НЕКОТОРЫЕ ОСОБЕННОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ МЕТОДОВ ЭКСПЕРИМЕНТАЛЬНОЙ ИДЕНТИФИКАЦИИ ЛИНЕЙНЫХ ПРОЦЕССОВ

А.П. Полищук, С.А. Семериков
г. Кривой Рог, Криворожский государственный педагогический университет

На примере динамических объектов, описываемых линейными дифференциальными уравнениями с постоянными коэффициентами, рассматриваются типовые трудности реализации последовательных алгоритмов коррекции параметров математической модели по экспериментальным данным и способы их преодоления.

Математическая модель линейной системы может быть представлена в одной из форм, позволяющих решить задачу анализа – вычисление реакции процесса на заданное управление произвольного типа при нулевых или ненулевых начальных условиях. Это может быть дифференциальное уравнение порядка n, передаточная функция в форме преобразования Лапласа, импульсная реакция процесса, заданная вектором своих значений на конечной последовательности значений времени, частотные характеристики процесса и др.

Неклассическая постановка задачи идентификации математической модели.

В традиционной постановке задача вычисления коэффициентов a_i модели вида

$$\sum_{i=0}^{n} a_i f^{(i)}(t) = u(t), \qquad (1)$$

где i – порядок производной;

$u(t)$ – входная управляющая (или возмущающая) функция;

$f(t)$ – реакция процесса на $u(t)$;

t – время,

требует для своего решения знания всех составляющих вектора состояния процесса; другими словами, необходимо измерение значений функции и всех входящих в уравнение ее производных в последовательные моменты времени. Если n – порядок

уравнения и N – количество отсчетов, то решение системы $N–n>n$ линейных уравнений по методу наименьших квадратов (в предположении наличия погрешностей измерений) дает вектор коэффициентов моделирующего уравнения. Но в реальных экспериментах на реальных процессах измерение производных реакции объекта со сколь-нибудь приемлемой точностью невозможно и в процедурах идентификации приходится оперировать только векторами значений управления и реакции на конечном интервале времени. Методы численного дифференцирования или предварительной аналитической аппроксимации с последующим многократным дифференцированием могут обсуждаться, но не могут эффективно использоваться из-за неприемлемо больших погрешностей вычисления производных, неузнаваемо искажающих результат идентификации.

О методе экспоненциальной аппроксимации.

И все же метод аналитических замен может эффективно использоваться для решения задач идентификации линейных процессов при условии, что порядок уравнения нам известен, переходом к разностной форме уравнения (1) [1]. Так как решение уравнения (1) предполагается в форме

$$f(t) = \sum_{i=0}^{n} A_i e^{p_i t} , \qquad (2)$$

справедливой для множества равноотстоящих значений t_j (j=1, 2, …, N), то как стоящие под знаком суммы экспоненты, так и сама функция $f(t)$ должны удовлетворять разностному уравнению с постоянными коэффициентами, имеющему характеристические корни $\rho = e^{p_i}$, вида:

$$\sum_{j=0}^{n} c_j f(t_i + j\delta t) = 0 , \qquad (3)$$

где t_i – время в i-м отсчете (i=0, 1, … , $N–n$);

j – сдвиг по последовательности отсчетов.

Получаем избыточную систему линейных алгебраических уравнений, решение которой методом наименьших квадратов дает вектор коэффициентов c_i характеристического уравнения

$$\sum_{j=0}^{n} c_j f(t_i + j\delta t) = 0 , \qquad (4)$$

а его корни дают нам значения ρ_i, логарифмы которых представляют собой коэффициенты в показателях экспонент p_i для (2). Коэффициенты A_i в (2) и решение в целом теперь легко отыскиваются решением составленной на базе этого уравнения системы. При необходимости получить коэффициенты левой части уравнения (1) достаточно перемножить двучлены $(p - p_i^*)$.

Но если порядок уравнения неизвестен, приходится переходить к методу проб, задаваясь порядком, идентифицируя модель для этого порядка, вычисляя реакцию этой модели на заданное управление и оценивая степень близости реакций модели и объекта – например, двигаясь от 2-го порядка к 3-му, затем к 4-му и т.д. Из всех попыток наилучшей придется считать ту, которая даст наилучшее совпадение реакций.

Об оценке близости функций, заданных векторами своих значений на конечном множестве значений аргумента.

Теория идентификации лукаво относит выбор критерия близости к прикладным областям – прежде всего в силу трудности дать общие обоснованные рекомендации – и в распоряжении разработчика по-прежнему в основном только Гауссовы оценки по минимуму суммы квадратов отклонений и Чебышевский минимакс. Привлекательность Гауссового критерия очевидна – мы получаем унимодальную функцию оцениваемых параметров модели с фиксированным нулевым значением абсолютного минимума (при отсутствии погрешностей), что дает возможность ранжировать результаты попыток при варьировании структуры модели. Но специалисты по идентификации неоднократно отмечали (например, [2]) возможность получения неудовлетворительных результатов при использовании этого критерия. Одна из причин этого, по нашему мнению, в следующем. Критерий Гаусса представляет собой квадрат модуля разностного вектора (в задачах идентификации – векторов значений реакций объекта и модели на одно и то же управление). При изменении вектора параметров модели вектор ее реакции на управление изменяет и направление и модуль. Нас интересует только направление (модуль вектора легко доопределяется нормированием по одному из идентифицируемых параметров). Но при сближении векторов по направлению (то есть при правильном шаге коррекции параметров модели) разностный вектор может не уменьшиться, а увели-

читься по модулю за счет увеличения модуля вектора реакции модели, и результат попытки будет оценен как отрицательный. Блокирование этой ситуации осуществляется просто фиксированием модуля разностного вектора, например, нормированием по модулю с последующим масштабированием модулем вектора реакции объекта. Мы получаем 2 вектора, отличающиеся только направлением и каждая попытка варьирования параметров модели получает адекватную оценку. Использование этого приема избавило нас от неоднозначности оценки результатов при исследовании ряда поисковых алгоритмов идентификации.

Фрагмент подпрограммы нормирования критерия близости по предложенному алгоритму (при наличии классов векторов с перегруженными операциями векторной арифметики) на языке C++ выглядит прозрачно (~ – операция нормирования вектора по модулю):

```
/*Приведем вектор реакции модели к модулю вектора реакции
объекта*/
VecModel=(~VecModel)*ModVecExper;
//Вычислим вектор разности экспериментального
//и модельного векторов решения
dvector res=VecExper-VecModel;
//Максимально возможный модуль разности равен
//удвоенному модулю экспериментального вектора
double mres=!res;
//Возвращаемое значение критерия – в пределах 0.0–1.0
return mres*(1.0/(2.0*ModVecExper));
```

О вычислении комплексных корней полиномов.

Эта процедура неизбежна в реализации алгоритмов идентификации и параметризованный класс полиномов с возможностями комплексной полиномиальной арифметики должен быть в библиотеке специалиста в области математического моделирования. Обычно используют численные методы, основанные на выделении в многочлене квадратичных трехчленов, вычислении их коэффициентов и использовании двумерного аналога метода Ньютона (метод Берстоу [1] и др.). По нашему мнению, ничуть не хуже классический метод Ньютона для полинома с комплексными коэффициентами. Мы приводим простейший рекурсивный вариант подпрограммы из нашей библиотеки [5, 6], который испытан и надежно работает для полиномов до 20-й степени (здесь cvector, cpolynom – комплексные вектор и полином, derive –

функция дифференцирования полиномов, арифметические операции перегружены в классе полиномов):

```
cvector newton(cpolynom p)
{
  double t=1,eps=1e-5; cvector result=p.getm()-1,tv;
  /*если вектор-результат одномерный, то имеем дело с
полиномом первой степени*.
  if(result==tv) { result[0]=-p[0]/p[1]; return result; }
  //начальное приближение к корню
  complex z(1.0,1.0);
  do{
    complex dz(0.0,0.0); //real(dz)==0 && imag(dz)==0;
    dz=derive(p,1)(z);
    z-=p(z)/dz; //модифицируем приближение
  }while(abs(p(z))>eps); /*повторяем до достижения задан-
ной точности*/
  /*В этом цикле находится только один корень z. Для
нахождения остальных делим исходный полином на x-z, пони-
жая тем самым степень на единицу, и находим ещё один ко-
рень, и т.д. до первой степени */
  result[0]=z; cpolynom z1=2.0; z1[0]=-z,z1[1]=1.0;
  cvector more=newton(p/z1); //рекурсивный вызов
  for(long i=1;i<result.getm();i++) result[i]=more[i-1];
  return result; //возвращаем вектор результата
}
```

В этом алгоритме начальное (в общем случае – произвольное) приближение обязательно должно задаваться с ненулевой мнимой частью, иначе поиск будет осуществляться только в вещественной области. Для блокирования случаев близкой к нулю 1-й производной можно использовать модифицированный алгоритм с переходом к вычислению высших производных (до первой ненулевой).

О выборе начального приближения вектора параметров модели в поисковых и рекуррентных алгоритмах последовательного уточнения параметров модели.

Дрейф характеристик реальных объектов в процессе функционирования из-за износа, старения, расходования рабочей среды и пр. неизбежен, и системы идентификации должны отслеживать этот дрейф по непрерывно обновляемым выборкам измерений с помощью достаточно быстродействующих алгоритмов, обеспечивающих получение адекватного результата при пренебрежимо малом изменении параметров объекта за время идентификации. На первый план выдвигается задача минимизации

объема вычислений для получения результата. При этом возникает потребность в 2-х типах алгоритмов: для быстрого вхождения в субоптимальную область в начальной фазе при сравнительно невысоких требованиях к точности полученного результата и для «доводки» параметров модели до максимально возможного удовлетворения требований к точности воспроизведения поведения объекта.

Наиболее привлекательным для выполнения первой фазы идентификации, особенно для систем высокого порядка, выглядит алгоритм последовательного симплексного поиска экстремума в многомерном пространстве [3]. Сущность алгоритма проста. В пространстве идентифицируемых параметров модели вычисляются координаты вершин регулярного симплекса (правильного многогранника) и для каждой вершины вычисляются значения критерия близости реакций модели и объекта. Наихудшая вершина с помощью несложных геометрических вычислений отражается относительно противоположной грани, находится новая наихудшая и весь процесс повторяется. Независимо от размерности симплекса (пространства поиска, порядка модели) на каждом шаге уточнения наиболее трудоемкая задача анализа выполняется однократно, в отличие, например, от классического метода градиента, в котором задачу анализа приходится решать отдельно на приращениях каждого параметра модели для оценки составляющих вектора градиента. Для иллюстрации ниже приведен упрощенный фрагмент основного цикла симплекс-поиска:

```
//Создаем начальный симплекс с центром в начале коорд.
//и смещаем его в центр области поиска
Simplex ss(Crit,r,off1,l);
do{//Получаем упорядоч. по качеству вектор номеров вершин
  qual=ss.Qual(); rv=qual[0];//Эту будем отражать
  bValPrev=ss.vertex[rv][r];
  //Отражаем худшую вершину
  ss.ReflSimpl(rv); qv=ss.GetCritVector();
  //Если критерий удовлетворителен – завершаем работу
  if(qv[rv]<=eps) break;
  //Если отражение не улучшило критерий в вершине –
  //перешагнули за оптимум и надо уменьшать размер
  //симплекса
  if(qv[rv]>bValPrev)
  { //Центр симплекса – в лучшую вершину
    for(i=0;i<r;i++) off1[i]=ss.vertex[qual[r]][i];
    //Уменьшаем длину ребра
```

```
    if(l>eps)l*=k;
    //Перестраиваем симплекс
    ss.SimplexBuilder(Crit,r,off1,l);
  }
}while(qv[rv]>eps);
```

Очевидно, для начала поиска необходимо определить стартовое положение центра симплекса и здесь нас поджидают неприятности, специфичные для задач идентификации динамических объектов. Исследуемые объекты обычно не бывают без потерь и в силу этого для них характерно затухание переходных процессов, вызванных внешними воздействиями. Но неудачный выбор стартового вектора параметров модели может привести к ее неустойчивости и неограниченному росту значений $f_{mod}(t)$ – в программе идентификации это означает неизбежное переполнение разрядной сетки и аварийное завершение. Поэтому выбор центра области поиска и ее границ должен базироваться либо на априорных сведениях о моделируемом процессе, либо установление таких границ пробными прогонками задач анализа.

Об особенностях идентификации импульсной реакции линейных объектов.

Для решения задачи анализа (имитационного моделирования) линейного объекта не обязательно знание коэффициентов дифференциального уравнения, достаточно определения его реакции на воздействие типа единичного импульса. Активный эксперимент с нанесением такого возмущения на реальном объекте провести затруднительно. В [4] описан предложенный в 1967-1969 г.г. Nagumo J. и Noda A. метод последовательного обучения модели в реальном масштабе времени, предназначенный для вычисления вектора значений импульсной реакции процесса по его реакции на произвольное воздействие и последующим использованием этого вектора для вычисления оценки дискретного варианта интеграла свертки в задачах анализа:

$$f_i = \sum_{i=1}^{N} g_i u_i \,. \qquad (5)$$

Алгоритм решения этой задачи состоит в следующем. Начальная оценка импульсной реакции вычисляется, например, по вектору параметров модели, выбранному с центре допустимой области поиска. По этой оценке вычисляется через свертку (5) реакция модели на первое измерение значения функции

управления и вектор импульсной реакции корректируется в направлении вектора управления, модуль корректирующего вектора берется равным модулю разностного вектора реакций объекта и модели. В начале процесса идентификации размерности векторов импульсной реакции, реакций объекта на управление и вектора управления растут на каждом шаге до достижения заданного интервала времени, на котором желательно иметь оценку импульсной реакции, а затем эта размерность может оставаться постоянной при скольжении вдоль последовательности измерений (при получении нового измерения самое старое отбрасывается). Иллюстрацией изложенному может служить следующий фрагмент C++ программы:

```
for(int t=1;t<T;t++)
{/*Формируем очередную составляющую временного вектора
управления*/
  VecCtrlT[t]=VecCtrl[t];
  for(int i=1;i<=t;i++) {
    VecModel[i]=0;
    for(int k=0;k<=i;k++) //Цикл вычисления свертки
      VecModel[i]+=VecImpulsM[i-k]*VecCtrlT[i]*Tstep;
  }
  /*Цикл кратного повторения вычисления коррекции на од-
них и тех же данных*/
  for(int rpt=0;rpt<5*Rngl;rpt++)
  {//Расчет коррекции импульсной функции на момент t
    for(int j=0;j<=t;j++)
      deltah[j]=(VecExper[j]-VecModel[j])*(~VecCtrlT)[j];
    //Корекция импульсной функции
    VecImpulsM+=deltah;
  }
}
```

Основная особенность – невозможность получения удовлетворительной точности коррекции фрагмента импульсной реакции за один проход; для одних и тех же данных вычисления необходимо повторять, причем количество необходимых повторений растет с ростом порядка уравнения процесса и с удалением от его начала. Сходимость алгоритма к оптимальным оценкам импульсной реакции довольно медленная, объем вычислений значителен и время обучения модели для объекта 2-го порядка при «дальнем старте» на 100 измерениях составляет на ПК Celeron – 366 с RAM 32mb около 2-х минут, но для уравнения 3-го порядка на 500 измерениях возрастает до 40 минут. Поэтому в

начальной фазе идентификации метод лучше применять в комбинации с последовательным симплекс – поиском для попадания в субоптимальную область, а затем для более точного приближения использовать последовательное обучение.

О выборе шага по времени между соседними измерениями.

Увеличение частоты измерений естественно приводит к повышению точности модели, но обходится очень дорого в объеме вычислительной работы. В соответствии с известной теоремой Котельникова эта частота должна быть вдвое выше самой высокой из «существенных» частот идентифицируемого процесса, по результатам наших экспериментов с объектами второго порядка «срыв» до полной неработоспособности алгоритма происходит при частотах опроса ниже собственной частоты процесса.

О влиянии погрешностей вычисления интеграла свертки.

Обычно при дискретном вычислении оценки свертки 2-х функций мы используем их ступенчатую или прямоугольную аппроксимацию – измеренное значение функции считаем постоянным до следующего измерения. Если процесс идентификации начинать после накопления хотя бы 2-х измерений, можно использовать для текущего измерения среднее между ним и предыдущим. Существуют методы повышения точности вычисления свертки за счет аппроксимации функций последовательностью треугольных импульсов. Использование этих приемов может существенно повысить результативность процессов идентификации.

Литература:

1. Хемминг Р.В. Численные методы для научных работников и инженеров. – М.: Наука, 1968.
2. Дейч А.М. Методы идентификации динамических объектов. – М.: Энергия, 1979.
3. Дамбраускас А.П. Симплексный поиск. – М.: Энергия, 1979.
4. Гроп Д. Методы идентификации систем. – М.: Мир, 1979.
5. Полищук А.П., Семериков С.А. Методы вычислений в классах языка С++: Учебное пособие. – Кривой Рог: Издательский отдел КГПИ, 1999.
6. Полищук А.П., Семериков С.А. Автоматика: Учебное пособие. – Кривой Рог: Издательский отдел КГПИ, 1999.