УДК 681.3.06

ПРОБЛЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ В СРЕДЕ MATLAB

В.А. Бороденко
Павлодарский государственный университет им. С. Торайгырова

Для компьютерного моделирования систем регулирования и управления используется программный продукт MATLAB® фирмы Math Works, Inc [1], который де-факто стал стандартом в проектных, исследовательских, научных и учебных организациях. Целесообразно проанализировать, насколько эта программа пригодна для использования в учебном процессе при преподавании теории автоматического управления (ТАУ).

Основные проблемы, затрудняющие использование указанной среды в учебном процессе высшей школы при подготовке бакалавров и магистров, можно разделить на три группы: это полное отсутствие некоторых общепринятых у нас методов анализа и синтеза систем управления в западной практике и, соответственно, в MATLAB; неодинаковая трактовка некоторых методов в зарубежной и отечественной практике; и, наконец, просто ошибки или неувязки, заложенные в среду MATLAB при её создании.

В первую группу, например, следует отнести полное отсутствие здесь таких методов оценки устойчивости, как алгебраические критерии Гурвица и Рауса, критерий Михайлова, метод D-разбиения по одному или двум параметрам. Эту проблему можно решить, вводя собственные функции в виде специально созданных m-файлов, что позволяет сделать открытая архитектура MATLAB, либо используя нетрадиционно штатные средства программы. Примером первого подхода являются разработанные нами в виде m-файлов функции hurwitz и routh [2], используемые для расчета устойчивости по критериям Гурвица и Рауса. Для получения кривой D-разбиения по одному параметру нами используется штатная функция nyquist, учитывая, что программа производит расчет характеристик в диапазоне частот от минус до плюс бесконечности.

С помощью этой же функции может быть выполнено построение годографа Михайлова, причем целесообразно самостоятельно задать диапазон частот f (в примере от 0 до 2 рад/с с шагом 0.1) и указать у дроби передаточной функции единицу в качестве знаменателя, поскольку вводится только числитель, равный характеристическому полиному.

```
>> f=0:0.1:2;
>> d=[1  2  3  4  1];
>> nyquist(d, 1, f)
```

На графике следует убрать кривую при отрицательных частотах (рисунок 1), для чего снять флажок в меню Show-Negative Frequencies программы.

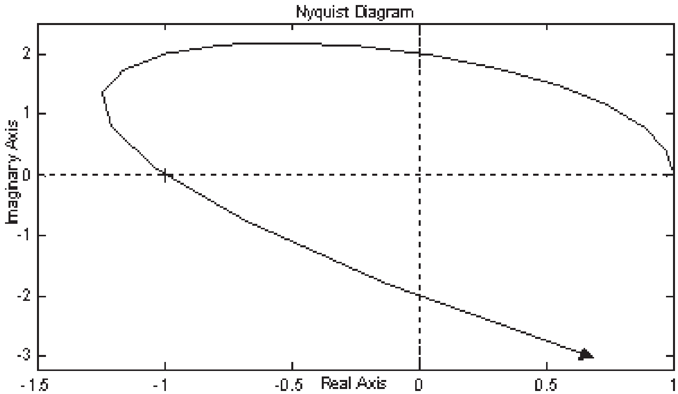

Рисунок 1

Аналогичный результат может быть получен без использования функции nyquist(). Надпись к графику, оси и подписи к ним придется делать самостоятельно, зато не будет креста в точке (-1,*j*0).

```
>> d=[1 2 3 4 1]; w=0:0.01:2; f=0+j*w; h=polyval(d,f);
>> plot(h),grid; title('Критерий Михайлова')
```

Приводимая далее специально разработанная функция uv() [3] реализует критерий Михайлова (вторую форму) путем вычисления коэффициентов четной и нечетной функций и выводом (если указано в левой части выражения) их корней, т.е. частот пересечения графиков $U(\omega)$ и $V(\omega)$ с осью абсцисс.

```
function [u,v,ur,vr] = uv(p)
% Входной параметр - полином p, выходной - векторы U и V

n=length(p);pp=p;p=fliplr(p);
m=n-(n~=fix(n/2)*2);
k=0;kk=1;
for i=1:n
    if k==2
        k=0;kk=kk*(-1);
    end
    p(i)=p(i)*kk; k=k+1;
end
ut=[];vt=[];urt=[];vrt=[];
```

```
i=1:2:n; ut(i)=[ut,p(i)];
i=1:2:m; vt(i)=[vt,p(i+1)];
ut=fliplr(ut);vt=[fliplr(vt),0];
urr=roots(ut);vrr=roots(vt);  % вычисление корней функций
mur=(real(urr)>=0 & imag(urr)==0); % выбор действительных частот
mvr=(real(vrr)>=0 & imag(vrr)==0);
for i=1:length(urr)  % формирование вектора частот четной функции
    if mur(i)==1
        urt=vertcat(urt,urr(i));
    end
end
for i=1:length(vrr)  % формирование вектора частот нечетной функции
    if mvr(i)==1
        vrt=vertcat(vrt,vrr(i));
    end
end
if nargout==0 % проверка необходимости строить график
     w=0:0.01:(max([urt vrt])*1.1);  % до максимальной частоты
    f=0+j*w; h=polyval(pp,f);
      plot(w,real(h),w,imag(h),'--',w,0,'k.-'), grid;
    title('Критерий Михайлова (форма 2)');
    xlabel('Частота, рад/с'); legend boxoff; % без рамки
   legend('U(w)','V(w)','location','best') % выбрать место
else
     u=ut;v=vt;ur=urt;vr=vrt;  % только вывод полиномов и частот
end
```

Результатом обращения к этой функции с указанием выходных аргументов (слева от знака равенства) будет вид четной и нечетной функций, представленных коэффициентами, и их корни (частоты)

```
>> den=[1 2 3 4 5]; [u,v,ur,vr]=uv(den)
u =
   1   0  -3   0   5
```

```
v =
   -2     0     4     0
ur =
    []
vr =
        0
   1.4142
```

откуда видно, что частот пересечения графиков с осями только две, это 0 и 1.4142 (при построении частотных характеристик мнимые, комплексные и отрицательные частоты должны отбрасываться, что и делает разработанная функция), причем обе принадлежат нечетной функции *V*(ω). Средствами MATLAB можно придать стандартный вид четной и нечетной функциям, используя символьную переменную 'w'.

```
>> u=poly2sym(u,'w')
u =
w^4-3*w^2+5
>> v=poly2sym(v,'w')
v =
-2*w^3+4*w
```

График (рисунок 2), построенный с помощью этой же функции, но при обращении без выходных аргументов, подтверждает сделанные выводы – система неустойчива, так как отсутствует поочередное пересечение характеристиками оси частот (чередование корней).

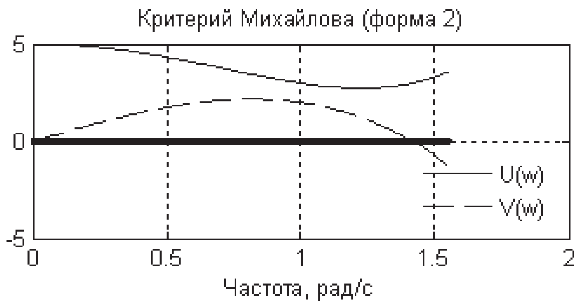

Рисунок 2

Рассмотрим пример D-разбиения по двум параметрам в MATLAB.

Для построения методом D-разбиения области устойчивости в пространстве параметров *T* и *k* системы четвертого порядка с характеристическим уравнением $D(s)=Ts^4 + (2T + 1)s^3 + (2T+2)s^2 + 2s + k = 0$ воспользуемся достаточным условием $\Delta_3 = a_3(a_1a_2 - a_0a_3) - a_1^2a_4 \geq 0$ критерия Гурвица и сформируем по нему функцию $Z(T, k)$, подставив значения параметров. При вычислениях заменяем T на x и k на y, чтобы не объявлять символьные переменные, поскольку переменные x, y и z для двумерных функций MATLAB принимает по умолчанию. Задав пределы изменения постоянной времени $0 \leq T \leq 2$ и коэффициента усиления $0 \leq k \leq 4$ с шагом 0.1, построили с помощью функции contourf() график с залитой зеленым цветом областью устойчивости $D(0)$ (рисунок 3).

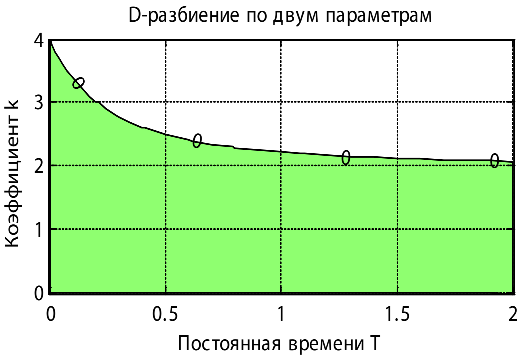

Рисунок 3

>> [x,y]=meshgrid(0:0.1:2,0:0.1:4); % сетка значений
>> z=2*((2*x+1).*(2*x+2)-x*2)-(2*x+1).^2.*y; % функция

% построить контурный график и залить область, выделенную на уровне 0-0

>> [c,h]=contourf(x,y,z,[0 0]);
>> set(h,'showtext','on'); % включить печать значений уровней
>> title('D-разбиение по двум параметрам'); grid;
>> xlabel('Постоянная времени T'); ylabel('Коэффициент k')

График не учитывает требований, вытекающих из необходимых условий устойчивости $a_i > 0$, а именно: $T > 0$ – вытекает из требования $a_0 = T > 0$, и $k > 0$ – обусловлено требованием $a_4 = k > 0$. Эти условия учтены выбором пределов изменения параметров, начиная от нуля.

В качестве примера проблем третьей группы укажем обработку программой транспортной задержки при использовании конструктора структурных схем Simulink. Звено чистого запаздывания (Transport Delay) моделируется в Simulink приближенно разложением в ряд Паде. При составлении схемы в свойствах типового блока необходимо указать время задержки τ Time Delay (секунд) и обязательно порядок разложения Pade Order. При нулевом порядке, который выставлен по умолчанию, задержка проявляется на экране приборов (Scope), но у линеаризированной модели коэффициент передачи звена будет равен единице и задержка отсутствует, независимо от того, какая величина ее была установлена. Это зачастую не выявляется студентами и приводит к ошибочному построению кривой переходного процесса или амплитудно-фазовой частотной характеристики.

Используя разложение Паде, MATLAB воспроизводит задержку в виде провала или отрезка с колебаниями в начале переходного процесса, что не соответствует реальному виду кривой. Во избежание этого можно использовать график с экрана Scope, открыв в нем кнопкой Parameters окно с панелью Data history, установив флажок в опции Save data to workspace и формат Array вывода данных в переменную ScopeData. График строим командой plot(ScopeData(:,1), ScopeData(:,2)), grid, к его недостаткам следует отнести неравномерный шаг по времени и отсутствие автоматического определения показателей качества.

Имеются проблемы и при работе с символьной математикой. Не всегда выражение, возвращаемое функцией преобразования, приемлемо для дальнейшего использования, например, при достаточно простом изображении получается маловразумительный оригинал

```
>> w=5/(s^3+2*s^2+3*s+4);
>> temp=ilaplace(w)
temp =
1/4*sum(_alpha*(-1+_alpha)*exp(_alpha*t),...
_alpha = RootOf(_Z^3+2*_Z^2+3*_Z+4))
```

Частично в этом случае помогают функция перевода в цифровой формат vpa() и ограничение числа разрядов вычисляемых чисел от 32, используемых по умолчанию, до меньшего значения, например, 4. Разовое применение vpa() к результату

```
>> temp=vpa(temp)
temp =
1.0938014797678091851515394567267*
exp(-1.6506291914393882188808009674262*t)- ...
```

выводит выражение, которое не приводится здесь целиком в связи с нецелесообразностью полного воспроизведения – его в таком виде сложно даже прочитать. Попробуем преобразовать выражение в уменьшенный цифровой формат, который можно задать и отдельной функцией digits(D), где D – число обрабатываемых разрядов. Однако, поскольку влияние этой команды на последующие операции нежелательно, лучше воспользоваться возможностью временного изменения числа разрядов в самой функции vpa(). Двойное преобразование функцией vpa()

обеспечивает эффект уменьшения разрядной сетки для всех чисел, входящих в преобразуемое выражение.

```
>> temp=vpa(temp,4)
temp =
1.094*exp(-1.651*t)-1.094*exp(-.1747*t)*cos(1.547*t)+1.044*exp(-.1747*t)*sin(1.547*t) +
+.2500*i*(2.087*exp(-.1747*t)*cos(1.547*t)+2.188*exp(-.1747*t)*sin(1.547*t)) +
+ .2500*i*(-2.087*exp(-.1747*t)*cos(1.547*t)-2.188*exp(-.1747*t)*sin(1.547*t))
```

Однако и это выражение усложнено, т. к. содержит семь составляющих переходного процесса, включая даже мнимые, в отличие от трех составляющих, возвращаемых численной функцией residue() и соответствующих оригиналу $1.0938e^{-1.6506t} - 1.0938e^{-0.1747t}cos(1.5469t) + 1.0436e^{-0.1747t}sin(1.5469t)$.

```
>> [r,p,k]=residue(5,[1 2 3 4])
r =
   1.0938
  -0.5469 - 0.5218i
  -0.5469 + 0.5218i
p =

  -1.6506
  -0.1747 + 1.5469i
  -0.1747 - 1.5469i
k =
     []
```

Еще пример. Приводимое ниже выражение с гиперболическим синусом менее информативно, чем возвращаемый функцией residue() в числовом варианте результат *-0.5exp(-3t) + 0.5exp(-t)*, хотя и полностью ему адекватно. Вдобавок оно включает корень -2, который на самом деле сократился с исключением соответствующей моды из переходного процесса, что может ввести пользователя в заблуждение.

```
>> w=(s+2)/(s^3+6*s^2+11*s+6);
>> temp=ilaplace(w)
temp =
exp(-2*t)*sinh(t)
```

Таким образом, прежде, чем использовать продукт MATLAB в учебном процессе, преподаватель должен соотнести результаты моделирования с теоретическими положениями, которые он будет излагать в ходе обучения, и разработать необходимые дополнения, приёмы работы с программой, либо приготовить объяснения на возможные вопросы обучаемых.

ЛИТЕРАТУРА

1. Дьяконов В., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. – СПб.: Питер, 2002.

2. Бороденко В.А. Оценка устойчивости линейной системы по критерию Рауса в среде MatLAB 5.x // Наука и техника Казахстана. – 2001. – №2. – С.180-185.

3. Бороденко В.А. Практический курс теории линейных систем автоматического регулирования. – Павлодар : Кереку, 2007. – 260 с.

Түйіндеме

Проблемаларды қарастырған, оқу процесінде бағдарламалық өнімде қолдануға қиналуы.

Resume

The problems, complicating use of software product MATLAB in educational process, are considered.

УДК 621.316.925

ЭВОЛЮЦИЯ ПРИНЦИПОВ ОЦЕНКИ НАДЕЖНОСТИ СРЕДСТВ АВТОМАТИКИ ЭНЕРГОСИСТЕМ

В.А. Бороденко

Павлодарский государственный университет им. С. Торайгырова

Устройства релейной защиты и автоматики (РЗА) электроэнергетических систем характеризуются эффективностью функционирования, под которой следует понимать способность выполнять предельное число функций, каждую с предельным эффектом [1]. При этом эффективность функционирования включает понятия технического совершенства и надежности, а техническое совершенство в свою очередь подразделяется на селективность и устойчивость функционирования. Отметим сразу, что если для устройств РЗ подход к оценке технического совершенства (селективности, чувствительности) представляется достаточно ясным – повреждение в защищаемой зоне, повреждение вне защищаемой зоны – то для устройств системной автоматики аналогичные показатели отсутствуют.

Простой перечень свойств, характеризующих надежность изделия (безотказность, долговечность, ремонтопригодность), говорит о том, что все они связаны не с идеей (способом) распознавания аварийного состояния объекта электроснабжения, заложенной в устройство, а с конструктивными особенностями компонентов и связей между ними. Между тем, объем