Алгоритм та програмна реалізація декартового множення як елемент побудови моделі гідрогазодинамічних дросельних схем

Роман Грудецький

Кафедра автоматизованого управління виробничими процесами, Луцький національний технічний університет, УКРАЇНА, м. Луцьк, вул. Потебні, 56,
E-mail: grudik@ukr.net

Abstract – In the design of modern technological parameters of transducers having trouble choosing scheme of the converters, which would provide the necessary technical and metrological characteristics of them.

One such problem is the lack to date methodologies describe outline of transducers that describe the relevant circuit solutions , with which to perform synthesis and optimization of transducers . Proposed to be used for this set-theoretic concept, extending the number of concepts and operations of set theory. This will formalize the description and modeling scheme of the transducers , and on this basis to develop a theory of synthesis circuit solutions that automate the process of decision -making circuit .

For a given or re- synthesized circuit building measuring transducer to explore its functionality and specifications.

It is necessary to develop a methodology for modeling of converters and make it algorithmization that will implement the following processes on the computer.

The result of research is to develop computer-aided design elements mentioned above transducers .

Ключові слова – ламінарний дросель, турбулентний дросель, дросельна вимірювальна схема, САПР, прямий декартовий добуток, непрямий декартовий добуток, гадрогазодинаміка.

I. Вступ

На сучасному етапі розвитку технологій у виробництві та вимог до якості готової продукції зростають вимоги до ефективності контролю за параметрами таких виробництв. Це у свою чергу підвищує вимоги до методів та засобів контролю різних технологічних параметрів. Досить часто сучасні прилади не задовольняють поставлені виробництвом вимоги щодо точності вимірювання.

Поставлені задачі можна вирішити за допомогою застосування дросельних елементів та приладів, побудованих на їх основі.

Перевагами таких схем є простота у використанні, надійність експлуатації, низькі затрати на виготовлення та широкий діапазон застосування.

Для вирішення задачі структурно-параметричної оптимізації при побудові таких вимірювальних пристроїв необхідно описати структуру дросельних вимірювальних схем, сформулювати правила їх синтезу. Це дозволить створювати нові схеми, на основі яких будувати газогідродинамічні пристрої, зокрема пристрої контролю складу речовин із розширеними функціональними можливостями.

Отже, для проектування дросельних вимірювальних схем виникає необхідність у розробці системи автоматизованого проектування (САПР). Це дасть змогу ще на етапі розробки (після побудови дросельної вимірювальної схеми) приладів досліджувати їх функціональні можливості, такі як область застосування, точність, чутливість тощо.

II. Побудова дросельних вимірювальних схем

Дросельні вимірювальні схеми можуть бути побудовані з ламінарних, турбулентних та дроселів змішаного типу.

Ламінарний дросель характеризується великим співвідношенням довжини дроселя до його діаметру. Протікання рідини через такий дросель по каналу є ламінарним [1].

Турбулентні дроселі, мають канал циліндричної форми з малим – співвідношенням довжини до діаметру, в яких ефект дроселювання викликається місцевими опорами на вході і виході, а вплив сил тертя при течії рідини по каналу практично не позначається. Як правило, потік рідини в цих дроселях буває турбулентним і зазвичай адіабатичним [2].

Дросель змішаного типу – це такий дросель, який не підпадає під визначення ламінарного чи турбулентного дроселя [3].

Розглянемо детальніше побудову дросельних вимірювальних схем з подальшою можливістю їх характеристик. Для цього будемо використовувати операції прямого та «непрямого» декартового множення. Для такої побудови за вхідні параметри використовуємо множину первинних дросельних елементів $D=\{Л,Т,З\}$, де $Л$ – ламінарний, $Т$ – турбулентний та $З$ – змішаний дросельний елемент. Будемо вважати, що кількість первинних дроселів необмежена.

Пряме або декартове множення двох множин – це множина, елементами якого є усі можливі впорядковані пари вхідних множин (1). Будемо позначати таку операцію символом «$\times$». Наведемо приклади: якщо множини вихідних елементів

$$A=\{a\} \text{ і } B=\{b\}, \text{ то } A\times B=\{<a,b>\},$$

якщо ж $A=\{a,b\}$, а $B=\{a,b,c\}$, то

$$A\times B=\left\{\begin{matrix} <a,a>,<a,b>,<a,c>, \\ <b,a>,<b,b>,<b,c> \end{matrix}\right\} \qquad (1)$$

Результатом таких операцій будуть пари елементів (дроселів), включених у схему послідовно. Тобто множина $<Л,Т>$ визначає послідовне з’єднання ламінарного та турбулентного дроселів, а $<Т,Л>$ турбулентного та ламінарного. Слід звернути увагу, що важливу роль відіграє послідовність включення дроселя у пару.

При паралельному з’єднані елементів послідовність включення пар в схему ролі не відіграє. Будемо позначати паралельні з’єднання $[Л,Т]$ - паралельне включення ламінарного та турбулентного дроселів. Причому $[Л,Т]$ та $[Т,Л]$ - це одна і та ж схема.

Для знаходження усіх пар, включених у схему паралельно застосуємо операцію «непрямого» множення (2). Будемо позначати таку операцію символом «*» [4]. Так, якщо

$A=\{a,b\}$, а $B=\{a,b,c\}$, то

$$A*B=\{\ [a,a],[a,b],[a,c],[b,b],[b,c]\ \} \qquad (2)$$

Для опису складніших схем будемо використовувати дроселі нижчих порядків [4]. Порядком дроселя будемо називати кількість дроселів, що включені в схему (незалежно від структури самої схеми).

Таким чином, вимірювальні дросельні схеми першого порядку – це самі дроселі, другого порядку – це всі можливі схеми, складені з дроселів першого порядку тощо.

Будь-яку вимірювальну дросельну схему N порядку можна отримати, застосувавши по черзі операції прямого та непрямого множення до множин менших порядків.

В зв'язку з цим дроселі другого порядку – це складені дроселі, утворені із будь-яких двох дроселів першого порядку ($N=1+1$), а дроселі третього порядку – дроселі, утворені або із будь-яких трьох дроселів першого порядку, або із будь-якого одного дроселя другого порядку і будь-якого одного дроселя першого порядку ($N=1+1+1$, $N=2+1$; $N=1+2$) і т.д.

Розглянута методологія побудови поширюється на дроселі будь-якого порядку. Так, наприклад, дроселі п'ятого порядку утворюють послідовним чи паралельним з'єднанням дроселів четвертого порядку і дроселів першого порядку, дроселів першого порядку і дроселів четвертого порядку, дроселів другого порядку і дроселів третього порядку, а також дроселів третього порядку і дроселів другого порядку ($N=4+1$; $N=1+4$; $N=2+3$; $N=3+2$). Дроселі шостого порядку утворюють аналогічно вже з п'яти різноманітних пар дроселів різних, але нижчих порядків ($N=5+1$; $N=4+2$; $N=3+3$; $N=2+4$; $N=1+5$). Таким чином дроселі N-го порядку утворюють із ($N-1$) кількості пар вихідних дроселів різних порядків.

Побудову дроселів четвертого порядку розглянемо по аналогії до дроселів третього порядку.

Для $D_1=\{Л\}$, коли $n=1$, а $k_4=15$ маємо:

$Л-Л-Л-Л$; $\begin{bmatrix}Л\\Л\end{bmatrix}-Л-Л$; $Л-\begin{bmatrix}Л\\Л\end{bmatrix}-Л$;

$\begin{bmatrix}Л\\Л\\Л\end{bmatrix}-Л$; $Л-Л-\begin{bmatrix}Л\\Л\end{bmatrix}$; $Л-\begin{bmatrix}Л-Л\\Л\end{bmatrix}$; $Л-\begin{bmatrix}Л\\Л\\Л\end{bmatrix}$;

$\begin{bmatrix}Л-Л-Л\\Л\end{bmatrix}$; $\begin{bmatrix}\begin{bmatrix}Л\\Л\end{bmatrix}-Л\\Л\end{bmatrix}$; $\begin{bmatrix}Л-\begin{bmatrix}Л\\Л\end{bmatrix}\\Л\end{bmatrix}$; $\begin{bmatrix}Л-Л\\Л\\Л\end{bmatrix}$;

$\begin{bmatrix}Л\\Л\\Л\\Л\end{bmatrix}$; $\begin{bmatrix}Л\\Л\end{bmatrix}-\begin{bmatrix}Л\\Л\end{bmatrix}$; $\begin{bmatrix}Л-Л\\Л-Л\end{bmatrix}$; $\begin{bmatrix}Л-Л\\Л\end{bmatrix}-Л$.

Отже складені дроселі четвертого порядку утворюють за п'ятнадцятьма різними схемами. Кожній такій схемі відповідає певна структура математичної моделі дроселя.

При $n=2$, наприклад коли $D_1=\{Л,T\}$, вже маємо $k_4=176$ різних схем. Враховуючи велику кількість дроселів їх розглядати не будемо.

III. Програмна реалізація алгоритму побудови дросельних схем N порядку

Для того щоб побудувати дросельну вимірювальну схему будь-якого порядку необхідно мати первинні дросельні елементи. Якщо відома кількість первинних елементів, застосовуючи покрокові операції прямого і непрямого множення дроселів, на 1 кроці отримаємо дві дросельні схеми другого порядку. Записавши результати обрахунків у базу даних, можемо використовувати їх у подальшому. Таким чином, якщо необхідно знайти дросель N порядку, то отримавши з бази даних усі рядки (номера яких в сумі дають N) і застосувавши операції прямого та непрямого добутку отримаємо шуканий результат.

Програмну реалізація вище описаних правил виконана за допомогою програмних продуктів Borland Delphi – CodeGear RAD Studio 2009 з використанням бази даних FireBird 1.5.

Для реалізації вище викладених правил створено програмний комплекс, який виконує наступну послідовність дій (функцій):

1. Задати множину первинних елементів.
2. Задати порядок синтезованого дроселя.
3. Знайти всі можливі пари дроселів нижчих порядків, які б в результаті складення давали дросель вищого порядку.
4. Попарно виконати прямий декартовий добуток рядків матриці, знайдених попередньо (див. пункт 3).
5. Аналогічно виконати непрямий добуток.
6. Записати результати, отримані прямим та непрямим декартовим множенням у відповідний рядок матриці результатів. Номер рядка цієї матриці буде визначати порядок складеного дроселя.
7. З отриманих варіантів виключити схеми, які повторюються.
8. Записати результат в новий рядок матриці (номер рядка матриці визначається порядком знайденого дроселя).
9. Якщо номер новоутвореного рядка співпадає з порядком заданої схеми (вхідний параметр), дії завершити.
10. Побудувати математичні моделі кожної зі знайдених дросельних схем.

Відповідно дії 3-10 повторюються поки порядок синтезованих схем буде нижчий заданого. При досягненні заданого порядку, синтез дросельних схем зупиняється.

Основними на першому етапі є функції прямого (procedure get_pramo(a, b: array of string; k, m: integer; radok: integer; pozition: integer)) та непрямого (procedure get_nepramo(a, b: array of string; k, m:

integer; radok: integer; pozition: integer)) добутку. Вхідними параметрами для цих процедур будуть дросельні схеми нижчих порядків a та b. Ці змінні містять у масивах символьного типу. Змінні k та m відповідно розміри цих масивів.

Результати генерування дросельних схем на кожному етапі записуються в головний масив BigBossMasiv : array [1..1000, 1..1000] of string. Змінні radok та pozition визначають в який рядок та на яку позицію даного записувати заново генеровану схему. Тобто у першому рядку цього масиву будемо записувати усі дроселі першого порядку, у другий рядок – другого тощо. Це дасть нам змогу в подальшому знаходити доданки (номера рядків), які в сумі визначать порядок шуканої дросельної схеми.

Нижче наведено програмну реалізацію пошуку усіх пар дроселів нижчих порядків

```
e := 1;
   for q := 1 to 99 do
   for w := 1 to 99 do
   if (q+w)=radok then
     begin
       pary_find[1,e]:=q;
       pary_find[2,e]:=w;
       inc(e);
     end;
   e := e-1;
```

де pary_find – це масив, що зберігаю номера шуканих пар.

Програмна реалізація процедури прямого добутку наведено нижче.

```
procedure TFrmMain.get_pramo(a, b: array of string; k, m: integer; radok: integer; pozition: integer);
var
  i, j, kilk : integer;
  res : array [1..1000] of string;
begin
```

Програмна реалізація процедури прямого добутку:

```
kilk := 1;
for I := 0 to k-1 do
  for j := 0 to m-1 do
    begin
      res[kilk] := '<'+a[i]+','+b[j]+'>';
      inc(kilk);
    end;
for I := pozition to pozition + kilk - 1 do
  BigBossMasiv[radok, i] := res[i-pozition+1];
BigBossMasivPozition:= pozition+kilk-1;
end;
procedure TFrmMain.get_nepramo(a, b: array of string; k, m, radok: integer; pozition: integer);
var
  i, j, kilk : integer;
  res : array [1..1000] of string;
begin
kilk := 1;
for I := 0 to k-1 do
  for j := i+1 to m-1 do
    begin
      res[kilk] := '['+a[i]+','+b[j]+']';
      inc(kilk);
    end;
for I := pozition to pozition + kilk - 1 do
  BigBossMasiv[radok, i] := res[i-pozition+1];
BigBossMasivPozition:= pozition+kilk-1;
end;
```

Зберігаючи результати моделювання у базу даних було введено правило перевірки на повторення. При знаходженні повторень при синтезі дросельних схем дублікат буде видалено. При цьому слід враховувати порядок включення у пару. При послідовному включенні елементів порядок відіграє важливу роль, а при паралельному – ні. Тобто, <a,b> та <b,a> - це різні схеми послідовного з’єднання елементів a та b, а з’єднання [a,b] та [b, a] – однакові.

При автоматичній генерації схем виникає питання вірності запису згідно введеним правилам. Щоб проаналізувати схему та отримати «чистий» варіант дросельної схеми у прийнятному для користувача вигляді була розроблена процедура фільтрації виведення результатів filter(a: string):string, вхідним параметром якої є запис формули схеми N порядку.

Програмна реалізація цієї функції:

```
function TFrmMain.filter(a: string):string;
var
  i,j,p : integer;
  s : string;
  count1, count2 : integer;
  mozhna : boolean;
begin
s:=a;
for i := 1 to length(s)-1 do
 begin
  if (s[i]='<')and(s[i+1]='<') then
  begin
   for j:=i+2 to length(s)-2 do
    if s[j]='>' then
     begin
      p:=j;
     end;
   Delete(s,i+1,1);
   Delete(s,p-1,1);
  end;
 end;
for i := 1 to length(s)-1 do
 begin
  if (s[i]='[')and(s[i+1]='[') then
  begin
   for j:=i+2 to length(s)-2 do
    if s[j]=']' then
     begin
      p:=j;
     end;
   Delete(s,i+1,1);
   Delete(s,p-1,1);
  end;
 end;
for i := length(s) downto 2 do
```

```
   begin
    if (s[i]='>')and(s[i-1]='>') then
    begin
     for j:=i-1 downto 2 do
      if (s[j]='<') and (s[j-1]<>'[') and (s[j-1]<>']') and (s[j+1]<>']') and (s[j+1]<>']') then
       begin
        p:=j;
        break;
       end;
     Delete(s,i-1,1);
     Delete(s,p,1);
    end;
   end;
  for i := length(s) downto 1 do
   begin
    if (s[i]=']')and(s[i-1]=']') then
    begin
     for j:=length(s)-2 downto 1 do
      if s[j]='[' then
       begin
        p:=j;
        break;
       end;
     Delete(s,i-1,1);
     Delete(s,p,1);
    end;
   end;
  result := s;
  end;
```

Таким чином, ми отримаємо результати генерації у наступному вигляді (Рис. 1).

Як бачимо на Рис. 1 при двох первинних дросельних елементах (ламінарний та турбулентний) можливо отримати 176 дросельних вимірювальних схем 4 порядку.

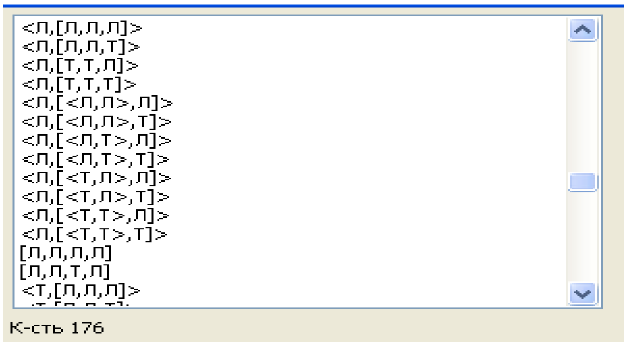

Рис. 1. Приклад генерації дросельної схеми 4 порядку при кількості первинних елементів два (Л, Т)

Висновок

Таким чином отримано алгоритм та його програмну реалізацію для декартового множення множин, як перший етап моделювання гідрогазодинамічних дросельних схем, що в подальшому дозволить досліджувати фізичні характеристики вимірювальних пристроїв, побудованих на їх основі.

References

[1] “Laminar throttle,” 2013. [Online]. Available: http://www.ngpedia.ru/id14351p1.html. [Accessed: Oct. 10, 2003].

[2] “Turbulent throttle,” 2013. [Online]. Available: http://www.ngpedia.ru/id14341p1.html. [Accessed: Oct. 10, 2003].

[3] “Pneumatic throttle,” 2013. [Online]. Available: http://www.ngpedia.ru/id14329p1.html. [Accessed: Oct. 10, 2003].

[4] E. Pistun, Doctoral Dissertation in Technical Sciences, LPI, Saint-Petersburg, Russia, 1992.