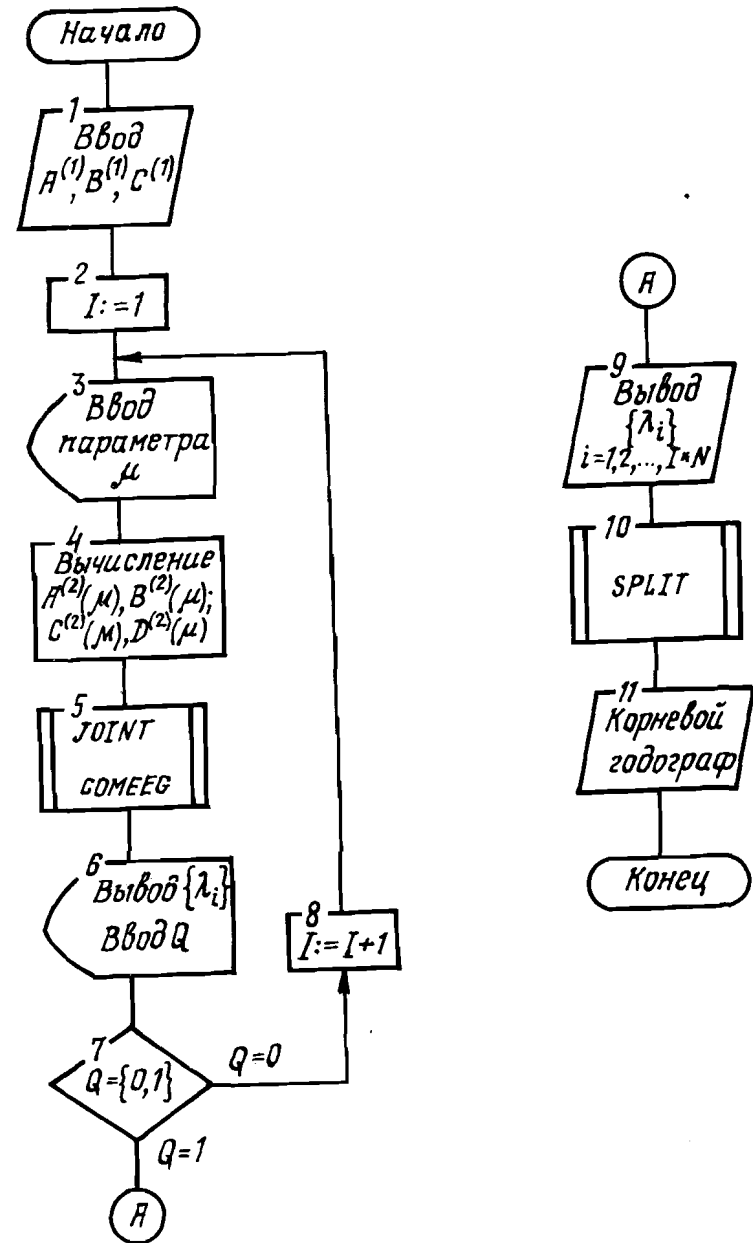


Рис. 12. Схема алгоритма расчета корневого годографа в режиме диалога

Применение процедуры SPLIT. Обращение к процедуре SPLIT имеет вид:

CALL SPLIT(U,V,N). (75)

В процедуре SPLIT выполняется автоматическая сортировка и масштабирование. По оси y введено 10 градаций, отмечаемых горизонтальными линиями, а внутри них еще по 10 градаций, таким образом, весь диапазон изменения переменной по оси y разбит на 100 делений.

Масштаб по оси x равен масштабу по оси y , так что результирующий график имеет квадратную разметку и углы могут быть измерены непосредственно. Это удобно при изображении корневых годографов и амплитудно-фазовых характеристик. Однако, если диапазоны изменения переменных x и y значительно различаются, то график может оказаться или слишком сжатым, или растянутым. Для отображения графиков с диапазоном изменения переменных, различающихся более чем на порядок, целесообразно применять процедуры, в которых масштабы по осям x и y различны, например V8V3X [19].

Входные параметры процедуры SPLIT:

- U - массив x -координат точек, выводимых на печать;
- V - массив y -координат точек, выводимых на печать;
- N - число точек.

Выходные параметры процедуры SPLIT:

Выходом процедуры SPLIT является отображение на бумажном носителе точек, координаты которых записаны в массивах U и V.

Программа процедуры SPLIT:

```

SPLIT:
PROC(U, V, N);
DCL K(1, N), LINE(101) CHAR(1), CORD(11), (X1, Y1) (N), (U1, V1) (*), JTK EXT;
PUT PAGE EDIT(TEPMINHA, JTK)(A, F(3)); DCL (IPLU, IZERO, IBLAN, IEVEN,
IEYS, MINUS, IAST) CHAR(1); IPLU='+'; IZERO='0'; IBLAN=
';'; IEVEN='I'; MINUS='-'; IAST='*'; KS=0;
DO I=1 TO N;
  X1(I)=U(I); Y1(I)=V1(I);
END;
YMAX=Y1(1); YMIN=Y1(1);
DO I=2 TO N;
  IF YMAX>=Y1(I) THEN GOTO M105; YMAX=Y1(I);
M105:; IF YMIN<=Y1(I) THEN GOTO M107; YMIN=Y1(I);
M107:;
END;
DO I=1 TO N;
  IF YMAX/=YMIN THEN GOTO M300; YMAX=YMAX+0.1*Y1(I); YMIN=YMIN-0.1*
Y1(I);
M300:;
END;
DO I=1 TO N-1;
  DO J=I+1 TO N;
    IF X1(I)<=X1(J) THEN GOTO M201; TEM=X1(I); X1(I)=X1(J);
    X1(J)=TEM; Y1(I)=Y1(J); Y1(J)=TEM;
M201:;
  ;
END;
END;
XMIN=X1(1); XMAX=X1(N); IF XMAX=XMIN THEN
DO;

```

```

XMAX=XMAX+0.1*XMIN; XMIN=XMIN-0.1*XMIN;
END;
IF XMAX=0 & XMIN=0 THEN
DO;
XMAX=1; XMIN=-1;
END;
IF YMAX=0 & YMIN=0 THEN
DO;
YMAX=1; YMIN=-1;
END;
TEM=YMAX-YMIN; YMAX=YMAX+0.1*TEM; YMIN=YMIN-0.1*TEM; TEM=XMAX-
XMIN; XMAX=XMAX+0.1*TEM; XMIN=XMIN-0.1*TEM; COR=(YMAX-YMIN)/10.0;
CRX=(XMAX-XMIN)/10.0; TEM=ABS((XMAX-XMIN)/(YMAX-YMIN));
IF TEM>0.5 & TEM<=2 THEN CRX=COR; M=6; M1=M+1; AM=M*10;
DO J=1 TO 10;
TEM=J; CORD(J+1)=TEM*COR+YMIN;
END;
CORD(1)=YMIN; PUT PAGE EDIT(CORD)(SKIP,X(10),(11)(X(2),E(8,1)));
ZK=100.*YMIN/(YMIN-YMAX); ZL=-AM*XMIN/CRX/10.0; LZ=LZ+1.5;
KZ=ZK+1.5; ITAL=M1;
DO K=1 TO N;
Y1(K)=100.*Y1(K)/(YMAX-YMIN)+ZK+1.5; X1(K)=AM*X1(K)/CRX/10.0+ZL+
1.5;
END;
K1,KTAL=1;
M109:
;
DO I=1 TO 101;
LINE(I)=IBLAN;
END;
DO J=1 TO 101 BY 10;
LINE(J)=IEYEN;

```

```

END;
M301:
; IF ITAL=M1 THEN GOTO M38;
DO J=2 TO 100;
LINE(J)=MINUS;
END;
DO J=11 TO 91 BY 10;
LINE(J)=IPLU;
END;
M38:
; IF LZ=KTAL THEN GOTO M164;
DO I=1 TO 101;
LINE(I)=IZERO;
END;
M164*IF KZ<=0 ! KZ>100 THEN GOTO M101; INE(KZ)=IZERO;
M101:
; NX=X1(K1); IF NX=KTAL THEN GOTO M102; NY=Y1(K1); LINE(NY)=
IAST; K1=K1+1; GOTO M101;
M102:
; IF ITAL=M1 THEN GOTO M32; AKS=KS; ST=AKS+CRX*XMIN;
KS=KS+1; PUT EDIT(ST,LINE)(SKIP,X(3),E(10,3),X(2),(101)A(4));
ITAL=1; GOTO M35;
M32:
; PUT EDIT(LINE)(SKIP,X(15),(101)A(4));
M35:
; ITAL=ITAL+1; KTAL=KTAL+1; IF K1<N THEN GOTO M109;
DO I=1 TO 101;
LINE(I)='-';
END;
PUT EDIT(LINE)(SKIP,X(15),(101)A(1)); RETURN;
END SPLIT;

```

9. ПЕРЕХОДНЫЕ ПРОЦЕССЫ

Исследование переходных процессов завершает этапы анализа и синтеза системы управления. Обычно исследуют реакцию системы на стандартные входные воздействия и выявляют характеристики качества непосредственно по виду переходного процесса.

Так же как и при построении корневого годографа, при построении переходного процесса необходимо знание векторно-матричного описания замкнутой системы в форме:

$$\left. \begin{aligned} \dot{\vec{x}} &= A \vec{x} + \vec{b} r; \\ \vec{y} &= C \vec{x}, \end{aligned} \right\} \quad (76)$$

где $r(t)$ - входное воздействие;
 \vec{x} - вектор переменных состояния;
 \vec{y} - выход системы;
 A - матрица замкнутой системы управления;
 \vec{b} - вектор коэффициентов передачи входа;
 C - матрица формирования выхода.

Стандартное входное воздействие $r(t)$ выбирают или из класса гармонических функций

$$r(t) = m e^{\alpha t} \sin \omega t, \quad (77a)$$

или степенных функций

$$r(t) = M_0 + M_1 t + M_2 t^2. \quad (77b)$$

Для того чтобы при расчете переходных процессов учесть вид входного воздействия, необходимо преобразовать систему (76) к виду, в котором генераторы входных воздействий входят в описание динамики в качестве дополнительных уравнений.

Для формирования гармонического сигнала (77a) введем вектор $\vec{r}^T = [r_1, r_2]$. Дифференциальные уравнения генерации принимают вид:

$$\left. \begin{aligned} \dot{\vec{r}} &= R \vec{r} = \begin{bmatrix} \alpha & \omega \\ -\omega & \alpha \end{bmatrix} \vec{r}; & \vec{r}_0 &= \begin{bmatrix} 0 \\ M \end{bmatrix}; \\ r &= \vec{C}^* \vec{r} = [1 \ 0] \vec{r}. \end{aligned} \right\} \quad (78a)$$

Для формирования степенного сигнала (77б) необходимо ввести дополнительный трехмерный вектор \vec{r} . Соответствующая система уравнений генерации принимает вид:

$$\left. \begin{aligned} \vec{r} = R\vec{r} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \vec{r}; \quad r_0 = \begin{bmatrix} M_0 \\ M_1 \\ 2M_2 \end{bmatrix}; \\ r = \vec{C}^T \vec{r} = [100] \vec{r}. \end{aligned} \right\} \quad (78б)$$

Объединяя уравнения (76) и (78), получаем следующую расширенную систему уравнения для описания автономной замкнутой системы:

$$\left. \begin{aligned} \begin{bmatrix} \vec{x} \\ \vec{r} \end{bmatrix} = \begin{bmatrix} A & \vec{b} \vec{C}^T \\ 0 & R \end{bmatrix}, \quad \begin{bmatrix} \vec{x} \\ \vec{r} \end{bmatrix}_0 = \begin{bmatrix} \vec{x}_0 \\ \vec{r}_0 \end{bmatrix}; \\ \vec{y} = [C \quad 0] \begin{bmatrix} \vec{x} \\ \vec{r} \end{bmatrix}. \end{aligned} \right\} \quad (79)$$

Для системы уравнений (79) алгоритм вычисления переходного процесса сводится к определению переходной матрицы и вычислению реакции системы на заданные начальные условия согласно рекуррентному соотношению:

$$\left. \begin{aligned} \begin{bmatrix} \vec{x}_k \\ \vec{r}_k \end{bmatrix} = \tilde{\Phi} \begin{bmatrix} \vec{x}_{k-1} \\ \vec{r}_{k-1} \end{bmatrix}, \quad \begin{bmatrix} \vec{x} \\ \vec{r} \end{bmatrix} = \begin{bmatrix} \vec{x}_0 \\ \vec{r}_0 \end{bmatrix}; \\ k=1, 2, \dots, n, \end{aligned} \right\} \quad (80)$$

где $\tilde{\Phi}$ - переходная матрица системы (79).

Схема алгоритма вычисления и построения переходного процесса приведена на рис. 13,а. Этот алгоритм, так же как и алгоритм построения корневого годографа, предусматривает диалог проектировщика с ЭВМ. Все блоки, в которых изображен знак "?", связаны с прерыванием процесса расчета и ожиданием ответа. Такой диалог предполагает применение алфавитно-цифровых или графических дисплеев. ЭВМ запрашивает порядок N , матрицу A , вектор \vec{b} и матрицу C замкнутой следящей системы (76), а также тип Q стандартного входного воздействия (77).

После этого в зависимости от признака Q запрашиваются параметры матрицы R , значения начальных условий \vec{r}_0 и модифицируется порядок результирующей системы. Затем формируется матрица \tilde{A} и поступает запрос на величину такта дискретности T , требуемого для вычисления переходной матрицы $\tilde{\Phi}$ и ординат переходного процесса. Вычисление $\tilde{\Phi}$ реализовано с помощью разложения в матричный ряд

$$\tilde{\Phi} = I + \tilde{A}T + \frac{\tilde{A}^2 T^2}{2!} + \frac{\tilde{A}^3 T^3}{3!} + \frac{\tilde{A}^4 T^4}{4!} \quad (81)$$

Максимально допустимое время переходного процесса, для которого сохраняется заданная точность вычислений, определяется [15] по следующей формуле:

$$t_{\text{макс}} = \frac{5 \ln(1 + \delta)}{T^4 |\lambda_{\text{макс}}|^5} \quad (82)$$

где δ - заданная погрешность вычислений;

$\lambda_{\text{макс}}$ - максимальное собственное значение матрицы A .

Начало построения переходного процесса соответствует моменту времени $T_0 = 0$. Предусмотрен запрос начального состояния \vec{x}_0 . Вычисление переходного процесса на интервале (T_0, T_K) реализовано в виде процедуры *TRESP* (рис. 13, б).

По окончании расчета переходного процесса известны реакции системы по всем переменным состояниям, что является исчерпывающей информацией для построения графиков переходных процессов по интересующим составляющим выхода.

Если время переходного процесса оказалось недостаточным или требуется изменить начальные условия по переменным состояниям исследуемой системы, или такт дискретности в алгоритме предусмотрены соответствующие запросы с терминала. Если полученные числовые массивы полностью характеризуют переходные процессы в системе, то построение их графиков выполняется процедурой *GRAPH*.

Применение процедуры *TRESP*. Обращение к процедуре *TRESP* имеет вид:

$$\text{CALL TRESP}(\Phi, XK, RK, T_0, T_K, T) \quad (83)$$

Входные параметры процедуры *TRESP* :

Φ - переходная матрица системы (79) для заданного значения такта дискретности;

XK - вектор начальных условий для системы (76);

RK - вектор начальных условий для системы (78 а, б);
 $T0$ - момент начала расчета переходных процессов;
 TK - момент окончания расчета переходных процессов;
 T - такт дискретности.

Выходные переменные процедуры $TRESP$:

X - массив состояний в моменты времени $T0, T1, \dots, TK$;
 $T0$ - переменная, которая при выходе из процедуры равна TK .

Это удобно для продолжения цикла вычислений переходного процесса, если оказалось, что назначенного значения TK недостаточно для завершения переходного процесса.

Применение процедуры $GRAPH$. Обращение к процедуре $GRAPH$ имеет вид:

$$CALL GRAPH(K, N, A). \quad (84)$$

Входные параметры процедуры $GRAPH$:

K - количество точек по оси абсцисс;
 N - количество выводимых функций;
 A - матрица выводимых значений размером $K \times (N+1)$
 значения абсцисс - в позициях $A(I, 1)$;
 значения ординат первой функции - $A(I, 2)$;
 значения ординат второй функции - $A(I, 3)$;

 значения ординат N -й функции - $A(I, N+1)$.

Выходные параметры процедуры $GRAPH$:

Выходом процедуры являются графики функций $F1(X), F2(X), \dots, FN(X)$. Для различения ординат используют следующие символы:

$F1(X) - 'A'$	$F5(X) - 'C'$
$F2(X) - 'B'$	$F6(X) - 'D'$
$F3(X) - 'A'$	$F7(X) - 'E'$
$F4(X) - 'B'$

10. ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ И ЕГО ОРГАНИЗАЦИЯ

Для организации процессов проектирования систем управления в пакетном или диалоговом режимах приведенные выше

процедуры целесообразно объединить в пакет прикладных программ исследования динамических систем. В состав пакета входят модули, написанные на языке ПЛ/1. Управляющая программа (монитор пакета) организует последовательность вызова модулей для решения конкретных задач исследования. В табл. 3 показано взаимодействие модулей пакета для решения этих задач.

Реализация многих из приведенных алгоритмов расчета наиболее эффективна при использовании режима диалога. В режиме диалога проектировщик получает возможность изменения исходных данных, последовательности расчета, выводимых переменных в ходе решения задачи на ЭВМ.

Имеются необходимые технические средства (алфавитно-цифровые и графические дисплеи, графопостроители) и системное программное обеспечение для организации диалога при проектировании систем управления. Однако требуется разработка универсальных и проблемно-ориентированных языков диалога, организация диалоговых средств работы с пакетами прикладных программ.

В настоящее время имеются первые результаты в этой новой области исследований. Разработана диалоговая система программирования ДИАЛ [10], которая предоставляет пользователю универсальный язык диалога, средства отладки программ, написанных на диалоговом языке; возможность хранения, модификации и коррекции программ и данных пользователя; средства работы с пакетами программ в режиме диалога; возможность вызова модулей на языках ПЛ/1, ФОРТРАН; возможность пополнения пакета программ.

Таблица 3

Пакет прикладных программ исследования динамических систем

Перечень задач	Модули пакета прикладных программ													
	MNFLT	DISCR	COMEIG	COMEEG	KH	MFUNC	JOINT	TRFUNG	BERSTOW	FREQLS	ISQERR	SPLIT	TRESP	GRAPH
Линеаризация	●													

Перечень задач	Модули пакета прикладных программ													
	MINFIT	DISCR	COMELG	COMEG	KH	MFUNG	JOINT	TRFUNG	BERSTOW	FREQLS	ISQERR	SPLIT	TRESP	GRAPH
Построение дискретной модели		●												
Спектральное разложение			●	●										
Оценка спектрального числа обусловленности					●									
Вычисление функций от матриц						●								
Соединения динамических систем							●							
Вычисление передаточных функций								●	●					
Вычисление нулей									●					
Вычисление частотных характеристик										●		●		●
Оценка качества процессов											●			
Построение корневого годографа				●			●					●		
Построение переходных процессов							●						●	●

В заключение следует отметить, что вопросы создания вычислительных методов и алгоритмов решения задач исследования и проектирования систем управления, описанные в пособии, характеризуют новое направление в развитии современной теории управления.

СПИСОК ЛИТЕРАТУРЫ

1. Алберт А. Регрессия, псевдоинверсия и рекуррентное оценивание. Пер. с англ. М.: Наука, 1977, 224 с.
2. Деруссо П., Рой Р., Клоуз Ч. Пространство состояний в теории управления. Пер. с англ. М.: Наука, 1970, 472 - 477 с.
3. Заде Л., Дезоер Ч. Теория линейных систем. Пер. с англ. М.: Наука, 1970, 704 с.
4. Ивашенко Н. Н. Автоматическое регулирование. Изд. 4-е. М.: Машиностроение, 1978, 736 с.
5. Квакернаак Х., Сиван Р. Линейные оптимальные системы управления. Пер. с англ. М.: Мир, 1977, 652 с.
6. Наумов Б. Н. Теория нелинейных автоматических систем. М.: Наука, 1972, 544 с.
7. Остром К. Введение в стохастическую теорию управления. Пер. с англ. М.: Мир, 1973, 322 с.
8. Пакет прикладных программ для моделирования на ЭВМ аналоговых систем и непрерывных процессов. М., Б. М., ПРО.309.006 ТУ-11с, Ф0- 19 с, ЭД-23с, Д-34с, Д1-78с, 1977, Д2-71с, 1978.
9. Потемкин В. Г. Машинное проектирование (математическое описание динамических систем в пространстве состояний). М.: Изд. МИФИ, 1978, 80 с.
10. Потемкин В. Г., Кравченко В. В. Диалоговая система программирования ДИАЛ. М.: Изд. МИФИ, 1980, 72 с.
11. Сейдж Э. П., Мелса Дж. Л. Идентификация систем управления. Пер. с англ. М.: Наука, 1974, 248 с.
12. Уилкинсон. Алгебраическая проблема собственных значений. Пер. с англ. М.: Наука, 1970, 564 с.
13. Уилкинсон, Райнш. Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра. Пер. с англ. М.: Машиностроение, 1976, 389 с.
14. Форсайт Дж., Молер К. Численное решение систем линейных алгебраических уравнений. Пер. с англ. М.: Мир, 1969, 218 с.

15. Чернецкий В. И., Дидук Г. А., Потапенко А. А. Математические методы и алгоритмы исследования автоматических систем. Л.: Энергия, 1970, 375 с.

16. Шеннон Р. Имитационное моделирование систем: искусство и наука. Пер. с англ. М.: Мир, 1978, 418 с.

17. Якубовски Р. Алгоритм моделирования динамических систем с помощью вычислительных машин, основанный на теории потоковых графов. Кибернетический сборник. Новая серия. Вып. 7. М.: Мир, 1970, 219 - 236 с.

18. *Conference on Computational-Aided Control System Design, IEEE Conference Publication, v, 96, 1973.*

19. *Melsa J.L., Jones S.K. Computer programs for computational assistance in the study of linear control theory. N.Y.: Mc Graw Hill, 1973, 198 p.*

ОГЛАВЛЕНИЕ

Предисловие	3
1. Линеаризация динамических систем	4
2. Построение дискретных моделей систем управления	13
3. Спектральное разложение произвольных действительных и комплексных матриц. Вычисление функций от матриц	23
4. Соединения динамических систем	42
5. Передаточные функции	49
6. Частотные характеристики	58
7. Оценка качества	64
8. Корневой годограф	70
9. Переходные процессы	76
10. Пакет прикладных программ и его организация	80
Список литературы	83

Валерий Георгиевич
Потемкин

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ
(алгоритмы исследования систем управления)

Редактор О. А. Сафронова
Техн. редактор Н. М. Генкина
Корректор В. З. Решетникова

Л-65502	Подписано в печать 2/1У-1980 г.
Формат 60x84 1/16	Объем 5,5 Уч.-изд.л. 5
Тираж 800 экз.	Цена 22 коп. Изд. № 021-1
	Заказ 2544

Типография МИФИ, Каширское шоссе, д. 1