

Veszprémi Egyetem

Műszaki Informatikai Kar

Számítástudomány Alkalmazása Tanszék

Műszaki informatika szak

DIPLOMADOLGOZAT

**Egzakt linearizációs szabályozó struktúra  
kiválasztása MIMO rendszerekre gráfelméleti  
módszerekkel**

Bujdosó Tibor

Témavezető: **Dr. Hangos Katalin**

Veszprém

2004.

# Nyilatkozat

Alulírott Bujdosó Tibor, diplomázó hallgató, kijelentem, hogy a diplomadolgozatot a Veszprémi Egyetem Számítástudomány Alkalmazása tanszékén készítettem mérnök-informatikus diploma (master of engineering in information technology) megszerzése érdekében.

Kijelentem, hogy a diplomadolgozatban foglaltak saját munkám eredményei, és csak a megadott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy a diplomadolgozatban foglalt eredményeket a Veszprémi Egyetem, valamint a feladatot kiíró szervezeti egység saját céljaira szabadon felhasználhatja.

Veszprém, 2004. május 13.

Bujdosó Tibor

## Köszönetnyilvánítás

Legelőször témavezetőmnek, Dr. Hangos Katalinnak mondok hálás köszönetet a gondos irányításért, és a sok támogatásért, amellyel munkámat, fejlődésemet segítette.

Köszönettel tartozom szüleimnek, hogy tanulmányaimat sok esztendőn át minden erejükkel segítették.

Hálás vagyok Németh Erzsébetnek, aki munkámat sok hasznos tanáccsal segítette.

## Tartalmi összefoglaló

Egzakt linearizációs szabályozó struktúra gráfelméleti módszerekkel történő kiválasztását vizsgálom ebben a diplomamunkában nemlineáris MIMO rendszerekre.

Egzakt linearizáció alkalmazásával lineáris szabályozóval szabályozhatunk egy SISO nemlineáris rendszert úgy, hogy a nemlineáris rendszerhez találunk egy olyan alkalmas statikus nemlineáris állapotvisszacsatolást, amivel elérhetjük, hogy a rendszer lineáris legyen az új bemenet és az eredeti kimenet között. Ha egy MIMO nemlineáris rendszert szeretnénk ezzel a módszerrel szabályozni, akkor a MIMO rendszert szét kell vágnunk olyan részrendszerekre, amelyek teljesítik az egzakt linearizáció relatív fokszám feltételét.

Ehhez felhasználok a nemlineáris MIMO rendszer struktúragráfját, amelyben megkeresem azokat a SISO részgráfokat, amelyek eleget tesznek a relatív fokszám feltételnek. Ezeket a részgráfokat megfelelően egy súlyozott páros gráfnak, és ezen a súlyozott páros gráfon maximális párosítást keresve megkapunk egy lehetséges optimális egzakt linearizációs szabályozó struktúrát, amelyben minimális a részrendszerek közötti kölcsönhatás.

Az egzakt linearizációs szabályozó struktúrát megkereső algoritmust C++ programozási nyelven implementáltam.

Több esettanulmányon vizsgálom, hogy az egzakt linearizációs szabályozó struktúra kiválasztása egyértelmű és optimális-e.

**Kulcsszavak:** *egzakt linearizáció, relatív fokszám feltétel, struktúra gráf, kölcsönhatás a részrendszerek között*

## Summary

Distributed controller structure selection for exact linearization is investigated in this diploma thesis by using graph theoretical methods on nonlinear MIMO systems. Using exact linearization, we could control a nonlinear SISO system with linear controllers in such a way that the use of static nonlinear state feedback is accomplished for the purpose of transforming a given nonlinear system into a linear and controllable one. If we want to control a MIMO nonlinear system with this method, the MIMO system should be decomposed into distributed subsystems which satisfy the relative degree condition of the exact linearization.

For this purpose we apply the structure graph of the nonlinear MIMO system, wherein SISO subgraphs are found that satisfy the relative degree condition. These subgraphs correspond to the edges of a weighted bipartite graph and maximal matching is found in this weighted bipartite graph to obtain a possible optimal exact linearization control structure wherein the interaction of the subsystems are minimal.

I have implemented the exact linearization control structure selection algorithm using C++ programming language.

By using case studies I have investigated whether the result of the proposed algorithm is unique and optimal.

**Keywords:** *exact linearization, relative degree condition, structure graph, interaction of the subsystems*

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>1</b>
<b>2. Nemlineáris rendszerek egzakt linearizálása állapot visszacsatolással</b>	<b>3</b>
2.1. Relatív fokszám . . . . .	3
2.2. Nemlineáris koordináta transzformáció és állapot visszacsatolás . . . .	6
2.3. Az állapottér egzakt linearizációs probléma SISO rendszerekre . . . .	8
2.3.1. Probléma kitűzés . . . . .	8
2.3.2. A megoldás létezési feltétele . . . . .	9
2.3.3. A megoldás menete . . . . .	11
<b>3. Egzakt linearizációs szabályozó struktúra tervezés MIMO rendszerekre</b>	<b>13</b>
3.1. Relatív fokszám vizsgálata struktúra gráfokon . . . . .	13
3.2. Folyamatrendszerek maximális relatív fokszámmal . . . . .	18
3.3. Súlyozott páros gráf struktúratervezésre . . . . .	18
3.4. Maximális párosítás keresésének feltételei . . . . .	20
3.4.1. A főátlóban lévő elemek kiválasztása . . . . .	20
3.4.2. Domináns párok kölcsönhatásának minimalizálása . . . . .	21
3.5. Algoritmus a maximális párosítás megkeresésére . . . . .	22
3.5.1. A páros gráf élsúlyainak meghatározása . . . . .	22
3.5.2. A maximális párosítás megkeresése . . . . .	24
3.6. Részrendszerek kapcsolata . . . . .	25
<b>4. Esettanulmányok</b>	<b>26</b>
4.1. Hőcserélő cella modellje . . . . .	26
4.2. Szabályozó struktúra kiválasztása a hőcserélő hálózaton . . . . .	27
4.3. Folytonosan kevert tartály reaktor . . . . .	30
4.4. Tennessee Eastman probléma . . . . .	35
4.5. Az esettanulmányok értékelése . . . . .	35
<b>5. A program felhasználói leírása</b>	<b>37</b>
<b>6. Összefoglalás</b>	<b>40</b>
<b>7. Függelék</b>	<b>42</b>

# 1. Bevezetés

Egy nemlineáris rendszert nemlineáris szabályozókkal szabályozni csak akkor érdemes, ha széles működési tartományban szeretnénk egy, a célfüggvényben megfogalmazott viselkedést elérni. Ez esetben a nemlineáris rendszer analízise nehéz feladat. Egzakt linearizáció alkalmazásával lineáris szabályozóval szabályozhatunk egy SISO nemlineáris rendszert úgy, hogy a nemlineáris rendszerhez találunk egy olyan alkalmas statikus nemlineáris állapot visszacsatolást, amivel elérhetjük, hogy a rendszer lineáris legyen az új bemenet és az eredeti kimenet között, feltéve, hogy a rendszer kielégíti az úgynevezett relatív fokszám feltételt (2.2.1. lemma). Ezután bármely lineáris szabályozótervezési módszer, úgymint PID, pólusáthelyezés, LQR alkalmazható a rendszer stabilizálására, vagy a dinamikus paraméterek módosítására.

Ha egy több bementű-több kimenetű (MIMO) nemlineáris rendszert szeretnénk ezzel a módszerrel szabályozni, akkor a MIMO rendszert szét kell vágnunk olyan részrendszerekre, amelyek teljesítik a fenti relatív fokszám feltételt. E diplomamunkában nemlineáris MIMO rendszerekhez szabályozó struktúra kiválasztásának egy lehetséges egzakt linearizáción alapuló módszerét gráfelméleti módszerekkel vizsgálom, felhasználva a folyamatrendszer struktúragráfját.

A diplomamunkámban vizsgált probléma elméleti alapjainak leírása megtalálható [1]-ben. Ez a munka a nemlineáris rendszer szabályozó struktúrájának kiválasztásához a relatív fokszám és a rendszer struktúragráfja közötti kapcsolatot írja le, de nem ad algoritmikus megoldást az optimális szabályozó struktúra meghatározására

A szabályozó struktúra kiválasztását más megközelítésben is lehet vizsgálni. Pl. [4]-ben a rendszer numerikus paramétereit is felhasználva kerül kiválasztásra a megfelelő szabályozó struktúra. Itt a relatív erősítési mátrixot megfelelően egy súlyozott páros gráfnak úgy kell kiválasztani a legjobb szabályozó struktúrát, hogy a párosításban szereplő élsúlyok abszolút értékének összege maximális legyen.

A munkám során felhasznált gráfelméleti módszerek megtalálhatók [6]-ben és [5]-ban. A vizsgálandó rendszer struktúragráfjában a bemenetektől a kimenetekig vezető legrövidebb utakat szélességi kereséssel határozza meg a program. Az átsúlyozott relatív fokszám mátrixnak megfelelően súlyozott párosgráfban pedig maximális súlyú párosítást keresek a legjobb szabályozó struktúra kiválasztására.

A diplomamunka a következőképpen épül fel. A 2. fejezetben SISO nemlineáris rendszerek egzakt linearizálásának feltételeit és megoldását írom le. A 3. fejezetben MIMO nemlineáris rendszerekhez egzakt linearizációs szabályozó struktúra kiválasztásának feltételeit és a szabályozó struktúra kiválasztására alkalmas algoritmust mutatom be. A 4. fejezetben az egzakt linearizációs szabályozó struktúra kiválasztására létrehozott algoritmust vizsgálom konkrét példákon. Az 5. fejezet a C++ programozási nyelven implementált algoritmus felhasználói dokumentációját tartalmazza.



## 2. Nemlineáris rendszerek egzakt linearizálása állapot visszacsatolással

[3]Az egzakt linearizáció célja az, hogy SISO nemlineáris rendszerhez találjunk egy olyan alkalmas statikus nemlineáris állapot visszacsatolást, amivel elérhetjük, hogy a rendszer lineáris legyen az új koordinátákban az új bemenet és az eredeti kimenet között. Ezután bármely lineáris szabályozótervezési módszer, úgymint PID, pólusáthelyezés, vagy LQR alkalmazható a rendszer stabilizálására, vagy a dinamikus paraméterek módosítására.

Az egzakt linearizáció állapot visszacsatolással egy egyszerű, de szigorú alkalmazási határokkal rendelkező technika nemlineáris rendszerek szabályozására, mert csak olyan időinvariáns SISO rendszereknél alkalmazható, amelyek kielégítik az úgynevezett relatív fokszám feltételt.

### 2.1. Relatív fokszám

A relatív fokszám fogalma nemcsak az egzakt linearizáció szükséges feltételeiben játszik központi szerepet, hanem a nemlineáris szabályozás elméletének más területein is.

**2.1.1. Definíció (Lie-derivált).** *Legyenek adottak a  $\lambda \in R^n \rightarrow R$ ,  $f \in R^n \rightarrow R^n$  függvények, és  $U = \text{dom}(f) \cap \text{dom}(\lambda) \subseteq R^n$  nyílt halmaz. Azt mondjuk, hogy a  $\lambda$  függvény Lie-deriváltja  $f$  mentén az alábbi függvény:*

$$L_f \lambda(x) = \frac{\partial \lambda(x)}{\partial x} f(x) = \sum_{i=1}^n \frac{\partial \lambda(x)}{\partial x_i} f_i(x) = \langle d\lambda(x), f(x) \rangle \quad (1)$$

*Láthatjuk, hogy a Lie-derivált skalár értékű, így a műveletet elvégezhethetjük a magasabb rendű deriváltakra is:*

$$L_g L_f \lambda(x) = \frac{\partial (L_f \lambda(x))}{\partial x} g(x) \quad (2)$$

$$L_f^k \lambda(x) = \frac{\partial (L_f^{k-1} \lambda(x))}{\partial x} f(x) \quad (3)$$

**2.1.2. Definíció (Relatív fokszám).** Azt mondjuk, hogy egy egy-bemenetű egy-kimenetű (SISO) nemlineáris rendszer

$$\dot{x} = f(x) + g(x)u \quad (4)$$

$$y = h(x) \quad (5)$$

relatív fokszáma  $r$  az  $x^0$  pontban, ha

1.  $L_g L_f^k h(x) = 0$  minden  $x$ -re  $x^0$  környezetében  $\forall k < r - 1$ -re, és

2.  $L_g L_f^{r-1} h(x) \neq 0$

**2.1.1. Lemma.** A relatív fokszám  $r$  pontosan egyenlő azzal a számmal, ahányszor  $t = t^0$ -nál deriválni kell a bemenet értékét, hogy az a kimeneten megjelenjen.

A definíció és a fenti állítás összevetéséhez felhasználjuk az (4) állapot egyenletet, amelyben a  $g(x)$  függvény vektor értékű:

$$g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_n(x) \end{bmatrix}$$

A kimeneti egyenlet  $k$ -adik deriváltját a következőképpen számoljuk:

- $k = 0$  :  $y = h(x)$

$$\frac{dy}{dt} = \frac{\partial h}{\partial x} \dot{x} = \frac{\partial h}{\partial x} (f(x) + g(x)u) = L_f h(x) + L_g h(x) \cdot u$$

- $k = 1$  : feltéve, hogy  $L_g h = 0$  (1. feltétel), akkor  $\dot{y} = L_f h$ , és

$$\frac{d^2 y}{dt^2} = \frac{\partial (L_f h)}{\partial x} (f(x) + g(x)u) = L_f^2 h + L_g L_f h \cdot u$$

Ha  $L_g L_f h \neq 0$ , akkor  $r > 1$  (lásd 2. feltétel)

- és így tovább.

Több-bemenetű több-kimenetű (MIMO) rendszerekre a relatív fokszámot definiálhatjuk az összes lehetséges be-kimenet párra.

**2.1.1. Példa (Lineáris időinvariáns SISO rendszerek relatív fokszáma).** Legyen a lineáris időinvariáns SISO rendszer a következő egyenletekkel adott:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}$$

Könnyű belátni azt, hogy

$$L_f^k h(x) = CA^k x$$

és

$$L_g L_f^k h(x) = CA^k B$$

Megjegyezzük azt, hogy a relatív fokszám feltételek függnek a lineáris időinvariáns SISO rendszer Markov paramétereitől.

Így az  $r$  relatív fokszám a következőképpen adható meg:

$$\begin{aligned}CA^k B &= 0 \quad \forall k < r - 1 \\ CA^{r-1} B &\neq 0.\end{aligned}$$

Jól ismert, hogy az az egész szám, amely egy lineáris rendszerben teljesíti ezt a feltételt, pontosan egyenlő a rendszer átviteli függvényében  $H(s) = C(sI - A)^{-1}B$  a nevező fokszámának ( $n$ ) és a számláló fokszámának ( $m$ ) különbségével, azaz  $r = n - m \geq 0$ .

A következő példa bemutatja egy egyszerű nemlineáris rendszer relatív fokszámának kiszámítását.

**2.1.2. Példa (Nemlineáris rendszer relatív fokszáma).** Kiszámoljuk a következő rendszer relatív fokszámát az  $[1 \ 1]^T$  pontban.

$$\begin{aligned}\dot{x} &= \begin{bmatrix} x_2 \\ 2\omega\zeta(1 - \mu x_1^2)x_2 - \omega^2 x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= h(x) = x_1\end{aligned}$$

ahol  $\omega$ ,  $\zeta$  and  $\mu$  nemnulla konstansok. Találjunk két olyan pontot az állapottéren, ahol a rendszer relatív fokszáma különböző.

A szükséges Lie-deriváltakat a következőképpen számolhatjuk:

$$\begin{aligned} L_g h(x) &= \frac{\partial h}{\partial x} g(x) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \\ L_f h(x) &= x_2 \\ L_g L_f h(x) &= \frac{\partial L_f h}{\partial x} g(x) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 \end{aligned}$$

Látható, hogy a rendszer relatív fokszáma 2 az állapottér bármely pontjában (az  $[1 \ 1]^T$  pontban is). Következésképpen nincsenek az állapottérnek olyan pontjai, ahol a rendszer relatív fokszáma különböző lenne.

## 2.2. Nemlineáris koordináta transzformáció és állapot visszacsatolás

SISO rendszereknél statikus állapot visszacsatolásos szabályzásnál az általános esetben az  $u$  bemenetet a következőképpen számoljuk:

$$u = \alpha(x) + \beta(x)v \quad (6)$$

,ahol  $v$  a külső referencia input. Az e szabályzóval visszacsatolt input-affin nemlineáris SISO rendszer a következő lesz:

$$\dot{x} = f(x) + g(x)\alpha(x) + g(x)\beta(x)v \quad (7)$$

$$y = h(x) \quad (8)$$

Az állapotvisszacsatolás alkalmazásának (és a koordináták váltásának az állapottéren) az a célja, hogy transzformáljuk az adott nemlineáris rendszert egy olyanná, amely lineáris és irányítható.

A következő lemma bevezeti a a koordináta transzformációt, amely alkalmazásával a rendszer lineáris lesz az új koordinátákban.

**2.2.1. Lemma.** *Tekintsük az (4) nemlineáris rendszert  $r=n$  relatív fokszámmal (ami pontosan egyenlő a az állapottér dimenziójával) valamely  $x = x^0$  pontban és a következő koordináta transzformációt*

$$\Phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_n(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix} \quad (9)$$

azaz a kimeneti függvény és annak első  $n - 1$   $f(x)$  irányú deriváltjának transzformáltja. Ekkor létezik nemlineáris statikus visszacsatolás úgy, hogy a zárt rendszer az új koordinátákban lineáris és irányítható.

Az új koordináták:

$$z_i = \phi_i(x) = L_f^{i-1}h(x) \quad 1 \leq i \leq n \quad (10)$$

Ezekkel a rendszermodell a következőképpen írható:

$$\dot{z}_1 = z_2 \quad (11)$$

$$\dot{z}_2 = z_3 \quad (12)$$

$$\vdots \quad (13)$$

$$\dot{z}_{n-1} = z_n \quad (14)$$

$$\dot{z}_n = b(z) + a(z)u \quad (15)$$

így

$$\frac{d}{dt}L_f^k h(x) = \frac{\partial L_f^k h(x)}{\partial x} \dot{x} = \frac{\partial L_f^k h(x)}{\partial x} (f(x) + g(x)u) \quad (16)$$

$$= L_f^{k+1}h(x) + L_g L_f^k h(x)u \quad (17)$$

Ha a rendszer relatív foka  $n$ , akkor

$$L_g L_f^k h(x) = 0 \quad \forall k < n \quad (18)$$

és a fenti egyenlet a következőképpen alakul:

$$\frac{d}{dt}L_f^k h(x) = \dot{z}_k = L_f^{k+1}h(x) = z_{k+1} \quad \forall k < n \quad (19)$$

végül az utolsó egyenlet az alábbi alakú:

$$\dot{z}_n = \frac{d}{dt}L_f^{n-1}h(x) + L_g L_f^{n-1}h(x)u \quad (20)$$

Feltéve, hogy a következő állapot visszacsatolási szabályt választjuk

$$u = \frac{1}{a(z)}(-b(z) + v) \quad (21)$$

a paraméterek a fenti egyenlet szerint:

$$\alpha(x) = -\frac{b(z)}{a(z)} = -\frac{L_f^n h(x)}{L_g L_f^{n-1} h(x)} \quad , \quad \beta(x) = \frac{1}{a(z)} = \frac{1}{L_g L_f^{n-1} h(x)}$$

Az eredményül kapott zárt rendszert meghatározzák a következő egyenletek:

$$\dot{z}_1 = z_2 \quad (22)$$

$$\dot{z}_2 = z_3 \quad (23)$$

$$\vdots \quad (24)$$

$$\dot{z}_{n-1} = z_n \quad (25)$$

$$\dot{z}_n = v \quad (26)$$

azaz a zárt rendszer valóban lineáris és irányítható.

*Fontos megjegyezni, hogy a transzformációt két alapelem alkotja úgy, hogy*

1. *a koordináta váltás az  $x^0$  lokális környezetében definiált*
2. *az állapot visszacsatolás is az  $x^0$  lokális környezetében definiált.*

### 2.3. Az állapottér egzakt linearizációs probléma SISO rendszerekre

Az (2.2.1) Lemmában a kritikus feltétel az, hogy a rendszer relatív fokszáma megegyezzen az állapotváltozók számával, azaz  $r = n$  legyen. Ha nem ez az eset áll fenn, azaz  $r < n$ , akkor megpróbálhatunk *mesterséges kimenetet* konstruálni  $y = \lambda(x)$  alakban. Azaz keresünk egy olyan  $\lambda(x)$  függvényt, amely különbözik az eredeti  $h(x)$ -től úgy, hogy a rendszer relatív fokszáma  $r = n$  legyen. Az elgondolás az, hogy az (2.2.1) Lemmával megadott alapeset létezéséhez megtaláljuk a feltételeket, és képesek legyünk  $\lambda(x)$  megkonstruálására.

Fontos megjegyezni, hogy nem várható, hogy a  $\lambda(x)$  függvény egyértelmű legyen (ha létezik), mert a nemlineáris koordináta transzformáció  $\Phi(x)$  sem egyértelmű. Ezért azt a transzformációt  $\Phi(x)$  szeretnénk megtalálni, amely egyszerű és invertálható.

#### 2.3.1. Probléma kitűzés

Tekintsük az (4) input-affin nemlineáris rendszer modellt a kimeneti egyenlete nélkül

$$\dot{x} = f(x) + g(x)u \quad (27)$$

és feltételezzük, hogy a következő probléma megoldható.

Adott  $x^0$  pont az állapot téren. Találjunk (ha lehetséges) az  $x^0$ -nak olyan  $U$  környe-

zetét, hogy a visszacsatolás

$$u = \alpha(x) + \beta(x)v \quad (28)$$

és a  $z = \Phi(x)$  koordináta transzformáció is értelmezve legyen  $U$ -n, úgy, hogy a megfelelő zárt rendszer

$$\dot{x} = f(x) + g(x)\alpha(x) + g(x)\beta(x)v \quad (29)$$

az új  $z = \Phi(x)$  koordinátákban lineáris és irányítható legyen. Azaz

$$\left[ \frac{\partial \Phi}{\partial x} (f(x) + g(x)\alpha(x)) \right]_{x=\Phi^{-1}(z)} = Az \quad (30)$$

$$\left[ \frac{\partial \Phi}{\partial x} (g(x)\beta(x)) \right]_{x=\Phi^{-1}(z)} = B \quad (31)$$

valamely alkalmas  $A \in R^{n \times n}$  mátrixra és  $B \in R^n$  vektorra teljesül:

$$\text{rank}(B \quad AB \quad \dots \quad A^{n-1}B) = n. \quad (32)$$

Ez a probléma a SISO verziója az úgynevezett *Állapottér Egzakt Linearizációs Problémának*.

### 2.3.2. A megoldás létezési feltétele

A következő lemma megadja az állapottér egzakt linearizációs probléma megoldásához szükséges és elégséges feltételt.

**2.3.1. Lemma.** *Az Állapottér Egzakt Linearizációs probléma megoldható akkor és csak akkor, ha létezik az  $x^0$ -nak olyan  $U$  környezete és olyan valós értékű  $U$ -n definiált  $\lambda$  függvény, hogy a rendszer*

$$\dot{x} = f(x) + g(x)u \quad (33)$$

$$y = \lambda(x) \quad (34)$$

relatív fokszáma  $n$   $x^0$ -ban.

A  $\lambda$  függvény megtalálásának problémája az, hogy a rendszer relatív fokszáma  $x^0$ -ban pontosan  $n$  legyen. Azaz a függvény olyan, hogy

$$L_g \lambda(x) = L_g L_f \lambda(x) = \dots = L_g L_f^{n-2} \lambda(x) = 0 \quad \forall x \quad (35)$$

$$L_g L_f^{n-1} \lambda(x^0) \neq 0 \quad (36)$$

A fenti egyenlet alapján a probléma megoldása egy parciális differenciál egyenletrendszer (PDE) megoldását igényli.

Ez jól látható, ha átírjuk az első (35) egyenletet PDE formára, felhasználva, hogy  $\lambda(x) = \lambda(x_1 \dots x_n)$  skalár értékű függvény.

$$L_g \lambda(x) = \sum_{i=1}^n \frac{\partial \lambda}{\partial x_i}(x) g_i(x) = 0 \quad (37)$$

**2.3.1. Definíció (Lie-szorzat).** *Legyenek  $f, g \in R^n \rightarrow R^n$  valós, vektor értékű függvények, és  $U = \text{dom}(f) \cap \text{dom}(g) \subseteq R^n$  nyílt halmaz. Az  $f$  és  $g$  függvények Lie-szorzata:*

$$[f, g](x) = \frac{\partial g(x)}{\partial x} f(x) - \frac{\partial f(x)}{\partial x} g(x) \quad (38)$$

ahol

$$\frac{\partial g(x)}{\partial x} = \begin{bmatrix} \frac{\partial g_1(x)}{\partial x_1} & \frac{\partial g_1(x)}{\partial x_2} & \dots & \frac{\partial g_1(x)}{\partial x_n} \\ \frac{\partial g_2(x)}{\partial x_1} & \frac{\partial g_2(x)}{\partial x_2} & \dots & \frac{\partial g_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n(x)}{\partial x_1} & \frac{\partial g_n(x)}{\partial x_2} & \dots & \frac{\partial g_n(x)}{\partial x_n} \end{bmatrix} \quad (39)$$

A (37) egyenlet megoldásához bevezetünk egy rövidített jelölést

$$L_{ad_f g} \lambda = \frac{\partial \lambda}{\partial x} [f, g] \quad (40)$$

ahol  $ad_{fg}^i$ -t rekurzívan definiáljuk a következőképpen:

$$ad_f^0 g(x) = g(x) \quad (41)$$

$$ad_f^i g(x) = [f, ad_f^{i-1} g(x)] \quad (42)$$

Például:

$$ad_f^1 g(x) = ad_f g(x) = [f, g] \quad , \quad ad_f^2 g(x) = [f, [f, g]]$$

A (35) egyenlet ekkor az alábbi ekvivalens formát ölti:

$$L_g \lambda(x) = L_{ad_f g} \lambda(x) = \dots = L_{ad_f^{n-2} g} \lambda(x) = 0 \quad (43)$$

és a nemtriviális feltétel (36) ekvivalens az alábbi egyenlettel:

$$L_{ad_f^{n-1} g} \lambda(x^0) \neq 0. \quad (44)$$

Most már kimondhatjuk az állapotter egzakt linearizációs probléma megoldásához szükséges és elégséges feltételt.



**2.3.1. Tétel.** *Feltételezzük, hogy a rendszer modell a következőképpen adott:*

$$\dot{x} = f(x) + g(x)u \quad (45)$$

*Az Állapottér Egzakt Linearizációs Probléma megoldható az  $x^0$  környezetében (azaz létezik olyan  $\lambda$  "kimeneti" függvény, amellyel a rendszer relatív foka  $n$   $x^0$ -ban) akkor és csak akkor, ha következő feltételek teljesülnek:*

1.  $a \Delta_c(x^0) = [g(x^0) \text{ ad}_f g(x^0) \dots \text{ad}_f^{n-2} g(x^0) \text{ ad}_f^{n-1} g(x^0)]$  mátrix rangja  $n$ ,
2.  $a D = \text{span}\{g, \text{ad}_f g, \dots, \text{ad}_f^{n-2} g\}$  disztribúció involutív  $x^0$  közelében.

A 2. feltétel  $D$  mátrixát írhatjuk a következő formában

$$D = \text{span}\{g, [f, g], [f, [f, g]], \dots\}$$

és a dimenziója legfeljebb  $n - 1$ , azaz  $\dim D \leq n - 1$ .  $D$  konstruálása során nehéz biztosítani, hogy involutív legyen.

### 2.3.3. A megoldás menete

Az előzőekben tárgyaltak alapján az  $u = \alpha(x) + \beta(x)v$  visszacsatolás és a  $z = \Phi(x)$  koordináta transzformáció megkonstruálásával az Állapottér Linearizációs probléma megoldása a következő lépésekből áll:

1.  $g(x)$ -ből és  $f(x)$ -ből létrehozuk a vektor mezőt

$$g(x) \text{ ad}_f g(x) \dots \text{ad}_f^{n-2} g(x) \text{ ad}_f^{n-1} g(x)$$

és ellenőrizzük az 1. és 2. feltételt.

2. Ha mindkettő teljesül  $\lambda(x)$ -et megkapjuk a (35)-as parciális differenciál egyenlet megoldásával.
3. Kiszámoljuk a visszacsatolási függvényeket

$$\alpha(x) = \frac{-L_f^n \lambda(x)}{L_g L_f^{n-1} \lambda(x)} \quad \beta(x) = \frac{1}{L_g L_f^{n-1} \lambda(x)} \quad (46)$$

4. Kiszámoljuk a koordináta transzformációt

$$\Phi(x) = \text{col}(\lambda(x), L_f \lambda(x), \dots, L_f^{n-1} \lambda(x)). \quad (47)$$

**2.3.2. Definíció (Linearizáló visszacsatolás és koordináták).** A (46)-ben definiált visszacsatolást linearizáló visszacsatolásnak, a (47) alapján megadott új koordinátákat linearizáló koordinátáknak nevezzük.

*Fontos megjegyezni, hogy a SISO nemlineáris rendszerek visszacsatolásos linearizációja nem robusztus, sőt néha meglehetősen érzékeny a paraméterek és a strukturális bizonytalanságok miatt, ezért csak korlátozottan használható folyamat rendszerek alkalmazásaiban*

### 3. Egzakt linearizációs szabályozó struktúra tervezés MIMO rendszerekre

Mivel az egzakt linearizáció csak egy-bemenetű egy-kimenetű (SISO) rendszerekre alkalmazható, ha egy több-bemenetű több kimenetű (MIMO) nemlineáris rendszert szeretnénk ily módon linearizálni, akkor a MIMO rendszert "szét kell vágnunk" olyan SISO részrendszerekre amelyek között a kölcsönhatás minimális. A következőkben egy lehetséges szétvágási módszer kerül leírásra.

#### 3.1. Relatív fokszám vizsgálata struktúra gráfokon

Egy adott bemenet és kimenet pár közötti kölcsönhatást jellemzi a közöttük számolható relatív fokszám. A rendszer struktúra gráfja a változók között lévő kapcsolatokat írja le. Eképpen a rendszer struktúra gráfja és a bemenetek, kimenetek között számolható relatív fokszámok között egyértelmű megfeleltetés létesíthető.

**3.1.1. Definíció (Struktúra gráf).** *Legyen adott a több-bemenetű több-kimenetű (MIMO) nemlineáris rendszer a következő egyenletekkel:*

$$\dot{x} = f(x) + \sum_{j=1}^m g_j(x)u_j + \sum_{k=1}^r w_k(x)d_k \quad (48)$$

$$y_i = h_i(x), i = 1 \dots p \quad (49)$$

ahol  $x \in R^n$  az állapot vektor  $d_k, u_j, y_i \in R$  a zavarás, a bemenet, és a kimeneti változók.

A nemlineáris rendszer struktúra gráfja egy irányított gráf  $G = (V, E)$ , ahol

- a csúcsok halmazát alkotja a bemeneti változók halmazának  $U = (u_1, \dots, u_m)$ , a zavarások halmazának  $D = (d_1, \dots, d_r)$ , az állapotváltozók halmazának  $X = (x_1, \dots, x_n)$  és a kimeneti változók halmazának  $Y = (y_1, \dots, y_p)$  uniója.

Ezek a részhalmazok páronként diszjunktak:

$$V = U \cup D \cup X \cup Y \quad (50)$$

$$X \cap U = X \cap D = X \cap Y = 0 \quad (51)$$

$$U \cap D = U \cap Y = D \cap Y = 0 \quad (52)$$

- az élek  $E$  halmaza leírja a közvetlen összeköttetéseket a csúcsok között. Létezik irányított él  $(v_i, v_j)$   $v_i$ -ből  $v_j$ -be, ha a  $v_i$  változó szerepel a  $v_j$ -t (ha kimenet), vagy  $v_j$ -t (ha állapotváltozó) meghatározó egyenlet jobb oldalán.

A gráfban nem futnak be élek a bemenetekbe, és a kimenetekből nem indulnak ki élek.

**3.1.1. Példa (Hőcserélő hálózat struktúragráfja).** Tekintsük az (1) ábrán adott hőcserélő hálózatot. Az energia mérlegek, amelyek leírják a rendszer dinamikus viselkedését a következő egyenlet struktúrával írhatók le:

$$\frac{dT_1}{dt} = \phi_1(T_1, T_2, T_{10}, F_1) \quad (53)$$

$$\frac{dT_2}{dt} = \phi_2(T_1, T_2, T_{20}, F_2) \quad (54)$$

$$\frac{dT_3}{dt} = \phi_3(T_3, T_4, T_{30}, F_3) \quad (55)$$

$$\frac{dT_4}{dt} = \phi_4(T_3, T_4, T_{40}, F_4) \quad (56)$$

$$\frac{dT_5}{dt} = \phi_5(T_5, T_6, T_{50}, F_5) \quad (57)$$

$$\frac{dT_6}{dt} = \phi_6(T_5, T_6, T_{60}, F_6) \quad (58)$$

ahol

$F_i$  = az  $i$ . ág tömegárama ,

$T_i$  = az  $i$ . ág kimeneti hőmérséklete

$T_{i0}$  = az  $i$ . ág bemeneti hőmérséklete

és  $\phi_i(\cdot)$  a változók egymástól való függését jelöli.

Az  $F_2$  és az  $F_4$  folyam összeadódik, így a következő algebrai egyenletek teljesülnek:

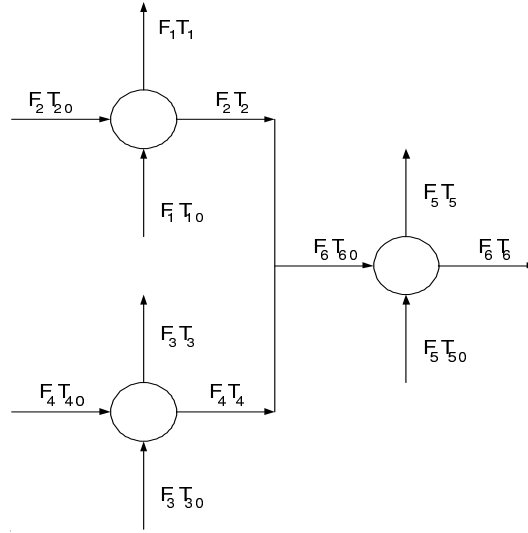
$$F_6 = \phi_7(F_2, F_4)$$

$$T_{60} = \phi_8(T_2, T_4, F_2, F_4).$$

Ezt figyelembe véve az (58) egyenlet a következő alakban írható:

$$\frac{dT_6}{dt} = \phi_9(T_2, T_4, T_5, T_6, F_2, F_4).$$

A szabályozási cél, hogy a kívánt értéken tartsuk  $T_1$  és  $T_6$  hőmérsékleteket. Fő zavarásnak a  $T_{30}$  és  $T_{50}$  hőmérsékleteket tekintjük. A rendelkezésre álló bemenetek az  $F_1$ ,  $F_2$  és  $F_4$  tömegáramok.



1. ábra. Hőcserélő hálózat

Ilyen feltételezésekkel a hőcserélő hálózat struktúra gráfja a 2. ábrán látható.

**3.1.2. Definíció (Relatív fokszám MIMO rendszerekre).** Tekintsük a (48), (49) egyenletekkel megadott MIMO nemlineáris rendszert.  $r_{ij}$  a relatív fokszám az  $(u_j, y_i)$  által meghatározott SISO rendszerben az  $x^0$  pontban, ha

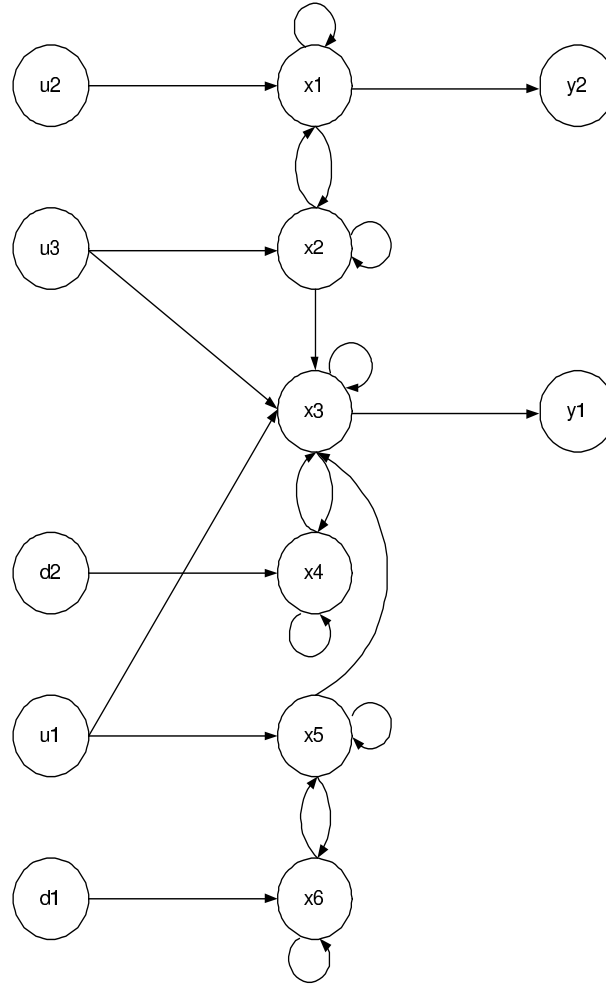
1.  $L_g L_f^k h_i(x) = 0$  minden  $x$ -re  $x^0$  környezetében  $\forall k < r_{ij} - 1$ -re, és
2.  $L_g L_f^{r_{ij}-1} h_i(x) \neq 0$

Egy MIMO nemlineáris rendszerhez megadhatjuk annak relatív fokszám mátrixát a következőképpen:

$$M_r = \begin{bmatrix} r_{11} & \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mp} \end{bmatrix} \quad (59)$$

amelynek elemei az  $r_{ij}$  relatív fokszámok az összes lehetséges bemenet és kimenet pár között.

Ha az adott bemenet és kimenet között a struktúra gráfban nem létezik út, akkor a megfelelő relatív fokszám  $r_{ij} = \infty$ .



2. ábra. A hőcserélő hálózat struktúra gráfja

**3.1.1. Lemma (Relatív fokszám MIMO rendszerekre a struktúra gráfon).**

*Tekintsük a (48), (49) egyenletekkel megadott MIMO nemlineáris rendszert és a neki megfelelő struktúra gráfot. Legyen  $l_{ij}$  a legrövidebb út  $u_j$  és  $y_i$  között. Legyen  $r_{ij}$  a relatív fokszám  $u_j$  és  $y_i$  között. Ekkor a következő egyenlőség teljesül:  $r_{ij} = l_{ij} - 1$ .*

Tehát egy adott  $u_j$  és  $y_i$  bemenet között számolható  $r_{ij}$  relatív fokszám azt jelenti, hogy a struktúra gráfban a bemenettől a kimenetig hány állapotváltozón megy keresztül a közöttük lévő legrövidebb irányított út.

**Bizonyítás:** Ha a relatív fokszám 3.1.2. Definícióját összevetjük azzal, ahogyan a

2.1.1. Lemmában a kimeneti egyenlet  $k$ -adik deriváltját számoljuk, akkor a következőt láthatjuk:

- $k = 0$ :  $y_i = h_i(x)$

$$\frac{dy_i}{dt} = \frac{\partial h_i}{\partial x} \dot{x} = \frac{\partial h_i}{\partial x} (f(x) + g(x)u) = L_f h_i(x) + L_g h_i(x) \cdot u$$

- $k = 1$  feltéve, hogy  $L_g h_i = 0$ , akkor  $\dot{y} = L_f h_i$ , és

$$\frac{d^2 y_i}{dt^2} = \frac{\partial (L_f h_i)}{\partial x} (f(x) + g(x)u) = L_f^2 h_i + L_g L_f h_i \cdot u$$

- és így tovább.

Ekkor a relatív fokszám 3.1.2. Definíciója szerint  $\forall k < r_{ij} - 1$ -re  $L_g L_f^k h(x) = 0$ , azaz a kimeneti egyenlet  $\forall k < r_{ij} - 1$ -re a következő alakú lesz:

$$\frac{d^k y_i}{dt^k} = L_f^k h_i$$

és  $k = r_{ij} - 1$ -re a kimeneti egyenlet a következő alakú lesz:

$$\frac{d^k y_i}{dt^k} = L_f^k h_i + L_g L_f^{k-1} h_i \cdot u$$

Ezek a deriváltak meghatározzák a struktúragráfban az  $u_j$ -ből az  $y_i$ -be vezető úton lévő állapotokat a következő módon:

$$u_j \rightarrow y_i^{(r_{ij})} \rightarrow \dots \rightarrow \ddot{y}_i \rightarrow \dot{y}_i \rightarrow y_i$$

Ez az út a legrövidebb  $u_j$  és  $y_i$  között, mert ha lenne olyan  $l_{ij} < r_{ij} + 1$ , akkor lenne olyan  $L_g L_f^{l_{ij}-2} \neq 0$ , de ez  $r_{ij}$  definíciójából következően ellentmondás, azaz  $l_{ij} = r_{ij} - 1$ .

Fontos megjegyezni, hogy a relatív fokszám kiszámításával csak a rendszer struktúrais tulajdonságairól kapunk információt. Így a relatív fokszám kevesebb információt tartalmaz, mint az állapottér modell, mert például nem tartalmaz információt a rendszer numerikus paramétereiről. A relatív fokszám azt jellemzi, hogy mennyire közvetlen a kölcsönhatás a bemenetek és a kimenetek között.

### 3.2. Folyamatrendszerek maximális relatív fokszámmal

Számos tipikus folyamatrendszer rendelkezik maximális relatív fokszámmal, azaz  $r = n$ . Folyamatrendszereknél ezek megtalálására felhasználjuk a rendszer struktúra gráfját.

**3.2.1. Lemma.** *Egy SISO rendszer relatív foka  $r = n$ , ha*

1. *A bemenettől a kimenetig vezető legrövidebb út átmegy az összes állapotváltozón.*
2. *A bemenet és a kimenet között számolható  $(n - 1)$ -ed rendű Lie-derivált nem nulla:*

$$L_g L_f^{n-1} h(x) \neq 0$$

A fenti két feltétel a 3.2.1. Lemma szerint ekvivalens egymással, így az egzakt linearizáció 2.2.1. Lemmában leírt alapesete áll elő: a koordináta transzformációt elvégezve a rendszer az új koordinátákban lineáris lesz és irányítható.

**Bizonyítás:**

1. Ha a bemenet és a kimenet között a legrövidebb út átmegy az összes állapotváltozón, akkor ez azt jelenti, hogy a bemenet és a kimenet között nincsen rövidebb út, amely csak néhány állapotváltozón menne át.
2. A 3.1.1. Lemmából egyértelműen következik.

### 3.3. Súlyozott páros gráf struktúratervezésre

Ha egzakt linearizációs szabályozó struktúrát szeretnénk alkalmazni egy MIMO rendszerrel, akkor a rendszer struktúra gráfjában találnunk kell a 3.2.1. Tételben leírt tulajdonságokkal rendelkező részgráfokat. Ezeknek az  $SG_i$  részgráfoknak diszjunkt utakat kell alkotniuk a rendszer struktúra gráfjában:

$$\begin{aligned} SG^{(i)} &= (V^{(i)}, E^{(i)}) \\ E^{(i)} &= \{(v_1, v_2) \in E \mid v_1, v_2 \in V^{(i)}\}, \quad i = 1, \dots, p \\ V^{(i)} \cap V^{(j)} &= \emptyset \quad \forall 1 \leq i < j \leq p, \end{aligned}$$



Az egzakt linearizációs szabályozó struktúra megtalálásához tehát, a bemenetek és kimenetek egy olyan párosítását keressük, amellyel a részrendszerek számát maximalizáljuk úgy, hogy azok között a kölcsönhatás minimális. Az összes lehetséges diszjunkt utat meghatározhatjuk, ha kiszámoljuk a rendszer relatív fokszám mátrixát. Annak érdekében, hogy megkapjuk a szabályozó struktúrákat az  $M_r$  relatív fokszám mátrixot megfeleltetjük egy súlyozott páros gráfnak, amelyben maximális párosítást keresünk.

A szabályozó struktúra tervezéséhez használt  $G = (U \cup Y, E)$  súlyozott páros gráf a következőképpen adott:

- $U$ : bemenetek,  $Y$ : kimenetek, két független csúcshalmaz;
- $E$ : élek halmaza. Létezik él  $u_j$ -ből  $y_i$ -be  $w_{ij}$  súllyal, ha az  $M_r$  relatív fokszám mátrix megfelelő  $r_{ij}$  eleme nem egyenlő  $\infty$ -el.
- Az  $e_{ij} = (u_j, y_i)$  él súlyát a relatív fokszám mátrix  $r_{ij}$  eleméből számoljuk a 3.5.1. alfejezetben később leírt módon.

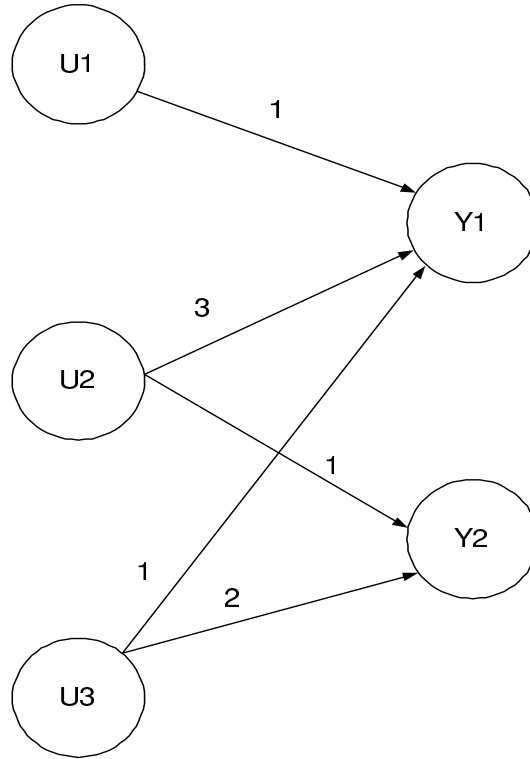
**3.3.1. Példa.** Az 1. ábrán látható hőcserélő hálózat relatív fokszám mátrixa a következőképpen adható meg:

$$M_r = \begin{bmatrix} 1 & \infty \\ 3 & 1 \\ 1 & 2 \end{bmatrix} \quad (60)$$

A 2. ábrán látható struktúra gráfban ugyanis az összes bemenettől az összes kimenetig vezető legrövidebb utak a következőképpen írhatók le:

$$\begin{aligned} l_{11} & : u_1 \rightarrow x_3 \rightarrow y_1 \\ l_{12} & : - - \\ l_{21} & : u_2 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow y_1 \\ l_{22} & : u_2 \rightarrow x_1 \rightarrow y_2 \\ l_{31} & : u_3 \rightarrow x_3 \rightarrow y_1 \\ l_{32} & : u_3 \rightarrow x_2 \rightarrow x_1 \rightarrow y_1 \end{aligned}$$

Ezek az utak a 3.1.1. Lemma szerint meghatározzák a relatív fokszám mátrix elemeit, amelyeket megfeleltetünk egy súlyozott páros gráf éleinek. Ez látható a 3. ábrán.



3. ábra. A relatív fokszám mátrixnak megfelelő súlyozott páros gráf

### 3.4. Maximális párosítás keresésének feltételei

Feltételezzük, hogy a kimenetek száma  $p$  kisebb, vagy egyenlő a bemenetek számánál  $m$ :  $p \leq m$ . Ehhez  $(m \times p)$ -es relatív rend mátrixból egy  $(p \times p)$ -es részre van szükségünk. Ennek a mátrixnak a főátlójában nem lehet  $\infty$  relatív fokszám, mert akkor van olyan kimenet amelyhez nincs megfelelő bemenet.

#### 3.4.1. A főátlóban lévő elemek kiválasztása

Ezek közül a  $(p \times p)$ -es mátrixok közül azt a mátrixot fogjuk kiválasztani, amelynek főátlójában maximalizáltuk:

- Az 1-es relatív fokszámok számát, majd a maradékban
- A 2-es relatív fokszámok számát, majd a maradékban
- ...
- A  $i$ -es relatív fokszámok számát, addig

amíg a lehetséges bemenet - kimenet párok maximális számát ( $p$ ) el nem érjük. Így azokat a részrendszereket kaptuk meg, amelyeken belül a bemenet és a kimenet között a kölcsönhatás a legerősebb.

### 3.4.2. Domináns párok kölcsönhatásának minimalizálása

Miután az  $(p \times p)$ -es mátrix főátlójában maximalizáltuk sorban az 1, 2, ...,  $i$ -es relatív fokszámokat, törekednünk kell arra, hogy a domináns bemenet kimenet párosítások minél kevésbé hassanak egymásra. Ha az előző részben leírt feltételek teljesítése után több lehetséges  $(p \times p)$ -es relatív fokszám mátrixunk marad, akkor ezek közül azt választjuk, amelynek főátlóján kívül a legtöbb  $\infty$  relatív fokszám található.

A 3.4. fejezetben leírt feltételek a következő egyszerű példán szemléletesen láthatók.

#### 3.4.1. Példa (Lehetséges párosítások közül a legmegfelelőbb kiválasztása).

*A (60). egyenlettel megadott relatív fokszám mátrixból 3 lehetséges párosítást választhatunk ki:*

- 1. konfiguráció:  $u_1/y_1, u_2/y_2$

$$M_{r1} = \begin{bmatrix} 1 & \infty \\ 3 & 1 \end{bmatrix}$$

- 2. konfiguráció:  $u_1/y_1, u_3/y_2$

$$M_{r2} = \begin{bmatrix} 1 & \infty \\ 1 & 2 \end{bmatrix}$$

- 3. konfiguráció:  $u_3/y_1, u_2/y_2$

$$M_{r3} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$

Látható, hogy a 3.4.1-es fejezetben megadott feltételnek csak az 1. és 3. konfiguráció tesz eleget. Ugyanis a 2. konfiguráció mátrixának főátlójában nem a lehetséges legtöbb 1-es relatív fokszám található.

A maradék két konfiguráció közül az 1-es a legmegfelelőbb, mivel ennél a párosításnál a legkisebb a részrendszerek között a kölcsönhatás. Az 1. konfigurációban az  $u_1$ -es bemenet csak az  $y_1$ -es kimenetre hat, az  $y_2$ -re nincs hatással. Míg a 3. konfigurációban az  $u_3$ -as bemenet az  $y_1$ -es kimeneten kívül hat az  $y_2$ -es kimenetre is az  $x_2 \rightarrow x_1$  állapotokon keresztül.

### 3.5. Algoritmus a maximális párosítás megkeresésére

A következőkben leírásra kerülő algoritmus a 3.4. fejezetben leírt feltételeknek megfelelően megkeresi a relatív foks szám mátrixnak megfeleltetett súlyozott páros gráfban a lehetséges legtöbb bemenet-kimenet párt.

#### 3.5.1. A páros gráf élsúlyainak meghatározása

Az algoritmus maximális párosítást keres maximális élsúlyokkal, ezért a relatív foks szám mátrixnak megfeleltetett súlyozott páros gráf élsúlyait (amelyek eredetileg a relatív foks szám mátrix elemei voltak) meg kell változtatnunk, hogy a 3.4. fejezet feltételei teljesüljenek a megfelelő párosítás keresése során. Az új  $w_{ij}$  élsúlyokat úgy határozzuk meg, hogy növekvő relatív foks számhoz csökkenő élsúly tartozik.

- Első lépésként meghatározzuk az új élsúlyok minimális és maximális értékét.

Vezessük be az alábbi jelöléseket:

$p$  - a kimenetek maximális száma;

$t$  - a legnagyobb relatív foks szám;

$\min(w_{ij}) = k$  - a legnagyobb relatív rendből származtatott új élsúly

$\max(w_{ij}) = k + t$  - az 1-es relatív foks számából származtatott új élsúly

A 3.4.1. fejezetben leírt feltételt, mely szerint a relatív foks szám mátrixból kiválasztott részmátrix főátlójában maximalizáljuk sorban az 1-es, 2-es stb. relatív foks számokat, a páros gráf új élsúlyaira vonatkoztatva következőképpen írhatjuk:

$$(p - 1) \cdot (k + t) < p \cdot k \quad (61)$$

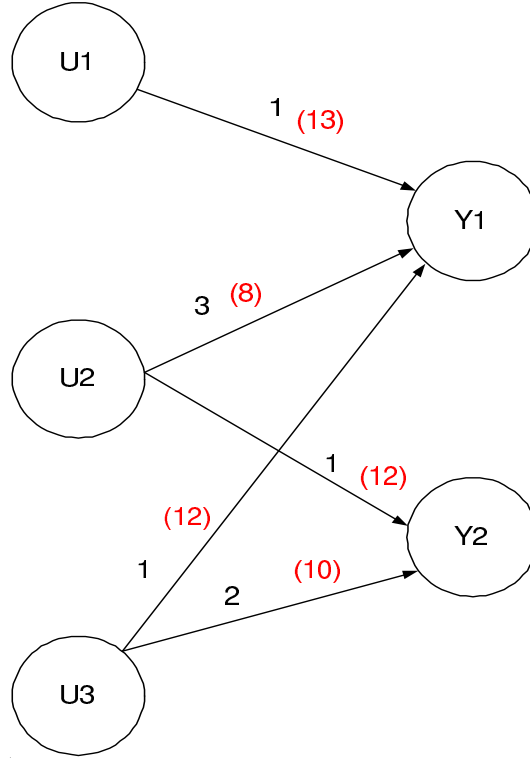
A (61). egyenletet kifejezve kapjuk:

$$k > \frac{t}{\frac{p}{p-1} - 1}$$

Ebből:

$$k > \frac{t}{\frac{1}{p-1}}$$

$$k + t > t \cdot p$$



4. ábra. A súlyozott páros gráf a maximális párosítás megkereséséhez való új élsúlyokkal

Ennek alapján a legkisebb relatív fokszámból származtatott legnagyobb  $\max(w_{ij})$  élsúlynak  $t \cdot p$  megfelel. A növekvő relatív fokszámokhoz csökkenő élsúlyokat rendelünk úgy, hogy 1-el csökkentjük az élsúlyokat a relatív fokszám 1-el történő növelésekor.

- A 3.4.2 fejezet feltételét úgy tudjuk figyelembe venni, hogy az így kapott élsúlyokat beszorozzuk  $p$ -vel, és mindegyikhez hozzáadjuk a relatív fokszám mátrix adott sorában ( $j$ -edik) lévő  $\infty$ -ek számát. Így maximalizálni tudjuk a főátlón kívüli  $\infty$ -ek számát ( $v_{\infty j}$ ).

Ezzel megkaptuk a relatív fokszám mátrixnak megfelelő páros gráf végleges élsúlyait, tehát pl: az 1-es relatív fokszámokhoz tartozó új élsúlyok  $t \cdot p^2 +$  a relatív fokszám mátrix adott sorában lévő  $\infty$ -ek száma.

$$w_{ij} = (tp - r_{jj} + 1)p + v_{\infty j} \quad (62)$$

$$w_{ij} = \begin{cases} tp^2 + v_{\infty j} & , ha \quad r_{jj} = 1 \\ (tp - 1)p + v_{\infty j} & , ha \quad r_{jj} = 2 \\ (tp - 1)p + v_{\infty j} & , ha \quad r_{jj} = 3 \end{cases} \quad (63)$$

### 3.5.2. A maximális párosítás megkeresése

A következő célfüggvény maximalizálásával megkapjuk az egzakt linearizációs szabályozó struktúrához szükséges bemenet-kimenet párokat.

- $x_{ij} = 1$ , ha  $ij$  eleme a párosításnak,
- $x_{ij} = 0$  egyébként.

$w_{ij}$  = él  $i$  és  $j$  csúcsok között.

A célfüggvény:

$$\max z = \sum_i \sum_j w_{ij} x_{ij} \quad (64)$$

$$\begin{aligned} \sum_{i=1}^n x_{ij} &= 1, \forall 1 \leq j \leq n \\ \sum_{j=1}^n x_{ij} &= 1, \forall 1 \leq i \leq n \\ x_{ij} &\in \{0, 1\} \end{aligned}$$

**3.5.1. Példa.** A 1. képen látható hőcserélő hálózat súlyozott páros gráfja látható a 4. ábrán az új élsúlyokkal:

$p = 2$  - a kimenetek száma,

$t = 3$  - a relatív fokszám mátrix legnagyobb eleme.

$w_{11} = (t * p) * p + 1 = (3 * 2) * 2 + 1 = 13$   $r_{11} = 1$ ,  $\infty$ -k száma az 1. sorban = 1

$w_{21} = (t * p - 2) * p = (6 - 2) * 2 = 8$   $r_{21} = 3$ ,  $\infty$ -k száma az 2. sorban = 0

$w_{22} = (t * p) * p = 12$   $r_{22} = 1$ ,  $\infty$ -k száma az 2. sorban = 0

$w_{31} = (t * p) * p = 12$   $r_{31} = 1$ ,  $\infty$ -k száma az 3. sorban = 0

$w_{32} = (t * p - 1) * p = 10$   $r_{32} = 2$ ,  $\infty$ -k száma az 3. sorban = 0

### 3.6. Részrendszerek kapcsolata

Az előzőekben ismertetett módszerrel meghatározhatjuk egy MIMO rendszernél azokat a SISO részrendszereket, amelyek segítségével egzakt linearizációs szabályozó struktúrát tudunk létrehozni.

Egy adott részrendszer az általa meghatározott bemenet-kimenet pár között egy erős, nemlineáris kölcsönhatást valósít meg. Felmerül azonban a kérdés, hogy a részrendszerek hatnak-e egymásra, és ha igen mennyire. A részrendszerek közötti kölcsönhatást és annak erősségét tükrözik a kiválasztott párosításhoz tartozó relatív fokszám mátrix off-diagonális elemei. Ez a kölcsönhatás annál gyengébb minél nagyobbak az off-diagonális elemek. Ideális esetben, ha nincs kölcsönhatás a részrendszerek között, a relatív fokszám mátrix főátlóján kívül az összes elem  $\infty$ .

$$M_r = \begin{bmatrix} r_1 & \infty & \dots & \infty \\ \infty & r_2 & \dots & \infty \\ \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \dots & r_p \end{bmatrix}$$

Valós rendszereknél ez igen ritkán fordul elő, viszont a 3. fejezetben leírt módszer amellett, hogy kiválasztja azokat a bemenet-kimenet párokat, amelyekben belül a kölcsönhatás a legerősebb, ügyel arra is, hogy a részrendszerek közötti kölcsönhatás a lehető legkisebb legyen.

A 1. ábrán látható hőcserélő hálózatban kiválasztott egzakt linearizációs szabályozó struktúra elemei az  $u_1/y_1$ ,  $u_2/y_2$  bemenet-kimenet párok, amelyek a legkevésbé hatnak egymásra a lehetségesek közül. A kiválasztott párosításnak megfelelő relatív fokszám mátrix:

$$M_r = \begin{bmatrix} 1 & \infty \\ 3 & 1 \end{bmatrix}$$

A 2. ábrán látható struktúra gráfról leolvasható, hogy az  $u_1$  bemenet nincs hatással az  $y_2$  kimenetre. Viszont az  $u_2$  bemenet az  $x_1 \rightarrow x_2 \rightarrow x_3$  állapotokon keresztül hat az  $y_1$  kimenetre. Ez a hatás azonban a leggyengébb a lehetségesek közül.

## 4. Esettanulmányok

Az egzakt linearizációs szabályozó struktúra megkeresését e fejezetben több esettanulmány segítségével mutatom be.

### 4.1. Hőcserélő cella modellje

Az először vizsgált példa egy hőcserélő hálózat, amelyet hőcserélő cellák összekapcsolásával kapunk. Az energia mérlegek, amelyek egy hőcserélő cella dinamikus viselkedését írják le a következők:

$$\begin{aligned}\frac{dT_h}{dt} &= \frac{F_h}{V_h}(T_{h0} - T_h) + \frac{UA}{c_{ph}\rho_h V_h}(T_m - T_h) \\ \frac{dT_m}{dt} &= \frac{F_m}{V_m}(T_{m0} - T_m) + \frac{UA}{c_{pm}\rho_m V_m}(T_h - T_m)\end{aligned}$$

ahol

- $h$  - hideg oldal,
- $m$  - meleg oldal,
- $T_{h0}$  - hideg oldal bemeneti hőmérséklete,
- $T_{m0}$  - meleg oldal bemeneti hőmérséklete,
- $T_h$  - hideg oldal kimeneti hőmérséklete,
- $T_m$  - meleg oldal kimeneti hőmérséklete,
- $F_h$  - hideg oldal tömegárama,
- $F_m$  - meleg oldal tömegárama,
- $V_h$  - hideg oldal térfogata,
- $V_m$  - meleg oldal térfogata,
- $c_{ph}$  - hideg oldal fajhője,
- $c_{pm}$  - meleg oldal fajhője,
- $\rho_h$  - hideg oldal sűrűsége,
- $\rho_m$  - meleg oldal sűrűsége,
- $A$  - hőátadási felület,
- $U$  - hőátadási együttható.

A hőcserélő cella modelljében mind a hideg ( $V_h$ ), mind a meleg oldalt ( $V_m$ ) egy tökéletesen kevert tartálynak tekintjük, amelyek között az  $A$  hőátadási felületen

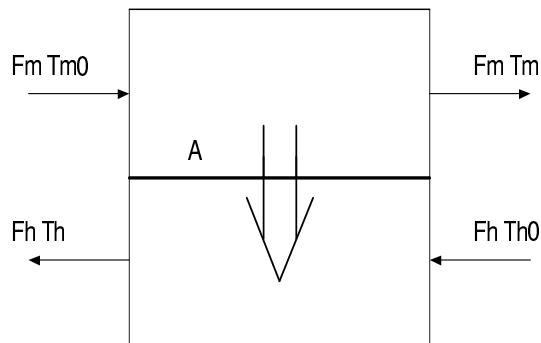


keresztül hőenergia adódik át a meleg oldalról a hideg oldalra. Az  $F_h$ ,  $F_m$  tömegáramokat állandónak tekintjük. Feltételezzük, hogy nincs hőveszteség és az  $U$  hőátadási együttható állandó.

A hőcserélő cella modellje az 5. ábrán látható.

Például egy konkrét hőcserélő cella paraméterei a következők lehetnek:

$$\begin{aligned} F_h &= 0.315 * 10^{-3} \frac{m^3}{s} \\ F_m &= 0.19 * 10^{-3} \frac{m^3}{s} \\ V_h &= 6.05 * 10^{-3} m^3 \\ V_m &= 3.2 * 10^{-3} m^3 \\ c_{ph} &= 1910 \frac{J}{Kkg} \\ c_{pm} &= 1590 \frac{J}{Kkg} \\ \rho_h &= 1000 \frac{kg}{m^3} \\ \rho_m &= 1000 \frac{kg}{m^3} \\ A &= 0.633 m^2 \\ U &= 476 \frac{J}{m^2 K s} \end{aligned}$$



5. ábra. Hőcserélő cella modellje

## 4.2. Szabályozó struktúra kiválasztása a hőcserélő hálózaton

A 6. ábrán látható hőcserélő hálózaton vizsgáljuk a szabályozó struktúra kiválasztását. Az energiamérlegek, amelyek leírják a rendszer dinamikus viselkedését, a

következő egyenletstruktúrával adottak:

$$\frac{dT_1}{dt} = \phi_1(T_1, T_2, T_{1\ 0}, F_1) \quad (65)$$

$$\frac{dT_2}{dt} = \phi_2(T_1, T_2, T_{2\ 0}, F_2) \quad (66)$$

$$\frac{dT_3}{dt} = \phi_3(T_3, T_4, T_{3\ 0}, F_3) \quad (67)$$

$$\frac{dT_4}{dt} = \phi_4(T_3, T_4, T_{4\ 0}, F_4) \quad (68)$$

$$\frac{dT_5}{dt} = \phi_5(T_5, T_6, T_{5\ 0}, F_5) \quad (69)$$

$$\frac{dT_6}{dt} = \phi_6(T_5, T_6, T_{6\ 0}, F_6) \quad (70)$$

$$\frac{dT_7}{dt} = \phi_7(T_7, T_8, T_{7\ 0}, F_7) \quad (71)$$

$$\frac{dT_8}{dt} = \phi_8(T_7, T_8, T_{8\ 0}, F_8) \quad (72)$$

$$\frac{dT_9}{dt} = \phi_9(T_1, T_2, T_{9\ 0}, F_9) \quad (73)$$

$$\frac{dT_{10}}{dt} = \phi_{10}(T_1, T_2, T_{10\ 0}, F_{10}) \quad (74)$$

ahol

$F_i$  = az  $i$ . ág tömegárama

$T_i$  = az  $i$ . ág kimeneti hőmérséklete

$T_{i\ 0}$  = az  $i$ . ág bemeneti hőmérséklete

és  $\phi_i(\cdot)$  a változók egymástól való függését jelöli.

Az  $F_2$  és az  $F_4$  tömegáramok összeadódnak, így a következő algebrai egyenletek teljesülnek:

$$F_6 = \phi_{11}(F_2, F_4)$$

$$T_{6\ 0} = \phi_{12}(T_2, T_4, F_2, F_4).$$

Ezt figyelembe véve a (70) egyenlet a következő alakban írható:

$$\frac{dT_6}{dt} = \phi_{13}(T_2, T_4, T_5, T_6, F_2, F_4).$$

Az  $F_6$  és az  $F_8$  tömegáramok szintén összeadódnak, így a következők teljesülnek:

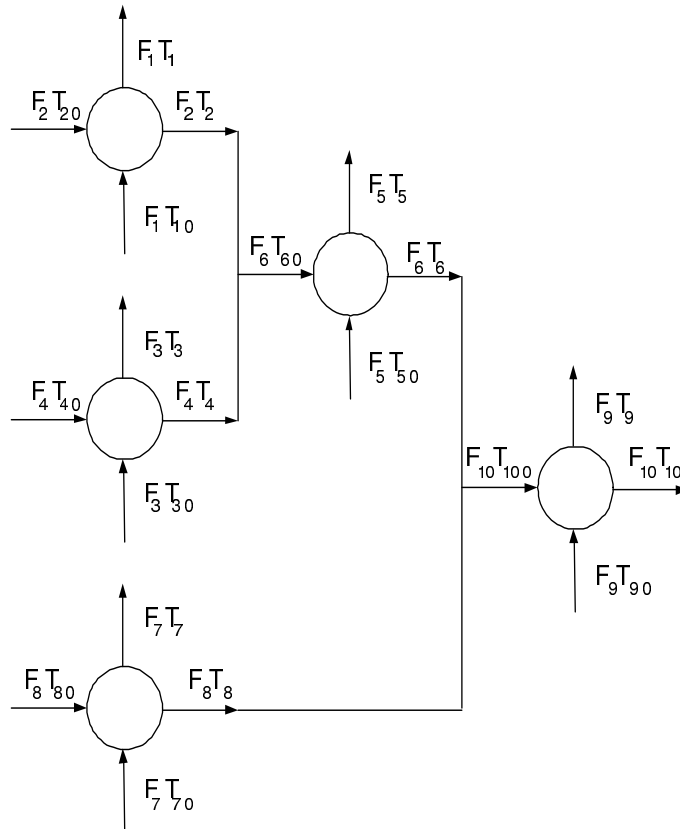
$$F_{10} = \phi_{14}(F_6, F_8) = \phi_{15}(F_2, F_4, F_8)$$

$$T_{10\ 0} = \phi_{16}(T_6, T_8, F_6, F_8) = \phi_{17}(T_2, T_4, T_8, F_2, F_4, F_8)$$

Így a (74) egyenlet a következő alakban írható:

$$\frac{dT_{10}}{dt} = \phi_{18}(T_2, T_4, T_8, T_9, T_{10}, F_2, F_4, F_8)$$

A szabályozási cél, hogy a kívánt értéken tartsuk a  $T_1$ ,  $T_5$  és  $T_{10}$  hőmérsékleteket, amelyek rendre az  $y_1$ ,  $y_2$ ,  $y_3$  kimenetek. Fő zavarásnak a  $T_{30}$ ,  $T_{70}$  és  $T_{90}$  hőmérsékleteket tekintjük ( $d_1$ ,  $d_2$ ,  $d_3$ ). A rendelkezésre álló bemenetek az  $F_1$ ,  $F_2$ ,  $F_4$ ,  $F_5$  és  $F_8$  tömegáramok ( $u_1$ ,  $u_2$ ,  $u_3$ ,  $u_4$ ,  $u_5$ ).



6. ábra. Hőcserélő hálózat

A szabályozó struktúra kiválasztásához először a hőcserélő hálózat 7. ábrán látható struktúragráfján megkeressük a bemenetektől a kimenetekig vezető legrövidebb utakat. A 3.1.1. Lemma szerint így megkapjuk a hőcserélő hálózat relatív fokszám

mátrixát:

$$M_r = \begin{bmatrix} 1 & 4 & 3 \\ 2 & 2 & 1 \\ \infty & 2 & 1 \\ \infty & 1 & \infty \\ \infty & \infty & 1 \end{bmatrix}$$

amelyet megfeleltetünk egy súlyozott páros gráfnak (8. ábra). Ebben a páros gráfban a 3.4. fejezetben leírt feltételeknek megfelelően maximális párosítást keresünk, amelyhez a súlyozott páros gráfot a 3.5.1. fejezetben ismertetett módon új élsúlyokkal látjuk el.

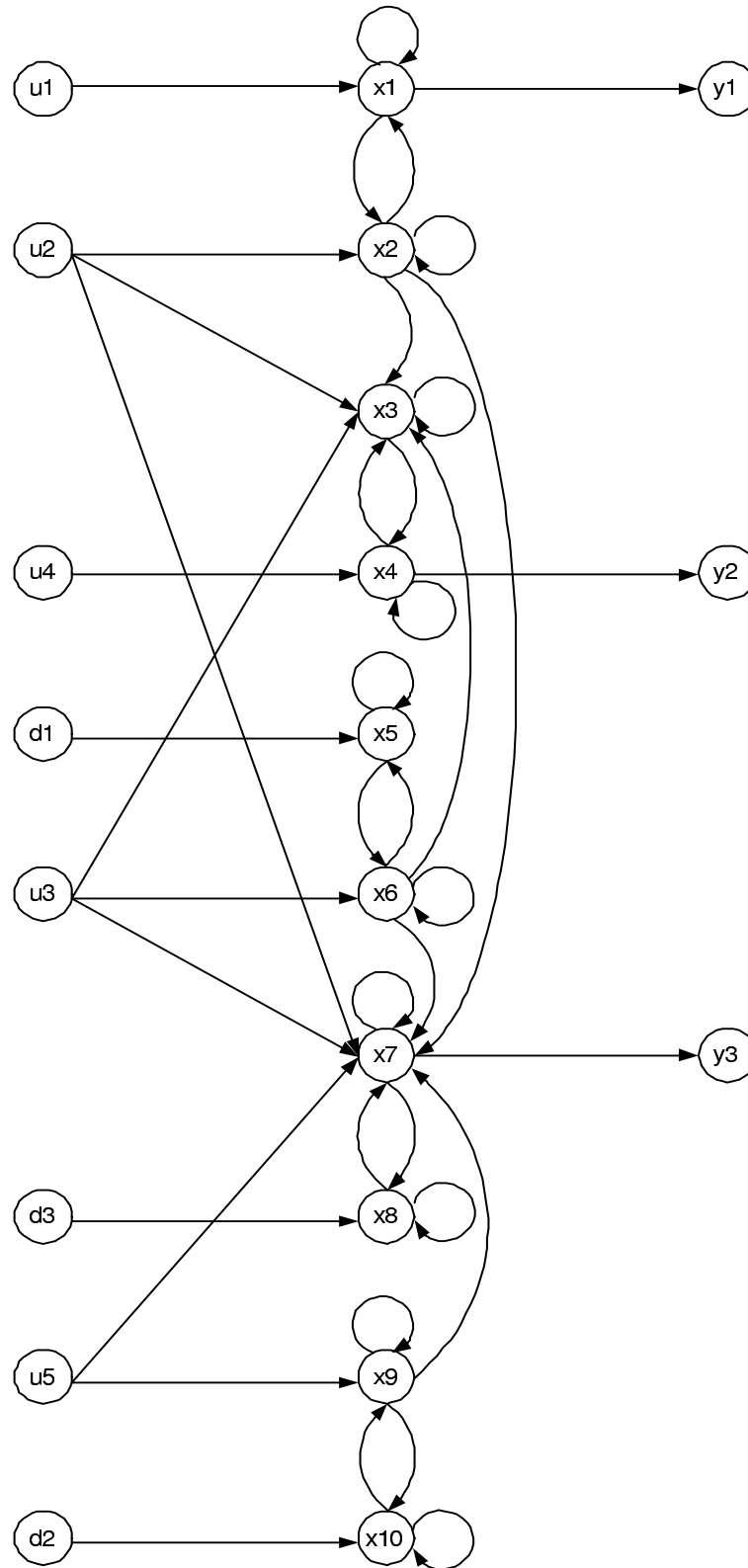
Ezek után a maximális súlyú párosítást kiválasztva megkapjuk a legmegfelelőbb egzakt linearizációs szabályozó struktúrát, amely a következő lesz:

$$u_1/y_1, u_4/y_2, u_5/y_3.$$

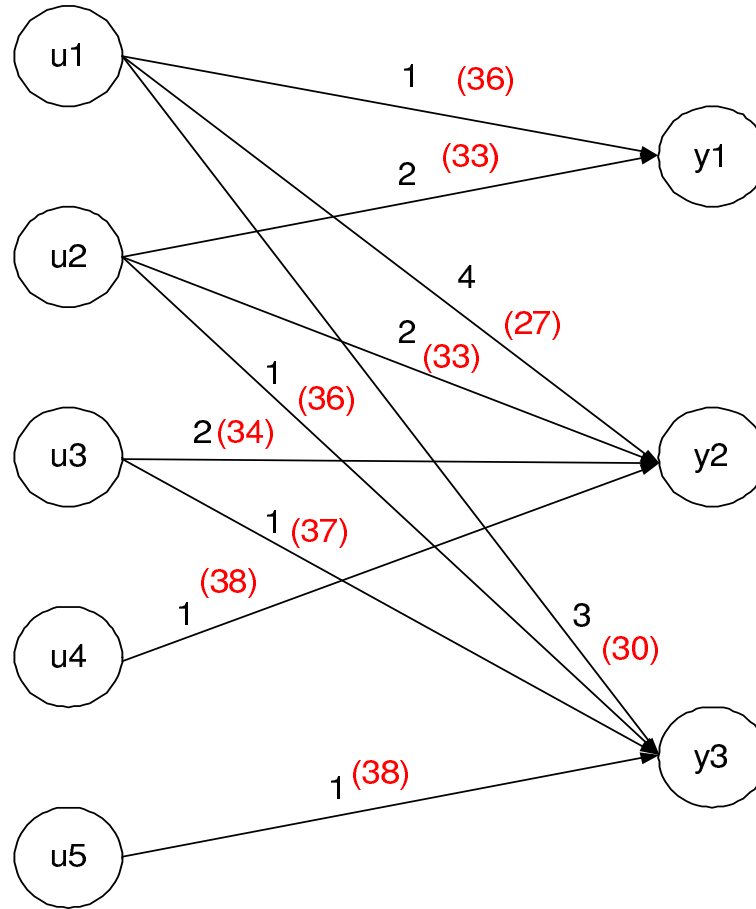
### 4.3. Folytonosan kevert tartály reaktor

Tekintsük a 9. ábrán adott folytonosan kevert tartály reaktort. Az  $A$  és  $B$  anyagot tartalmazó oldatot táplálunk be a reaktorba  $F_A$  és  $F_B$  tömegáramokkal,  $T_A$ ,  $T_B$  hőmérsékleteken és  $c_{A0}$ ,  $c_{B0}$  koncentrációval. A reaktorban  $A+B \rightarrow C+D$  elsőrendű reakció játszódik le. A kimeneti folyam  $F$  térfogatárammal,  $c_A$ ,  $c_B$ ,  $c_C$ ,  $c_D$  koncentrációkkal és  $T$  hőmérséklettel hagyja el a reaktort. A megfelelő fűtő vagy hűtő rendszert használva  $Q$  hőmennyiséget táplálhatunk be, vagy vehetünk el a rendszerből. Feltételezzük, hogy a  $\rho$  sűrűség és a  $C_p$  hőkapacitás a folyadéokban állandó; a hőveszteséget elhanyagoljuk. A tömeg és energia mérlegek, amelyek leírják a rendszer dinamikus viselkedését, a következők:

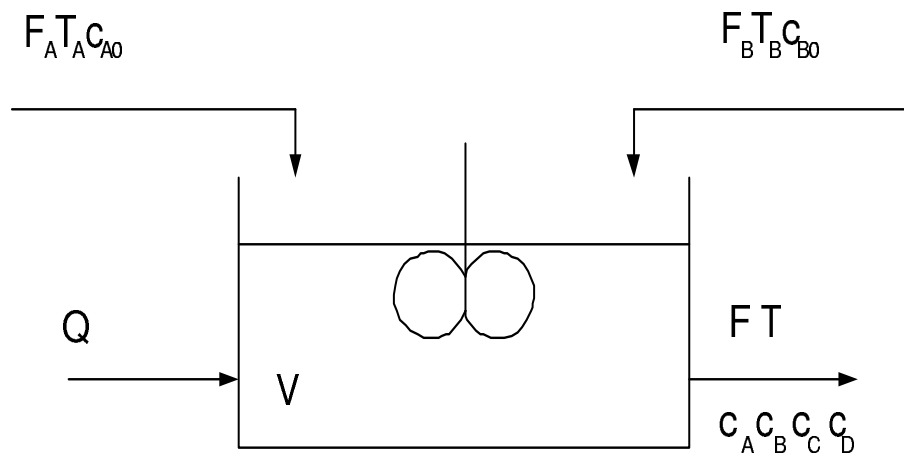
$$\begin{aligned} \frac{dV}{dt} &= F_A + F_B + F \\ \frac{dc_A}{dt} &= \frac{F_A}{V}(c_{A0} - c_A) - c_A \frac{F_B}{V} - kc_{AC}c_B e^{-\frac{E}{RT}} \\ \frac{dc_B}{dt} &= \frac{F_B}{V}(c_{B0} - c_B) - c_B \frac{F_A}{V} - kc_{AC}c_B e^{-\frac{E}{RT}} \\ \frac{dc_C}{dt} &= -c_C \frac{F_A + F_B}{V} + kc_{AC}c_B e^{-\frac{E}{RT}} \\ \frac{dT}{dt} &= \frac{F_A}{V}(T_A - T) + \frac{F_B}{V}(T_B - T) + \frac{-(\Delta H)}{\rho C_p} kc_{AC}c_B e^{-\frac{E}{RT}} + \frac{1}{V\rho C_p} Q \end{aligned}$$



7. ábra. Hőcserélő hálózat struktúragráfja



8. ábra. A relatív fokszám mátrixnak megfelelő páros gráf a maximális párosítás megkereséséhez szükséges élsúlyokkal



9. ábra. Folytonosan kevert tartály reaktor

ahol

$$\begin{aligned}
 C_p &= \text{hőkapacitás,} \\
 E &= \text{aktivációs energia,} \\
 F_A, F_B, F &= \text{tömegáramok,} \\
 Q &= \text{hő betáplálás a tartályba,} \\
 T_A, T_B, T &= \text{hőmérsékletek,} \\
 V &= \text{térfogat,} \\
 -\Delta H &= \text{reakció hő,} \\
 c_i &= \text{az } i. \text{ alkotórész moláris koncentrációja,} \\
 \rho &= \text{sűrűség.}
 \end{aligned}$$

A tartályban lévő folyadék  $V$  térfogatát, a kimeneti folyam  $c_A$ ,  $c_C$  koncentrációit és  $T$  hőmérsékletét szeretnénk szabályozni. A rendelkezésre álló bemeneti változók az  $F_A$ ,  $F_B$ ,  $F$  térfogatáramok és a  $Q$  hőbevitel.

Így kapjuk a következőket:

$$\begin{aligned}
 \text{az állapot változók: } x_1 &= V - V_s, x_2 = c_A - c_{AS}, x_3 = c_B - c_{BS}, \\
 x_4 &= c_C - c_{CS}, x_5 = T - T_S; \\
 \text{a bemenetek: } u_1 &= F_A - F_{AS}, u_2 = F_B - F_{BS}, u_3 = F - F_S, u_4 = Q - Q_S; \\
 \text{a kimenetek: } y_1 &= x_1, y_2 = x_2, y_3 = x_4, y_4 = x_5
 \end{aligned}$$

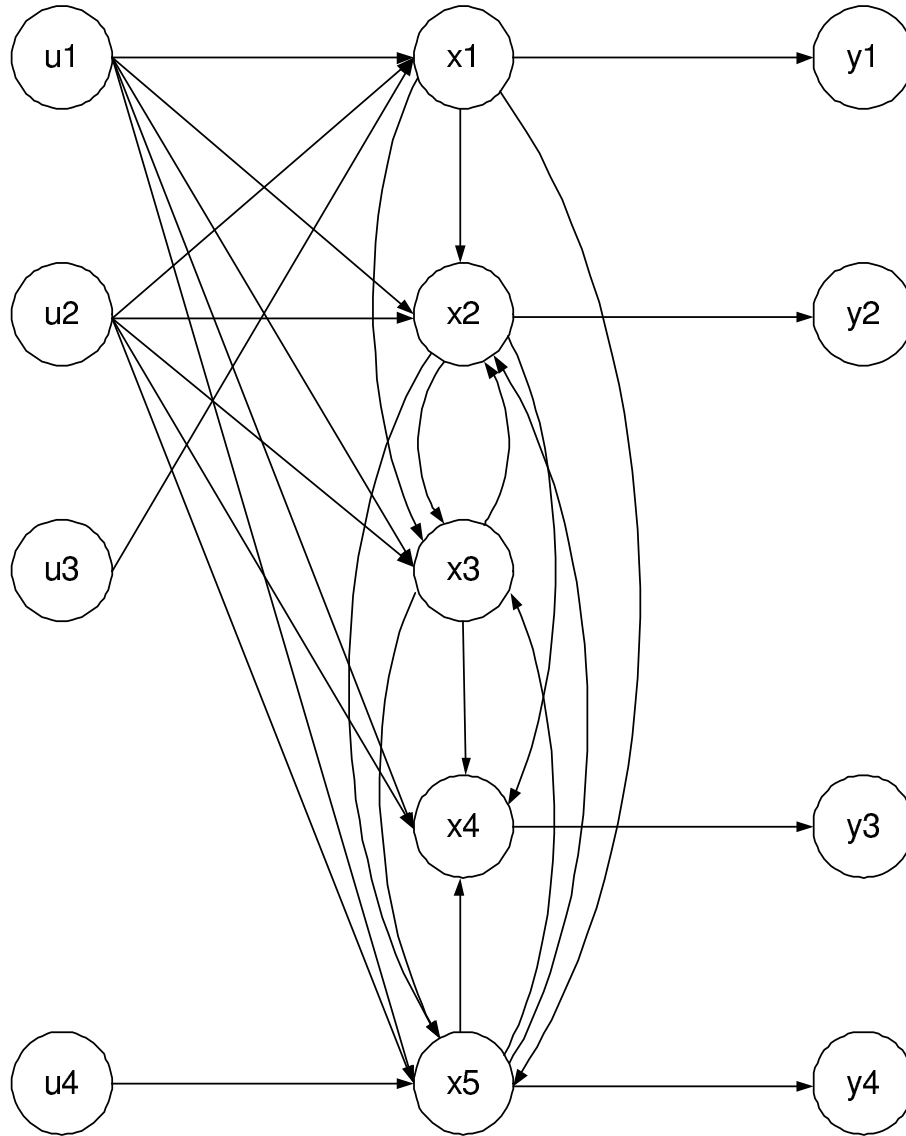
ahol az  $S$  alsóindex a névleges állandósult állapot értéke. A dinamikus egyenletek átírhatók a (48) (49) egyenletekkel megadott állapottér modell formára. A rendszer struktúragráfja a 10. ábrán látható.

Ezután a relatív foksámokat kiszámolva a relatív foksám mátrix a következő lesz:

$$M_r = \begin{bmatrix} 1 & 1 & 1 & \infty \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$

Követve a 3.4.1. fejezet feltételét, a relatív foksám mátrix a következő alakba írható át:

$$M_r = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & \infty \\ 1 & 1 & 2 & 1 \end{bmatrix}$$



10. ábra. Folytonosan kevert tartály reaktor struktúra gráfja

Az egzakt linearizációs szabályozó struktúrát kiszámolva két, relatív fokszám értelemben azonos megoldást kapunk:

$$\begin{pmatrix} u_1/y_2 & u_2/y_3 & u_3/y_1 & u_4/y_4 \\ F_A/c_A & F_B/c_C & F/V & Q/T \end{pmatrix}$$

és

$$\begin{pmatrix} u_1/y_3 & u_2/y_2 & u_3/y_1 & u_4/y_4 \\ F_A/c_C & F_B/c_A & F/V & Q/T \end{pmatrix}$$



#### 4.4. Tennessee Eastman probléma

Tekintsük az [2]-ben leírt Tennessee Eastman problémát. A bemeneti és az állapotváltozók halmaza a következők:

$$U = \{V_D, V_E, V_A, V_C, V_B, V_{co}\}$$

$$X = \{R, L_r, P_r, Y, C_B, T_{co}\}$$

ahol

- $V_i$   $i \in C$  térfogatáram az  $i$ . komponensben,
- $C$  komponensek halmaza,  $C = \{A, B, C, D, E\}$ ,
- $V_{co}$  a hűtő közeg térfogatárama,
- $R$  a termék koncentrációjának aránya,
- $L_r$  szint a reaktorban,
- $P_r$  reaktor nyomása,
- $Y$  termék térfogatárama,
- $C_B$  semleges komponens koncentrációja,
- $T_{co}$  hűtő közeg hőmérséklete.

Kiszámolva a rendszer relatív fokszám mátrixát a következőt kapjuk:

$$M_r = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Látható, hogy ebben az esetben ezzel a módszerrel nem lehet eldönteni, hogy milyen egzakt linearizációs szabályozó struktúrát válasszunk.

#### 4.5. Az esettanulmányok értékelése

A 4.2. fejezetben bemutatott hőcserélő hálózaton egyértelműen eldönthető, hogy milyen egzakt linearizációs szabályozó struktúrát válasszunk.

Az  $y_1$  és  $y_2$  kimenetekhez a 3.4.1. fejezetben leírt feltétel alapján egyértelműen adódnak az  $u_1$  és  $u_4$  bemenetek. Az  $u_2$ ,  $u_3$ ,  $u_5$  bemeneteknek ugyanolyan hatása van az

$y_3$  kimenetre. Az  $u_5$  bemenet a legmegfelelőbb, mivel ennek nincs hatása a többi kimenetre, míg az  $u_2$ -es bemenetnek van hatása az  $y_1$ ,  $y_2$ -es kimenetekre is, az  $u_3$ -as bemenetnek pedig az  $y_2$ -es kimenetre van még hatása.

Az eredmény optimális mivel a legjobb azok között, amelyek közül a részrendszerek legkevésbé hatnak egymásra.

A 4.3. fejezetben vizsgált folytonosan kevert tartály reaktor esetében a relatív fokszám mátrixnak van két azonos sora. A megoldás nem egyértelmű, mivel két, relatív fokszám értelemben azonosan jó megoldást kapunk, amelyet felhasználhatunk a szabályozó struktúra kiválasztására.

A 4.4. fejezetben vizsgált Tennessee Eastman problémánál ezzel a módszerrel nem tudjuk eldönteni, hogy milyen egzakt linearizációs szabályozó struktúrát válasszunk, mivel a relatív fokszám mátrix minden eleme 1-es, így ez alapján nem tudjuk a megfelelő bemenet-kimenet párokat kiválasztani.

Ilyenkor más módszert kell találnunk a párosítások kiválasztására, olyat, amely figyelembe veszi a rendszer numerikus paramétereit. Ilyen lehet például [4]-ben leírt módszer, amely az állandósult állapot mátrix erősítéseit használja fel a bemenet-kimenet párok kiválasztására.

A Tennessee Eastman probléma állandósult állapot mátrixa  $G(0)$ :

	$V_D$	$V_E$	$V_A$	$V_C$	$V_B$	$V_{co}$
$R$	0.054	0.012	-0.007	-0.060	0.001	-0.005
$L_r$	105	106	-24.3	-176	2.06	-24.2
$P_r$	652	673	-184	-1053	4.70	-170
$Y$	-0.151	-0.124	0.067	0.566	-0.151	0.499
$C_B$	-2.42	-2.52	0.614	3.96	-0.349	0.758
$T_{co}$	6.40	6.662	-1.64	-10.0	0.022	-3.16

Ezzel a módszerrel kiválasztott bemenet-kimenet párok:

$$(R/V_A), (L_r/V_E), (P_r/V_C), (Y/V_B), (C_B/V_{co}), (T_{co}/V_D)$$

## 5. A program felhasználói leírása

Az egzakt linearizációs szabályozó struktúrát kiválasztó algoritmus implementálására a C++ nyelvet választottam. A C++ nyelv az egyik legrugalmasabb programozási nyelv, amelyet platform függetlennek tekinthetünk.

A program használata nagyon egyszerű. A program az `<input.txt>` fileból olvassa be a bemeneti adatokat. Meg kell adni a vizsgálni kívánt folyamatrendszer struktúragráfját. A bemenetként megadott struktúragráfból el kell hagyni a hurokéleket, de mivel ezek nincsenek hatásal az eredményre, nem is szükségesek. A struktúragráf megadásához éllistas reprezentációt használtam, mivel a struktúragráfok általában nem túl sok éllel rendelkeznek, így a bemenő adatok mennyisége kevesebb, mint adjacencia mátrixos ábrázolás esetén. Az éllistas tároláshoz szükség van a struktúragráf csúcscsámára is. Ezen felül a bemenetek, és a kimenetek számát, valamint a bemeneti és kimeneti csúcsok sorszáma van még szüksége a programnak. Az `<input.txt>` fileba `#`-el kezdve megjegyzéseket is írhatunk.

A struktúragráf csúcsait számokkal kell megadnunk 1-től kezdve. Meg kell adnunk legalább egy bemenetet és egy kimenetet. A program ellenőrzi, hogy számbemenetet és nullától különböző számokat adtunk e meg. Ha nem így járunk el hibajelzést ad: **Nem szam, vagy nulla!**

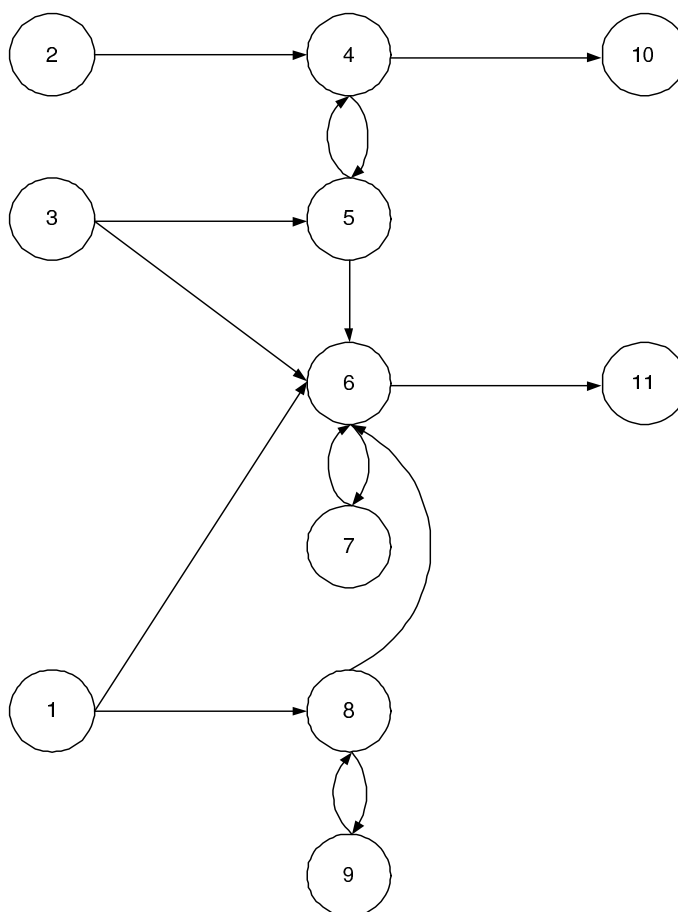
A program kiszámolja a struktúra gráfon a bemenetektől a kimenetekig vezető legrövidebb utak hosszát, majd a 3.1.1. Lemmát ( $r_{ij} = l_{ij} - 1$ ) felhasználva megadja a folyamatrendszer relatív fokszám mátrixát.

Ezután a (62). egyenletnek megfelelően kiszámolja a folyamatrendszer átsúlyozott relatív fokszám mátrixát. Ez a mátrix megfeleltethető egy olyan súlyozott páros gráfnak, amelyre teljesülnek a 3.4. fejezet feltételei.

A program ezután megkeresi ebben a súlyozott páros gráfban a legnagyobb súlyú párosítást, amely megadja a legjobb egzakt linearizációs szabályozó struktúrát. Ezt oly módon teszi, hogy a lehetséges párosítások közül kiválasztja a legnagyobb súlyút. A súlyozás biztosítja, hogy a legjobb egzakt linearizációs szabályozó struktúrát válasszuk ki. A program eredményül az `<output.txt>` fileban a relatív fokszám mátrixot, az átsúlyozott relatív fokszám mátrixot, valamint a legjobb szabályozó struktúrát adja vissza.

Az `<input.txt>` fileban és az `<output.txt>` fileban nincsenek ékezetes betűk, mivel

ezeket a C++ fordító nem ismeri.



11. ábra. A programnak bemenetként megadott struktúra gráf

A 1. ábrán látható hőcserélő hálózathoz a következő <input.txt> file tartozik:

```
#csucsk_szama:
```

```
11
```

```
#honnan->hova_a_graf_elei:
```

```
1 6
```

```
1 8
```

```
2 4
```

```
3 5
```

```
3 6
```

```
4 5
```

```
4 10
```

```
5 4
```

```
5 6
```

```
6 7
```

```
6 11
7 6
8 6
8 9
9 8
-1 1
#bemenetek_szama:
3
#kimenetek_szama:
2
#1.bemenet:
1
#2.bemenet:
2
#3.bemenet
3
#1.kimenet:
11
#2.kimenet:
10
```

Az eredményül kapott <output.txt> file:

A relativ fokszám matrix:

```
1 0
3 1
1 2
```

Az atsulyozott relativ fokszám matrix:

```
13 0
8 12
12 10
```

A legjobb szabalyozo struktura: u1y1 u2y2

## 6. Összefoglalás

Egzakt linearizációs szabályozó struktúra gráfelméleti eszközökkel történő kiválasztását vizsgáltam ebben a diplomamunkában nemlineáris MIMO rendszerekre.

Először meghatároztam a relatív fokszám és a struktúragráfon a bemenetek és a kimenetek között lévő legrövidebb út kapcsolatát. A bemenetek és a kimenetek között számolható relatív fokszámok egyenlők a struktúragráfon a bemenettektől a kimenetekig vezető legrövidebb utak eggyel csökkentett hosszával. Az egzakt linearizációs szabályozó struktúra kiválasztásához ennek segítségével a MIMO nemlineáris rendszer struktúragráfján meghatározhatjuk azokat a SISO részgráfokat, amelyek teljesítik az egzakt linearizáció relatív fokszám feltételét. Ezek közül a részgráfok közül azokat kell kiválasztanunk az egzakt linearizációs szabályozó struktúrához, amelyek között a kölcsönhatás minimális. Ezeket a részgráfokat megfelelően egy súlyozott páros gráfnak, és ezen a súlyozott páros gráfon maximális párosítást keresve megkapunk egy lehetséges optimális egzakt linearizációs szabályozó struktúrát. Az egzakt linearizációs szabályozó struktúra megkeresését több esettanulmányon vizsgáltam. Az eddig vizsgált példák közül az a következtetés vonható le, hogy ha a MIMO nemlineáris rendszer relatív fokszám mátrixában nincsenek azonos sorok, akkor az ebben a munkában leírt algoritmus egyértelmű és optimális egzakt linearizációs szabályozó struktúra kiválasztást tesz lehetővé. Ellenkező esetben több, relatív fokszám értelemben azonosan jó megoldást kaphatunk. Amennyiben a relatív fokszám mátrix minden eleme azonos, ezzel a módszerrel nem tudjuk kiválasztani a megfelelő egzakt linearizációs szabályozó struktúrát. Ilyenkor más, a nemlineáris rendszer numerikus paramétereit is figyelembe vevő módszert kell találnunk a szabályozó struktúra kiválasztására.

Végül meg kell jegyezni azt, hogy további munka témája lehet a kiválasztott egzakt linearizációs szabályozó struktúra szabályozóinak összehangolása, amelyet egy felsőbb hierarchia szinten lévő szabályozóval valósíthatunk meg. Egy másik vizsgálat célja lehet az az eset, amikor a nemlineáris rendszerben valamely szenzor hibája miatt kiesik egy bemenet, vagy egy kimenet. Ekkor újra ki kell választani a legmegfelelőbb szabályozó struktúrát.

---

## Hivatkozások

- [1] Kravaris C. Daoutidis P. Structural evaluation of control configurations for multivariable nonlinear processes. *Chemical Engineering Science*, 47(5):1091, 1992.
- [2] Vogel E.F. Downs J.J. A plant-wide industrial process control problem. *Computers and Chemical Engineering*, (17), 1993.
- [3] Szederkényi G. Hangos K.M., Bokor J. *Analysis and Control of Nonlinear Process Systems*. Springer, 2004.
- [4] Tuza Zs. Hangos K.M. Optimal control structure selection for process systems. *Computers and Chemical Engineering*, page 1521, 2001.
- [5] Szabó R. Iványos G., Rónyai L. *Algoritmusok*. Műegyetemi jegyzet, 1997.
- [6] Plummer M.D. Lovász L. *Matching Theory*. Akadémiai Kiadó, 1986.

## 7. Függelék

A <structsearch.exe> file forráskódja:

```
<structsearch.cpp>

#include <stdio.h>
#include <stdlib.h>
#include "sor.h" //FIFO stack

//globális változók:
FILE * inputfile; //az input adatokhoz tartozó filemutató
//a struktúragráf egy adott csúcsát bejárta e már a szélességi keresés
bool *bejarva;
lista *ellist; //éllista a struktúragráf tárolására
int **rorder; //relatív fokszám mátrix
int **srorder; //súlyozott relatív fokszám mátrix
int **permmatrix; //a permutációs mátrix bemenete
int **permmatrix2; //a permutációs mátrix kimenetei
int *eredmeny; //részeredmények
int *eredmeny2; //relative fokszám (eredmény mártix)
int input; //bemenetek száma
int output; //kimenetek száma

//elvégzi a gráf szélességi bejárását, v kezdőcsúcs
void szb(int v)
{
    lista q,t; //az épp most bejárt csúcsok listája
    listaelem *temp; //ideiglenes tároló
    int x; //csúcsok
    int elozo=0; //már bejárt csúcsok
    bejarva[v]=true; //kezdőcsúcs bejárva
    q.insertlast(v); //kezdőcsúcsot beszúrja az épp most
    bejárt csúcsok listájába
    //v csúcsból bejárja annak gyerekeit, v-t kiszedi
    while (!q.isempty())
    {
        //visszaadja a sor első elemét, és ki is törli
        q.getfirst(x);
        temp=ellist[x].gethead();
        if (temp!=NULL) elozo=rorder[v][x];
        while (temp!=NULL)
        {
            if (!bejarva[temp->hova])
            {
```



```

        bejarva[temp->hova]=true;
        q.insertlast(temp->hova); //gyerekeket a sor végére
        rorder[v][temp->hova]+=elozo+1;
    };
    temp=temp->next;
} //while vége
} //while vége
} //szb vége

//beolvasó függvény
//ellenőrzi, hogy számbemenetet adtunk e meg
int beolv()
{
    char s[255];
    int x;
    bool ok=false;

    while(!ok)
    {
        fscanf(inputfile,"%s",&s); //beolvassa az input.txt adatait
        if (s[0]!='#') //megjegyzés az input fileban
        {
            if ((x=atoi(s))==0) //ellenőrzi a számbemenetet
            {
                printf("Nem szam vagy nulla!\n");
                exit(-1);
            }
            else ok=true;
        } //else vége
    } // while vége
    return x;
} //beolv vége

//A függvény elvégzi n szám permutációját
void perm (char *a, const int k, const int n)
// n a mérete
// generálja a permutációkat: a[k], ..., a[n-1]-ig
{
    if (k == n-1) // kimeneti permutáció
    {
        for(int i=0;i<input;i++)
            for(int j=0;j<output;j++)
                permmatrix2[i][j]=permmatrix[a[i]][j];

        eredmeny2[0]=0;
        int p=1;
    }
}

```

```

    for(i=0;i<input;i++)
        for(int j=0;j<output;j++)
            {
                eredmeny2[0]+=permmatrix2[i][j]*srorder[i][j];
                if (permmatrix2[i][j]==1)
                    {
                        eredmeny2[p]=i;eredmeny2[p+1]=j;p+=2;
                    }
            };
    if (eredmeny2[0]>eredmeny[0])
        for(i=0;i<2*output+1;i++) eredmeny[i]=eredmeny2[i];
}
else // a[k], ..., a[n-1]-nek több mint 1 permutációja van
{
// ezeket rekurzívan generálja
    for (int i = k; i < n; i++)
        {
            // felcseréli a[k]-t a[i]-vel
            char temp = a[k]; a[k] = a[i]; a[i] = temp;
            //a[k+1], ..., a[n-1] minden permutációja
            perm(a, k+1, n);
            // felcseréli a[k]-t a[i]-re ismét,
            temp = a[k]; a[k] = a[i]; a[i] = temp;
        }
    } //else vége
} //perm vége

//main függvény, adatok bekérése, kiíratás
void main()
{
    int csucok; //A struktúragráf összes csúcsának száma
    int i,j; //futóindexek
    int honnan=0; //A struktúragráfban egy él kiindulási pontja
    int hova; //és vége
    int tav; //távolság az adott bemenet és kimenet között
    int max; //a legnagyobb relatív fokszám

//input file nyitás
if(!(inputfile=fopen("input.txt","r")))
{
    fprintf(stderr,"sikertelen input file nyitas\n");
    exit(-1);
}

//a struktúragráf csúcsszámának beolvasása
csucok=beolv();

```

```

//a struktúragráf éllistás tárolása
ellist=new lista[csucsok];

int *inputcs; //bemenetek
int *outputcs; //kimenetek
//a relatív fokszám mátrixban található nullák száma
int *nullaszam;

while(honnan>=0) // éllista feltöltése
{
    honnan=beolv()-1;hova=beolv()-1;
    if (honnan>=0) {ellist[honnan].insertlast(hova);}
};

bejarva=new bool[csucsok];
//relative fokszám matrix létrehozása:
//sorok: bemenetek, oszlopok : kimenetek
input=beolv(); //bemenetek száma
output=beolv(); //kimenetek száma
//bemeneteknek, kimeneteknek foglal egy tömböt
inputcs=new int[input];
//a relatív fokszám mátrix egy sorában lévő 0-k száma
nullaszam=new int[input];
outputcs=new int[output];

//relatív fokszám mátrix mátrix létrehozása,
rorder=(int **)(new int[csucsok]);
//kinullázzuk
for(i=0;i<csucsok;i++) rorder[i]=new int[csucsok];
for (i=0;i<csucsok;i++) for(j=0;j<csucsok;j++) rorder[i][j]=0;

//feltölti a rorder matrix egy sorát:
for(j=0;j<input;j++)
{
    //az adott bemenettől az összes kimenethez
    inputcs[j]=beolv()-1;
    nullaszam[j]=0; //nullák száma az adott sorban
};

//a kimenetek beolvasása
for (j=0;j<output;j++)
{
    outputcs[j]=beolv()-1;
};

//szélességi bejárás az összes bemenetre

```

```
for(j=0;j<input;j++)
{
    for(i=0;i<csucsock;i++) bejarva[i]=false;
    szb(inputcs[j]);
}

//sulyozott relatív fokszám mátrix létrehozása
srorder=(int **)(new int[input]);
//permutációs mátrix létrehozása
permmatrix=(int **)(new int[input]);
//permutációs eredmény mátrix létrehozása
permmatrix2=(int **)(new int[input]);
for(i=0;i<input;i++)
{
    srorder[i]=new int[output];
    permmatrix[i]=new int[output];
    permmatrix2[i]=new int[output];
}

//kiíratás

FILE * outputfile; //output adatok filemutatója
//output file megnyitása
outputfile=fopen("output.txt","wt");
if (outputfile==NULL)
{
    fprintf(stderr, "sikertelen file nyitas\n");
    exit(-1);
}
//a relatív fokszám mátrix kiíratása
fprintf(outputfile,"A relatív fokszam matrix:\n");
max=0; //a legnagyobb relatív fokszám
for (i=0;i<input;i++)
{
    for (j=0;j<output;j++)
    {
        if (rorder[inputcs[i]][outputcs[j]]==0)
        {
            tav=0;nullaszam[i]++;
        }
        else tav=rorder[inputcs[i]][outputcs[j]]-1;
        if (max<tav) max=tav;
        fprintf(outputfile,"%d ",tav);
    }
    fprintf(outputfile,"\n");
}
```

```
};
//az átsúlyozott relatív fokszám mátrix kiíratása
fprintf(outputfile, "\n");
fprintf(outputfile, "Az atsulyozott relatív fokszam matrix\n");
fprintf(outputfile, "\n");
for (i=0; i<input; i++)
    for(j=0; j<output; j++)
    {
        if (rorder[inputcs[i]][outputcs[j]]==0) srorder[i][j]=0;
        else srorder[i][j]=(output*max-rorder[inputcs[i]]
        [outputcs[j]]+2)*output+nullaszam[i];
        if (i==j) permatrix[i][j]=1;
        else permatrix[i][j]=0;
    }

for (i=0; i<input; i++)
{
    for(j=0; j<output; j++) fprintf(outputfile, "%d ", srorder[i][j]);
    fprintf(outputfile, "\n");
}

//eredmény kiíratása
eredmeny=new int[2*output+1];
eredmeny2=new int[2*output+1];
char* p=new char[input];
for (j=0; j<input; j++) p[j]=j;
perm(p, 0, input);
fprintf(outputfile, "\n");
fprintf(outputfile, "A legjobb szabalyozo struktura: ");
for(j=1; j<(2*output+1); j+=2)
fprintf(outputfile, "u%dy%d ", eredmeny[j]+1, eredmeny[j+1]+1);
fprintf(outputfile, "\n");

//input, output fileok lezárása
fclose(outputfile);
fclose(inputfile);
} //main vége
```