ТЕХНОЛОГИЯ МАШИНОСТРОЕНИЯ

УДК 681.3:519.71

А.В. Степанов

РЕШЕНИЕ УНИВЕРСАЛЬНОЙ СТРУКТУРНОЙ СИСТЕМЫ проф. Л.Т. ДВОРНИКОВА

Одним из этапов создания механизма, работающего в соответствии с заданными входными и выходными условиями, является синтез его структуры. Набор необходимых для этого исходных данных, чаще всего, представлен несколькими параметрами: требуемой подвижностью механизма, числом наложенных на систему связей, количеством звеньев, допустимыми классами применяемых кинематических пар, а также ограничением на максимальную сложность звеньев.

Выполнение выходных условий может быть обеспечено правильным подбором числа звеньев той или иной сложности, образующих механизм, а также способом и порядком соединения их между собой.

Для определения числа звеньев требуемой сложности, не превышающей заданной, и числа кинематических пар, разрешенных к применению видов, используют различного рода зависимости.

Разработкой теории структур механизмов активно занимался проф. Дворников Л.Т. Он объединил несколько зависимостей, связывающих числа звеньев различной сложности с числом кинематических пар того или иного вида при заданных условиях, и получил изящную систему уравнений, названную им универсальной структурной системой [1].

$$\begin{cases} \sum\limits_{k=1}^{5} p_k \cdot h(k-m) = \tau + \sum\limits_{i=1}^{\tau-1} i \cdot n_i \\ n = 1 + n_{\tau-1} + \ldots + n_i + \ldots + n_2 + n_1 \\ W = (6-m)n - \sum\limits_{k=1}^{5} (k-m) \cdot h(k-m) \end{cases} \quad (1)$$

Здесь: n – общее число звеньев;

W – подвижность системы;

m – число наложенных на систему связей;

k – ограничение на классы применяемых кинематических пар;

p_k – число кинематических пар того или иного класса;

τ – количество вершин наиболее сложного звена;

n_1, n_2, ..., n_i – числа звеньев различной сложности;

h – функция вида:

$$h(k-m) = \begin{cases} 1, & m < k, \\ 0, & m \le k. \end{cases} \quad (2)$$

Скрупулезный анализ системы показывает, что это – необычная система, не приводимая к “классическому” виду. Количество одночленов, входящих в левые и правые части уравнений, и число неизвестных может меняться от расчета к расчету. Чаще всего, число неизвестных превышает число уравнений системы. Значения искомых неизвестных могут быть только целыми числами.

При небольшом числе звеньев решение системы получить достаточно просто, но с увеличением общего числа звеньев механизма её решение становится далеко не тривиальным.

Ясно, что для решения системы аналитические методы не могут быть применены. В отсутствии аналитических методов решения задачи применяют поисковые методы, в частности, методы поисковой оптимизации. Наиболее разработанными методами поисковой оптимизации являются методы деформируемых конфигураций, развивающие идеи симплексной оптимизации [2]. Однако, методы поисковой оптимизации предполагают, что качество очередного шага поисковой процедуры оценивается некоторым числовым показателем - целевой функцией, являющейся аналоговой величиной. В данной же системе нет ни одной непрерывной переменной. Все параметры, входящие в систему, дискретного типа. Разработка критерия, оценивающего правильность направления движения к области искомых решений и представляющего собой непрерывную в диапазоне изменения настраиваемых параметров функцию, не представляется возможной.

Отсюда следует вывод, что направленный поиск с использованием целевой функции не может быть организован.

Теоретически возможным является метод простого перебора всех возможных комбинаций чисел звеньев той или иной сложности и кинематических пар разрешенных к применению видов. Он также как и любой другой, требует разработки компьютерного алгоритма, на основе которого может быть создано действующее приложение.

Однако, попытки реализации простого перебора возможных комбинаций значений неизвестных в нескольких вложенных циклах не дают ожидаемых результатов, поскольку это требует перебора

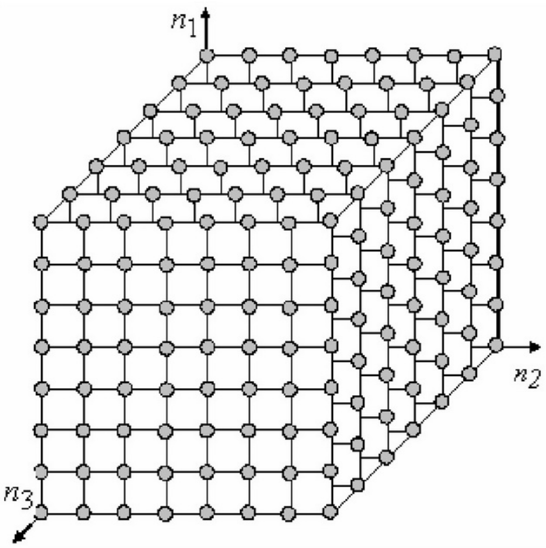

Рис.1. Дискретное пространство (гиперкуб) возможных вариантов комбинаций звеньев

невероятно огромного количества вариантов. Для поиска целочисленных решений этой системы был разработан оригинальный алгоритм, логика работы которого описывается далее.

Рассмотрим первое уравнение универсальной структурной системы. Искомые неизвестные $n_{\tau-1}, \cdots n_2, n_1$ в правой части могут принимать значения в диапазоне от нуля до числа n. Количество неизвестных в ней равно τ - 1 . Отсюда следует, что перебор вариантов наборов звеньев различной сложности, не превышающей τ, должен осуществляться внутри дискретного пространства - гиперкуба размерности τ - 1 и длиной ребра, равной n единиц. Поскольку количество звеньев любой сложности – есть целое число, виртуальный гиперкуб представляет собой полую конструкцию, состоящую из узлов. Для иллюстрации приведенных здесь рассуждений на рисунке 1 изображен гиперкуб при τ =3 и n=7. Передняя грань гиперкуба и его сечения вдоль координаты n_3 выполнены непрозрачными.

Не все узлы гиперкуба должны участвовать в дальнейшем поиске целочисленных решений системы. Те узлы, в которых сумма значений искомых переменных не равна общему количеству звеньев, должны быть исключены из рассмотрения.

Очередной набор чисел звеньев различной сложности, подставленный в правую часть первого уравнения, дает общее количество кинематических пар. Этому числу будет соответствовать некоторое количество сочетаний числа кинематических пар разрешенных видов. Эти сочетания располагаются в дискретном пространстве (гиперкубе), размерность которого равна числу видов кинематических пар. Длина ребра этого виртуального гиперкуба равна общему числу кинематических пар, полученному из первого уравнения системы.

Таким образом, задача поиска целочисленных решений универсальной структурной системы сводится к организации движения в двух дискретных пространствах и проверке тождественности третьего уравнения системы на каждом шаге.

При обращении в тождество третьего уравнения системы наборы звеньев различной сложности и кинематических пар различных видов являются одним из вариантов решения.

Как обеспечить движение в первом из дискретных пространств и гарантированно полный перебор вариантов наборов звеньев различной сложности, не превосходящей τ?

Для определения стратегии этого процесса вновь рассмотрим первое уравнение системы. Правая часть его представляет собой целое число. Количество слагаемых в ней всегда равно значению τ. Значение любого из сомножителей n_i находится в диапазоне от нуля до n.

Известно, что любое число в конкретной системе счисления может быть записано в виде суммы его разрядов:

$$N = \sum_{i=1}^{k} c_i \cdot b_i, \qquad (3)$$

где N – рассматриваемое число;

k – количество разрядов числа;

c_i – цифра i –го разряда;

b_i – вес i –го разряда.

Таким образом, правая часть первого уравнения системы (1) может быть представлена в виде виртуального счетчика, содержащего τ разрядов и работающего в системе счисления с основанием $n + 1$, поскольку в каждом из разрядов может быть записано любое число от нуля до n. В такой интерпретации каждый одночлен правой части уравнения соответствует одному разряду счетчика, а содержимое каждого из его разрядов есть число из диапазона от нуля до n - то есть n_i .

Для исключения проблем, связанных с переводом чисел из одной системы счисления в другую, представляется полезным отказаться от применения “чистой” системы счисления и принять к использованию смешанную систему счисления. Поскольку значение $\boldsymbol{n}$ в данной задаче меняется от решения к решению логично использовать десятично- n -ичную систему счисления.

Добавляя единицу в младший разряд счетчика можно обеспечить гарантированно полный перебор всех возможных вариантов наборов звеньев. В связи с тем, что в каждом из разрядов счетчика может быть записано целое число в диапазоне от нуля до общего числа звеньев, количество его состояний значительно превышает количество возможных “реальных” наборов звеньев.

Для организации “решета” может быть использовано второе уравнение системы. Варианты наборов, для которых сумма числа звеньев, зафиксированных в счетчике, не равна их общему числу, должны быть отброшены. К дальнейшему анализу принимаются те наборы звеньев различной сложности, для которых сумма числа звеньев n_i равна их общему числу - n .

Если на множество состояний счетчика наложить ограничение, что сумма цифр (чисел) его разрядов согласно второму уравнению системы, равна n, то такое виртуальное решето отсеет все неприемлемые варианты наборов звеньев.

Иными словами, если сумму значений разрядов счетчика обозначить как *sum1*, то должно выполняться равенство:

$$sum1 = 1 + n_{\tau-1} + ... + n_i + ... + n_2 + n_1 = n. \quad (4)$$

Однако не все отобранные варианты наборов звеньев могут являться решениями системы. Из этого многообразия вариантов должны быть отобраны только те, которые обеспечивают требуемую подвижность кинематической цепи, которая, в свою очередь зависит от состава применяемых кинематических пар. Нужны дополнительные условия, выполняя которые мы получим варианты

```
no:=StrToInt(pl3ed1.Text);
tau:=StrToInt(pl3CB1.Text);
W:=StrToInt(pl3CB2.Text);
m:=StrToInt(pl3CB3.Text);
kmin:=StrToInt(pl3CB4.Text);
// преобразовали строковые данные в целые числа
// делаем видимыми панели pl2 и pl4
pl2.Visible:=True;
pl4.Visible:=True;
kolvar:=0; // обнулили счетчик количества вариантов
pl4ListBox1.Clear; // очистили окно списка вариантов
for i:=1 to tau do ni[i]:=0;
ni[1]:=no-1;
 repeat // внешний цикл перебора комбинаций звеньев
  sum1:=1;
  for i:=1 to tau-1 do sum1:=sum1+ni[i];
  if sum1<>no then goto m1;
   sum2:=tau; // поиск  общего числа кинематических пар
   for i:=1 to tau-1 do sum2:=sum2+i*ni[i];
   for i:=1 to 6 do pk[i]:=0; // обнуление массива pk
   pk[kmin]:=sum2;
   repeat // цикл по pk
   sum3:=0;
   for i:=1 to 5 do sum3:=sum3+pk[i];
   // продолжение анализа
   if sum3=sum2 then
    begin // последняя проверка
    sum4:=0;
    for i:=kmin to 5 do sum4:=sum4+(i-m)*pk[i];
    if ((6-m)*sum1-sum4)=W then
  begin // регистрация решения
  kolvar:=kolvar+1;
  stroka:='';
 pl4ListBox1.Items.Add('вариант'+IntToStr(kolvar));
  pl4ListBox1.Items.Add('n='+IntToStr(sum1));
 for i:=1 to tau-1 do
stroka:=stroka+'n['+IntToStr(i)+']='+IntToStr(ni[i])+' ';
 pl4ListBox1.Items.Add(stroka);
  stroka:='';
for i:=1 to 5 do
stroka:=stroka+'p['+IntToStr(i)+']='+IntToStr(pk[i])+' ';
 pl4ListBox1.Items.Add(stroka);
  end  // регистрация решения
    end; // последняя проверка
// изменение состояния счетчика числа кинематических пар
   pk[kmin]:=pk[kmin]+1;
   for i:=kmin to 5 do if pk[i]=sum2+1 then
    begin // переносы
    pk[i]:=0;
    pk[i+1]:=pk[i+1]+1;
    end; // переносы
   until pk[6]>0; // цикл по pk
m1: ni[1]:=ni[1]+1;
  for i:=1 to tau-1 do if ni[i]=no+1 then
   begin // переносы в счетчике звеньев
   ni[i]:=0;
   ni[i+1]:=ni[i+1]+1;
   end; // переносы в счетчике звеньев
 until ni[tau]>0;
// закончен внешний цикл перебора комбинаций звеньев
// все компоненты панели сделали невидимыми
 pl4lb1.Visible:=False;  pl4lb2.Visible:=False;
 pl4lb3.Visible:=False;  pl4lb4.Visible:=False;
 pl4lb5.Visible:=False;  pl4lb6.Visible:=False;
 pl4lb7.Visible:=False;  pl4lb8.Visible:=False;
 pl4ListBox1.Visible:=False;
// если решений нет сделали видимой первую метку
 if kolvar=0 then pl4lb1.Visible:=True else
  begin // решения есть
   pl4lb8.Visible:=True;
pl4lb2.Visible:=True;
   pl4lb3.Caption:=IntToStr(kolvar);
pl4lb3.Visible:=True;
   pl4lb4.Visible:=True;
pl4lb7.Visible:=True;
   if (kolvar>1) and (kolvar<5) then
   pl4lb5.Visible:=True;
   if kolvar>4 then pl4lb6.Visible:=True;
   pl4ListBox1.Visible:=True;
  end; // решения есть
```

Рис.2. Процедура поиска целочисленных решений с использованием описанного алгоритма

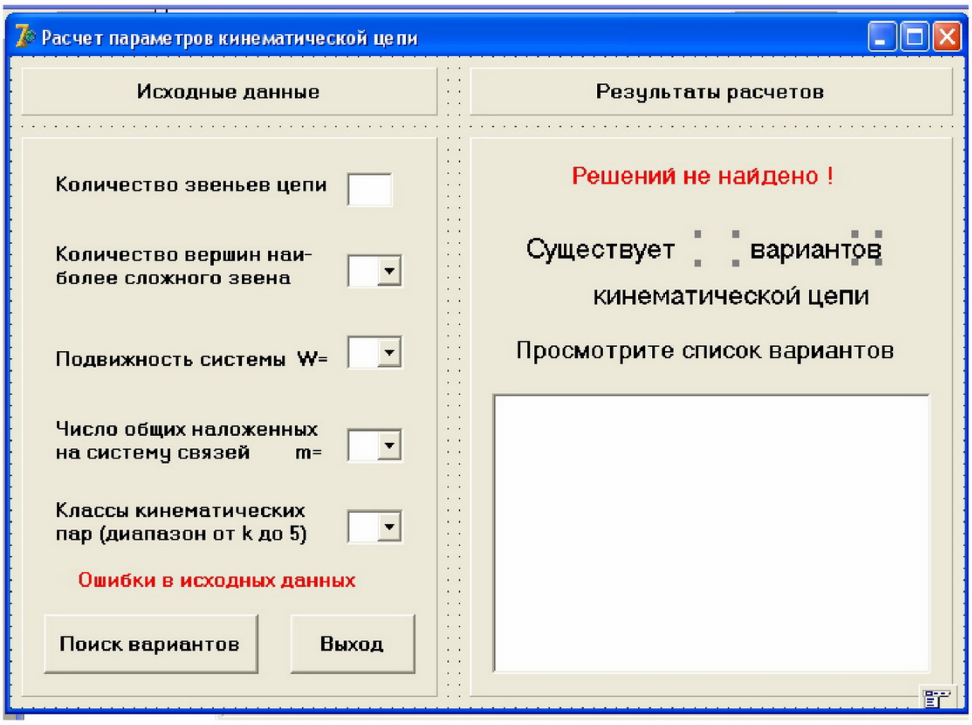

Рис. 3. Вид формы с набором компонентов.

кинематических цепей, отвечающих заданным требованиям. Наложим ограничения на генерируемые варианты звеньев различной сложности, приняв во внимание соотношения, записанные в форме третьего уравнения системы.

Умножив значение соответствующего разряда счетчика на его вес, и просуммировав произведения, получим число, значение которого представляет собой общее число кинематических пар цепи без учета того, сколько пар будет того класса или другого

$$sum2 = \tau + (\tau - 1)n_{\tau - 1} + \ldots + in_i + \ldots + 2n_2 + n_1. \quad (5)$$

Существует множество различных сочетаний кинематических пар с общим количеством *sum2.* Для перебора всех возможных сочетаний организуем еще один счетчик (второй). Каждый из разрядов этого счетчика может принимать значения от 0 до *sum*2. Количество разрядов в счетчике должно быть 6. Пять разрядов необходимы для хранения количеств пар соответствующих классов. Шестой разряд нужен для регистрации переполнения, которое будет свидетельствовать о том, что перебор всех возможных сочетаний закончен. При заданном m в каждом конкретном расчете будет использоваться $6 - m$ разрядов счетчика.

При каждом изменении состояния счетчика необходимо получать сумму его разрядов, равную общему числу кинематических пар

$$sum3 = \sum_{k=m+1}^{5} p_k \quad (6)$$

Полученное значение необходимо проверять на допустимость дополнительной проверкой на равенство:

$$sum3 = sum2 \quad (7)$$

Как только будет найдено сочетание кинематических пар, удовлетворяющее условию (7), организуем накопление еще одной суммы – *sum*4. В этот момент содержимое каждого разряда счетчика равно числу пар соответствующего класса.

$$sum4 = \sum_{k=m+1}^{5} (k - m)\, p_k \quad (8)$$

Если значение полученной суммы *sum*4 равно значению выражения $6 - m - W$, то это сочетание кинематических пар разрешенных классов вместе с комбинацией звеньев – есть решение системы.

Полученная комбинация звеньев и кинематических пар регистрируется и осуществляется поиск следующего возможного решения.

Добавляется единица в последний разряд счетчика до тех пор, пока сумма его разрядов не будет равна числу звеньев цепи. Затем осуществляются заново шаги, описанные выше. Это продолжается до тех пор, пока не произойдет переполнение первого счетчика. В этом случае зарегистрированы все варианты возможных целочисленных решений системы

Таким образом, описанный выше метод поиска возможных целочисленных решений универсальной структурной системы предполагает использование следующих технических устройств или программных модулей: генератора множества вариантов комбинаций звеньев, на основе счетчика, работающего по основанию $n + 1$; генератора сочетаний кинематических пар разрешенных классов (тоже счетчик, но с другим основанием), четырех сумматоров и нескольких блоков проверки условий.

На основе приведенного выше описания, был разработан соответствующий модуль в системе программирования Delphi (рис. 2).

Приложение имеет одну единственную форму, внешний вид которой представлен на рис. 3.

На разработанную программу получено свидетельство об официальной регистрации программ для ЭВМ за № 2006611506

СПИСОК ЛИТЕРАТУРЫ

1. *Дворников Л.Т.* Начала теории структуры механизмов. СибГГМА, Новокузнецк, 1994, -102 с., ил.
2. *Рыков А.С.* Поисковая оптимизация. Методы деформируемых конфигураций. – М.: Физматлит., 1993. – 216 с.
3. *Дворников Л.Т., Степанов А.В.* О компьютерном алгоритме решения задачи синтеза структур кинематических цепей // Материалы тринадцатой научно-практической конференции по проблемам механики и машиностроения. Новокузнецк, 2003, с.58-64

УДК 681.3: 621.01

А.В. Степанов

ВИРТУАЛИЗАЦИЯ В ЗАДАЧАХ КОМПЬЮТЕРНОГО СИНТЕЗА СТРУКТУР МЕХАНИЗМОВ

.

Виртуализация – это способ абстрагирования, при котором объекты реального мира, обладающие многочисленными параметрами и характеристиками, заменяются образами искусственных, как правило, воображаемых, объектов. Такие образы, в отличие от реальных объектов, называют виртуальными объектами. Использование виртуальных объектов, имеющих минимальное количество параметров, характеризующих их, в ряде случаев существенно упрощают работу исследователя над проблемой.

Виртуализация широко используется в компьютерных технологиях. В процессе выполнения задания на компьютере пользователь практически всегда вынужден использовать его периферийные устройства, с помощью которых он считывает данные с различных носителей и выдает результаты. Рядовой пользователь при этом работает не с реальными, а с виртуальными устройствами, что позволяет освободить его от знания технических особенностей реальных устройств. Виртуальное устройство имеет, чаще всего два параметра: имя и назначение. Для примера, печатающее устройство имеет имя - **prn** и назначение - **печать**. При необходимости вывода данных пользователь манипулирует его именем. При этом в качестве реального печатающего устройства может быть назначен не только принтер, но и любое другое вовсе не печатающее устройство компьютера. Перед исполнением компьютерной программы осуществляется процедура связывания виртуального устройства с реальным и ввод-вывод осуществляется на конкретном физическом устройстве. В случае выхода из строя реального устройства, эта связь может быть легко изменена путем переназначения, и ввод-вывод будет производиться уже на другом физическом устройстве.

Опосредовано виртуализация применяется практически во всех областях знаний. Теория механизмов не является здесь исключением. Сопоставляя реальный механизм с его структурной схемой, можно легко понять, где реальные звенья и кинематические пары, а где их виртуальные двойники. В этом случае реальный механизм является частью реального мира, а структурная схема её виртуальным отображением. Изображения звеньев и кинематических пар структурной схемы – виртуальные объекты. Виртуальное звено на схеме характеризуется только количеством вершин многоугольника, а кинематическая пара – конфигурацией её значка.

В приведенном случае виртуализация позволяет упростить решение задачи структурного синтеза механизма, освободив исследователя от проблем расчета и изготовления реальных звеньев механизма, и дает возможность сосредоточиться только на анализе получаемых структур.

Если же саму структурную схему принять в качестве части реального мира, то можно, при необходимости, использовать виртуализацию вновь и создать уже иные виртуальные объекты для самой структурной схемы. Это дает возможность спроектировать новые алгоритмы для решения задачи компьютерного синтеза структур механизмов.

Покажем это на примере поиска полного многообразия структур плоских рычажных механизмов при заранее определенной, тем или иным образом, номенклатуре и количестве звеньев, различной сложности.