

Modell alapú diagnosztika diszkrét módszerekkel

Hangos Katalin, Lakner Rozália, Starkné Werner Ágnes

Veszprém, 2012.

COPYRIGHT: ©2012, Dr. Hangos Katalin, dr. Lakner Rozália, Starkné dr. Werner Ágnes, Pannon Egyetem Műszaki Informatikai Kar

A szerző nevének feltüntetése mellett nem kereskedelmi céllal szabadon másolható, terjeszthető, megjelentethető és előadható, de nem módosítható.

TÁMOGATÁS: Készült a TAMOP-4.2.1/B-09/1/ KONV-2010-0003 számú "Mobilitás és környezet: Járműipari, energetikai és környezeti kutatások a Közép- és Nyugat-Dunántúli Régióban" című projekt támogatásával.

ISBN 978-615-5044-55-7

KÉSZÜLT: a Pannon Egyetem gondozásában

Tartalomjegyzék

| | |
|--|-----------|
| 1. Bevezetés | 7 |
| 2. A diagnosztikai feladat | 9 |
| 2.1. Dinamikus rendszerek modelljei | 9 |
| 2.1.1. Dinamikus rendszerek, jelek és paraméterek | 9 |
| 2.1.2. Állapottér és input-output modellek | 11 |
| 2.2. A diagnosztikai feladat fajtái | 12 |
| 2.2.1. Meghibásodás detektálás és azonosítás | 12 |
| 2.2.2. Predikción alapuló diagnosztika | 12 |
| 2.2.3. Paraméterbecslésen alapuló diagnosztika | 13 |
| 2.2.4. Állapot-becslésen alapuló diagnosztika | 14 |
| 3. Fizikai alapú diszkrét dinamikus modellek | 15 |
| 3.1. Dinamikus modellek felállítása | 15 |
| 3.1.1. A modellezési feladat | 15 |
| 3.1.2. Dinamikus mérlegegyenletek | 19 |
| 3.1.3. Kiegészítő algebrai egyenletek | 22 |
| 3.1.4. Modellezési feltételezések | 23 |
| 3.1.5. A modell összetevői | 23 |
| 3.1.6. Állapottér modell alak | 24 |
| 3.1.7. Egyszerű példa: szabad kifolyású tartály | 24 |
| 3.1.8. A kávéfőző gép modellje | 26 |
| 3.2. Kvalitatív dinamikus modellek | 28 |
| 3.2.1. Előjel és intervallum aritmetikák | 28 |
| 3.2.2. A kvalitatív dinamikus modellek származtatása | 30 |
| 3.2.3. Konfluenciák és a belőlük származtatott szabályok | 30 |
| 3.2.4. Kvalitatív differencia-algebrai egyenletek | 31 |
| 4. SDG modelleken alapuló diagnosztika | 37 |
| 4.1. SDG modellek származtatása | 37 |
| 4.1.1. Állandósult állapot körüli linearizálás | 38 |
| 4.1.2. Az állapottér modellek szerkezete | 39 |
| 4.2. SDG modellek és tulajdonságaik | 42 |
| 4.2.1. Utak és körök a struktúra gráfban | 43 |
| 4.2.2. Az SDG modellek diagnosztikai célú értelmezése | 43 |

| | |
|---|-----------|
| 4.2.3. Az egységgrás válaszfüggvény jellemzése | 44 |
| 4.3. Predikción alapuló diagnosztika SDG modellekkel | 47 |
| 4.3.1. Diagnosztika a legrövidebb utak előjele alapján | 47 |
| 4.3.2. Diagnosztika az állandósult állapotbeli eltéréssel | 47 |
| 4.3.3. Példa: kávéfőző gép diagnosztikája | 48 |
| 4.4. Direkt diagnosztika SDG modelleken | 48 |
| 4.4.1. A módszer elve | 48 |
| 4.4.2. Egyszerű példa | 49 |
| 5. Diagnosztika szabályok alkalmazásával | 51 |
| 5.1. Szabályrendszerek, következtetési módszerek | 51 |
| 5.1.1. Predikátumok és szabályok | 51 |
| 5.1.2. Szabályrendszerek | 52 |
| 5.1.3. Adatvezérelt és célvezérelt következtetés | 53 |
| 5.1.4. Példa: kávéfőző gép szabályrendszere | 54 |
| 5.2. Diagnosztika adatvezérelt következtetéssel | 55 |
| 5.2.1. Predikción alapuló diagnosztika szabályokkal | 55 |
| 5.2.2. Kávéfőző gép predikción alapuló diagnosztikája | 55 |
| 5.3. Diagnosztika célvezérelt következtetéssel | 56 |
| 5.3.1. Direkt diagnosztika szabályokkal | 56 |
| 5.3.2. Kávéfőző gép direkt diagnosztikája | 57 |
| 6. Diagnosztika HAZOP és FMEA alapján | 59 |
| 6.1. Kockázat menedzsment | 59 |
| 6.1.1. Veszély és működőképességi analízis – HAZOP | 60 |
| 6.1.2. Meghibásodás és hatás analízis – FMEA | 61 |
| 6.2. Szabály alapú diagnosztika | 62 |
| 6.2.1. Megfeleltetés a HAZOP és FMEA táblák elemei között | 62 |
| 6.2.2. Szabályok a HAZOP és FMEA táblákból | 63 |
| 6.2.3. Példa: Kávéfőző gép – HAZID szabályok | 63 |
| 6.2.4. A diagnosztikai következtetés | 65 |
| 7. Diagnosztika színes Petri háló modellekkel | 67 |
| 7.1. Diszkrét eseményű rendszerek modelljei | 67 |
| 7.1.1. Diszkrét eseményű rendszerek fogalma | 67 |
| 7.1.2. Diszkrét eseményű rendszerek állapotter modelljei | 68 |
| 7.2. Színes Petri hálók és tulajdonságaik | 68 |
| 7.2.1. Petri hálók | 68 |
| 7.2.2. Magasabb szintű Petri hálók | 72 |
| 7.3. Színes Petri hálók és kvalitatív egyenletek | 75 |
| 7.3.1. Kvalitatív algebrai egyenletek és színes Petri hálók | 75 |
| 7.3.2. Kvalitatív differencia egyenletek és színes Petri hálók | 77 |
| 7.3.3. Kvalitatív differencia-algebrai egyenletekből származó színes Petri háló modellek megoldása | 77 |
| 7.4. Diagnózerek és tervezésük | 80 |

| | | |
|-----------|--|-----------|
| 7.4.1. | Karakterisztikus kvalitatív bemenet-kimenet jel-nyomok | 80 |
| 7.4.2. | Diagnózerek fogalma és szerkezete | 81 |
| 7.4.3. | Diagnózer felállítása színes Petri háló alakú modell alapján | 81 |
| 8. | Folyamatbányászati módszerek | 83 |
| 8.1. | Bevezetés | 83 |
| 8.2. | Folyamatbányászat | 83 |
| 8.3. | Eseménynapló - log | 84 |
| 8.4. | Modellezési módszerek - Petri háló | 85 |
| 8.5. | Folyamatbányászati lehetőségek | 85 |
| 8.6. | Információszerzés a Prom 6.0 keretrendszerrel | 90 |
| 8.6.1. | Példa log elemzése | 90 |
| 8.6.2. | A repairExamples.xes log elemzése | 91 |
| 8.6.3. | Esettanulmány - Operátori eljárással irányított garázkapu | 93 |
| 8.6.4. | A keretrendszer ismertetése | 97 |

1. fejezet

Bevezetés

A diagnosztikai módszerek nemcsak az orvostudományban, hanem a mindennapi élet szinte valamennyi területén alapvető fontosságúak. Felhasználásukkal állapíthatjuk meg a megfigyelt jelenségek és a diagnosztizálandó objektumról rendelkezésre álló a priori ismeretek alapján, hogy az objektum normális vagy hibás állapotban van-e, és ha igen, milyen hiba ok(ok) lépett/léptek fel.

A műszaki rendszerek diagnosztikájával számos közlemény foglalkozik a szakirodalomban (lásd például [2]), és igen sokféle diagnosztikai módszert és eljárást fejlesztettek ki. A diagnosztikai módszerek egy széles családja a rendszerről rendelkezésre álló tudást matematikai modellek formájában használja, ezeket a módszereket nevezzük modell alapú diagnosztikai módszereknek.

A diagnosztikára használatos modellek matematikai formája meghatározza a diagnosztikai eljárásban alkalmazott matematikai módszerek fajtáját is. A legelterjedtebbek a folytonos idejű úgynevezett állapotter modelleken alapuló diagnosztikai eljárások, amelyek algebrai vagy geometriai módszereket alkalmaznak a diagnosztikára. Viszonylag kevés közlemény foglalkozik olyan modell alapú diagnosztikai módszerekkel, ahol a modell diszkrét jellegű, például gráf vagy szabályrendszer alakjában adott, pedig ezek a modellek kiválóan alkalmasak igen nagyméretű, összetett rendszerek diagnosztizálására.

A könyv célja, hogy megismertessük az Olvasót a diszkrét dinamikus modellek legfontosabb fajtáival (időfüggő szabályok, Petri hálók, kvalitatív differenciálegyenletek, előjeles irányított gráfok), és megmutassuk, hogyan lehet ilyen modelleket felállítani és ellenőrizni, majd diagnosztikai célra felhasználni.

Az anyag épít az Olvasó alapvető irányítástechnikai (jelek és redszerek, alapvető rendszerleírási módok) és mesterséges intelligencia (tudásábrázolás, szabályok, szabályalapú következtetés) ismereteire, de ezeket az ismereteket a hivatkozó fejezetekben röviden összefoglaljuk.

A modell alapú diszkrét diagnosztikai módszerek eszköztárából igyekeztünk azokat a módszereket kiválasztani, amelyek biztos elméleti alapokon állnak, és széles alkalmazási területen használhatóak. Az elméleti anyagot minden fejezetben egyszerű példákon is bemutatjuk, hogy segítsük a megértést. Ezek a példák is több alkalmazási területet (például hőtani, gépészeti, közlekedési stb.) ölelnek fel.

A monográfia, amelyben számos saját kutatási eredményünk is szerepel, a TAMOP-

4.2.1/B-09/1/ KONV-2010-0003 számú "Mobilitás és környezet: Járműipari, energetikai és környezeti kutatások a Közép- és Nyugat-Dunántúli Régióban" című projekt támogatásával készült el.

Veszprém, 2012. február 29.

Hangos Katalin, Lakner Rozália és Starkné Werner Ágnes
Pannon Egyetem
Műszaki Informatikai Kar

2. fejezet

A diagnosztikai feladat

A mindennapi életben a "diagnosztika" szót elsősorban egészségügyi kontextusban használjuk: egy betegség fajtáját és kiváltó okait határozza meg a szimptómák és az esetleges orvosi vizsgálatok eredményei alapján. A műszaki gyakorlatban a "diagnosztika" egy hasonló célú módszercsaládot foglal magában: egy dinamikus rendszerről tudja a mért adatok és az esetleges kiegészítő mérések eredményei alapján megállapítani, hogy az meghibásodott-e, és ha igen, "mi a baja".

Ebben a fejezetben megismerkedünk a diagnosztikához felhasználható rendszermodellek tulajdonságaival, és megfogalmazzuk kissé pontosabban is, hogy milyen diagnosztikai feladat-típusok ismeretesek.

2.1. Dinamikus rendszerek diagnosztikai célú modelljei

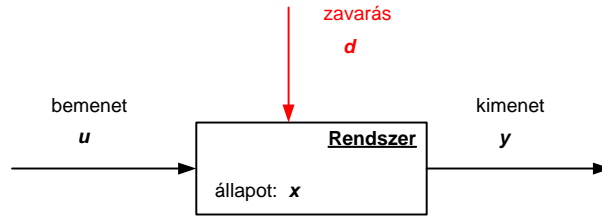
A dinamikus rendszerek matematikai modelljei a rendszer- és irányításelmélet alapvető fogalmaira, a jelek, rendszerek és paraméterek fogalmára épülnek.

2.1.1. Dinamikus rendszerek, jelek és paraméterek

A rendszer egy olyan, időben változó egység, amely a környezetével áll kapcsolatban, amit időfüggvények, úgynevezett *jelek* segítségével írunk le. A környezet a rendszerre az ún. *bemenetek* (ezeket befolyásolhatjuk) és a *zavarások* által hat, a rendszer hatását a környezetre az ún. *kimenetekkel* írjuk le.

A 2.1. ábra egy úgynevezett jel-folyam ábrán adja meg a rendszert a legfontosabb jeleivel együtt. A rendszerhez a környezettel összekötő kimenet, bemenet és zavarás jeleken kívül az úgynevezett *állapot*-változók, mint jelek is hozzátartoznak, ezekkel egyszerűbb, a szabályozótervezési és diagnosztikai alkalmazások szempontjából fontos rendszerleírási formák adhatók.

Jeltípusok. A jeleket értelmezési tartományuk (esetünkben az idő) és érték-készletük milyensége alapján sorolhatjuk típusokba. A diszkrét módszerekkel történő diagnosztikai

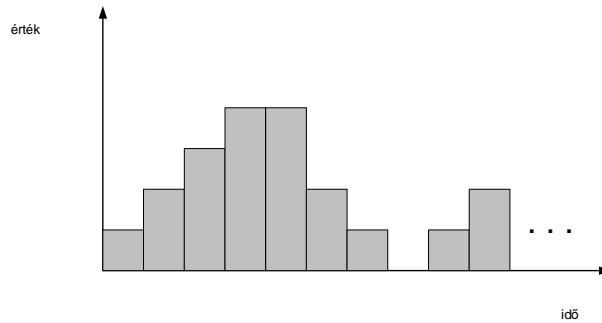


2.1. ábra. A rendszer jel-folyam ábrája

alkalmazásokban az úgynevezett diszkrét idejű jelek fontosak, amikor az időhalmaz elemei diszkrét (nem feltétlenül ekvidisztáns) időpontok. Egy diszkrét idejű skalár bemeneti jel például az alábbi sorozattal adható meg:

$$U(0, \dots) = \{u(k) = u(t_k) \mid k = 0, 1, 2, \dots\}$$

A diszkrét idejű jeleken belül is külön érdekességgel bírnak azok a jelek, amelyeknek az érték-készlete is diszkrét (*diszkrét idejű diszkrét érték-készletű jelek*). Erre egy skalár értékű példát láthatunk a 2.2. ábrán.



2.2. ábra. Diszkrét idejű diszkrét érték-készletű jel

Matematikai szempontból a rendszereket legáltalánosabban jeleken ható operátorokkal (**S**) írhatjuk le, amelyek a bemenetek jelteréből a kimenetek jelterébe képeznek, azaz

$$y = \mathbf{S}[u]$$

A rendszerek matematikai leírásában szereplő, nem jel típusú változókat *paramétereknek* nevezzük. A *paraméterek* a diagnosztikai alkalmazásokban gyakran *időtől nem függő konstansok*, az ilyen rendszereket időinvariáns rendszereknek (lásd alább) nevezzük.

Rendszerosztályok. A rendszerek osztályozhatóak jeleik típusa vagy a nekik megfelelő **S** operátor matematikai tulajdonságai alapján. A diszkrét módszerekkel történő diagnosztikai alkalmazásokban az alábbi rendszerosztályok játszanak fontos szerepet.

- *diszkrét idejű rendszerek*, amelyeknél a jelek (bemenet, kimenet, állapot) diszkrét idejűek,
- *lineáris időinvariáns (LTI) rendszerek*, amelyeknél az \mathbf{S} operátor lineáris és a rendszermodell paraméterei állandóak. Ez az alapeset, amelyre a legtöbb szabályozási és diagnosztikai feladatnak analitikus megoldása van.

2.1.2. Állapottér és input-output modellek

A gyakorlatban kétféle matematikai modellt használunk dinamikus rendszerek leírására.

1. Az ún. *input-output modell* csak a bemenet (esetleg a zavarás) és kimenet jeleket és ezek időbeli deriváltjait (a folytonos idejű esetben) vagy véges differenciáit (a diszkrét idejű esetben) tartalmazza.

Így a diszkrét idejű egy-bemenetű egy-kimenetű (SISO) esetben az általános input-output modell alakja

$$y(k) = H(u(k); y(k-1), u(k-1); y(k-2), u(k-2); \dots) \quad (2.1)$$

2. Az ún. *állapottér modell*ben egyaránt szerepelnek a bemenet (esetleg a zavarás) és kimenet jelek, valamint az állapot jelek. Az állapottér modellben az állapotváltozók időfejlődését az ún. koncentrált paraméterű esetben egy elsőrendű közönséges differenciál vagy differencia egyenlet-rendszer, az úgynevezett *állapotegyenletek* írják le, a kimenet(ek) pedig az állapotoktól és a bemenetektől az algebrai jellegű úgynevezett *kimeneti egyenlet*(ek) szerint függ(nek).

A diszkrét idejű esetben az általános állapottér modell az alábbi alakú:

$$\begin{aligned} x(k+1) &= F(x(k), u(k)) && \text{(állapot egyenlet)} \\ y(k) &= G(x(k), u(k)) && \text{(kimenet egyenlet)} \end{aligned} \quad (2.2)$$

adott $x(0)$ kezdeti feltétellel és nemlineáris F állapot, valamint G kimeneti függvényekkel.

Az input-output és állapottér modellekre a legegyszerűbb diszkrét idejű LTI esetben mutatunk példát.

Diszkrét idejű lineáris időinvariáns input-output modell. Tekintsük a legegyszerűbb, egy-bemenetű egy-kimenetű esetet! Ekkor a modell általános formája könnyen az úgynevezett *prediktív alakra* hozható

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) \cdots - a_n y(k-n) + b_0 u(k) + \cdots + b_m u(k-m)$$

ahol n és m ($n \geq m$) adott egész számok (a modell rendjei), a paramétervektor pedig:

$$p = [-a_1 \quad -a_2 \quad \dots \quad -a_n \quad b_0 \quad b_1 \quad \dots \quad b_m]^T$$

A *prediktív alakban felírt modellek az aktuális k -edik időpillanatbeli $y(k)$ kimenetet az aktuális $u(k)$ bemenet, és a régebbi bemenet és kimenet értékek függvényében írják fel.*

Diszkrét idejű lineáris időinvariáns állapotter modellek. Ezen modellek általános alakja az alábbi alakban írható, amelyen jól látszik az állapotter modellek fent leírt szerkezete.

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma u(k) && (\text{állapot egyenlet}) \\y(k) &= Cx(k) + Du(k) && (\text{kimenet egyenlet})\end{aligned}$$

adott $x(0)$ kezdeti feltétellel. A leírásban szereplő diszkrét idejű vektorértékű jelek

$$x(k) \in \mathcal{R}^n, \quad y(k) \in \mathcal{R}^p, \quad u(k) \in \mathcal{R}^r$$

A rendszer paraméterei az alábbi konstans mátrixok:

$$\Phi \in \mathcal{R}^{n \times n}, \quad \Gamma \in \mathcal{R}^{n \times r}, \quad C \in \mathcal{R}^{p \times n}, \quad D \in \mathcal{R}^{p \times r}$$

Az egymást követő diszkrét időpillanatokban tekintjük a jeleket, azaz

$$x(k) = x(t_k), \quad u(k) = u(t_k), \quad y(k) = y(t_k)$$

A modellbeli diszkrét idősorozat lehet ekvidisztáns ($t_k - t_{k-1} = \Delta h$), de ez nem feltétlenül teljesül.

2.2. A diagnosztikai feladat fajtái

A dinamikus modelleken alapuló diagnosztikai feladatok közös jellemzője valamely, speciális tulajdonságokkal rendelkező rendszermodell felhasználása. A modell fajtája az alkalmazott irányítástechnikai módszerrel együtt jellemzi a diagnosztikai feladatot.

2.2.1. Meghibásodás detektálás és azonosítás

A dinamikus rendszerek diagnosztikája a rendszer- és irányításelmélet egy speciális feladat-osztálya, ahol a rendszer modelljének vagy modelljeinek ismeretében a mért adatok segítségével állapítjuk meg, hogy a rendszer hibamentes (normál vagy referencia) állapotban van-e egy adott időpillanatban (ez az ún. *meghibásodás detektálás*), és ha nem, akkor milyen hiba vagy hibák léphettek fel (ez az ún. *meghibásodás-azonosítás*).

A fenti meghatározás rámutat arra, hogy a meghibásodás detektálás egyszerűbb feladat, mint a meghibásodás azonosítás. Ha ugyanis rendelkezünk a hibamentes referencia működést leíró modellel, akkor a mérési adatok és a modell ismeretében jó eséllyel következtethetünk arra, hogy a rendszer meghibásodott-e, vagy sem. *A hiba azonosításához további információkra, azaz a meghibásodott állapotban érvényes modellekre van szükség.*

2.2.2. Predikción alapuló diagnosztika

A predikción alapuló diagnosztika elve, hogy a rendszer mérhető kimenetét a rendszer bemeneti jelével gerjesztett rendszermodell(ek) kimenetével hasonlítjuk össze. Meghibásodás detektálásnál elegendő a hibamentes referencia modell kimenetével történő összehasonlítás és egy eltérési küszöbérték, meghibásodás azonosításnál több modellre van szükség. Ez utóbbi eset elvi feladatkitűzése az alábbi formában adható meg.

Predikción alapuló meghibásodás azonosítás: elvi feladatkitűzés*Adott:*

- A meghibásodási módok száma N_F (0=normál)
- **Prediktív** rendszermodellek minden meghibásodási módra

$$y^{(Fi)}(k+1) = \mathcal{M}^{(Fi)}(\mathcal{D}[1, k], u(k+1)), \quad k = 1, 2, \dots$$

ahol $y^{(Fi)}(k+1)$ az i -edik modellel jóslt kimenet a $(k+1)$ -edik időpillanatban,

- A mért adatok egy véges rekordja, amely bemenet-kimenet párokból áll:

$$D[0, k] = \{ (u(\tau), y(\tau) \mid \tau = 0, \dots, k) \}$$

- Egy $J^{(Fi)}$, $i = 0, \dots, N_F$ veszteségfüggvény, amely az i -edik modellel jóslt $y^{(Fi)}(\tau)$ és a mért $y(\tau)$ kimenet különbségéből képzett $r^{(i)}(\tau) = y(\tau) - y^{(Fi)}(\tau)$ reziduáltól függ:

$$J^{(Fi)}(y - y^{(Fi)}, u) = \sum_{\tau=1}^k [r^{(i)T}(\tau)Qr^{(i)}(\tau)]$$

Kiszámítandó: A rendszer aktuális meghibásodási módja, amely az az i modell index, amelyikre a veszteségfüggvény minimális.

2.2.3. Paraméterbecslésen alapuló diagnosztika

A paraméterbecslésen alapuló diagnosztika során a mért adatokból becsült, és a különböző hibamódokhoz rendelhető modellbeli paraméter-értékeket hasonlítjuk össze. Itt is elég a hibamentes nominális modell paramétereinek ismerete és egy hibakorlát a meghibásodás detektálásához, az azonosításhoz ugyanakkor valamennyi feltételezett hibához tartozó dinamikus modellt meg kell adni.

Paraméterbecslésen alapuló meghibásodás azonosítás: elvi feladatkitűzés*Adott:*

- A meghibásodási módok száma N_F (0=normal)
- **Parametrikus** rendszermodellek minden meghibásodási módra

$$y^{(Fi)}(k+1) = \mathcal{M}^{(Fi)}(\mathcal{D}[1, k]; p^{(Fi)}), \quad k = 1, 2, \dots$$

- A mért adatok egy rekordja: $D[0, k] = \{ (u(\tau), y(\tau) \mid \tau = 0, \dots, k) \}$
- A paraméterektől függő veszteségfüggvény $J^{(Fi)}$, $i = 0, \dots, N_F$

$$J^{(Fi)}(p^{(estFi)} - p^{(Fi)}) = \rho^{(i)T}Q\rho^{(i)}, \quad \rho^{(i)} = p^{(estFi)} - p^{(Fi)}$$

Kiszámítandó: A rendszer aktuális meghibásodási módja, amely az az i modell index, amelyikre a veszteségfüggvény minimális.

2.2.4. Állapot-becslésen alapuló diagnosztika

Sok esetben a meghibásodást nem egy statikus hibamóddal, hanem egy speciális zavarással (*hibajel*) írják le, amely a legáltalánosabb esetben akár időfüggő, sőt véletlenszerű (sztochasztikus) is lehet. A hibamentes normál működést ilyenkor a χ hibajel hiánya, azaz azonosan nulla értéke jelenti. Ennek fényében az állapotbecslésen alapuló diagnosztika egy speciális, a zavaró hibajellel kiegészített állapotter modellt igényel, ami a diszkrét idejű esetben az alábbi alakot ölti.

$$\begin{aligned} x(k+1) &= F(x(k), u(k); \chi(k)) && \text{(állapot egyenlet)} \\ y(k) &= G(x(k), u(k)) && \text{(kimenet egyenlet)} \end{aligned} \quad (2.3)$$

A diagnosztika elve ez esetben az, hogy a hibamentes dinamikus modellt felhasználva a mért bemenetek és kimenetek értékeiből megbecsüljük az állapot változók értékét minden időpillanatban az irányítástechnikában használatos állapot-becslési (szűrési, pl. Kálmán-szűrő) vagy megfigyelő-tervezési módszerek valamelyikével, és ebből következtetünk az adott időpillanatban a meghibásodás jelenlétére vagy hiányára.

Fontos megjegyezni, hogy a χ hibajel vektorértékű is lehet, így több meghibásodás is, sőt ezek kombinált hatása is leírható a hibajel koordinátaival.

Az állapot-becslésen alapuló diagnosztikai módszer finom időbeli felbontásával is kiténik a többi módszer közül, hátránya nagy számításigényében van. Ez a hátrány azonban a diszkrét diagnosztikai alkalmazásokban nem játszik nagy szerepet, és itt az állapotbecslésen alapuló módszereket széles körben elterjedten alkalmazzák.

3. fejezet

Fizikai alapú diszkrét dinamikus modellek

3.1. Dinamikus modellek felállítása mérnöki alapelvek alapján

A diagnosztikai célú modellek egy jelentős része fizikai alapú, mérnöki alapelvek felhasználásával felépített modellekből származik, ezért ebben a fejezetben röviden áttekintjük az ilyen modellek felállításának alapelveit, módszerét és lépéseit. Erről a módszertanról az érdeklődő Olvasó bővebb ismertetést talál a szakirodalomban [7].

A diszkrét diagnosztikai módszerek természetüknél fogva diszkrét modelleket igényelnek, ezért a fejezet második részében rövid összefoglaló található az úgynevezett kvalitatív (azaz minőségi) modellezésről, aminek segítségével a mérnöki alapelvekből származó modellek diszkrét értékészletű modellekké alakíthatóak.

3.1.1. A modellezési feladat

Amikor egy modellezési feladat felmerül, például diagnosztikai célra lenne szükségünk modellre, akkor a munka kezdetén célszerű, sőt szükséges is, hogy a megoldandó feladatot pontosan megfogalmazzuk.

Egy modellezési feladat általános kitűzése az alábbi alakban fogalmazható meg.

- **Adott:**
 - a modellezendő rendszer,
 - a modellezési cél.
- **Meghatározandó:** egy olyan matematikai modell, amely leírja a rendszer viselkedését az adott célra.

Fontos megjegyezni, hogy a modellezendő rendszer megadása önmagában nem definiálja a modellezési feladatot. Ez a tény rámutat a modellezési cél (például diagnosztika)



$$\frac{dh}{dt} = \frac{v_{be}}{A} sz_{be} - \frac{v_{ki}}{A} sz_{ki}$$

$$\frac{dT}{dt} = \frac{v_{be}}{Ah} (T_{be} - T) sz_{be} + \frac{Q_H}{c_p \rho Ah} k$$

$$h(0) = h_0, \quad T(0) = T_0$$

3.1. ábra. Modellezési példa

fontosságára a modell felállításánál: ugyanazon objektumnak különböző modellezési célokra nagyon különböző modelljei lehetségesek.

Egy egyszerű rendszert és annak elkészült matematikai modelljét mutatja be a 3.1. ábra. Az ábra bal oldalán a modellezendő rendszer, egy szamovár képe látható, a jobb oldalon pedig a szamovár időbeli viselkedését irányítási vagy diagnosztikai célra leíró dinamikus modell matematikai egyenletei. A modellegyenletekben az alábbi változók és paraméterek szerepelnek: h vízszint, t idő, v_{be} beáramlási térfogat-sebesség, v_{ki} kiáramlási térfogat-sebesség, sz_{be} beáramlási szelep állapota, sz_{ki} kiáramlási szelep állapota, A keresztmetszet, ρ sűrűség, T hőmérséklet, T_{be} beáramlási hőmérséklet, Q_H fűtőteljesítmény, k fűtéskapcsoló állapota, c_p fajhő, h_0 és T_0 kezdeti értékek.

A modellezési cél

A modellezési cél kiemelt jelentőségű a modell matematikai formájának és típusának, a modell változóinak és elvárt pontosságának megválasztásánál.

Az alábbiakban felsorolunk néhány tipikus modellezési célt és a célhoz illeszkedő matematikai modell típust.

- *Szimulációs célra* explicit formában adott, predikcióra alkalmas, közepes pontosságú, dinamikus vagy statikus modellek használatosak, utóbbiak az állandósult állapotok meghatározására szolgálnak.
- A *tervezési célra* készült modellek nagy pontosságot követelnek meg, ezeknek a modelleknek az ún. inverz feladat megoldását, azaz a modell változóinak ismeretében a paraméterek meghatározását is támogatniuk kell.
- Az *irányítási* (predikciós, értéktartó szabályozási, identifikációs, diagnosztikai) célra készített modellek dinamikus típusúak, itt általában nem szükséges nagy pontosság.

A modellezési cél meghatározza a modell értelmezési tartományát, és befolyásolja

- a figyelembevett jelenségeket (mechanizmusokat),
- modell matematikai formáját,
- modell pontosságát (a jellemző változókra nézve).

Mechanizmusok

A modellezés során figyelembevett jelenségeket vagy mechanizmusokat a modellezendő rendszer és a modellezési cél együttesen határozza meg. A rendszer fizikai, kémiai, biológiai stb. tulajdonságai meghatározzák, hogy miféle folyamatok jöhetnek egyáltalán szóba, ha a dinamikus viselkedésének okait próbáljuk megragadni. Egy pohár forró tea időbeli viselkedésének leírásához a párolgás jelenségére például biztosan szükség lesz, az ivaros szaporodás törvényszerűségeire pedig bizonyosan nem.

A szóbajöhethető jelenségek köréből a modellezési cél által meghatározott értelmezési tartomány, felhasználás és pontosság igények figyelembevételével választjuk ki azokat a mechanizmusokat, amelyek törvényszerűségeit a modellben felhasználjuk.

Az egyes mechanizmusok lefolyásának törvényszerűségeit a vonatkozó szaktudományok ismeretanyaga tartalmazza, például az előző példabeli párolgásét a hőtan vagy termodinamika. Ebben a jegyzetben olyan egyszerű fizikai és kémiai jelenségeket tartalmazó példákat fogunk csak használni, amelyek az alábbi mechanizmusokat tartalmazzák.

- *Áramlások*: amelyeken belül a konvekciós és a diffúziós áramlásokat különböztethetjük meg. Az egyszerű diagnosztikai modellekben a konvekciós áramok szerepelnek, ezek leírásával az áramlástan foglalkozik.
- Az energia-változásokat is leíró modellekben a különböző fázisok közötti *hő-, illetve energiaátadást* is szerepeltetjük, ezek tulajdonságait a hő- és áramlástan írja le. A hőközlési folyamatokhoz tartoznak a *hűtést és fűtést* leíró mechanizmusok is.
- Az anyagok minőségi változását a *kémiai reakciók* mechanizmusával írhatjuk le a reakciókinetika tudományága által meghatározott módon.
- A *fázisátalakulásokkal* (párolgás, forrás, olvadás, ...) a termodinamika tudományága foglalkozik.
- A termodinamikában tárgyalják a diagnosztikai modellekben elvétve szereplő elegyedési és szétválasztási mechanizmusokat (pl. desztilláció, oldódás, kristályosítás, ...) is.
- Igen ritkán még a következő mechnizmusok is szerepelnek diagnosztikai célú modellekben: hőszugárzás, bioreakció, adszorpció, abszorpció, ...

A 7 lépéses modellezési eljárás

A matematikai modellek felállításának módszere már kialakult a szakirodalomban, ahol a modell elkészítésének folyamatát (vagy eljárását) az alábbi hét lépésre bontják.

1. Probléma definiálása

Ennek során a modellezési feladat formális leírása készül el, amelynek része

- a rendszer körülhatárolása,
- modellezési cél meghatározása,
- a folyamatábra elkészítése (a berendezések és legfontosabb változók megadásával).

2. Mechanizmusok meghatározása

Itt a feladat a

- leírandó jelenségek összegyűjtése (pl. áramlás, átadás, reakció, párolgás).

3. Adatok összegyűjtése és értékelése

A rendelkezésre álló adatok lehetnek

- állandók táblázatokból (ezek pontosságát is meg kell adni!),
- berendezések és működésük jellemzői,
- mért adatok (előkísérletek eredményei).

4. Modell elkészítése

Hagyományosan ezt tekinthetjük a modellezés fő lépésének, de az ezt megelőző és követő lépések a modell minősége, konzisztenciája és használhatósága szempontjából alapvető fontossággal bírnak. Ennek a lépésnek al-lépései is vannak, amelyek az alábbiak.

- 4.1. mérlegelési térfogatok meghatározása,
- 4.2. modellezési feltételezések megfogalmazása,
- 4.3. modellegyenletek felírása (mérlegegyenletek, kiegészítő egyenletek),
- 4.4. kezdeti- és peremfeltételek megadása.

5. Modell megoldása

Ez a lépés magában foglalja a megoldást előkészítő és követő rész-lépéseket is, amelyek során szükséges

- a megoldó eljárás készítése vagy testreszabása,
- a megoldás hihetőségének és pontosságának ellenőrzése.

6. Modell megoldásának ellenőrzése

Ezt a lépést idegen szóval *modell verifikációnak* is nevezik. A verifikáció során elvégezzük például a

- kvalitatív viselkedés ellenőrzését mérnöki intuíció alapján,
- dinamikus tulajdonságok (pl. stabilitás) meglétének ellenőrzését.

7. Modell érvényességének ellenőrzése

A modell érvényességének ellenőrzése két részlépésből áll, mindkettőhöz a modellezendő rendszerről származó mért adatok szükségesek.

- 7.1. *modell kalibráció*, amelynek során a modellbeli bizonytalan/ismeretlen paraméterek meghatározása történik paraméterbecsléssel,
- 7.2. *modell validáció*, mikor a matematikai modellt és a valós rendszer mérésekkel meghatározott viselkedését hasonlítjuk össze statisztikai módszerekkel.

A 3.2. ábra a modellezési lépések sorrendi kapcsolatait mutatja. Megfigyelhető, hogy fő szabályként sorrendben hajtjuk végre a lépéseket. A 6. és a 7. ellenőrző lépés után azonban, ha az ellenőrzés negatív eredménnyel zárult, visszalépünk, és megismétlünk előző lépéseket a modell kibővítése, kiegészítése vagy javítása céljából.

3.1.2. Dinamikus mérlegegyenletek

Diagnosztikai célra szinte kizárólag úgynevezett koncentrált paraméterű modelleket használnak, amelyek közöséges differenciálegyenlet alakjában tartalmazzák a dinamikus mérlegegyenleteket. Az ilyen modellekben az időbeli változás leírására használatos jelek skalár- vagy vektorértékű időfüggvények.

A koncentrált paraméterű modellek a rendszerelméletben és az irányítástechnikában szokásos elnevezés alapján olyan közelítést alkalmaznak, amelynél az egyes rész-rendszereket jellemző mennyiségek térbeli eloszlását nem vesszük figyelembe, azaz fizikailag úgy tekintjük, mintha minden modellezendő térrész tökéletesen kevert lenne.

Mérlegelési térfogatok

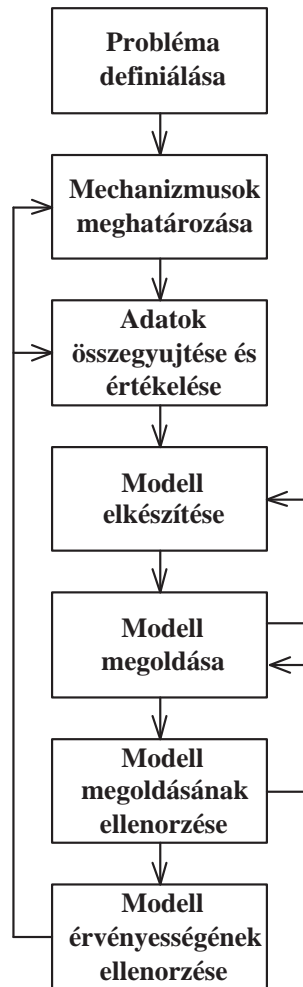
A dinamikus mérlegegyenletek a dinamikus modellek alap-egyenletei. Ezek felállításához a modellezendő rendszert úgynevezett *mérlegelési térfogatokra* bontjuk fel, és ezekre írunk fel dinamikus mérlegegyenleteket. A mérlegelési térfogatok diagnosztikai célra készített modellekben az alábbi tulajdonságokkal rendelkeznek.

- Általában (*időben*) *állandó térfogatúak*.
- Feltételezzük, hogy *tökéletesen keverték*, így koncentrált paraméterű modelleket kapunk közöséges differenciálegyenlet alakú mérlegegyenletekkel.

Megmaradó mennyiségek

A dinamikus mérlegegyenletekkel csak *megmaradó* (extenzív) *mennyiségek* időbeli változása írható le. Ezek a megmaradó extenzív mennyiségek az alábbiak:

- összes tömeg,
- energia (entalpia, belső energia),
- komponens tömeg,
- impulzus, de ezzel diagnosztikai célra készített modellekben nem számolunk.



3.2. ábra. A 7 lépéses modellezési eljárás

Dinamikus mérlegegyenlet

A dinamikus mérlegegyenlet valamely megmaradó extenzív mennyiségre felírt általános alakja az alábbi tagokat tartalmazza.

$$\left\{ \begin{array}{l} idobeli \\ valtozas \end{array} \right\} = \left\{ \begin{array}{l} be- \\ aramlas \end{array} \right\} - \left\{ \begin{array}{l} ki- \\ aramlas \end{array} \right\} + \left\{ \begin{array}{l} forras \\ nyelo \end{array} \right\}$$

Ezeknek a tagoknak a pontos matematikai formája az egyes mechanizmusok által létrehozott áramlások és forrástagok függvénye. A mérlegegyenletbeli tagok leírásához szükséges alapvető fogalmak és tudnivalók az alábbiakban foglalhatók össze.

Intenzív mennyiségek

Az intenzív mennyiségek az áramlások hajtóerői, ezért leírásuk és jellemzőik alapvető fontosságúak modellezési szempontból.

Az intenzív mennyiségek legfőbb fizikai jellemzője, hogy a részrendszerek egyesítésekor kiegyenlítődnek. Az intenzív mennyiségek közül a legfontosabbak ezen túlmenően

- potenciál típusú mennyiségek, amelyek
- áramlások és átadások hajtóerői (leggyakrabban lineáris közelítést alkalmazunk a hajtóerőknél "kereszthatás" nélkül), és
- mérhetőek.
- Minden *megmaradó extenzív mennyiségnek van intenzív párja* az alábbi felsorolás szerint:
 - összes tömeg m - nyomás p ,
 - energia U - hőmérséklet T ,
 - komponens tömeg m_X - koncentráció c_X (kémiai potenciál).

A termodinamikából (hőtanból) ismeretesek az alábbi, extenzív - intenzív mennyiségek közötti összefüggések:

- $U = c_P m T$ (c_P fajhő),
- $m_X = \frac{m}{\rho} c_X$ (ρ sűrűség).

Ezek a megmaradási elvekből származó modelleknél kiegészítő algebrai egyenletekként szerepelnek (lásd később).

Megmaradási egyenletek tökéletesen kevert esetben

A tökéletesen kevert mérlegelési térfogatok esetén közösleges differenciálegyenletekből és algebrai egyenletekből álló modell adódik. Ez a koncentrált paraméterű dinamikus modell kiválóan alkalmas diagnosztikai célra. Ebben a speciális esetben a dinamikus mérlegegyenletek egy mérlegelési térfogatra igen egyszerű alakot öltenek.

Az összes tömegre felírt dinamikus mérlegegyenlet. Jellemzői az alábbiak:

- *Nincs forrás/nyelő.*
- Az összes tömeg m mérhető (pl. szintmérés).
- A tökéletesen kevert esetben a be- (v_B) és kiáramlások (v_K) tömegáramok [kg/s].

A fentieket egy egyszerű, egy be- és kiáramlásos mérlegegyenlet példáján szemléltethetjük.

$$\frac{dm}{dt} = v_B - v_K$$

Az energiára felírt dinamikus mérlegegyenlet. Az összes tömeg mérlegegyenletére épül, az alábbi jellemzőkkel:

- A forrás/nyelő lehet külső (pl. elektromos) fűtés-hűtés vagy hőátadás Q ($[J/s]$).
- A tökéletesen kevert esetben a be- ($c_{pB}v_B T_B$) és kiáramlások ($c_{pK}v_K T$) energiaáramok [J/s].
- Az energia U közvetlenül **nem mérhető**, helyette a hőmérsékletet használjuk a végső modellegyenletben. Ezt a formát *intenzív alakra hozással* állítjuk elő (lásd később).

Az alábbi egyszerű, egy be- és egy kiáramlást és egy külső fűtést leíró tagot tartalmazó példával szemléltethetjük a fentieket:

$$\frac{dU}{dt} = c_{pB}v_B T_B - c_{pK}v_K T + Q$$

3.1.3. Kiegészítő algebrai egyenletek

A modell teljes matematikai leírásához általában szükségesek további összefüggések, amelyekkel a modell matematikailag megoldhatóvá, zárttá válik. Ezek matematikai formájukat tekintve általában algebrai egyenletek, amelyek tartalmi szempontból az alábbi csoportokba sorolhatók.

- *Extenzív-intenzív összefüggések:* ezekre már a fentiekben láttunk példát ($U = c_p m T$).
- *Átadást leíró egyenletek:* két egymással érintkező fázis határfelületén keresztüli nem-konvektív áramok leírására szolgálnak. Hőátadás esetén például az időegység alatt az A felületen a T_1 és T_2 hőmérsékletű fázisok között $j = K_T A (T_1 - T_2)$ mennyiségű energia áramlik a magasabb T_1 hőmérsékletű helyről az alacsonyabb felé.
- *Termodinamikai relációk,* például az általános gáztörvény vagy a párolgáshő hőmérsékletfüggését leíró algebrai egyenlet.
- Mérlegelési térfogatokra vonatkozó összefüggések.
- Berendezésre és szabályozóra vonatkozó összefüggések.
- Reakciókinetikát leíró egyenletek, stb.

A fenti felsorolásban a fontosabb kiegészítő egyenlet típusokat magyarázattal és példával is elláttuk.

3.1.4. Modellezési feltételezések

Modellezési feltételezések formájában fogalmazzuk meg azokat a körülményeket, feltételeket, megszorításokat stb., amelyek mellett az elkészült modell érvényes, azaz leírja a modellezendő objektum viselkedését a modellezési célban megfogalmazott szempontból. Ezért *a modellezési feltételezések a modell integráns részét képezik*, a leíró egyenletek mellett ezek megadása is feltétlenül szükséges a modellezés végeredményeként.

A modellezési feltételek származhatnak közvetlenül a feladat megfogalmazásából, rögzíthetik a modellező döntéseit a modellezés során figyelembevett vagy elhanyagolt mechanizmusokról, megszabhatják a modell értelmezési tartományát vagy valamely paraméterének a modellben megengedhető értékét. A fentiekből látható, hogy a modellezési feltételezések a modellezési eljárás során fokozatosan kerülnek megfogalmazásra, így ezek listáját *fokozatosan kell összegyűjteni*.

A leggyakoribb modellezési feltételezések az alábbiak:

- rendszer időbeli viselkedésére vonatkozó feltételezések (pl. dinamikus, állandósult állapotú),
- mérlegelési térfogatokra vonatkozó feltételezések (pl. csak folyadékfázis, gáz és folyadékfázis),
- térbeli eloszlásra vonatkozó feltételezések (pl. tökéletesen kevert/koncentrált paraméterű),
- jelenségekre (mechanizmusokra) vonatkozó feltételezések (pl. nincs párologás, van hőátadás),
- elhanyagolható hatásokkal kapcsolatos feltételezések (pl. sűrűség csak a hőmérséklettől függ, fajhő állandó),
- állapotok kívánt tartományai, pontosságuk (pl. a hőmérséklet 20 és 30 °C közötti).

3.1.5. A modell összetevői

Az elkészített modell a szabványos 7 lépéses modellezési eljárás végrehajtása során nyeri el végső formáját. Ebben az alábbi kötelező összetevők szerepelnek (*ezeket kell megadni a modellezés eredményeképpen*).

1. Rendszer leírása (folyamatábra változókkal),
2. Modellezési cél,
3. Mechanizmusok felsorolása,
4. Modellezési feltételezések listája,
5. A rendelkezésre álló adatok (adat, mértékegység, lelőhely, pontosság),
6. Mérlegelési térfogatok (folyamatábrán jelölve),

7. Modellegyenletek (megmaradási egyenletek, kiegészítő egyenletek, kezdeti- és peremfeltételek),
8. Modell változók és paraméterek listája (azonosító, jelentés, mértékegység).

3.1.6. Állapottér modell alak

A diagnosztikai célra kifejlesztett mérnöki dinamikus modelleket a rendszer- és irányításméletben szokásos úgynevezett **állapottér modell formára kell hozni** felhasználás előtt a kiegészítő egyenletek behelyettesítésével a dinamikus mérlegekből származó differenciálegyenletekbe, hogy az alábbi általános alakot kapjuk

$$\begin{aligned} \frac{dx}{dt} &= f(x, u) && \text{(állapot egyenlet)} \\ y &= h(x, u) && \text{(kimeneti egyenlet)} \end{aligned} \quad (3.1)$$

ahol x az állapotok, u a bemenetek, y a kimenetek vektora.

Ezek után a rendszer műszerezettségét és a feladat kiírást is figyelembe véve kell meghatározni, hogy az állapotter modell jeleinek mely modellbeli változók feleltethetők meg. A fizikai alapú modell jelei lehetnek

- állapotok (x): ezek a megmaradó mennyiségek (vagy intenzív párjuk), amelyek a dinamikus mérlegekből származó differenciálegyenletek differenciális változói,
- bemenetek (u): a differenciálegyenletek jobboldalán szereplő, befolyásolható vagy legalább mérhető változók,
- kimenetek (y): olyan mérhető, közvetlenül nem befolyásolható változók, amelyek állapotváltozók, vagy ezektől függenek.

3.1.7. Egyszerű példa: szabad kifolyású tartály

Rendszer leírása

Adott egy állandó keresztmetszetű tartály, amelyben tiszta vizet tárolunk. A vizet a csapból egy beömlő szeleppel ellátott nyíláson át ereszthetjük bele, a fogyasztáskor szükséges vizet egy kiömlő szelep segítségével ereszthetjük ki, a kiömlés sebessége a tartálybeli vízszint függvénye. Készítsük el a rendszer diagnosztikai célú modelljét, ha a vízszintet és a kapcsolók állapotát mérni tudjuk!

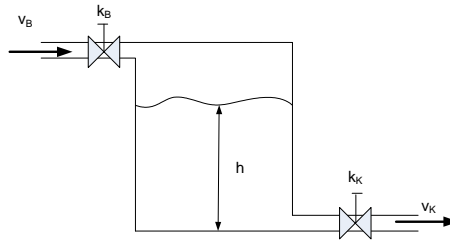
Mechanizmusok

- be- és kiáramlás,
- szabad kifolyás (hidrosztatikus nyomás által).

Modellezési feltételezések

F1 tökéletesen kevert,

F2 csak víz (csak összes tömegmérleg),



3.3. ábra. A folyamatába

F3 szabad kifolyás,

F4 állandó keresztmetszet A ,

F5 sűrűség (ρ) állandó.

Mérlegegyenletek

Csak összes tömegmérlegre van szükség:

$$\frac{dm}{dt} = v_B - v_K \quad (3.2)$$

Kiegészítő egyenletek

- $m = A \cdot h \cdot \rho$ (a h vízszint mérhető),
- $v_B = v_B^* k_B$ (a k_B $[0,1]$ kapcsoló állás mérhető),
- $v_K = K \cdot h \cdot k_K$ (szabad kifolyás, a k_K $[0,1]$ kapcsoló állás mérhető).

Modellegyenlet mérhető változókkal

$$\frac{dh}{dt} = \frac{v_B^*}{A\rho} k_B - \frac{K}{A\rho} h \cdot k_K \quad (3.3)$$

Állapottér alak

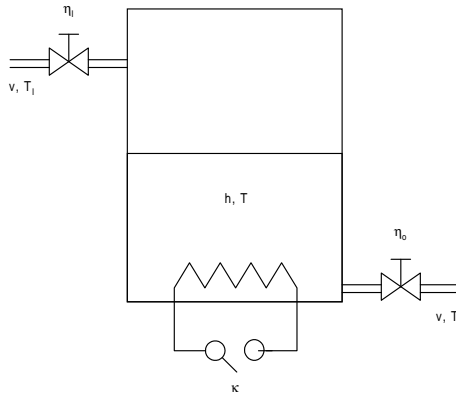
- állapotváltozó: h vízszint,
- bemenetek: k_B és k_K kapcsoló állások,
- kimenet: h vízszint.

A modellben szereplő változók és paraméterek

| | | | |
|---------|--|-------|-------------------------------------|
| t | idő [s] | m | víz tömege [kg] |
| h | vízszint [m] | K | kiáramlási állandó [m^2/sec] |
| v_B | beömlő térfogatsebesség [m^3/s] | v_K | kiömlő térfogatsebesség [m^3/s] |
| ρ | sűrűség [kg/m^3] | A | keresztmetszet [m^2] |
| k_B | beömlő szelepállás [1/0] | k_K | kiömlő szelepállás [1/0] |
| v_B^* | max. beömlő térfogatsebesség [m^3/s] | | |

3.1.8. A kávéfőző gép modellje**Rendszer leírása**

A kávéfőző gép egy tartály, amelyben egy elektromos, ki-bekapcsolható fűtőtest melegíti a vizet, amelyet a csapból egy beömlő szeleppel pótolhatunk, és az elkészült forró vizet egy kiömlő szelep segítségével eresztjük rá az őrlött kávéra (a folyamatábrát ld. a 3.4. ábrán).



3.4. ábra. A kávéfőző gép folyamatábrája

Készítsük el a rendszer diagnosztikai célú modelljét, ha a vízszintet és a víz hőmérsékletét mérni tudjuk!

Mechanizmusok

- be- és kiáramlás,
- külső fűtés.

Modellezési feltételek

1. csak a vízforraló részt írjuk le,
2. tökéletesen kevert,

3. konstans fizikai tulajdonságok,
4. állandó keresztmetszetű (A) tartály,
5. állandó H fűtőteljesítmény, bináris (κ) kapcsolóval.

Változók és paraméterek a modellben

| | | | |
|----------|---------------------------------|-------|---------------------------------|
| t | idő [s] | T_I | befolyási hőmérséklet [K] |
| h | vízszint [m] | H | fűtőteljesítmény [Joule/sec] |
| v_I | beömlő térfogatáram [m^3/s] | v_O | kiömlő térfogatáram [m^3/s] |
| c_p | fajhő [Joule/kgK] | A | keresztmetszet [m^2] |
| ρ | sűrűség [kg/m^3] | T | hőmérséklet [K] |
| κ | bináris fűtéskapcsoló [1/0] | | |

Modellegyenletek

Dinamikus mérlegegyenletek a vízre

- tömegmérleg

$$\frac{dM}{dt} = \rho v_I - \rho v_O \quad (3.4)$$

- energiamérleg

$$\frac{dE}{dt} = c_p \rho T_I v_I - c_p \rho T v_O + \kappa H \quad (3.5)$$

Kiegészítő egyenletek

$$M = \rho A h \quad (3.6)$$

$$E = c_p \rho A h T \quad (3.7)$$

A kiegészítő egyenleteket behelyettesíthetjük a mérlegegyenletekbe a mérhető változókat tartalmazó modell alak előállításának érdekében. Algebrai átalakítások után, és a (3.5) egyenletbe a (3.4) átalakított alakját beírva az alábbi két differenciálegyenlethez jutunk.

$$\frac{dh}{dt} = \frac{1}{A} v_I - \frac{1}{A} v_O \quad (3.8)$$

$$\frac{dT}{dt} = \frac{1}{A} v_I T_I \frac{1}{h} - \frac{1}{A} v_O T \frac{1}{h} + \frac{H}{c_p \rho A} \kappa \frac{1}{h} \quad (3.9)$$

Kezdeti értékek: $h(0) = h_0$, $T(0) = T_0$

Paraméterek: A , H , c_p , ρ

Változók (jelek):

- állapotváltozók (x): h , T ,
- bemenetek (u): v_I , v_O , T_I , κ ,
- kimenetek (y): h , T .

3.2. Kvalitatív dinamikus modellek

A kvalitatív modellek az előző alfejezetben tárgyalt közönséges, mindennapi modellek olyan változatai, amelyekben a változók és paraméterek *kvalitatív (minőségi) értékeket* is felvehetnek. Ilyen kvalitatív érték leggyakrabban egy intervallum, amikor pl. egy paraméter értékéről csak annyit tudunk, hogy milyen határok közé esik.

Ebben az alfejezetben a diagnosztikában használt legfontosabb kvalitatív dinamikus modellel és származtatásukkal foglalkozunk, ezek a konfluenciák és a kvalitatív differenciaegyenletek.

Az állapotter modellek szerkezeti leírásából származtatott úgynevezett előjeles irányított gráf (SDG) modellel viszont nem itt, hanem majd a 4. fejezetben foglalkozunk.

3.2.1. Előjel és intervallum aritmetikák

A kvalitatív modellekben a közönséges, jól megszokott aritmetikai műveletek (pl. összeadás, szorzás) helyett a változók diszkrét értékkészlet halmaza, mint alaphalmazon értelmezett műveleteket használunk. Diagnosztikai célokra a *logikai (igaz és hamis)* értékkészlet mellett *előjel* (0,+,-) és *intervallum értékkészletű* változókat és paramétereket szokás használni.

Előjel aritmetika

Az előjel aritmetika az ún. kibővített előjel érték-halmazon

$$U_S = \{ +, -, 0; ? \}, \quad ? = + \cup 0 \cup - \quad (3.10)$$

értelmezett összeadásra (\oplus_S) és szorzásra (\otimes_S) épül, amelyeket - a logikai műveletekhez hasonlóan - ún. *műveleti táblák* segítségével adunk meg. A *határozatlan előjel* halmaz (?) az egymással diszjunkt pozitív +, negatív - és nulla 0 halmazok uniójaként áll elő.

Az előjel összeadás. Műveleti táblája a 3.1. táblázatban látható, ahol a hagyományos "elemi" előjelekre vonatkozó részt kettős vonallal különítettük el. Innen leolvasható, hogy ez a művelet kommutatív (a tábla a főátlójára szimmetrikus), de *növeli a bizonytalanságot*, mert bizonyos esetekben határozott előjelű operandusok esetén a művelet eredménye határozatlan előjelű lesz, például $+ \oplus - = ?$.

3.1. táblázat. Az előjel összeadás műveleti táblája

| $a \oplus_S b$ | + | 0 | - | ? |
|----------------|---|---|---|---|
| + | + | + | ? | ? |
| 0 | + | 0 | - | ? |
| - | ? | - | - | ? |
| ? | ? | ? | ? | ? |

Az előjel szorzás. Műveleti táblája (lásd 3.2. táblázat) hasonló az előjel összeadáséhoz, ez a tábla is mutatja azt, hogy a művelet kommutatív. Ez esetben a bizonytalanság

csökkenése figyelhető meg, hiszen van eset, amikor határozatlan előjelű operandus 0-val történő szorzása határozott előjelű (azaz 0) eredményt produkál.

3.2. táblázat. Az előjel szorzás műveleti táblája

| $a \otimes_S b$ | + | 0 | - | ? |
|-----------------|---|----------|---|----------|
| + | + | 0 | - | ? |
| 0 | 0 | 0 | 0 | 0 |
| - | - | 0 | + | ? |
| ? | ? | 0 | ? | ? |

A fenti összeadás és szorzás, valamint az ezekből származtatott előjel kivonás és osztás műveletekkel generált aritmetika az ún. előjel aritmetika.

Intervallum aritmetikák

Az intervallum aritmetikákhoz választanunk kell egy értékkészletet valós fix vagy szabad végpontokkal rendelkező intervallumokból az alábbi formában

$$U_{\mathcal{I}} = \{[a_\ell, a_u] \mid a_\ell, a_u \in \mathcal{R}, a_\ell \leq a_u\} \quad (3.11)$$

ahol a fenti intervallumok (érték-halmazok) végpontjait az ún. **határpont halmazban** (angolul *landmark set*) gyűjthetjük össze:

$$L_{\mathcal{I}} = \{a_i \mid a_i \leq a_{i+1}, i \in I \subseteq \mathcal{N}\} \quad (3.12)$$

Ezután a fenti értékkészleten (ami egy intervallumokból álló halmaz) definiáljuk az intervallum összeadás ($\oplus_{\mathcal{I}}$) és intervallum szorzás ($\otimes_{\mathcal{I}}$) műveleteket.

Normalizált intervallumok. Ezek *nem fix végpontú* intervallumok, ahol a változóknak és paramétereknek meghatározható egy úgynevezett "normális" (jelölve N) értéke, és a szóba jöhető egyéb értékei ehhez a normális értékhez képest írhatóak le az alábbiak szerinti értékkészlet halmazokkal

$$\mathcal{Q} = \{H, N, L, 0\}, \quad \mathcal{Q}_{\mathcal{E}} = \{H, N, L, 0, e+, e-\} \quad (3.13)$$

ahol a jelölések a *High, Low, Normal, error* kvalitatív értékeknek felelnek meg.

A normalizált intervallumokra is definiálhatunk aritmetikai műveleteket műveleti táblák formájában az előjel aritmetikánál megismert módon, de itt - az intervallumok szabad végpontja miatt - több lehetséges műveleti tábla is megfeleltethető egy műveletnek, amelyek közül *heurisztikus* megfontolásokkal választhatjuk ki a céljainknak megfelelőt. A 3.3. táblázat a normalizált intervallumok összeadásának egy lehetséges műveleti táblája. Ez a definíció is megőrzi az összeadás kommutatív tulajdonságát, hiszen a tábla a főátlójára szimmetrikus. Úgyszintén mutatja a tábla az intervallum összeadás *bizonytalanság-növelő* tulajdonságát, hiszen például $N \oplus_{\mathcal{Q}} N = e+$, azaz két véges hosszúságú intervallum összege végtelen hosszúságú.

3.3. táblázat. A normalizált intervallumok összeadásának műveleti táblája

| $[a] \oplus_{\mathcal{Q}} [b]$ | 0 | L | N | H |
|--------------------------------|-----|------|------|------|
| 0 | 0 | L | N | H |
| L | L | N | H | $e+$ |
| N | N | H | $e+$ | $e+$ |
| H | H | $e+$ | $e+$ | $e+$ |

3.2.2. A kvalitatív dinamikus modellek származtatása

A kvalitatív modellek *szisztematikusan* levezethetők a mérnöki modellekből

- intervallum értékű változók és paraméterek használatával és
- az egyenletek egyszerűsítésével.

3.2.3. Konfluenciák és a belőlük származtatott szabályok

A konfluenciák a mesterséges intelligencia területén de Kleer and Brown [18] által kidolgozott ún. "kvalitatív fiziká"-ból származnak, akik a fizikai modellekben szereplő algebrai és differenciálegyenleteket előjel aritmetika felhasználásával alakítottak át *kvalitatív differenciál-algebrai egyenletekké*.

A konfluenciák az alábbi egyszerű lépésekben algoritmikusan is származtathatók egy közös differenciál és algebrai egyenleteket tartalmazó dinamikus modellből.

1. Minden modellbeli $q(t)$ változóhoz definiáljuk a $[q]$ és δq *kvalitatív változókat* a következőképpen:

$$q \sim [q] = \text{sign}(q), \quad dq/dt \sim \delta q = \text{sign}(dq/dt)$$

azaz tekintjük a változó és annak időbeli deriváltja előjel-értékét.

2. A modellbeli *műveleteket előjel műveletekre cseréljük*, azaz

$$+ \sim \oplus_S, \quad * \sim \otimes_S \quad \text{stb.}$$

3. A modellben található *paramétereket* a $+$, $-$ vagy 0 *előjel konstansokra cseréljük* a konfluenciák egyenleteiben, azaz ezek látszólag eltűnnek az egyenletekből.

A konfluenciák megoldása. A megoldást az előjel műveletek megadásához hasonlóan *igazságtábla* vagy műveleti tábla formájában adjuk meg, ahol a tábla első oszlopában a konfluencia egyenlet baloldalán szereplő kvalitatív változó értéke áll, a többi oszlopban pedig felsoroljuk az egyenlet jobboldalán szereplő összes változó összes lehetséges érték-kombinációját.

Fontos megjegyezni, hogy egy *konfluencia megoldás táblájának* mérete a benne szereplő változók számával exponenciálisan nő.

3.4. táblázat. A (3.15) konfluencia megoldás táblája

| δh | $[\eta_I]$ | $[\eta_O]$ |
|------------|------------|------------|
| 0 | 0 | 0 |
| - | 0 | + |
| + | + | 0 |
| ? | + | + |

Példa: Kávéfőző gép tömegmérlegéből származtatott konfluencia.

Tekintsük a kávéfőző gép dinamikus modelljében szereplő tömegmérleget (lásd (3.8) egyenlet) abban az esetben, ha a beömlő v_I térfogatáramot egy η_I kétállású szelep, a kiömlő v_O térfogatáramot pedig a η_O kétállású szelep vezérli, azaz $v_I = \eta_I v$ és $v_O = \eta_O v$, ahol v egy konstans. Ekkor:

$$\frac{dh}{dt} = \frac{v}{A} \eta_I - \frac{v}{A} \eta_O \quad (3.14)$$

Ebből a fenti algoritmussal az alábbi lépésekben készíthetünk konfluenciát:

1. *kvalitatív változók*: $[\eta_I] \in \{0, +\}$, $[\eta_O] \in \{0, +\}$
2. minden *előjel konstans* értéke ”+”
3. a *konfluencia*

$$\delta h = [\eta_I] \ominus_S [\eta_O] \quad (3.15)$$

A konfluencia igazság- vagy megoldás táblája a 3.4. táblázatban látható.

Szabályok generálása. A *konfluenciák megoldás táblájának egy sora szabályként értelmezhető*, ha azt jobbról balra haladva olvassuk ki.

Például a kávéfőző gép tömegmérlegének (3.15) konfluenciájából az $\eta_I = 0$, $\eta_O = +$ érték-kombináció a $\delta h = -$ értéket adja, amiből a

$$\text{if } (\eta_I = \text{closed}) \text{ and } (\eta_O = \text{open}) \text{ then } (h = \text{decreasing})$$

szabály generálható.

Fontos megjegyezni, hogy a *konfluenciák megoldás tábláiból generált szabályrendszerek mindig teljesekek és ellentmondásmentesek*.

3.2.4. Kvalitatív differencia-algebrai egyenletek

A kvalitatív differencia és algebrai egyenletekből álló dinamikus modellek biztosítják a ”legfinomabb” leírást a kvalitatív modellek közül. Ebben az alfejezetben megismerkedünk ezek származtatásával és megoldási módjával.

Később, a 7. fejezetben tárgyaljuk azt, hogy ezek a modellek hogyan alakíthatóak színes Petri háló modellekké, és hogyan használhatóak diagnosztikai célra.

A diszkrét idejű kvalitatív differencia-algebrai egyenletek (DAE) származtatása

Mint azt már a 3.1. alfejezetben korábban láttuk, a megmaradási elvekből levezetett mérnöki dinamikus modellek folytonos idejű differenciál- és algebrai egyenletekből állnak. Ezekből az alábbi lépésekkel nyerhetünk kvalitatív differencia és algebrai egyenletekből álló dinamikus modellt.

1. A differenciál egyenletek a dinamikus mérlegegyenletekből származnak: ezeket **differencia egyenletekké kell alakítani** (időbeli diszkrétizálással).
2. A változóknak és paramétereknek **kvalitatív értékkészletet** (univerzum) kell választani.
3. Le kell vezetni az egyenletek kvalitatív formáját.

Kvalitatív jelek. Diagnosztikai célra célszerű olyan intervallum-univerzumot és erre épülő aritmetikát (lásd 3.2.1 pont) használni, amelynél a változóknak van egy "normális" N értékük, és az ennél alacsonyabb illetve magasabb kvalitatív értékek szerepelnek a nem rögzített végpontú univerzumban. Így a kvalitatív differencia és algebrai egyenletekből álló dinamikus modelleknél az alábbi univerzumokat használjuk:

$$\mathcal{Q} = \{H, N, L, 0\}, \quad \mathcal{B} = \{0, 1\}, \quad \mathcal{Q}_\varepsilon = \{e+, H, N, L, 0, e-\}$$

ahol *High*, *Low*, *Normal*, *error*.

Egy *kvalitatív jel* egy olyan jel (bemenet, kimenet, állapot vagy **zavarás (fault indicator!)**), amely értékeit minden időpillanatban egy véges kvalitatív értékkészletből veszi.

Egy *esemény* következik be, ha egy kvalitatív jel értéke megváltozik. Egy e_X eseményt formálisan az $e_X(t, q_X) = (t, [x](t) = q_X)$ párral jellemezhetünk, ahol t az az időpillanat, amikor az $[x]$ kvalitatív jel felveszi a q_X értéket.

Az egyenletek átalakítása. Ha a fenti kvalitatív érték univerzumokat és az azokon értelmezett műveleteket meghatározzuk, akkor a közöséges differencia és algebrai egyenletekből álló modellekből úgy kaphatjuk ezek kvalitatív megfelelőit, hogy

- a változók (jelek) adott időpillanatbeli értékét a megfelelő kvalitatív értékkel helyettesítjük,
- a paraméterek értékeit is a megfelelő kvalitatív értékkel helyettesítjük,
- az egyenletekben szereplő műveletek helyett az adott univerzum feletti kvalitatív műveletet tekintjük.

Kvalitatív DAE-k megoldása

A kvalitatív differencia és algebrai egyenletekből álló modellek megoldásához először a jel-nyomok fogalmát vezetjük be.

Jel-nyomok – esemény-sorozatok. Egy $[x]$ kvalitatív jel *jel-nyoma* (*signal trace*) egy esemény-sorozat

$$\mathcal{T}_{(x)}(t_0, t_F) = \{(t_0, [x](t_0) = q_{x0}), (t_1, [x](t_1) = q_{x1}), \dots, (t_F, [x](t_F) = q_{xF})\}$$

a (t_0, t_F) időintervallumon, ahol $q_* \in \mathcal{Q}_x$.

Ha egy jel-nyomban minden diszkrét időpillanathoz tartozik egy esemény, akkor az időpillanatok azonosítóit elhagyhatjuk, azaz a fenti jel-nyom leírása egyszerűsíthető

$$\mathcal{T}_{(x)}(t_0, t_F) = \{(q_{x0}), (q_{x1}), \dots, (q_{xF})\}$$

Több jelnek együttesen is lehet jel-nyoma, például az $u \in \mathcal{Q}$, $d \in \mathcal{Q}$, $y \in \mathcal{Q}$ esetben egy lehetséges jel-nyom a

$$\mathcal{T}_{(u,d,y)}(1, 3) = \{(N, N, N), (N, L, L), (N, L, 0)\}$$

alakban írható.

Diagnosztikai célra definiálhatunk

- nominális (normális viselkedést leíró) jel-nyomokat,
- karakterisztikus (jellemző) jel-nyomokat (valamilyen adott meghibásodásra).

A megoldás előállítás. A kvalitatív differencia és algebrai egyenletekből álló modellek megoldásának megadása - hasonlóan a konfluenciák igazság-táblájához - **megoldás-táblával** történik. Ennek előállításához az alábbi lépéseket kell elvégezni.

1. Összegyűjtjük valamennyi *független* (az egyenlet jobboldalán lévő) *változót* (ezek időben változó értékűek).
2. Felsoroljuk mindegyikhez az ő összes lehetséges jel-nyomait.
3. Szisztematikusan felsoroljuk az **összes lehetséges jel-nyom kombinációt**.
4. Minden ilyen kombinációra megoldjuk az egyenletet, és a megoldás jel-nyomot is feltüntetjük a megfelelő oszlopban.

Fontos megjegyezni, hogy exponenciálisan növekszik a táblázat mérete a változók számával és a jel-nyomok hosszával is.

Statikus példa: érzékelő additív hibával

Tekintsünk egy érzékelőt, amely a valódi v értéket egy konstans additív hibával terheli. Ekkor egy *algebrai modellegyenlet* írható fel az érzékelő működésére az alábbi alakban:

$$v^m = v + \chi \cdot E$$

ahol $[v] \in \mathcal{Q}$, $[v]^m \in \mathcal{Q}_e$, $\chi \in B_{-1} = \{-1, 0, 1\}$ és $[E] = L$ (kis torzítás).

A 3.5. táblázat mutatja az additív hibával terhelt érzékelő kvalitatív modellegyenletének megoldás-tábláját.

3.5. táblázat. Additív hibával terhelt érzékelő kvalitatív modellegyenletének megoldás-táblája

| $[v^m]$ | $[\chi]$ | $[v]$ | mode |
|---------|----------|-------|--------|
| N | 0 | N | normal |
| H | 0 | H | normal |
| L | 0 | L | normal |
| 0 | 0 | 0 | normal |
| $e+$ | 1 | H | faulty |
| H | 1 | N | faulty |
| N | 1 | L | faulty |
| L | 1 | 0 | faulty |
| N | -1 | H | faulty |
| L | -1 | N | faulty |
| 0 | -1 | L | faulty |
| $e-$ | -1 | 0 | faulty |

Dinamikus példa: a kávéfőző tömegmérlege

Dinamikus differencia egyenlet modellként a 3.1.8. pontban bemutatott kávéfőző (3.14) egyenletbeli tömegmérlegének kvalitatív változatát készítjük el.

A *differencia egyenlet* modellt a folytonos idejű differenciálegyenlet alakú modellből ekvidisztáns mintavételezéssel állíthatjuk elő az alábbi alakban:

$$[h]^T(k+1) = [h](k) + \chi_I(k) \cdot [v] - \chi_O(k) \cdot [v]$$

ahol $[h], [h]^T \in \mathcal{Q}_e$, $\chi_I, \chi_O \in \mathcal{B}$ és $[v] = L$.

A megoldás-tábla **azonosan konstans** inputokra a 3.6. táblázatban látható.

3.6. táblázat. Kávéfőző tömegmérleg kvalitatív modellegyenletének megoldás-táblája

| $[h]^T$ | $[h](t_0)$ | χ_I | χ_O |
|---------------|------------|-----------|-----------|
| (N, N, N) | N | $(1,1,1)$ | $(1,1,1)$ |
| (L, L, L) | L | $(1,1,1)$ | $(1,1,1)$ |
| ... | ... | ... | ... |
| (N, N, N) | N | $(0,0,0)$ | $(0,0,0)$ |
| ... | ... | ... | ... |
| $(e+, e+, H)$ | N | $(1,1,1)$ | $(0,0,0)$ |
| $(e+, H, N)$ | L | $(1,1,1)$ | $(0,0,0)$ |
| ... | ... | ... | ... |
| $(e-, 0, L)$ | N | $(0,0,0)$ | $(1,1,1)$ |
| $(e-, e-, 0)$ | L | $(0,0,0)$ | $(1,1,1)$ |
| ... | ... | ... | ... |

4. fejezet

Előjeles irányított gráf (SDG) modelleken alapuló diagnosztika

A legegyszerűbb diagnosztikai célokra használt diszkrét modell csak a változók közötti közvetlen vagy közvetett hatások meglétét és előjelét írja le előjelekkel súlyozott irányított gráfok, angol rövidítésükkel **S**igned **D**irected **G**raph (SDG) formájában. Ebben a fejezetben az SDG modellek származtatásával, tulajdonságaival, és diagnosztikai célokra történő felhasználásuk módjával foglalkozunk.

4.1. Az előjeles irányított gráf (SDG) modellek származtatása

Az előző fejezetben láttuk, hogy a diagnosztikai célra alkalmas dinamikus koncentrált paraméterű modelleket hogyan állíthatjuk fel mérnöki alapelvekből kiindulva a dinamikus mérlegegyenleteket felhasználva. Azt is láttuk, hogy ezek a modellek felírhatóak **állapot-tér modell formában**:

$$\begin{aligned} \frac{dx}{dt} &= F(x, u) && \text{(állapot egy.)} \\ y &= h(x, u) && \text{(kimeneti egy.)} \end{aligned} \tag{4.1}$$

ahol F és h adott nemlineáris függvények, és a modellbeli jelek (időfüggvények) közül x az állapotok, u a bemenetek és y a kimenetek vektora.

A dinamikus megmaradási modelleknél:

- az állapotegyenletek a dinamikus mérlegegyenletekből származnak,
- a bemenetek és kimenetek pedig a műszerezés függvényei (is).

4.1.1. Állandósult állapot körüli linearizálás

Az SDG moelleket a (4.1) állapotter modellből annak egy állandósult állapota körüli linearizálás segítségével származtathatjuk, ezért először ezt a műveletet tárgyaljuk.

Az állandósult állapot(ok)

A (4.1) állapotter modell egy x_0 *állandósult állapotát* úgy kapjuk, hogy az állapot egyenletből származó

$$0 = F(x_0, u_0)$$

nemlineáris algebrai egyenletet megoldjuk egy adott konstans állandósult u_0 bemenetre.

Az úgynevezett *input-affin rendszereknél* a fenti általános F függvény speciális $F(x, u) = f(x) + g(x)u$ alakú, ezért az alábbi nemlineáris algebrai egyenletet kell ilyenkor megoldani adott u_0 -ra

$$0 = f(x_0) + g(x_0)u_0 = F(x_0, u_0) \quad (4.2)$$

Fontos megjegyezni, hogy a fenti egyenletnek lehet több megoldása is, vagy előfordulhat, hogy egyáltalán nincs megoldása.

Az x_0 állandósult állapot ismeretében az állandósult kimenet az

$$y_0 = h(x_0)$$

kimeneti egyenletből már könnyűszerrel meghatározható.

Az állandósult állapot segítségével képezhetjük az $\tilde{x} = x - x_0$ *centrál állapotváltozót*. Hasonlóan centrál bemenet ($\tilde{u} = u - u_0$) és kimenet ($\tilde{y} = y - y_0$) is értelmezhető.

Linearizált állapotter modellek

Az állapotter modellek állandósult állapot körüli linearizálásánál a *többszörös függvények linearizálásából* indulunk ki, ahol az adott $y = h(x_1, \dots, x_n)$, $h : \mathcal{R}^n \mapsto \mathcal{R}^m$ függvényt egy x_0 pontja körül Taylor sorba fejtünk, és elhanyagoljuk a $\tilde{x} = x - x_0$ -ben a lineárisnál magasabb rendű tagokat. Ekkor a sorfejtésből az alábbi úgynevezett *linearizált alakot* kapjuk

$$\begin{aligned} \tilde{y} &= J^{(h,x)} \Big|_{x_0} \cdot \tilde{x} \\ J_{ji}^{(h,x)} &= \frac{\partial h_j}{\partial x_i}, \end{aligned}$$

ahol $J^{(h,x)}$ a h függvény Jacobi-mátrixa és $y_0 = h(x_0)$.

A fentiek segítségével a *nemlineáris állapotter modellek* egy adott (u_0, x_0) állandósult állapot körüli linearizált alakját úgy kaphatjuk meg, hogy linearizáljuk a nemlineáris többszörös függvényeket a

$$\dot{x} = f(x) + g(x)u = F(x, u)$$

$$y = h(x)$$

egyenletekben az állandósult állapot körül.

A gyakorlati alkalmazások szempontjából fontos *input-affin* esetben az $\eta = F(x, u) = f(x) + g(x)u$ függvényt és az $y = h(x)$ függvényt linearizáljuk az (x_0, u_0) pont körül

$$\tilde{y} = J^{(F,x)}\Big|_{x_0, u_0} \cdot \tilde{x} + J^{(F,u)}\Big|_{x_0, u_0} \cdot \tilde{u} \quad (4.3)$$

$$\tilde{y} = \left(J^{(f,x)}\Big|_0 + J^{(g,x)}\Big|_0 u_0 \right) \cdot \tilde{x} + g(x_0) \cdot \tilde{u} \quad (4.4)$$

A fentiek alapján az input-affin nemlineáris állapottér modellek LTI állapottér modell formája az alábbi alakban írható:

$$\begin{aligned} \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}\tilde{u} \\ \tilde{y} &= \tilde{C}\tilde{x} + \tilde{D}\tilde{u} \end{aligned}$$

ahol

$$\tilde{A} = J^{(f,x)}\Big|_0 + J^{(g,x)}\Big|_0 u_0, \quad \tilde{B} = g(x_0), \quad \tilde{C} = J^{(h,x)}\Big|_0, \quad \tilde{D} = 0$$

4.1.2. Az állapottér modellek szerkezete

Az előző alfejezetben láttuk, hogy a nemlineáris állapottér modellekhez munkapont körüli linearizálással *linearizált* LTI állapottér modelleket rendelhetünk

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu \quad (\text{állapot egy.}) \\ y &= Cx + Du \quad (\text{kimeneti egy.}) \end{aligned} \quad (4.5)$$

Ezek segítségével pedig egy tetszőleges, akár LTI akár nemlineáris állapottér modell szerkezete is jellemezhető.

Az állapottér modellek szerkezetének leírásához az úgynevezett *előjeles struktúra mátrixok*ot használjuk. Egy A valós mátrix $[A]$ előjeles struktúra mátrixát az alábbi módon definiáljuk:

$$[A]_{ij} = \begin{cases} + & \text{if } a_{ij} > 0 \\ 0 & \text{if } a_{ij} = 0 \\ - & \text{if } a_{ij} < 0 \end{cases} \quad (4.6)$$

A struktúra gráf

Az állapottér modellek szerkezetének leírására szolgál a $S = (V, \mathcal{E}; w)$ *struktúra gráf*, amely egy előjeles irányított gráf.

- *Csúcshalmaza* tartalmaz egy-egy csúcsot az állapot, kimenet és bemenet változóknak

$$\begin{aligned} V &= X \cup U \cup Y \\ X \cap U &= X \cap Y = U \cap Y = \emptyset \end{aligned}$$

- \mathcal{E} *irányított élei* a változók közötti *közvetlen* hatásokat írják le.
- A w *élsúlyok* a hatás *előjelének* felelnek meg.

A *struktúra gráf* megmutatja, hogy melyik kimenetre (állapotra) mely más jelek vannak *közvetlen, illetve közvetett* hatással.

A struktúra gráf előfordulási mátrixa. Algebrai kapcsolat létezik a struktúra gráf O előfordulási mátrixa és az LTI állapotter modell vagy a nemlineáris állapotter modell linearizált változatának (4.5) egyenlettel leírt (A, B, C, D) paraméterei között. Az O előfordulási mátrix o_{ij} eleme a definíció alapján

$$o_{ij} = \begin{cases} w_{ij} & , \text{ ha} \\ 0 & , \text{ egyebkent} \end{cases} \quad (v_i, v_j) \in E$$

Ezért az (A, B, C, D) (linearizált) LTI állapotter modellre a változóknak megfeleltetett csúcsok (u, x, y) sorrendjében az

$$O = \begin{pmatrix} 0 & 0 & 0 \\ [B] & [A] & 0 \\ [D] & [C] & 0 \end{pmatrix}$$

alak adódik.

Példa: Fröccsöntő gép anyagtartály

Az alábbi egyszerű példával azt illusztráljuk, hogy a szerkezeti gráf felrajzolásakor nem feltétlenül szükséges felállítani a nemlineáris állapotter modellt, hiszen a változók közötti közvetlen hatás vagy annak hiányának megállapításához, sőt a hatás előjelének meghatározásához is sokszor elegendő a közvetlen mérnöki intuíció.

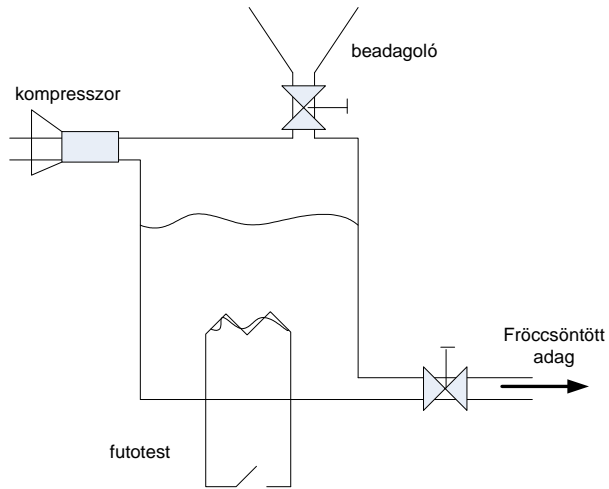
Folyamatábra Tekintsük a 4.1. ábrán látható fröccsöntő gépet, amelynek tartályába felülről egy adagoló szeleppel ellátott nyíláson keresztül adhatunk be műanyag granulátumot. A tartály nyomás alatt működik, a nyomást egy kompresszorral állítjuk elő (nyomásesés esetén egy motor lép üzembe). A tartályban lévő anyagot egy ki-be kapcsolható elektromos fűtőszállal fűtjük, hogy megolvadjon. Időről-időre egy adagoló szelep kinyitásával az olvadt műanyagot a fröccs-formába sajtoljuk.

Struktúra gráf Válasszuk a fröccsöntő gép jellemzésére az alábbi változókat:

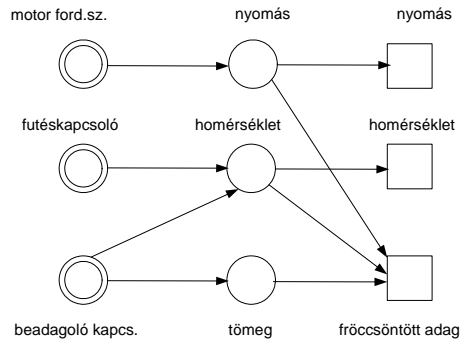
- állapotváltozók: a tartályban lévő műanyag massa tömege, hőmérséklete és nyomása,
- bemenetek: a motor fordulatszáma, a beadagoló kapcsoló állapota és a fűtéskapcsoló állapota,
- kimenetek: nyomás, hőmérséklet és a fröccsöntött adag tömege.

Ezekkel a változókkal a 4.2. ábrán látható SDG modellt rajzolhatjuk fel. *Az ábrán az állapotváltozóknak megfeleltetett csúcsokat körökkel, a bemeneteket kettős körökkel, míg a kimeneteket négyzetekkel jelöltük.*

A 4.2. ábrán látható struktúra gráfról könnyen leolvashatóak a változók közötti közvetlen és közvetett kapcsolatok a gráf csúcspontjainak és éleinek segítségével. Például közvetlen hatás (irányított él) van a motor fordulatszáma és a tartályban lévő nyomás, valamint a tartályban lévő nyomás és a mért nyomás között. A motor fordulatszáma és a mért nyomás közötti közvetett hatásnak a két csúcs között található irányított út felel meg. Azt is láthatjuk az ábráról, hogy nincs semmilyen (azaz se közvetlen se közvetett) hatás például a motor fordulatszáma és a hőmérséklet között, mert a fordulatszámnak megfelelő csúcsból nem vezet irányított út hőmérsékletnek megfeleltetett csúcsba.



4.1. ábra. A fröccsöntő gép folyamatábrája



4.2. ábra. A fröccsöntő gép struktúra gráfja

Kávéfőző gép SDG modellje

Példaként állapotter modell alapján felírt szerkezeti gráfra bemutatjuk a 3.1.8. alfejezetben megismert kávéfőző gép SDG modelljének levezetését.

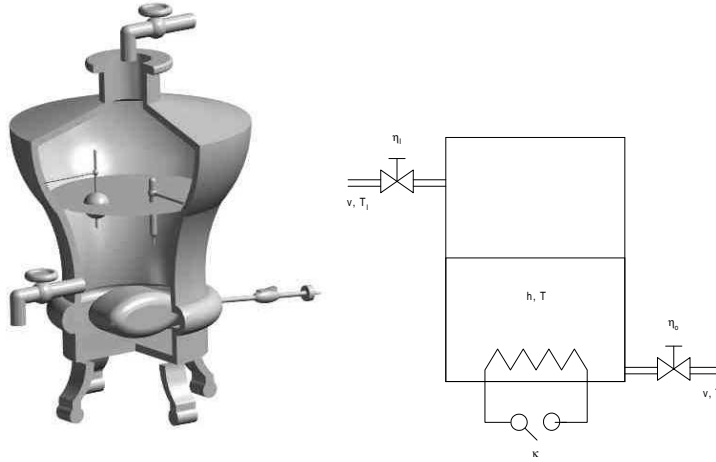
A kávéfőző gép grafikus rajzát és folyamatábráját a 4.3. ábrán láthatjuk.

Állapotter modell. A kávéfőző gép állapotter modelljét a 3.1.8. alfejezetben megismert módon a tartályban lévő vízre felírt tömeg- és energiamérlegekből származtathatjuk és az alábbi alakban írhatjuk:

$$\begin{aligned} \frac{dh}{dt} &= \frac{v}{A} \eta_I - \frac{v}{A} \eta_O && \text{(tömeg)} \\ \frac{dT}{dt} &= \frac{v}{Ah} (T_I - T) \eta_I + \frac{H}{c_p \rho h} \kappa && \text{(energia)} \end{aligned} \quad (4.7)$$

A modellben szereplő változók és paraméterek az alábbiak.

t idő [s]



4.3. ábra. A kávéfőző gép és a folyamatábrája

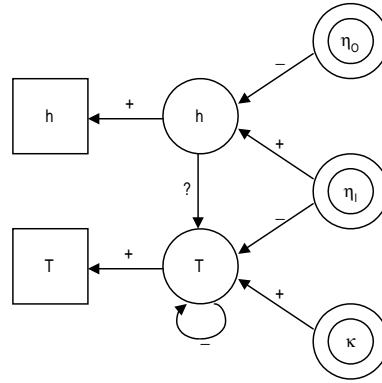
| | |
|----------|----------------------------------|
| h | tartálysztint [m] |
| v | térfogatsebesség [m^3/s] |
| c_p | fajhő [$Joule/kgK$] |
| ρ | sűrűség [kg/m^3] |
| T | hőmérséklet [K] |
| T_I | befolyó hőmérséklet [K] |
| H | fűtőteljesítmény [$Joule/sec$] |
| A | keresztmetszet [m^2] |
| η_I | bináris bemeneti szelep [1/0] |
| η_O | bináris kimeneti szelep [1/0] |
| κ | bináris kapcsoló [1/0] |

SDG modell. A fenti nemlineáris állapotter modell állapotváltozói a h vízszint és a T hőmérséklet, bemenetei a be- és kimeneti szelepek kapcsolóinak η_I és η_O állásai és a fűtőkapcsoló κ állása, kimenetei pedig az állapotváltozók ($y = x$).

Ezekkel a változókkal a kávéfőző gép SDG modellje a 4.4. ábrán látható alakot ölti.

4.2. SDG modellek és tulajdonságaik

Az előző alfejezetben láthattuk, hogyan rajzolhatók fel a szerkezeti gráfok (SDG modellek) az állapotter modell ismeretében vagy közvetlenül mérnöki intuíció alapján. Ebben az alfejezetben adottnak tekintjük a diagnosztizálandó dinamikus rendszer SDG modelljét, és azt vizsgáljuk, hogy ennek milyen tulajdonságai használhatóak fel diagnosztikai célra.



4.4. ábra. A kávéfőző gép SDG modellje

4.2.1. Utak és körök a struktúra gráfban

Egy dinamikus rendszer struktúra gráfja (SDG modellje) alkalmas a rendszer bizonyos dinamikus tulajdonságainak kvalitatív (minőségi) meghatározására: ez használható fel diagnosztikai célokra. Ezek a dinamikus tulajdonságok a gráfban található

$$P = (v_1, v_2, \dots, v_n), \quad v_i \in V, \quad e_{i,i+1} = (v_i, v_{i+1}) \in \mathcal{E}$$

irányított utakkal kapcsolatosak. Egy SDG modellbeli P irányított út

- a v_1 változó közvetett (indirekt) hatásának felel meg a v_n változóra,
- az út *értéke* pedig a

$$W(P) = \prod_{i=1}^{n-1} w(e_{i,i+1})$$

összefüggéssel számítható.

Fontos megjegyezni, hogy a *legrövidebb ut(ak)* és az *irányított körök* bírnak jelentőséggel a dinamikus tulajdonságok szempontjából. A stabilitás az irányított körök és körcsaládok értékével, az egységugrás válaszfüggvény jellemzői pedig az irányított utak értékével vannak kapcsolatban.

4.2.2. Az SDG modellek diagnosztikai célú értelmezése

Mint azt már eddig is láttuk, a szerkezeti gráfok olyan előjelekkel súlyozott irányított gráfok, azaz SDG modellek, amelyek a változók közötti hatásokat írják le. A 4.1.2. alfejezetben példát is láthattunk arra, hogy mérnöki intuíció alapján is felállítható SDG modell.

Most azt a kérdést vizsgáljuk meg közelebbről, hogy milyen következtetést lehet levonni a változók előjelekkel kifejezett kvalitatív értékére, pontosabban értékének megváltozására egy adott SDG modell segítségével.

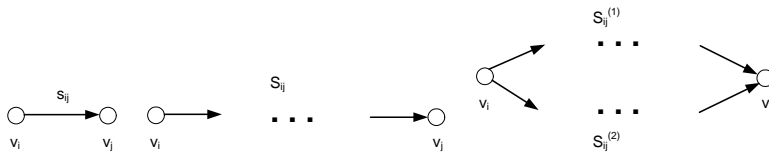
Jelöljük egy adott v_i változó eltérését az állandósult állapotbeli értékétől $\Delta v_i = v_i - \bar{v}_i$ -vel. Ekkor az eltérés $[\Delta v_i]_S$ előjel-értéke

$$[\Delta v_i]_S = [v_i - \bar{v}_i]_S$$

Ha a v_i változó hatással van a v_j változóra, azaz van olyan irányított út az SDG modellben, amely v_i -ből v_j -be megy, akkor ezen irányított út vagy utak (mert több is lehet!) értékétől függően kiszámítható a $[\Delta v_i]_S$ előjel-érték ismeretében a $[\Delta v_j]_S$ előjel-értéke az alábbiak szerint.

- Ha az összekötő út egy s_{ij} súlyú él, akkor $[\Delta v_j]_S = s_{ij} \otimes_S [\Delta v_i]_S$.
- Egy S_{ij} értékű összekötő út esetén $[\Delta v_j]_S = S_{ij} \otimes_S [\Delta v_i]_S$.
- Alternatív összekötő utaknál $[\Delta v_j]_S = (S_{ij}^{(1)} \oplus_S S_{ij}^{(2)}) \otimes_S [\Delta v_i]_S$, ami az előjel-összeadás tulajdonságai miatt *ellenkező előjelű közvetett hatásoknál határozatlan (?) előjel értéket eredményez.*

A fenti lehetőségeket a 4.5. ábra szemlélteti.



4.5. ábra. Élek, utak és hatások

4.2.3. Az egységugrás válaszfüggvény jellemzése

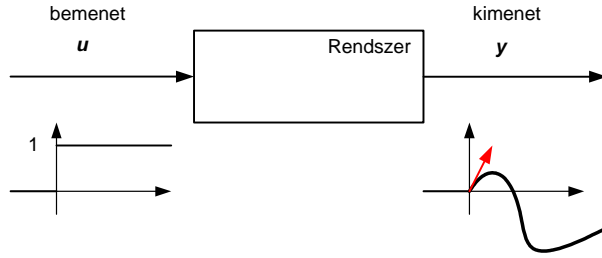
Az egységugrás válaszfüggvény a lineáris időinvariáns egy bemenetű-egy kimenetű (SISO) rendszerek jellemzésének egyik klasszikus módja, amikor a rendszer bemenetére egyensúlyi állapotban azonosan nulla kimenet mellett egy egységugrás függvényt (az egység-impulzus integrál-függvényét) adva regisztráljuk a rendszer kimenetén megjelenő nem feltétlenül monoton függvényt (ezt szemlélteti a 4.6. ábra).

Stabil rendszer esetén az egységugrás válaszfüggvény egy, a statikus erősítés által meghatározott állandó értékhez tart, ezt nevezzük *állandósult állapotbeli eltérésnek*. Az egységugrás válaszfüggvénynek az egységugrás időpillanatában vett meredekségét pedig a *kezdeti eltérés előjelének* hívjuk.

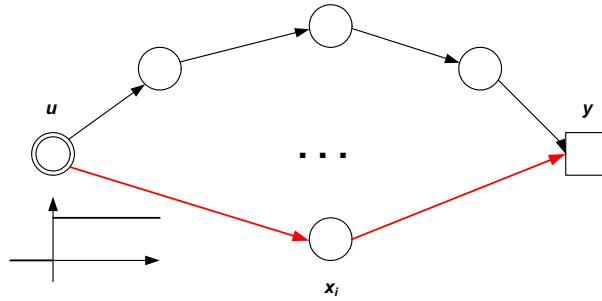
A kezdeti eltérés előjele

Az egységugrás válaszfüggvény kezdeti eltérésének előjelét a *legrövidebb út vagy legrövidebb utak előjel-értéke* segítségével határozhatjuk meg.

Az u bemenet és az y kimenet közötti legrövidebb út az SDG modellben az az irányított út u -ból y -ba, amelynek élszáma minimális. A legrövidebb út - ha nincs közvetlen hatás a bemenetektől a kimenetekre, azaz $D = 0$ - legalább egy állapotváltozón



4.6. ábra. Az egységugrás válaszfüggvény



4.7. ábra. A legrövidebb út fogalma

keresztül kell menjen (lásd 4.7 ábra), de elképzelhető, hogy több, azonos hosszúságú legrövidebb út is van.

Tegyük fel, hogy csak egy állapotváltozóhoz (x_i) tartozó csúcs van a legrövidebb úton, és ez az út egyértelmű (azaz nincs másik kettő hosszú út a gráfban). Ekkor egységugrás bemenet esetén a bemenet állandósult állapotbeli 0 értékétől való eltérése $[\Delta u]_S = +$, így ebben a pillanatban a bemenet által közvetlenül befolyásolt x_i állapotváltozó deriváltjának előjele (minden más változó állandósult állapotát feltételezve) az alábbi alakban írható:

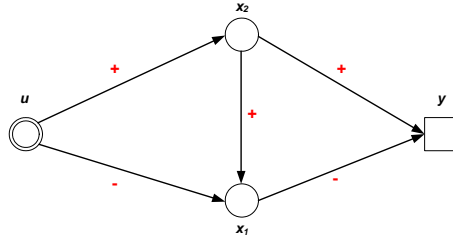
$$\left[\frac{dx_i}{dt}\right]_S = \delta x_i = s_{u,x_i} \otimes_S [\Delta u]_S = s_{u,x_i}$$

ahol s_{u,x_i} az összekötő él előjel-értéke az SDG modellben. Ezután a kimenet *kezdeti eltérése*nek előjele a

$$\left[\frac{dy}{dt}\right]_S = \delta y = S_{u,y}^* \otimes_S [\Delta u]_S = s_{u,x_i} \otimes_S s_{x_i,y}$$

összefüggéssel számítható, ahol $S_{u,y}^*$ a legrövidebb út előjel-értéke.

Fontos megjegyezni, hogy *több legrövidebb út esetén ezek értékének előjel-összegét kell képezni, ami az előjel-összeadás algebrai tulajdonságai miatt határozatlan (?) előjel-értéket is eredményezhet.*



4.8. ábra. Egy egyszerű SDG modell

Állandósult állapotbeli eltérés

Egy dinamikus LTI állapottér modellel adott rendszer *állandósult állapotát* ($\frac{dx}{dt} = 0$) az alábbi algebrai egyenletrendszer megoldásával kaphatjuk meg:

$$\begin{aligned} 0 &= A\bar{x} + B\bar{u} \\ \bar{y} &= C\bar{x} + D\bar{u} \end{aligned}$$

Egységugrás bemenet alkalmazásakor a rendszert bemenetének egy új állandósult $[\Delta u]_S = +$ értékével egy új állandósult állapotba visszük. Ebben az új állandósult állapotban a *jelek eltéréseinek előjel-értékeire* az alábbi egyenletrendszer írható fel:

$$\begin{aligned} 0 &= [A]_S \otimes_S [\Delta x]_S + [B]_S \otimes_S [\Delta u]_S \\ [\Delta y]_S &= [C]_S \otimes_S [\Delta x]_S + [D]_S \otimes_S [\Delta u]_S \end{aligned}$$

Matematikai szempontból a fenti egyenletrendszer egy lineáris előjel-egyenletrendszer, amely adott $[\Delta u]_S = +$ érték mellett megoldható az ismeretlen $[\Delta x]_S$ -re és $[\Delta y]_S$ -ra.

A megoldást kétféle módon állíthatjuk elő:

1. Algebrai úton, amikor a szokásos lineáris egyenletrendszer megoldás lépéseit előjel műveletekkel hajtjuk végre, vagy
2. algoritmikusan, az összes alternatív utak (közvetett hatások) megkeresésével és előjel-összegzésével.

Egyszerű példa

A fentieket egy, a 4.8. ábrán látható egyszerű példa segítségével illusztráljuk. Valamennyi közvetlen hatás nagyságát 1-nek véve, az alábbi egyszerű LTI állapottér modell tartozik a 4.8. ábrához:

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 - u \\ \frac{dx_2}{dt} &= u \\ y &= -x_1 + x_2 \end{aligned}$$

Így az algebrai módszerrel az alábbi eredményt kapjuk a változók állandósult állapotbeli eltéréseire a $[\Delta u]_S = +$ értéket feltételezve

$$\begin{aligned} [\Delta x_2]_S &= [\Delta u]_S = +, & [\Delta x_1]_S &= -[\Delta u]_S \oplus_S [\Delta x_2]_S = ?, \\ [\Delta y]_S &= [\Delta x_2]_S - [\Delta x_1]_S = ? \end{aligned}$$

Az algoritmikus módszerrel ugyanez az eredmény adódik:

$$[\Delta y]_S = (S_{u,x_1,y}^{(1)} \oplus_S S_{u,x_2,y}^{(2)} \oplus_S S_{u,x_2,x_1,y}^{(3)}) \otimes_S [\Delta u]_S = ?$$

4.3. Predikción alapuló diagnosztika SDG modellekkel

Ez a diagnosztikai módszer azon alapul, hogy az egységugrás válaszfüggvény jellemzőinek SDG modellből (különböző hibamódok mellett) megjósolt (predikált) értékét a valódi rendszeren megfigyelt értékkel hasonlítjuk össze.

4.3.1. Diagnosztika a legrövidebb utak előjele alapján

Ebben az esetben az egységugrás válaszfüggvény kezdeti meredekségére építjük a diagnosztikát, amelynek előjel-értékét az SDG modellbeli legrövidebb $P_{u,y}$ út vagy utak előjel-értékéből lehet meghatározni. A diagnosztikai módszer jellemzői az alábbiak.

- A módszer *predikción alapul*, hiszen a kezdeti eltérés előjelét az SDG modellből határozzuk meg, és ezt hasonlítjuk össze a valós megfigyelésből származó adattal.
- *Egyértelmű legrövidebb út esetén mindig határozott* értéke van a kezdeti eltérésnek az előjel-szorzás algebrai tulajdonságai miatt.
- Meghibásodás azonosításra is alkalmas a módszer, ha a hibamódok SDG modelljeit is megcsináljuk.

Meg kell jegyezni, hogy a hiba azonosítása nem nagy felbontó képességű, hiszen a kezdeti eltérés előjele csak + és - értékeket vehet fel, így SISO esetben csak két hibamód különíthető el.

Értelemszerűen több bemenetű – több kimenetű rendszerek esetében is használhatjuk a legrövidebb utak előjele alapján működő diagnosztikai módszert úgy, hogy egyenként adunk a rendszer bemeneteire egységugrás vizsgálójelet, a kimenetek válaszait pedig az adott bemenettel értelmezett SISO részrendszerek rész-SDG modelljei segítségével értékeljük.

4.3.2. Diagnosztika az állandósult állapotbeli eltéréssel

Az állandósult állapotbeli eltéréseken alapuló diagnosztikai módszert az alábbiakkal jellemezhetjük.

- Ez a módszer is *predikción alapuló*, itt az állandósult állapotbeli eltérések előjelét az SDG modellből határozzuk meg, és ezt vetjük össze a valós rendszeren tapasztalt állandósult állapotbeli eltérésekkel.

- Sajnos a jóslott eltéréseknek több alternatív hatásút esetén, azaz majdnem mindig **nem határozott** értéke van az előjel-összeadás algebrai tulajdonságai miatt, ezért
- sokkal rosszabbul használható, mint a kezdeti eltérés előjelén alapuló diagnosztika.

4.3.3. Példa: kávéfőző gép diagnosztikája

A kávéfőző gép 4.4. ábrán látható SDG modelljén mutatjuk be a fenti SDG modellen alapuló két diagnosztikai módszer alkalmazásához szükséges SDG modellen alapuló predikciót. Az ábráról megállapítható, hogy a rendszernek három bemenete (η_I , η_O és κ) és két kimenete (a T hőmérséklet és a h szint) van, de a κ fűtéskapcsoló bemenet nincs hatással a h kimenetre (nem köti őket össze irányított út). Így öt egységugrás válaszfüggvény értelmezhető ezen a rendszeren, amelyek az alábbiak:

- $\eta_I \mapsto h$, $\eta_O \mapsto h$
- $\eta_I \mapsto T$, $\eta_O \mapsto T$ és $\kappa \mapsto T$

Példaképpen tekintsük az $\eta_I \mapsto T$ egységugrás válaszfüggvényt, amelyből - a többi bemenet nulla állandósult állapotban tartása mellett - az η_I -re adott egységugrás hatása jóslható meg a T kimenetre.

- A kezdeti eltérés előjele a $S_{\eta_I, T, T}^{(1)}$ egyértelmű legrövidebb út mentén "-".
- Az állandósult állapotbeli eltérés értékét az $S_{\eta_I, T, T}^{(1)}$ legrövidebb, "-" értékű hatásút, és a (hurokélet is tartalmazó) $S_{\eta_I, T, T, T}^{(2)}$ "-" értékű hatásút előjel-összege, aminek értéke "?", azaz határozatlan.

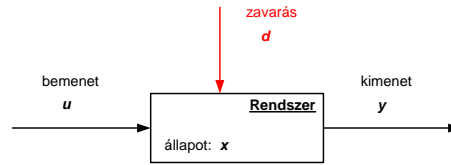
4.4. Direkt diagnosztika SDG modelleken

Ha a meghibásodás azonosításhoz olyan modellel rendelkezünk, amelyben a meghibásodás hiba-okait zavaró változókkal le tudjuk írni, akkor alkalmazhatjuk a direkt diagnosztikai módszert.

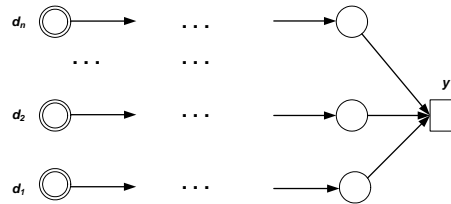
4.4.1. A módszer elve

A módszer alkalmazásához először kiterjesztjük a dinamikus rendszerek jel-folyam ábráját úgy, hogy a bemeneteknek két osztályát különböztetjük meg: a tényleges, okként szereplő és mérhető u bemeneteket, és a nem mérhető, de okként szereplő d zavarásokat (lásd 4.9. ábra). A hiba okokat azért lehet és célszerű zavarásokként modellezni, mert ezek - természetes módon - sajnos közvetlenül nem mérhetőek, de befolyásolják a rendszer viselkedését.

Ha a diagnosztizálandó rendszernek rendelkezünk az SDG modelljével, akkor ebben is meg kell különböztetnünk a zavarás jeleknek megfeleltetett csúcsokat az egyéb bemeneteket leíró csúcsoktól. Egy ilyen esetre látunk példát a 4.11. ábrán. *A zavarás jelek állandósult állapotának a hibamentes állapotot, azaz a nulla értéket feleltetik meg.*



4.9. ábra. A kibővített rendszermodell



4.10. ábra. Zavarás hatásutak

A diagnosztikai módszer

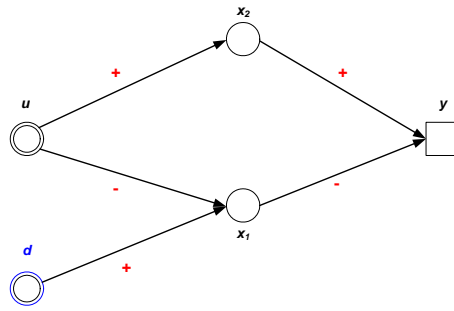
A módszer használatához feltételezzük, hogy rendelkezésünkre áll a fentiekben leírt, a hiba okokat zavarásokként leíró kiterjesztett SDG modell, valamint a megfigyelt kimenet állandósult állapotától való eltérésének $[\Delta y]_S$ előjele adott, általunk beállított bemenetekre. A diagnosztikai algoritmust az alábbi lépések definiálják.

- Az SDG modellben a megfigyelt kimenettől visszafelé bejárjuk a hatásutakat (az alternatív utak mindegyikét sorra véve) az érintett változókra visszafelé terjesztve azok állandósult állapotától való eltérésének előjel-értékét. A hatásutak az SDG modellek természetéből következően zavarás vagy bemenet változóban végződnek (lásd 4.10. ábra).
- Levágjuk az utat (eldobjuk a hibamód feltételezést), ha az érintett állapot változók megfigyelt vagy más kimenetből származtatott eltérés-értéke nem egyezik, illetve ha a bemenetekre nem a diagnosztikai kísérlethez választott értéket számoltuk ki.
- Azok a nem levágott hatásutak (lehet több ilyen is), amelyek $[\Delta d_i]_S \neq 0$ értéket eredményeznek, adják a diagnosztikai eredményt, azaz a ténylegesen felléphető hiba okok listáját (azokét, amelyek nincsenek ellentmondásban a diagnosztikai kísérlet eredményével).

4.4.2. Egyszerű példa

A 4.11. ábrán látható egyszerű, zavarással kiegészített SDG modell segítségével szemlél-tetjük a diagnosztikai algoritmus működését.

A rendszernek csak egy u bemenete és egy y kimenete van, amelyet a d hiba ok zavarás befolyásol. Tételezzük fel, hogy a bemenetet az állandósult nulla étéken tartjuk, azaz $[\Delta u]_S = 0$. Ekkor a kimenet állandósult állapotától való eltérésének előjel-értéke



4.11. ábra. Zavarással kiegészített SDG modell

($[\Delta y]_S$) az $S_{d,x_1,y}$ zavarás hatásút előjel-értékét felhasználva a

$$[\Delta y]_S = S_{d,x_1,y} \otimes_S [\Delta d]_S = -[\Delta d]_S$$

Ha tehát a $[\Delta u]_S = 0$ bemenet mellett a $[\Delta y]_S$ érték 0, akkor hibamentes állapotban van a rendszer, egyébként fellépett a d zavarással leírt hiba.

5. fejezet

Diagnosztika szabályok alkalmazásával

A diagnosztizálendő rendszernek szabályokkal történő leírása a legegyszerűbb, és egyben a legnagyobb hagyományokkal rendelkező formája a modell alapú, diszkrét módszereket használó diagnosztikának. Ebben a fejezetben a szabályok és szabályrendszerek, valamint a szabály-alapú következtetési módszerek rövid áttekintése után először a dinamikus rendszerek szabályokkal történő leírásával, majd a szabályokon alapuló, prediktív és közvetlen diagnosztikai eljárásokkal foglalkozunk.

5.1. Szabályrendszerek, következtetési módszerek

A diagnosztikai célra használt modellek legegyszerűbb, logikai értékészletű változók segítségével történő leírása szabályrendszerek segítségével történhet.

A szabályok a mesterséges intelligencián alapuló módszerek legelterjedtebb tudásreprezentációs eszközei, amelyekkel számos tankönyv foglalkozik részletesen (lásd például [8]). Ezért ebben a szekcióban csak a diagnosztikai módszerek megértéséhez és alkalmazásához szükséges legfontosabb ismereteket foglaljuk röviden össze.

5.1.1. Predikátumok és szabályok

A szabályok szintaxisának alap építőkövei a predikátumok.

Predikátumok

A szabályokban előforduló "atomi", azaz tovább nem bontható logikai (**true**, **false**) értékű változókat predikátumoknak nevezzük.

A diagnosztikai célra használt szabályokban a leggyakrabban az úgynevezett aritmetikai időfüggő predikátumok fordulnak elő. Ezek szintaxisa a

$$\langle jel \rangle \langle relaciojel \rangle \langle ertekek \rangle$$

hármassal írható le, ahol a $\langle jel \rangle$ a diagnosztizálandó rendszer egyik jele (időfüggő változója). Egy ilyen predikátum jellemzője, hogy

- mért vagy elvben mérhető jelekre támaszkodik, pl. $T > 90^\circ C$,
- a mért jel típusától függően lehet bemeneti, zavarás, állapot és kimeneti predikátum

Példák Diszkrét és valós értékészletű jele támaszkodó, valamint szoftver hibajelet tartalmazó predikátumokra egyszerű példák az alábbiak.

$$p_1 = (\kappa = \mathbf{on}) ; p_2 = (T < 300) ; p_3 = (h = \mathbf{low})$$

$$p_4 = (error = \text{"tank over flow"})$$

Szabályok szintaxisa

A szabályok szintaxisa az implikáció (\rightarrow) logikai művelet fogalmára épül. Szöveges formájában egy szabály a

HA *feltétel* **AKKOR** *kovetkezmény*
vagy
AKKOR *kovetkezmény* **HA** *feltétel*

alakú állítás, ahol a *feltétel* és a *kovetkezmény* egy-egy logikai kifejezést jelöl. Egy szabály felfogható egy

$$feltétel \rightarrow kovetkezmény$$

alakú logikai kifejezésként is, ahol \rightarrow a logikai implikáció műveleti jele.

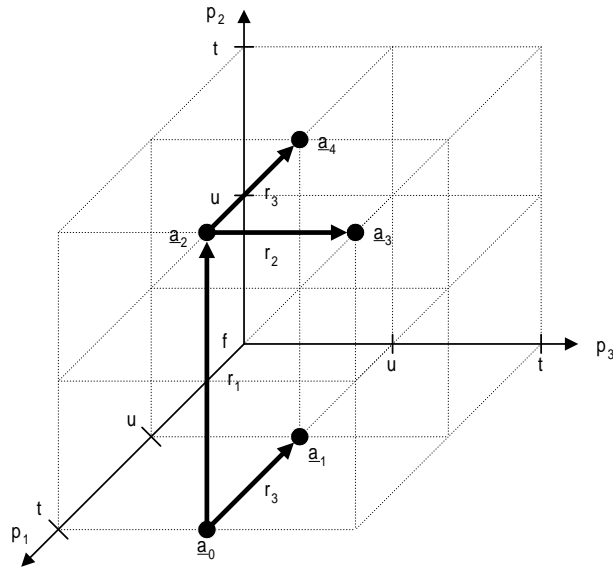
5.1.2. Szabályrendszerek

Szabályrendszernek nevezzük közös predikátum halmazra épülő szabályok összességét. A közös predikátum halmaz alkotja az úgynevezett *tény-bázist*. A *szabályok pedig relációkat határoznak meg a predikátumok között*. Ily módon egy szabályalapú tudásbázis időben változó tény-részét (jelek) alkotják a predikátumok, az időben állandó összefüggéseket (modellek) pedig a szabályok.

Datalog szabályrendszerek

A datalog szabályrendszerek a szabályrendszerek kanonikus (szabványos alakú) leírására és a szabályrendszerek tulajdonságainak analizésére szolgálnak. A datalog szabályrendszerek

- szabályainak formája: $(p_1 \wedge \dots \wedge p_k) \mapsto q$.
- függőségi gráfja egy súlyozott irányított gráf, amelynek csúcsai a predikátumoknak, irányított élei pedig a szabályoknak felelnek meg. A $(p_1 \wedge \dots \wedge p_k) \mapsto q$ alakú szabályhoz k darab $p_i \rightarrow q$ irányított él tartozik, amelyek súlya a szabály azonosítója.
- gyökér predikátumai a függőségi gráf kiinduló (azaz bemenő él nélküli) csúcsai, amelyek input vagy zavarás típusú predikátumoknak feleltethetők meg.



5.1. ábra. A következtetési folyamat az állapottérben

5.1.3. Adatvezérelt és célvezérelt következtetés

Egy szabályrendszer tény-bázisának állapottere a predikátumok értékeiből képezett $\mathbf{p} = [p_1, \dots, p_n]^T$ állapotvektorral járható be. A következtetéssel, azaz a szabályok által leírt összefüggések használatával módosítjuk a tény-bázis állapotát, az ismert predikátumok segítségével meghatározunk új predikátum értékeket.

Az 5.1. ábra illusztrálja azt, hogy hogyan mozgunk az állapottérben a következtetés folyamata alatt. Egy adott állapottérbeli pontból egy másikba egy szabály alkalmazásával jutunk, a két pontot összekötő irányított élre a szabály azonosítóját írhatjuk.

A szabályok használata, azaz a következtetés kétféle módon valósítható meg.

1. Az ún. *előrefelé haladó következtetés* az ismert tények (predikátumok) következményeit határozza meg.
2. Az ún. *visszafelé haladó következtetéssel* pedig a tények okait deríthetjük fel.

Fontos megjegyezni, hogy *diagnosztikai* célokra a visszafelé haladó következtetés mutatkozik célravezetőnek.

Adatvezérelt (előrefelé haladó) következtetés

Az adatvezérelt következtetési módszert az alábbiak jellemzik.

- A következtetés célja egy célállapot elérése vagy megkonstruálása a kezdőállapotból.
- Az új következtetések előállítása a modus ponens alkalmazásával történik.
- A következtetés a terminálási feltétel eléréséig vagy az összes következmény előállításáig (nincs több alkalmazható szabály) folytatódik.

Célvezérelt (visszafelé haladó) következtetés

A célvezérelt következtetési módszer az alábbiakkal jellemezhető:

- A következtetési cél egy feltételezett célállapot érvényességének igazolása kezdetben érvényes tényekre támaszkodva.
- A modus ponens alkalmazásával állítunk elő új részcélokat.
- A következtetés az összes rész cél igazolásáig folyik vagy addig, amíg nincs több igazolható rész cél (nincs több alkalmazható szabály).

5.1.4. Példa: kávéfőző gép szabályrendszere

Egy adott rendszer működését diagnosztikai célra leíró szabályokat heurisztikusan könnyen létrehozhatunk. Ez esetben a diagnosztizálandó *hiba-okokat zavarásokkal írjuk le*, a hiba-ok felléptének egy úgynevezett *indikátorváltozót* (amelynek értéke hibamentes esetben 0, az 1-es érték jelzi a hiba felléptét) feleltethetünk meg.

Az alábbiakban két különböző hiba-okot feltételezve adunk példát a 3.1.8. alfejezetben bemutatott kávéfőző gép szabályokkal történő leírására.

1. hiba-ok: alul lyukas a tartály

A leíró szabályokat a fizikai alapú modell *"tömegmérlegéből származtatjuk"* egyszerű mindennapi logikával.

Predikátumok:

- bemeneti és zavarás: $p_{Isz} = (\eta_I = 1)$, $p_{Osz} = (\eta_O = 1)$, $p_{hly} = (\chi_I = 1)$
- kimeneti: $p_{hno} = (\Delta h > 0)$, $p_{hall} = (\Delta h = 0)$, $p_{hcs0} = (\Delta h < 0)$

Szabályok:

$$\begin{aligned}
 & \mathbf{HA} (p_{Isz} \wedge \neg p_{Osz} \wedge \neg p_{hly}) \mathbf{AKKOR} p_{hno} & (5.1) \\
 & \mathbf{HA} (\neg p_{Isz} \wedge p_{Osz} \wedge \neg p_{hly}) \mathbf{AKKOR} p_{hcs0} \\
 & \mathbf{HA} (\neg p_{Isz} \wedge \neg p_{Osz} \wedge p_{hly}) \mathbf{AKKOR} p_{hcs0} \\
 & \dots
 \end{aligned}$$

2. hiba-ok: ereszt a beömlő szelep

Ez a hiba-ok a fenténél összetettebben írható le: *ereszt a beömlő szelep, azaz akkor is folyik be a hideg víz, ha el van zárva η_I , azaz $\eta_I = 0$.*

Ez esetben az *"energiamérlegéből származó"* szabályok néhány példáját mutatjuk be.

Predikátumok:

- bemeneti és zavarás: $p_{Isz} = (\eta_I = 1)$, $p_f = (\kappa = 1)$, $p_{Ier} = (\chi_I = 1)$
- kimeneti: $p_{Tno} = (\Delta T > 0)$, $p_{Tall} = (\Delta T = 0)$, $p_{Tcs0} = (\Delta T < 0)$

Szabályok:

$$\begin{aligned}
 & \mathbf{HA} (p_{Isz} \wedge \neg p_f \wedge \neg p_{Ier}) \mathbf{AKKOR} p_{Tcso} \\
 & \mathbf{HA} (\neg p_{Isz} \wedge p_f \wedge \neg p_{Ier}) \mathbf{AKKOR} p_{Tno} \\
 & \mathbf{HA} (\neg p_{Isz} \wedge \neg p_f \wedge p_{Ier}) \mathbf{AKKOR} p_{Tcso} \\
 & \dots
 \end{aligned} \tag{5.2}$$

5.2. Diagnosztika adatvezérelt következtetéssel

Az adatvezérelt következtetéssel ismert tények (predikátum-értékek) következményeit állítjuk elő következtetéssel. Ha a szabályrendszert heurisztikus modellnek tekintjük, akkor a bemenet és zavarás predikátumok értékéből az ezektől függő állapot és kimenet predikátumok értéke számítható ki a következtetés segítségével. Ily módon tehát predikció alapú diagnosztikára nyílik lehetőség.

5.2.1. Predikción alapuló diagnosztika szabályokkal

Feladatkitűzés

A szabályokkal végzett predikción alapuló diagnosztikai feladat elvi kitűzése az alábbi.

Adott

- a diagnosztizálandó rendszert leíró időfüggő szabályrendszer,
- a mért értékekre támaszkodó input és output predikátumok.

Detektálandó a rendszer hibás állapota, illetve meghatározandó, hogy mely hibamód(ok) lép(tek)ett fel.

A diagnosztikai eljárás

A diagnosztikai eljárás két fő logikai lépésből áll.

1. A mért inputokra támaszkodó predikátumok értékeiből adatvezérelt (előrefelé haladó) következtetéssel kiszámítjuk az outputokra támaszkodó predikátumok értékét (minden hibamódban, ha van modell).
2. Összehasonlítjuk a mért értékekre támaszkodó és a szabályokkal számított output predikátumok értékeit: egyezés esetén az adott hibamódo(oka)t fogadjuk el érvényesnek.

5.2.2. Kávéfőző gép predikción alapuló diagnosztikája

A predikción alapuló diagnosztikai eljárás működését a 3.1.8. alfejezetben leírt kávéfőző gép példáján, az 5.1.4. alfejezetben ismertetett szabályokat felhasználva mutatjuk be.

Legyen a feladat az **1. hiba-ok: alul lyukas a tartály** szabályokkal végzett predikción alapuló diagnosztikája a **”tömegmérlegből származó”** szabályokkal.

A feladat megoldásához az alábbi elemeket használjuk fel.

- *Predikátumok:*

- bemeneti és zavarás: $p_{Isz} = (\eta_I = 1)$, $p_{Osz} = (\eta_O = 1)$, $p_{hly} = (\chi_l = 1)$
- állapot és kimeneti: $p_{hcso} = (\Delta h < 0)$, $p_{hall} = (\Delta h = 0)$, $p_{hkicsi} = (h < 1cm)$,
 $p_{hnormal} = (h \geq 15cm)$

- *Szabályok:*

$$HA (p_{Isz} \wedge \neg p_{Osz} \wedge \neg p_{hly}) AKKOR \neg p_{hcso} \quad (5.3)$$

$$HA (\neg p_{Isz} \wedge \neg p_{Osz} \wedge \neg p_{hly}) AKKOR p_{hall} \quad (5.4)$$

$$HA (\neg p_{Isz} \wedge \neg p_{Osz} \wedge p_{hly}) AKKOR p_{hcso} \quad (5.5)$$

$$HA (p_{hnormal} \wedge p_{hcso}) AKKOR p_{hkicsi} \quad (5.6)$$

$$HA (p_{hkicsi} \wedge p_{hall}) AKKOR p_{hkicsi} \quad (5.7)$$

$$HA (p_{hnormal} \wedge p_{hall}) AKKOR p_{hnormal} \quad (5.8)$$

- *Mért adatok:* $D_{\eta_I, \eta_O, h}(1, 2) = \{(0, 0, 17cm), (0, 0, 0.5cm)\}$

Először a mért adatokból a bemenetekre és a kimenetre támaszkodó predikátumokat határozzuk meg a két diszkrét időpillanatban:

$$\mathcal{D}_{\eta_I, \eta_O, h}(1, 2) = \{(\neg p_{Isz}, \neg p_{Osz}, p_{hnormal}), (\neg p_{Isz}, \neg p_{Osz}, p_{hkicsi})\} \quad (5.9)$$

A kimeneti predikátumok szabályokkal becsült értékei

- Hibamentes esetben ($\neg p_{hly}$) az első két időlépésben a szintre, mint kimenetre az alábbi becslés adódik $\{p_{hnormal}, p_{hnormal}\}$ az (5.4) és az (5.8) szabályok alkalmazásával.
- Hibás esetben (p_{hly}) a becslés $\{p_{hnormal}, p_{hkicsi}\}$ lesz az (5.3) és az (5.6) szabályok alkalmazásával, ami megegyezik a mért adatokból képezhető predikátum értékekkel, így ezt a hibamódot fogadjuk el diagnosztikai eredménynek.

5.3. Diagnosztika célvezérelt következtetéssel

Célvezérelt következtetéssel predikátumok okait határozhatjuk meg, ami közvetlenül, azaz direkt módon alkalmazható diagnosztikai célokra.

5.3.1. Direkt diagnosztika szabályokkal

A szabályok segítségével végzett közvetlen diagnosztika *elvi feladatkitűzése* az alábbi módon adható meg formálisan.

Adott

- a diagnosztizálandó rendszert leíró időfüggő szabályrendszer,
- a mért értékekre támaszkodó input és output predikátumok.

Detektálandó a rendszer hibás állapota, illetve meghatározandó, hogy mely hibamód(ok) lép(tek)ett fel.

A diagnosztikai eljárás

A szabályokra támaszkodó direkt diagnosztikai eljárás során az alábbi lépéseket hajtjuk végre.

1. A mért outputokra támaszkodó predikátumok értékeiből célvezérelt (visszafelé haladó) következtetést végzünk, és megállapítjuk a zavarásoknak megfelelő predikátumok értékét (ha van modell) úgy, hogy az a mért inputokra támaszkodó predikátumokkal konzisztens legyen.
2. Ha találunk ilyen esete(ke)t, akkor az e(zek)nek megfelelő hibamódo(ka)t fogadjuk el érvényesnek.

5.3.2. Kávéfőző gép direkt diagnosztikája

A direkt diagnosztikai eljárás működését újfent a 3.1.8. alfejezetben leírt kávéfőző gép példáján mutatjuk be.

Feladatunk most is az **1. hiba-ok: alul lyukas a tartály** szabály alapú diagnosztikája a **”tömegmérlegből származó”** szabályokkal, de most direkt diagnosztikai módszerrel.

Hasonlóan az 5.2.2. alpontban ismertetett predikción alapuló diagnosztikai módszerhez, az alábbi elemeket használjuk fel a direkt diagnosztikához.

- *Predikátumok:*

- bemeneti és zavarás: $p_{I_{sz}} = (\eta_I = 1)$, $p_{O_{sz}} = (\eta_O = 1)$,
 $p_{hly} = (\chi_l = 1)$

- állapot és kimeneti: $p_{hcso} = (\Delta h < 0)$, $p_{hall} = (\Delta h = 0)$, $p_{hkicsi} = (h < 1cm)$,
 $p_{hnormal} = (h \geq 15cm)$

- *Szabályok:*

$$HA (\neg p_{I_{sz}} \wedge p_{O_{sz}} \wedge \neg p_{hly}) AKKOR p_{hcso} \quad (5.10)$$

$$HA (\neg p_{I_{sz}} \wedge \neg p_{O_{sz}} \wedge \neg p_{hly}) AKKOR p_{hall} \quad (5.11)$$

$$HA (\neg p_{I_{sz}} \wedge \neg p_{O_{sz}} \wedge p_{hly}) AKKOR p_{hcso} \quad (5.12)$$

$$HA (p_{hnormal} \wedge p_{hcso}) AKKOR p_{hkicsi} \quad (5.13)$$

$$HA (p_{hkicsi} \wedge p_{hall}) AKKOR p_{hkicsi} \quad (5.14)$$

$$HA (p_{hnormal} \wedge p_{hall}) AKKOR p_{hnormal} \quad (5.15)$$

- *Mért adatok:* $D_{\eta_I, \eta_O, h}(1, 2) = \{(0, 0, 17cm), (0, 0, 0.5cm)\}$

Először ebben az esetben is a mért adatokból a bemenetekre és a kimenetre támaszkodó predikátumokat határozzuk meg a két diszkrét időpillanatban, amelyek azonosak az (5.9) egyenletbeli $\mathcal{D}_{\eta_I, \eta_O, h}(1, 2)$ kvalitatív mérési rekorddal.

Ezután a *diagnosztikai következtetés* menete az alábbi.

- (i) A hibamentes eset ($\neg p_{hly}$) feltételezésével az (5.13) szabályon visszafelé következtetve a p_{hcso} predikátum adódik, amely semmilyen adott szabály baloldalával sem egyeztethető össze, ha a bemenetre támaszkodó $\neg p_{I_{sz}}$, $\neg p_{O_{sz}}$ predikátumokat is figyelembe vesszük, ezt a feltételezést tehát el kell vetnünk.

- (ii) A hibás esetet (p_{hly}) feltételezve és az (5.13) szabályon visszafelé következtetve, itt is a p_{hcs_o} predikátum adódik. A következő lépésben az (5.12) szabályt használhatjuk, amelynek feltételi része a bemenetre támaszkodó $\neg p_{I_{sz}}$, $\neg p_{O_{sz}}$ predikátumokat is figyelembe véve illeszkedik a p_{hly} hipotézisünkre, így a lyukas hibamódot fogadhatjuk el.

6. fejezet

Diagnosztika HAZOP és FMEA felhasználásával

Ebben a fejezetben - a szabályalapú diagnosztikai módszerekre épülő - olyan diagnosztikai módszerekkel foglalkozunk, ahol a diagnosztikához szükséges tudást a kockázat menedzsment eljárások során kapott eredményekből nyerjük ki.

6.1. Kockázat menedzsment

Az úgynevezett kockázat menedzsment (risk assesment) módszerek eszköztára az 1990-es években alakult ki és vált általánosan elfogadottá a biztonság-kritikus (safety critical) iparágakban, például a járműgyártásban, valamint a vegyi- és atomenergia-iparban. Ez a terület azóta is intenzíven és látványosan fejlődik, hiszen a termékek és eljárások biztonságosságát a legjobban megelőzéssel, a biztonsági kockázatok feltárásával, elemzésével és az ellenük történő megelőző védekezéssel lehet növelni. Mára már minden biztonság-kritikus termékre és eljárásra, illetve ipari üzemre el kell végezni, és rendszeresen felül kell vizsgálni a biztonsági kockázatok feltárását és elemzését, így ezek naprakészen rendelkezésre is állnak.

A kockázat menedzsment keretében a lehetséges meghibásodások (okok) és veszélyek (következmények)

- szisztematikus felmérése,
- valószínűségi és veszélyességi értékelése,
- ok-okozati kapcsolatok feltárása,
- a lehetséges megelőző és rehabilitációs lehetőségek feltárása

történik meg szisztematikus és ellenőrzött formában.

HAZID elemzések. A kockázat menedzsmenten belüli fontos részterület a veszélyelemzés, angol nevén hazard identification (rövidítve HAZID). A veszélyelemzést

- adott "szabványos" módszertan alapján
- multidiszciplináris szakértői csoport (team) végzi.
- Az elemzés eredménye verbális, táblázatba rendezett, ami
- hatósági engedélyezés alapja, és ílymódon rendszeresen felülvizsgálandó.

A veszélyelemzési módszertanok közül az úgynevezett *HAZOP* és *FMEA* módszertanok a legelterjedtebbek, a következőkben ezeket vesszük kissé részletesebben is szemügyre.

6.1.1. Veszély és működőképességi analízis – HAZOP

A veszély és működőképességi analízis - angol nevén "hazard and operability analysis", rövidítve HAZOP - elsősorban a vegyi és rokon iparágakban terjedt el.

Jellemzői az alábbiak:

- Az elemzés eredményét egy táblázatba foglaljuk.
- A *mérhető* jellemző változók alapján haladunk,
- ezek eltéréseit (**Deviation**) vizsgáljuk (ez a "kulcs" oszlop).
- Változó típusonként *szabványos eltérés-típusokat* (**Guideword**) határoz meg a módszertani útmutató.
- Az eltérések lehetséges okait (**Causes**) és veszélyes következményeit (**Consequences**) (valamint a megelőzésre alkalmas akciókat) gyűjtjük össze.

A HAZOP táblázat formátuma. Diagnosztikai célra a fent említett oszlopok közül csak a **Deviation**, **Causes** és **Consequences** oszlopok tartalmát használjuk az alábbi formátumú táblázatba rendezve:

| Guideword | Deviation | Causes | Consequences |
|-----------|-----------|--------|--------------|
|-----------|-----------|--------|--------------|

A táblázat "Guideword" felíratú oszlopa az éppen vizsgált változóra vonatkozó egyik szabványos eltérés típust tartalmazza, a **Deviation** oszlopban pedig "<valtozo><guideword>" szintaxissal szerepel a vizsgált eltérés.

Példa: Kávéfőző gép HAZOP analízis

Az alábbiakban a 3.1.8. fejezetben bemutatott kávéfőző gép mérhető *h* szintjének példáján mutatjuk be egy lehetséges HAZOP táblázat részletét.

Rendszer: (folyamatos működésű) kávéfőző gép

Változó: szint *h*

| Guideword | Deviation | Causes | Consequences |
|-----------|-------------------|-----------------|---|
| alacsony | <i>h</i> alacsony | beáramlás nincs | kiáramlás alacsony hőmérséklet magas |
| | <i>h</i> alacsony | kiáramlás magas | hőmérséklet magas <i>h</i> nincs |

Fontos megjegyezni, hogy - noha az eredeti módszertan ezt nem követeli meg - az eltérések okai (**Causes**) és következményei (**Consequences**) is kifejezhetőek az eltérések leírására alkalmas ”<valtozo><guideword>” szintaxissal.

6.1.2. Meghibásodás és hatás analízis – FMEA

A meghibásodás és hatás analízis, angolul ”Fault Mode and Effect Analysis” (rövidítve FMEA), elsősorban a jármű és energetikai iparokban használatos.

Jellemzői több ponton hasonlóak a már megismert HAZOP jellemzőihez, de lényeges eltérések is vannak. Az FMEA elemzés során

- a rendszer komponensei alapján haladunk.
- Ezek lehetséges meghibásodási módjait (**Failure**) vizsgáljuk (ez az elsődleges ”kulcs” oszlop).
- Az elemzés eredményét egy táblázatba foglaljuk.
- Komponens típusonként szabványos meghibásodási mód-típusokat tartalmaz a módszertan.
- A meghibásodások lehetséges okait (Failure mode causes), veszélyes lokális következményeit (**Local effects**) és rendszerszintű következményeit (**System effects**) gyűjtjük össze.

Az FMEA táblázat formátuma. Diagnosztikai célokra az FMEA elemzés **Failure mode**, **Local effects** és **System effects** oszlopai használhatóak, amelyeket a következő formátumú táblázatban foglalhatunk össze.

| Component | Failure mode | Failure mode causes | Local effects | System effects |
|-----------|--------------|---------------------|---------------|----------------|
|-----------|--------------|---------------------|---------------|----------------|

Példa: Kávéfőző gép FMEA analízis

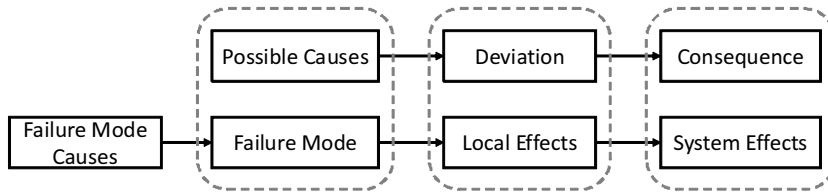
A HAZOP analízishez hasonlóan most is a 3.1.8. alfejezetben bemutatott kávéfőző gép mérhető *h* szintjének példáját használjuk egy lehetséges FMEA táblázat részletének bemutatásához.

Rendszer: (folyamatos működésű) kávéfőző gép

Komponens: beáramlási szelep η_I

| Component | Failure mode | Failure mode causes | Local effects | System effects |
|----------------------------|--------------|---------------------|-----------------|----------------|
| beáramlási szelep η_I | beragadt | törés | beáramlás nincs | szint alacsony |
| η_I | kiakadt | törés | beáramlás magas | szint magas |

Megfigyelhető, hogy - noha az eredeti módszertan ezt nem követeli meg - a meghibásodások lokális és rendszerszintű következményei (**Local effects** és **System effects**) is kifejezhetőek a ”<valtozo><guideword>” szintaxissal.



6.1. ábra. Megfeleltetés a HAZOP és FMEA adatok között

6.2. Szabály alapú diagnosztika HAZOP és FMEA információkból

Miután a kockázat- és veszélyelemzési információk biztonsági okok miatt a legtöbb biztonság-kritikus rendszerről rendelkezésre állnak, kézenfekvő az, hogy ezeket megpróbáljuk diagnosztikai célokra felhasználni. Ehhez először megvizsgáljuk, hogyan lehet a HAZOP és FMEA információkat egy keretrendszerben felhasználni. Ezután megmutatjuk, hogyan lehet a HAZOP és FMEA táblázatokból szabályokat előállítani.

6.2.1. Megfeleltetés a HAZOP és FMEA táblák elemei között

A HAZOP és FMEA táblázatok adatainak *együttes* használatához először megvizsgáljuk ezen táblázatban szereplő adatelemek *közös szintaxisát*. Az előzőekben megismert példák is azt sugallják, hogy ezt célszerű *adat-kettesek* formájában megadni, ahol

<**Deviation**> = (<Measured variable> <Guide word>)

<**Failure mode**> = (<Component id> <Failure type>)

<**Cause**> = (<Variable> <Guide word>)

vagy (<Component id> <Failure type>)

<**Consequence**> = (<Variable> <Guide word>)

<**Local effect**> = (<Variable> <Guide word>)

<**System effect**> = (<Variable> <Guide word>)

A kávéfőző gépre felírt HAZOP és FMEA táblázatokban az alábbi példák szerepeltek:

<szint> <alacsony>

<bearamlasi szelep> <beragadt>

Ha a fenti szintaxist is felhasználva szeretnénk megfeleltetést találni a HAZOP és FMEA táblázatok oszlopai között, akkor abból indulhatunk ki, hogy egy **Deviation** sohasem feleltethető meg egy **Failure mode**-nak, hiszen előbbi kötelezően mért változóra támaszkodik, de ennek egy oka (**Cause**) az már lehet egy komponens meghibásodása (**Failure mode**). Így a 6.1. ábrán szemléltetett megfeleltetést hozhatjuk létre a HAZOP és FMEA táblázatok oszlopai között.

Predikátumok és az elemek szintaxisa. A fenti megfeleltetésen kívül a táblázat elemek adat-kettesek által leírt szintaxisa lehetőséget teremt arra is, hogy a HAZOP és

FMEA táblázatok soraiból formális módszerekkel szabályokat alkossunk. Ehhez először észrevehetjük, hogy az adat-kettesek predikátumokká alakíthatóak:

- <Deviation>: mért változókra támaszkodó aritmetikai predikátummá alakítható,
- <Consequence>, <Local effect> etc.: változókra támaszkodó aritmetikai predikátumnak felel meg,
- <Failure mode>: speciális predikátum.

A kávéfőző gépre felírt HAZOP és FMEA táblázatokban szereplő adat-kettesek közül kettőnek adjuk meg a predikátum alakját példaképpen:

$$\begin{aligned} \langle \text{szint} \rangle \langle \text{alacsony} \rangle &\mapsto p_{hkicsi} = (h < h_{alacsonylimit}) \\ \langle \text{bearamlasi szelep} \rangle \langle \text{beragadt} \rangle &\mapsto p_{beragadt} = (\chi_{beragadt} = 1) \end{aligned}$$

6.2.2. Szabályok a HAZOP és FMEA táblákból

A fentiekben már láttuk, hogy a HAZOP és FMEA táblázatokban szereplő elemek predikátumokká alakíthatóak, a táblázatok soraiban pedig ok-okozati összefüggéseket írunk le. A HAZOP táblázatokban egy eltérés okait és következményeit, az FMEA táblázatokban pedig egy komponens meghibásodási mód lokális és rendszerszintű következményeit tüntetjük fel.

Ez lehetőséget ad arra, hogy a HAZOP és FMEA táblázatok egy-egy sorából *data log szabályokat* készítsünk. Ennek érdekében formális szintaxissal írjuk le a szabályok feltétel (*Precondition*) és következmény (*Conclusion*) részeinek, valamint a HAZOP és FMEA táblázatok oszlopaiban szereplő predikátumoknak a kapcsolatát:

$$\begin{aligned} \text{Precondition} = & \mathbf{Failure\ mode} \mid \mathbf{Cause} \mid \mathbf{Deviation} \\ & \mid \text{Precondition} \wedge \text{Precondition} \end{aligned}$$

$$\text{Conclusion} = \mathbf{Consequence} \mid \mathbf{Deviation} \mid \mathbf{Cause}$$

A logikai ok-okozati összefüggések származtatása alapján alapvető különbség van a HAZOP és FMEA táblázatok között az alábbiak szerint.

- *HAZOP*: a mérhető eltérések (**Deviation**) okait (**Cause**) keressük az analízis során, ez egy *visszafelé haladó következtetésnek* felel meg;
- *FMEA*: a rejtett meghibásodások (**Failure mode**) következményeit (**Local and System effects**) keressük, ami logikailag egy *előrefelé haladó következtetést* jelent.

6.2.3. Példa: Kávéfőző gép – HAZID szabályok

Az alábbiakban a 3.1.8. alfejezetben bemutatott kávéfőző gép egy folyamatos működésű változatának HAZID szabályait, azaz a HAZOP és FMEA táblázatokból kinyerhető szabályokra mutatunk példákat.

Folyamatos működésű kávéfőző gép

Az eredeti 3.1.8. alfejezetben bemutatott változathoz képest a folyamatos működésű kávéfőző gép az alábbi jellemzőkkel rendelkezik.

- folyamatos, 0 és 1 közötti értékészletű η_I beömlő és η_O kiömlő, valamint κ fűtés szelep,
- folytonos $v_I = \eta_I v$ beáramlás, $v_O = \eta_O v$ kiáramlás és $f = \kappa H$ fűtés.

A mérnöki modell egyenletei az alábbiak:

$$\begin{aligned} \frac{dh}{dt} &= \frac{v}{A}\eta_I - \frac{v}{A}\eta_O && \text{(tomeg)} \\ \frac{dT}{dt} &= \frac{v}{Ah}(T_I - T)\eta_I + \frac{H}{c_p \rho h}\kappa && \text{(energia)} \end{aligned} \quad (6.1)$$

ahol h a mérhető vízszint, T a mérhető hőmérséklet, A a tartály konstans keresztmetszete, c_p a fajhő, ρ a sűrűség és T_I a beáramlási hőmérséklet.

HAZOP szabályok

Tekintsük a h vízszintre, mint mérhető változóra vonatkozó HAZOP táblázat alábbi sorát:

| Guideword | Deviation | Causes | Consequences |
|-----------|--------------|-----------------|--------------------|
| alacsony | h alacsony | beáramlás nincs | kiáramlás alacsony |

Ebből az alábbi szabályok képezhetőek, ha a v_I és v_O változók mérhetőek:

- HA** ($v_I = 0$) **AKKOR** ($h < h_{min}$)
HA ($h < h_{min}$) **AKKOR** ($v_O < v_{min}$)

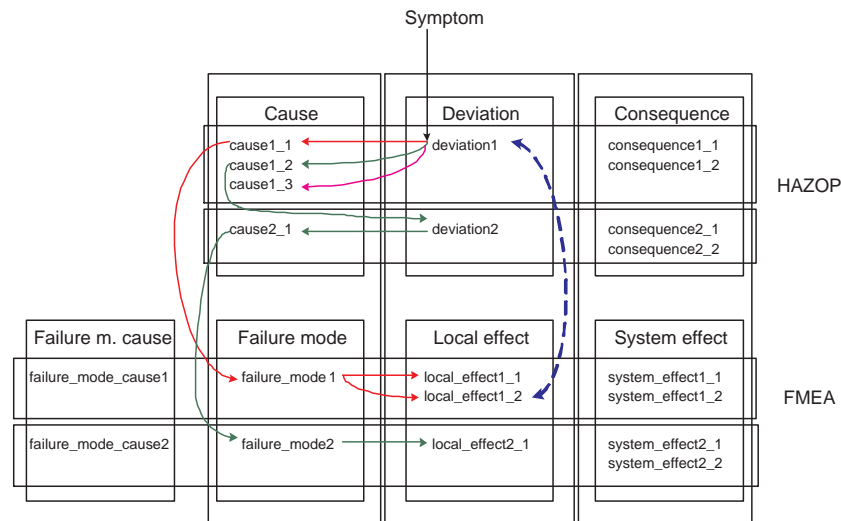
FMEA szabályok

Ebben az esetben a beáramlási szelep beragadásának következtében előállt következményeket leíró FMEA sort tekintjük példának.

| Component | Failure mode | Failure mode causes | Local effects | System effects |
|-------------------|--------------|---------------------|---|----------------|
| beáramlási szelep | beragadt | törés | beáramlás nincs ($v_I = \eta_I v = 0$) | szint alacsony |

Ebből a sorból az alábbi szabályt kaphatjuk:

- HA** ($\eta_I = 0$) **AKKOR** ($v_I = 0$)



6.2. ábra. A következtetési folyamat

6.2.4. A diagnosztikai következtetés

Miután a HAZOP és FMEA táblázatok soraiból a fenti módon szabályok képezhetőek, a diagnosztika egyik lehetséges módja az, hogy az összes lehetséges szabályból egy szabályrendszert képezzünk, és ezekkel - az 5. fejezetben megismert módszerek egyikével - szabály alapú diagnosztikai következtetést végzünk.

Ennél lényegesen hatékonyabban használhatjuk azonban a HAZOP és FMEA táblázatokban rejlő diagnosztikai információt, ha a diagnosztikai következtetést közvetlenül a HAZOP- FMEA táblázatokon, mint adatstruktúrákon végezzük el. Az ilyen közvetlen diagnosztika elvi feladatkitűzése az alábbi alakban fogalmazható meg.

Adott:

- a mért értékekből előállított *Deviation*-ok,
- az egyesített HAZOP-FMEA táblázatok.

Keressük: azokat a "gyökér-okokat" (*Failure mode*), amelyek összhangban vannak a *Deviation*-okkal (és az egyéb mért értékekkel).

A fenti feladatra *kétfázisú megoldási módszer* adható:

1. *visszafelé haladó következtetéssel* előállítjuk a lehetséges okokat,
2. *előrefelé haladó következtetéssel* szűkítjük ezek halmazát.

A 6.2. ábra egy példán mutatja be a kétfázisú diagnosztikai következtetés menetét egy hipotetikus, HAZOP és FMEA táblázatot is tartalmazó példán. A mért eltéréstől kiindulva először a HAZOP táblázatban esetleg egymás után csatolt visszafelé haladó következtetési lépésekkel addig keresünk, míg egy rejtett komponens meghibásodási okra

akadunk. Ekkor áttérünk az FMEA táblázat megfelelő sorára, és összevetjük ennek következményeit előrefelé haladó következtetéssel a mért értékekhez tartozó eltérésekkel és egyéb predikátumokkal.

7. fejezet

Színes Petri háló modelleken alapuló diagnosztika

A diszkrét modelleken alapuló diagnosztikai módszerek közül a színes Petri hálókat alkalmazóak rendelkeznek a legnagyobb diagnosztikai felbontással. Használatuk mégsem terjedt el széles körben, ami az itt alkalmazható módszerek kevésbé ismert voltával magyarázható elsősorban.

A színes Petri hálók az úgynevezett diszkrét eseményű rendszerek leírására alkalmas, könnyen olvasható, felhasználóbarát modellek, ezért a fejezet első részében a diszkrét eseményű rendszereket tárgyaljuk röviden.

7.1. Diszkrét eseményű rendszerek modelljei

Nevükből is látható módon a diszkrét eseményű rendszerek olyan rendszerek, amelyek működését események sorozatával jellemezhetjük.

7.1.1. Diszkrét eseményű rendszerek fogalma

Egy adott rendszer-típust a hozzá tartozó jelek tulajdonságai segítségével jellemezhetünk. Ennek alapján a diszkrét eseményű rendszerek jeleinek az alábbi jellemző tulajdonságai adhatók meg.

- a jelek (bemenet, kimenet, állapot) *értékkészlete diszkrét*:
 $x(t) \in \mathbf{X} = \{x_0, x_1, \dots, x_n\}$,
- az *idő diszkrét*: $T = \{t_0, t_1, \dots, t_n\} = \{0, 1, \dots, n\}$, de nem feltétlenül ekvidisztáns.

Eseménynek nevezzük egy diszkrét jelérték-változás bekövetkezését, amelyet a bekövetkezési idő - új jelérték párral adunk meg. Így például egy x diszkrét jelhez tartozó $e_x(t_i, x_j)$ eseménynél a jel a t_i időpillanatban az x_j diszkrét értéket veszi fel.

A diszkrét eseményű rendszerek leírásánál az *események sorrendje* számít, a rendszermodellnek képesnek kell lennie soros és párhuzamos események leírására. A diszkrét

eseményű rendszermodellek legfontosabb *alkalmazási területei* az ütemezés, operátori eljárások leírása, valamint az erőforrás-kezelési feladatok megoldása.

7.1.2. Diszkrét eseményű rendszerek állapotter modelljei

A diszkrét eseményű rendszerek állapotter modellje a diszkrét idejű lineáris állapotter modell általánosításaként írható fel az alábbi alakban

$$\begin{aligned} x(k+1) &= \Psi(x(k), u(k)) && (\text{állapot egyenlet}) \\ y(k) &= h(x(k), u(k)) && (\text{kimenet egyenlet}) \end{aligned} \quad (7.1)$$

adott $x(0)$ kezdeti feltétellel és nemlineáris Ψ állapot-, valamint h kimeneti függvényekkel. A fenti rendszermodellben szereplő jelek diszkrét idejűek, azaz az idő a $\mathbb{T} = \{t_0, t_1, \dots, t_n, \dots\}$ diszkrét halmazból veszi értékeit, és a t_i időpont helyett csak annak i indexét tüntetjük fel az egyenletekben.

Diszkrét eseményű rendszereket a diszkrét idejű rendszerosztályon belül az alábbiakkal jellemezhetjük:

1. nem feltétlenül ekvidisztáns mintavételezésűek,
2. a jelek értékészlete diszkrét,
3. a diszkrét jel-érték változásokat *esemény*-nek tekintik.

7.2. Színes Petri hálók és tulajdonságaik

A színes Petri hálók a diszkrét eseményű rendszerek egyik népszerű, szemléletes és gyakran használt formájának tekinthetjük. Történetileg a színes Petri hálók a közönséges Petri hálók általánosításaként kerültek definiálásra, ezért először a közönséges Petri hálókkal ismerkedünk meg.

7.2.1. Petri hálók

A Petri hálókat Adam Petri események előfeltételeinek és következményeinek formális leírására, és az események rendszere tulajdonságainak (például holtpon-mentességének) vizsgálatára dolgozta ki.

Petri háló modell – absztrakt leírás

Egy Petri háló szerkezetét a $\mathbf{PN} = (P, T, I, O)$ négyes írja le, ahol

- *Helyek (feltételek)* halmaza: P ,
- *Átmenetek (események)* halmaza: T ,
- *Bemeneti (előfeltétel) függvény:* $I : T \rightarrow P^\infty$,
- *Kimeneti (következmény) függvény:* $O : T \rightarrow P^\infty$.

A fenti leírás statikus, hiszen csak a rendszerben lehetséges események előfeltételeit, következményeit és egymáshoz történő kapcsolódását írja le, konkrét eseménysorozatok leírásához a háló működésének definiálására is szükség van (lásd alább).

A formális leírás mellett a Petri háló szerkezetét grafikusan is ábrázolhatjuk egy *páros irányított gráffal*, amelyben a

- *csúcsok*: a helyeknek (P) és átmeneteknek (T) felelnek meg (ezek partíciókat képeznek a csúcshalmazban $P \cap T = \emptyset$),
- *élek*: a bemeneti és kimeneti függvény (I, O) által meghatározottak (egy p_i feltételi csúcsból vezet él a t_j esemény csúcsba, ha az adott feltétel előfeltétele az eseménynek, valamint egy t_j esemény csúcsból vezet él a p_i feltételi csúcsba, ha az adott feltétel következménye az eseménynek).

Petri hálók dinamikája. A Petri hálóval modellezett rendszer dinamikus viselkedésének leírásához először bevezetjük az úgynevezett *jelölőpontokat* (vagy *token*-eket) a háló helyeire, ezeket fekete körlapokkal jelezzük az érintett helyeken. Ha egy p_i helyen van jelölőpont, akkor a megfelelő feltétel "igaz" értékű.

A μ *jelölőfüggvény* írja le a jelölőpontok elhelyezkedését a Petri háló helyein

$$\begin{aligned} \mu : \mathbf{P} &\rightarrow \mathcal{N}, \quad \mu(p_i) = \mu_i \geq 0 \\ \underline{\mu}^T &= [\mu_1, \mu_2, \dots, \mu_n], \quad n = |\mathbf{P}| \end{aligned}$$

Egy működési időlépésben egy esemény játszódik le, azaz az eseménynek megfelelő átmenet *tüzel*. A t_j átmenet akkor "engedélyezett" ha az előfeltételei "igaz"-ak, azaz van *token* a bemeneti helyeken

$$\underline{\mu}^{(i)}[t_j > \underline{\mu}^{(i+1)}]$$

Egy adott időpillanatban több átmenet is lehet engedélyezett, ezek egyike végrehajtódik, azaz az átmenet "tüzel". Tüzelés után az átmenet következményei lesznek "igaz"-ak, tehát a tüzelés eltünteti a jelölőpontokat az előfeltételeknek megfelelő helyekről, és jelölőpontokat generál a következményeknek megfelelő helyekre.

Értelmezhetünk *tüzelési (működési) sorozatokat* több átmenet egymást követő tüzeléseivel, amelyet a következőképpen jelölünk:

$$\underline{\mu}^{(0)}[t_{j_0} > \underline{\mu}^{(1)}[t_{j_1} > \dots[t_{j_k} > \underline{\mu}^{(k+1)}] \quad (7.2)$$

Petri háló modellek állapottere. A fenti jelölőfüggvény segítségével kapcsolatot teremthetünk a Petri háló $\underline{\mu}$ jelölő vektora és a (7.1) egyenletbeli diszkrét idejű állapotter modell változói (jelei) között.

Az *állapotvektor* a $\underline{\mu}$ jelölő vektor azon elemeiből áll, amelyeknél a jelölés a Petri háló gráfjában *belső* helyeken (azaz olyan helyeken, amelyeknek a ki- és be-foka legalább 1) van, azaz

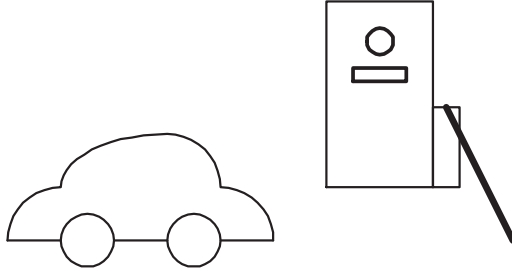
$$x(k) \sim \underline{\mu}_x^{(k)}$$

Hasonlóan a *bemenetek* a $\underline{\mu}$ jelölő vektor azon elemeihez tartoznak, ahol jelölés az *input* helyeken (azaz olyan helyeken, amelyeknek a be-foka nulla) van, tehát

$$u(k) \sim \underline{\mu}_u^{(k)}$$

Példa: parkoló-garázs kapu automata

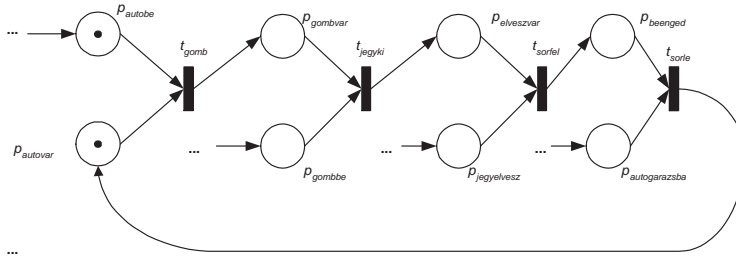
A mindennapi életből jól ismert diszkrét eseményű rendszer az úgynevezett parkoló-garázs kapu automata, amely a parkoló-garázs kapujának működését vezérli és kiadja a parkolójegyeket. Ennek egy sematikus rajza a 7.2. ábrán található a jegykiadó tálca, a jegykiadást kezdeményező gomb és a sorompó vázlatos ábrázolásával. Az automata



7.1. ábra. A parkoló-garázs kapu automata

működtetését a garázsba behajtani kívánó autó, illetve annak sofőrje végzi.

Az automata működésével kapcsolatos eseményeket, valamint ezek előfeltételeit és következményeit könnyen leírhatjuk Petri hálóval. A Petri háló modell grafikus leírása a 7.2. ábrán látható.



7.2. ábra. A parkoló-garázs kapu Petri háló modellje

Petri háló modell - formális leírás. A formális leírás elemei az alábbiak:

- Helyek (állapot; input)

$$P = \{p_{autovar}, p_{gombvar}, p_{elveszvar}, p_{beenged} ; p_{autobe}, p_{gombbe}, p_{jegyvesz}, p_{autogarazsba}\}$$

- Átmenetek:

$$T = \{t_{gomb}, t_{jegyki}, t_{sorfel}, t_{sorle}\}$$

- Bemeneti függvény:

$$I(t_{gomb}) = \{p_{autobe}, p_{autovar}\} \quad , \quad I(t_{jegyki}) = \{p_{gombbe}, p_{gombvar}\}$$

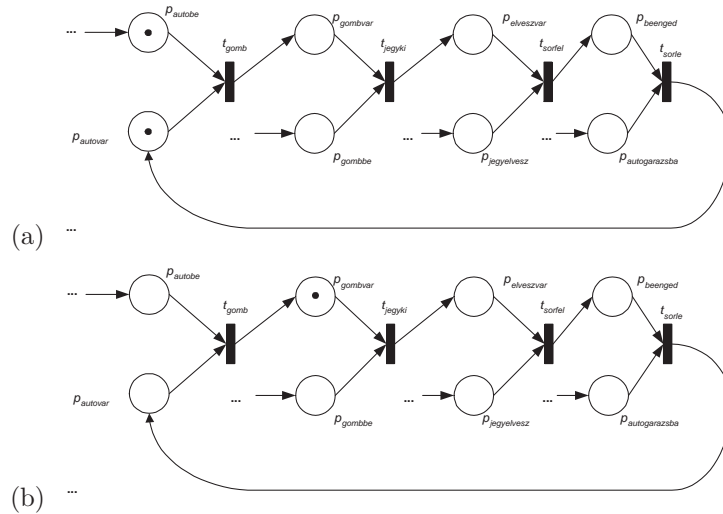
$$I(t_{sorfel}) = \{p_{jegyelvesz}, p_{elvezsvar}\} \quad , \quad I(t_{sorle}) = \{p_{beenged}, p_{autogarazsba}\}$$

- Kimeneti függvény:

$$O(t_{gomb}) = \{p_{gombvar}\} \quad , \quad O(t_{jegyki}) = \{p_{elvezsvar}\}$$

$$O(t_{sorfel}) = \{p_{beenged}\} \quad , \quad O(t_{sorle}) = \{p_{autovar}\}$$

Egy működési lépés. Tételezzük fel, hogy az automata alapállapotban van, azaz vár egy új autó megjelenésére, és az be is gurul a belépési érzékelő hurokba. Ezt a helyzetet írja le a 7.3. ábra (a) rész-ábrája. Ebben a helyzetben a t_{gomb} átmenet előfeltételei teljesülnek, azaz ez lesz engedélyezett. A t_{gomb} átmenet tüzelése utáni jelölést a 7.3. ábra (b) rész-ábrája mutatja.



7.3. ábra. A parkoló-garázs kapu Petri háló modelljének működése

A működési lépés formális leírása a $\underline{\mu}$ jelölővektor

$$\underline{\mu}^T = [\mu_{autovar}, \mu_{gombvar}, \mu_{elvezsvar}, \mu_{beenged} ;$$

$$\mu_{autobe}, \mu_{gombbe}, \mu_{jegyelvesz}, \mu_{autogarazsba}]$$

segítségével adható. A t_{gomb} átmenet működésének hatására

$$\underline{\mu}^{(1)}[t_{gomb} > \underline{\mu}^{(2)}$$

$$\underline{\mu}^{(1)} = [1, 0, 0, 0 ; 1, 0, 0, 0]^T$$

$$\underline{\mu}^{(2)} = [0, 1, 0, 0 ; 0, 0, 0, 0]^T$$

Petri háló modellek elérhetőségi gráfja

A Petri háló modellek megoldásával állíthatjuk elő egy adott kezdeti jelölésből (azaz kezdőállapotból) lehetséges tüzelési vagy működési sorozatokat (lásd (7.2) egyenlet), amelyek megadják a lehetséges lejátszódó esemény-sorozatokat.

A Petri háló modell *megoldásával* tehát jelölés- (vagy rendszerállapot) szekvenciákat állítunk elő, amelyeket egy súlyozott irányított gráf, az úgynevezett *elérhetőségi gráf* formájában adunk meg. Az elérhetőségi gráf elemei az alábbiak:

- a *csúcsok*: megfelelnek a Petri hálóban lehetséges jelöléseknek,
- az *irányított élek* akkor kötnek össze két jelölésnek megfelelő csúcsot, ha van átmenet, aminek tüzelése összeköti őket,
- az *élsúlyok* pedig az átmenet és az esetleges külső események (jelölések).

Az *elérhetőségi gráf előállítás*a algoritmikus módon, a Petri háló végrehajtásával (szimulációjával) történik az alábbi lépésekben:

1. *start*: az adott kezdeti jelölés,
2. *új csúcs hozzávétele*: az egyik engedélyezett átmenet tüzelésével (input hatása is!).

Fontos megjegyezni, hogy az előállítás lehet NP-nehéz konfliktushelyzet vagy nem véges működés esetén.

Az elérhetőségi gráf bejárásával végezhető el a Petri háló modell ún. viselkedési tulajdonságainak analízise, amelyről bővebben a [8] tankönyvben talál az Olvasó.

7.2.2. Magasabb szintű Petri hálók

A magasabb szintű Petri hálók az előző alpontban tárgyalt közönséges Petri hálók kiterjesztett változatai. A kiterjesztés célja a Petri háló leíró képességének növelése, azaz olyan elemek bevezetése, amelyekkel komolyabb időbeli vagy finomabb érték-készletű leírás érhető el.

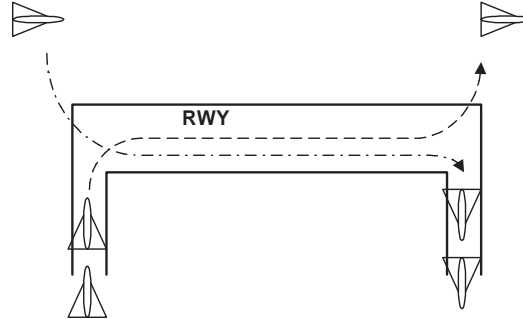
Időzített Petri háló modellek

Az időzített Petri háló modellekkel a dinamikus viselkedés finomabb időbeli leírása válik lehetségessé. A közönséges Petri hálóknál csak az események egymás utániságát tudjuk meghatározni, míg az időzített Petri hálókkal az események között eltelt idő leírása is megoldható.

Formálisan egy időzített Petri háló úgy származtatható egy közönséges Petri hálóból, hogy annak elemeit időzítő *feliratokkal* látjuk el. Ennek keretében értelmezzünk

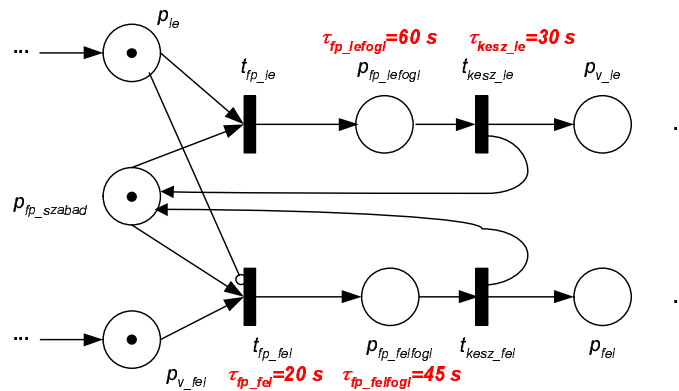
- egy *órát*, amely a háló mögött láthatatlanul beépített vagy egy speciális "forrás" hely formájában megvalósított elem,
- az *átmenetekhez tüzelési időt*,
- esetleg a *helyekhez várakozási időt*.

Futópálya időzített Petri háló modellje. A futópálya a repülőtereken használt speciális útpálya, amelyen a repülőgépek a fel- és leszállást végzik (lásd 7.4. ábra). Mivel



7.4. ábra. A futópálya

a fel- és leszálló gépek ugyanazt az útpályát, mint erőforrást kénytelenek használni, ún. *konfliktus-helyzet* alakul ki a rendszerben egyidejűleg jelenlévő fel- és leszálló gépek esetén. Ha a konfliktus-helyzetet fel szeretnénk oldani úgy, hogy mindig a leszálló gépeknek legyen előnye, akkor ezt a speciális él, az ún. *inhibitor él* segítségével tehetjük meg, ami a rendszer 7.5. ábráján a p_{le} hely és a t_{fp_fel} átmenet között látható. Ez az él tiltja a t_{fp_fel} átmenet engedélyezettségét, ha a p_{le} helyen van jelzőpont.



7.5. ábra. A futópálya időzített Petri háló modellje

A fel- és leszállási műveletek különböző időigényének leírására az átmenetekhez rendelt, és a 7.5. ábrán feliratokkal jelölt végrehajtási időket, a pálya foglaltsági időszükségletének leírására pedig a megfelelő helyhez rendelt várakozási időket vezettünk be.

Színes Petri háló modellek

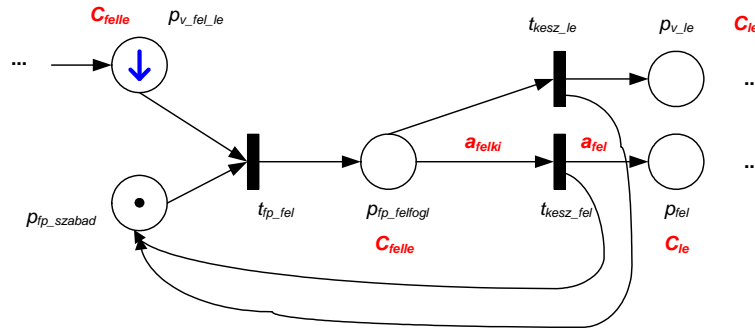
A színes Petri háló modellek a változók, azaz a helyekhez tartozó feltételek érték-készleteinek kiterjesztésével adnak lehetőséget egy diszkrét eseményű rendszer dinamikus viselkedésének finomabb leírására, és képesek fentiekben leírt időkezelés megvalósítására is.

A színes Petri háló modellek formális leírására itt is a közönséges Petri háló modellek elemeire alkalmazott *feliratokat* használjuk. A színes Petri háló modellek az alábbi jellemzőkkel rendelkeznek.

- A jelzőpontok (token-ek) diszkrét értékűek, a diszkrét érték-halmazokat (univerzumokat) *színhalmazoknak* nevezzük.
- Az egyes helyekhez megengedett színhalmaz rendelhető, ezt a hely melletti felirat jelzi.
- Az átmenetekhez és élekhez diszkrét függvények rendelhetőek, ezekkel írjuk le, hogy egy él milyen értékű (színű) jelzőpontok(at) érzékel, illetve közvetít a csatlakozó helyeken/helyekre, illetve, hogy egy átmenet milyen feltételek mellett tüzel. Ezeket az úgynevezett él- illetve átmenet függvényeket szintén felirat jelzi a hálón.

Futópálya színes Petri háló modellje. Példaképpen a 7.4. ábrán látható futópálya egy lehetséges színes Petri háló modelljét mutatjuk be, ahol a fel- és leszálló gépeket különböző színű jelzőpontok jelzik, azaz az ennek megfelelő színhalmaz $C_{felle} = \{\uparrow, \downarrow\}$.

Magát a színes Petri háló modellt a 7.6. ábrán láthatjuk. A színhalmazt megadó felirattal el nem látott helyeken logikai (Boolean) értékészlettel számolunk. A színezett



7.6. ábra. A futópálya színes Petri háló modellje

Peri háló modell ábrán jelölt feliratait valamennyien élfüggvények.

- a $p_{fp_felfogl}$ helytől a t_{kesz_fel} átmenethez vezető élfüggvény értéke akkor igaz (azaz akkor engedélyezett az átmenet), ha a helyen felszálló gépnek megfelelő token tartózkodik, formálisan:

$$a_{felki} : \text{if } val(p_{fp_felfogl}) = "\uparrow" \text{ then } "true"$$

- a $t_{kész-fel}$ átmenettől a p_{fel} helyre vezető élfüggvény az átmenet előfeltételi helyén lévő felszálló gép tokenet teszi a p_{fel} helyre, azaz

$$a_{fel} = val(p_{fp_defogl}), \quad val(p_{fel}) = a_{fel}$$

Színes Petri hálók megoldása és tulajdonságai

A közönséges Petri háló modellekhez hasonlóan a színes Petri háló modellek megoldását is algoritmikusan, a háló egy adott kezdeti jelöléséből kiindulva szimulációval határozhatjuk meg.

A megoldás itt is egy úgynevezett *elérhetőségi gráf* formájában adható meg, amelynek csúcspontjai a jelölések, azaz a diszkrét rendszerállapotok, irányított élei pedig az eseményeknek, azaz az egyes jelöléseket egymásba alakító lejátszódó állapotoknak felelnek meg. Az elérhetőségi gráf előállításában ebben az esetben is lehet NP-nehéz, ha konfliktus helyzet vagy nem véges működés szerepel a hálóban.

7.3. Színes Petri hálók és kvalitatív differencia-algebrai egyenletek

A kvalitatív differencia- és algebrai egyenletekből álló modellekkel már a 3.2.4. alfejezetben megismertedtünk, ahol ezen modellek származtatása mellett a megoldásukról is szó esett. Ebben az alfejezetben pedig megmutatjuk, hogy egy ilyen modell egy-egy értelmű módon megfeleltethető egy színes Petri háló modellnek.

A megfeleltetés lépései

Ha adott egy kvalitatív differencia- és algebrai egyenletekből álló modell, akkor az alábbi előkészítő lépések után készíthető el a neki megfeleltetett színes Petri háló modell.

- A szereplő *szín*halmazokat a változók kvalitatív értékkészletéből (univerzumok) állíthatjuk elő.
- A változókhoz rendeljük a színes Petri háló helyeit, *minden változóhoz egy helyet*.
- Az *átmeneteket a modell egyenleteihez rendeljük*, a statikus és dinamikus egyenletekhez egyaránt.
- A diagnosztikai célra készült modelleknél a meghibásodásokhoz is helyeket kell rendelni, ez célszerűen *a hiba-indikátor változókhoz tartozó hely* lehet.

7.3.1. Kvalitatív algebrai egyenletek és színes Petri hálók

Egy kvalitatív algebrai egyenletnek megfelelő átmenet a színes Petri hálóban előfeltételi helyekként az egyenlet jobb oldalán szereplő független változóknak megfeleltetett helyekhez kapcsolódik, míg következményként az egyenlet bal oldalán szereplő függő változó értékét eredményezi tüzeléskor. A kvalitatív algebrai egyenlet megoldás-tábláját az átmenethez rendelt átmeneti függvény formájában használhatjuk fel arra, hogy az átmenet a tüzeléskor valóban a megoldásnak megfelelő értéket eredményezze.

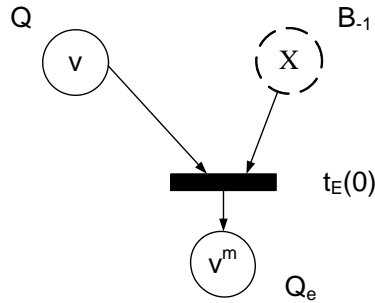
Statikus példa: érzékelő additív hibával

Tekintsük újra a 3.2.4. alfejezetben már megismert érzékelőt, amely a valódi v értéket egy additív E mérési hibával terhelve méri. Az érzékelő működését leíró *kvalitatív algebrai modell egyenlet* az alábbi alakú:

$$[v^m] = [v] + \chi \cdot [E]$$

ahol $[v] \in \mathcal{Q}$, $[v^m] \in \mathcal{Q}_e$, $\chi \in B_{-1} = \{-1, 0, 1\}$, $[E] = L$.

A fenti egyenletnek megfelelő *színes Petri háló modell* egy átmenetet tartalmaz, és a 7.7. ábrán látható. A színes Petri háló megfelelő működését úgy biztosíthatjuk, hogy a



7.7. ábra. Additív hibájú érzékelő színes Petri háló modellje

t_E átmenet átmeneti függvényét a kvalitatív algebrai modell egyenlet megoldás-táblájával tesszük azonosná, amelyet az alábbi táblázatban adtunk meg.

| $[v^m]$ | χ | $[v]$ | mód |
|-----------|--------|-------|--------|
| N | 0 | N | normál |
| H | 0 | H | normál |
| L | 0 | L | normál |
| 0 | 0 | 0 | normál |
| $e+$ | 1 | H | hibás |
| \bar{H} | 1 | N | hibás |
| N | 1 | L | hibás |
| L | 1 | 0 | hibás |
| N | -1 | H | hibás |
| L | -1 | N | hibás |
| 0 | -1 | L | hibás |
| $e-$ | -1 | 0 | hibás |

Fontos megjegyezni, hogy a kvalitatív algebrai egyenleteknek megfelelő átmenetekhez nem tartozik végrehajtási idő, ezek engedélyezetté válásuk után azonnal tüzelhetnek: ezt jelzi az átmenet azonosítója után zárójelben megadott 0.

7.3.2. Kvalitatív differencia egyenletek és színes Petri hálók

Hasonlóan az előző 7.3.1. alfejezetben tárgyalt kvalitatív algebrai egyenletet színes Petri hálóval leíró esethez, egy kvalitatív differencia egyenlet leírása is egy átmenetet igényel a színes Petri hálós modellben, amelynek átmeneti függvénye a kvalitatív differencia egyenlet megoldás-táblája.

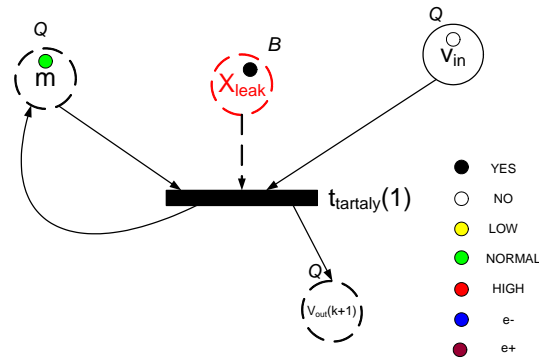
Egyszerű dinamikus példa: szabad kifolyású tartály. Dinamikus példaként egy egyszerű, szabad kifolyású tartályt választhatunk, amelynek mérnöki elvek alapján felállított dinamikus modelljét a 3.1.7. alfejezetben írtuk fel.

Az ott szereplő tömegmérlegből származtathatjuk az alábbi kvalitatív differencia egyenletet, ha egy kis lyukadást, mint lehetséges hiba-okot is figyelembe veszünk:

$$[m](k+1) = [m](k) + [v_{in}](k) - K \cdot [m](k) - \chi_{leak} \cdot [B]$$

ahol $[B] = L$ a lyukon kifolyó tömegáram, m a tartályban lévő folyadék tömege, v_{in} a betáplálási tömegáram, $K \cdot m$ a kifolyási tömegáram, és χ_{leak} a lyuk hiba-indikátor változója.

A fenti egyenletnek megfelelő *színes Petri háló modell* egy egységnyi működési idejű átmenetet tartalmaz, és a 7.8. ábrán látható. A színes Petri háló megfelelő működését úgy



7.8. ábra. Szabad kifolyású tartály színes Petri háló modellje

biztosíthatjuk, hogy a $t_{tartaly}$ átmenet átmeneti függvényét a kvalitatív modell egyenlet megoldás-táblájával tesszük azonosná.

7.3.3. Kvalitatív differencia-algebrai egyenletekből származó színes Petri háló modellek megoldása

Az előzőekben már láttuk, hogy mind a kvalitatív algebrai, mind a kvalitatív differencia egyenletek egyértelműen leírhatóak színes Petri háló formájában. Ez lehetővé teszi, hogy egy bonyolult kvalitatív differencia-algebrai egyenletekből álló modell megoldását úgy keressük meg, hogy összeállítjuk az egyenleteket leíró átmenetekből és a változóknak

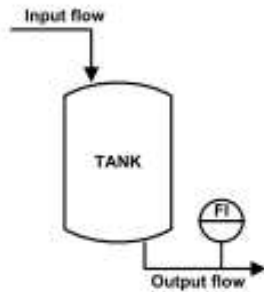
megfeleltetett helyekből álló színes Petri háló modellt, és annak megoldását a háló szimulációjával konstruáljuk meg.

Ha a megoldást predikció alapú diagnosztikai célra akarjuk felhasználni, akkor a szimulációval a hibamentes és a vizsgált hibaállapotokat tartalmazó színes Petri hálót is vizsgálni kell.

A 3.2.4. alfejezetben már megmutattuk, hogy egy kvalitatív differencia-algebrai egyenletekből álló modell megoldása a modell változóinak kvalitatív jel-nyomai formájában adható meg, így a színes Petri háló megoldásaként ilyen jel-nyomokat kell előállítani.

Példa: szabad kifolyású tartály additív hibával terhelt érzékelővel

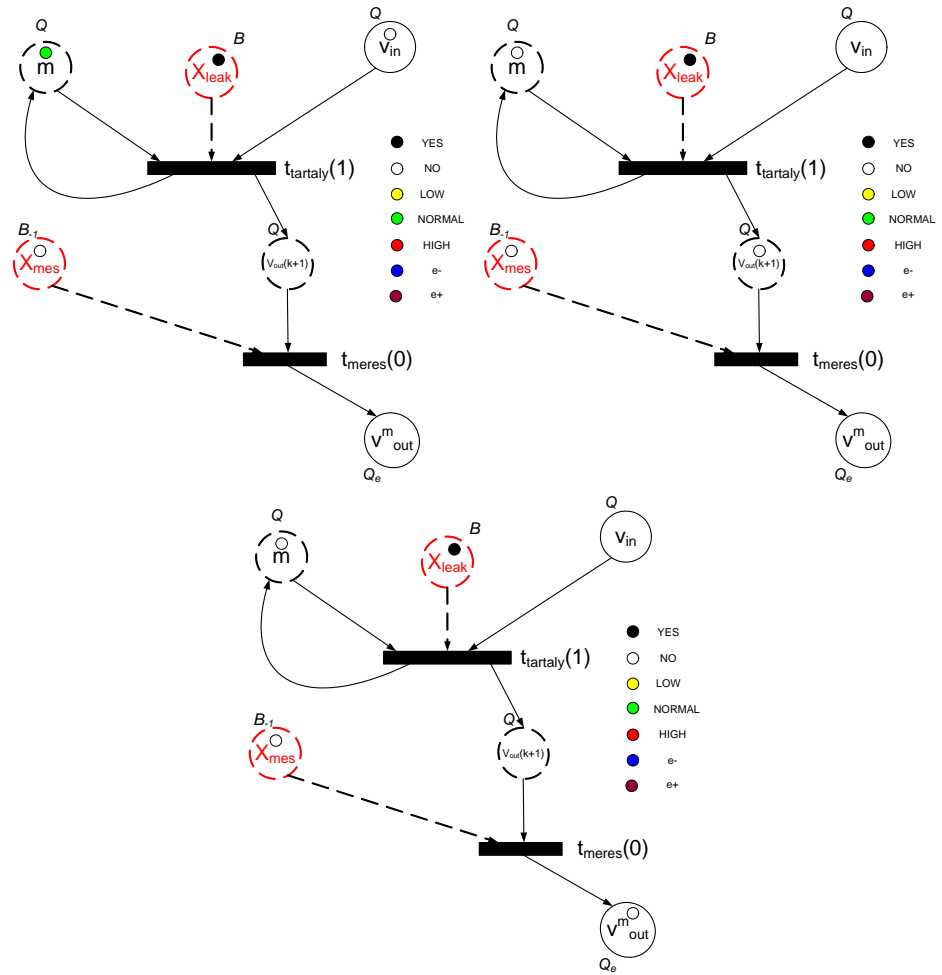
Példaként tekintsük a 7.9. ábrán látható összetett rendszert, amely az előbbieken már tárgyalt szabad kifolyású tartályból, és az ugyancsak megvizsgált additív hibával terhelt érzékelőből áll, amely a tartály v_O kiáramlási tömegsebességét méri.



7.9. ábra. Szabad kifolyású tartály érzékelővel

Az összetett rendszer *színes Petri háló modellje* a 7.10. ábrán látható, ahol az egyes rész-ábrák a háló egymást követő jelöléseit mutatják be. A színes Petri háló modell két átmenetből áll, a $t_{tartaly}$ átmenet a tartály viselkedését leíró kvalitatív differencia-, a t_E átmenet pedig az érzékelőnek megfelelő kvalitatív algebrai egyenlethez tartozik.

Az összetett rendszer egyenleteinek megoldását *kvalitatív jel-nyomokkal* a 7.11. ábrán látható táblázat foglalja össze azonosan konstans kvalitatív bemenet és időben állandó hiba indikátor változók esetére.



7.10. ábra. Szabad kifolyású érzékelős tartály színes Petri háló modelljének működése

| $[m]$ initial mass in tank | $[v_{in}]$ input flow sequence | $[Z_{leak}]^*$ tank leakage | $[Z_{meas}]^*$ sensor failure | $[v_T^m]$ measured flow sequence |
|----------------------------------|-----------------------------------|-----------------------------------|-------------------------------------|-------------------------------------|
| LOW | (NORMAL,NORMAL,NORMAL) | 0 | NEG | (LOW, LOW, LOW) |
| LOW | (NORMAL,NORMAL,NORMAL) | 1 | POS | (LOW, LOW, LOW) |
| HIGH | (LOW, LOW, LOW) | 1 | 0 | (LOW, NO, NO) |
| HIGH | (LOW, LOW, LOW) | 0 | NEG | (LOW, NO, NO) |
| ... | ... | ... | ... | ... |
| D NORMAL | (NO, NO, NO) | 0 | 0 | (LOW, NO, NO) |
| D NORMAL | (NO, NO, NO) | 1 | POS | (LOW, LOW, LOW) |
| NORMAL | (NO, NO, NO) | 1 | NEG | (e-, e-, e-) |
| NORMAL | (NO, NO, NO) | 0 | POS | (NORMAL, LOW, LOW) |
| D NORMAL | (NO, NO, NO) | 0 | NEG | (NO, e-, e-) |
| D NORMAL | (NO, NO, NO) | 1 | 0 | (NO, NO, NO) |
| ... | ... | ... | ... | ... |

7.11. ábra. Szabad kifolyású érzékelős tartály színes Petri háló modelljének megoldása

7.4. Diagnózerek és tervezésük

A diagnózer egy olyan egység (berendezés és/vagy algoritmus), amely egy diagnosztizálandó rendszerről rendelkezésre álló mért bemenet- és kimenet értékekből, valamint a szóba jöhető meghibásodási módok leírásából vagy modelljéből előállítja a mért értékeknek megfelelő meghibásodási módo(k)a)t.

Ha a meghibásodási módok leírása kvalitatív differencia-algebrai egyenletekből áll, akkor a mért adatokat is kvalitatív jelek formájában kell megadni.

7.4.1. Karakterisztikus kvalitatív bemenet-kimenet jel-nyomok

A 3.2.4. alfejezetben már megismerkedtünk a kvalitatív jel-nyomokkal. Egy $[x]$ kvalitatív jel **jel-nyoma** (**signal trace**) egy esemény-sorozat

$$\mathcal{T}_{(x)}(t_0, t_F) = \{(t_0; [x](t_0) = q_{x0}), (t_1; [x](t_1) = q_{x1}), \dots, (t_F; [x](t_F) = q_{xF})\}$$

a (t_0, t_F) időintervallumon, ahol $q_* \in \mathcal{Q}_x$ és \mathcal{Q}_x a jelhez tartozó kvalitatív érték-készlet halmaza, avagy univerzum.

Több jelnek együttesen is lehet jel-nyoma, például $\mathcal{T}_{(u,d,y)}(t_0, t_F)$. Diagnosztikai célra a mérhető $\mathcal{T}_{(u,y)}(t_0, t_F)$ bemenet-kimenet jelnyomokat használjuk.

Ha a diszkrét idejű esetben minden jelhez minden időpillanatban tartozik érték, akkor egyszerűsített jelölést vezethetünk be úgy, hogy az időt elhagyjuk, például

$$\mathcal{T}_{(h,T)}(1, 3) = \{(N, N), (L, H), (L, e+)\}$$

Diagnosztikai célra definiálhatunk

- nominális (normális viselkedést leíró) bemenet-kimenet jel-nyomokat,
- *karakterisztikus (jellemző) bemenet-kimenet jel-nyomokat* (valamilyen adott meghibásodásra).

7.4.2. Diagnózerek fogalma és szerkezete

A diagnózer egy olyan diszkrét eseményű rendszer, ami "felismeri" egy meghibásodási mód jellemző jel-nyomát, és így azonosítja azt. Esetünkben a diagnosztizálandó rendszert egy kvalitatív differencia-algebrai egyenletekből származó színes Petri háló formájában írjuk le, ezért célszerű a diagnózert is ilyen formában megadni.

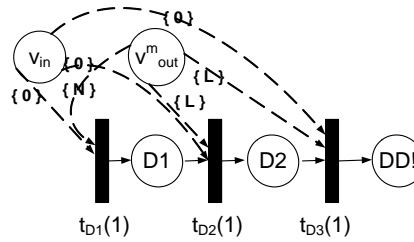
A színes Petri háló alakú diagnózer szerkezete a jellemző bemenet-kimenet jel-nyom ismeretében könnyen meghatározható az alábbiak szerint.

- Minden eseményhez a jellemző jel-nyomban külön átmenet t_{t_i} tartozik a diagnózerben.
- A diagnózer p_{Dt_i} belső helyei minden időlépés után az időlépés következményei, annak felelnek meg, hogy a jellemző jel-nyom i -edik eseménye megtörtént.
- A diagnózer a mért jelekhez tartozó helyekhez teszt-él segítségével kapcsolódik.

Példa. A fenti leírás alapján a

$$\mathcal{T}_{(v_{in}, v_{out}^m)}(1, 3) = \{(0, N), (0, L), (0, L)\}$$

jellemző bemenet-kimenet jel-nyom (ahol v_{in} a rendszer bemenete, v_{out}^m pedig a kimenet) felismerésére a 7.12. ábrán bemutatott diagnózer konstruálható.



7.12. ábra. Egy egyszerű diagnózer

7.4.3. Diagnózer felállítása színes Petri háló alakú modell alapján

Ha rendelkezünk egy olyan színes Petri háló alakú modellel, amely leírja a diagnosztizálandó rendszer dinamikus viselkedését a számunkra érdekes hibamódokban, akkor az alábbi lépések segítségével konstruálhatunk ehhez olyan diagnózereket, amelyek felismerik a hibamódokat.

1. Állítsunk elő a színes Petri háló szimulációjával minden hibamódhoz karakterisztikus bemenet-kimenet jel-nyomokat. Megjegyezzük, hogy a bemenetek megfelelő (okos) megválasztásával általában elkerülhető, hogy az összes lehetséges megoldást elő kell-jen állítani (így heurisztikával oldjuk meg az NP-nehéz lépést).
2. Minden karakterisztikus bemenet-kimenet jel-nyomhoz egy-egy külön diagnózert lehet konstruálni, azaz egy diagnózer egy hibamódot fog felismerni.

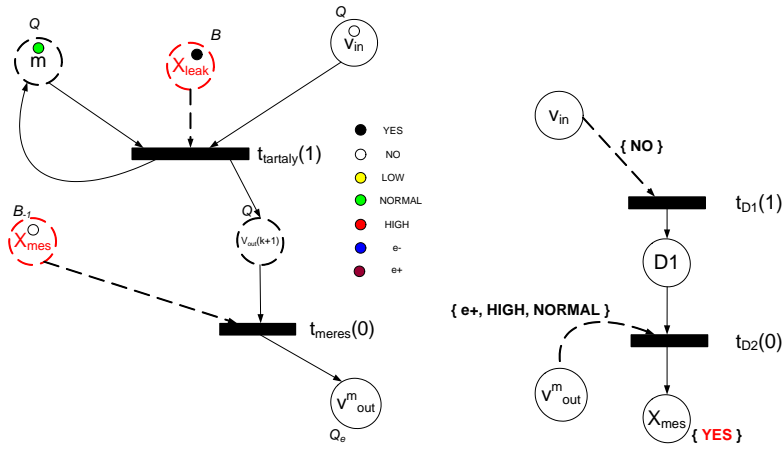
Szabad kifolyású érzékelős tartály színes Petri hálós diagnózere

Példaként konstruáljunk diagnózert a szabad kifolyású érzékelős tartály "pozitív hiba az érzékelőben", azaz $\chi = 1$ hibamódjának detektálására.

A rendszer bemenete a v_{in} beáramlási tömegáram, kimenete pedig a v_{out}^m mért kifolyási tömegáram. A "pozitív hiba az érzékelőben" egy *jellemző bemenet-kimenet jel-nyoma* pedig

$$\mathcal{T}_{(v_{in}, v_{out}^m)}(0, 1) = \{(0; [v_{in}] = 0), (1; [v_{out}^m] = H)\}$$

A 7.13. ábra bal oldali részábrája mutatja a diagnosztizálandó rendszer, a jobb oldali pedig a "pozitív hiba az érzékelőben" hibamódot felismerő diagnózert.



7.13. ábra. Szabad kifolyású érzékelős tartály színes Petri hálós diagnózere

8. fejezet

Folyamatbányászati módszerek a diagnosztikában

8.1. Bevezetés

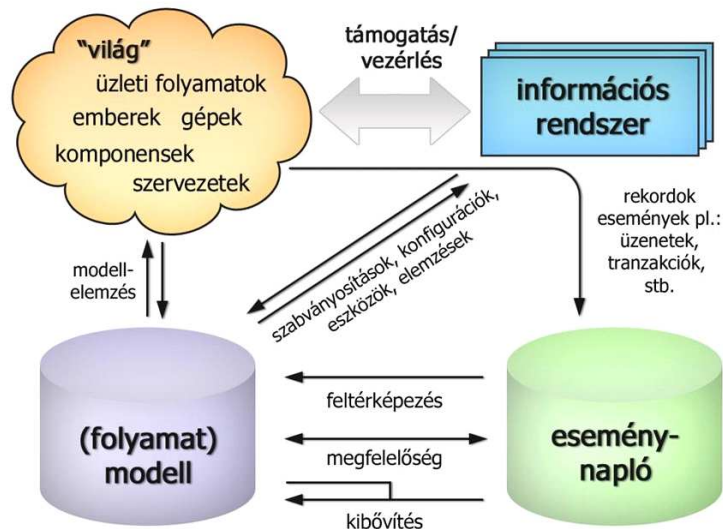
Az informatika területén az eseménynaplózás ma már korántsem számít új keletű fogalomnak, míg az eseménynaplózáson alapuló folyamatbányászaton rejülő lehetőségeket még csak most kezdjük kihasználni. A mai folyamatkezelő információs rendszerek meglehetősen részletes információkat tárolnak a végrehajtott folyamatokról, eseményekről. Ez a fajta információ különféle célokra széles körben feldolgozható és felhasználható.

A folyamatbányászat tudománya az adatbányászat egyfajta változatának is tekinthető, ahol a legfőbb cél, hogy a folyamatokból származó eseménynaplózásból valódi hasznosítható tudást nyerjünk ki. Számos, a témában járatos szakember fejlesztett már ki egészen kifinomult folyamatbányászati eszközöket. Ilyenek például a ProM, a Futura Reflect, az EMiT, a MiMo, az Aris PPM, a HP Business Cockpit vagy az ILOG JViews, melyek a rendelkezésre álló adatokból képesek lényeges összefüggéseket feltárni.

A különféle információs rendszerek egymástól különböző módon hajtják végre az események naplózását. Ennek következtében az interfész a folyamatkezelő rendszerek és a folyamatbányászati eszközök között nem egyértelmű. Ennek következtében a kutatók teljesen új folyamatbányászati technikákon dolgoznak, amelyek képesek létrehozni megfelelő bányászati infrastruktúrát akár felületes adatokból kiindulva.

8.2. Folyamatbányászat

Az információt folyamatokból nyerjük, és ezen forrásból pedig a tudást szerezhethetjük meg. Segítségével felderíthetővé és megjeleníthetővé válnak azon összefüggések, amelyek az elhúzódó átfutási időért, a magas költségekért, esetleg a rossz minőségért felelősek. A folyamatbányászatot alkalmazva a problémákat korábban azonosítani lehet, és előbb van lehetőség azok orvoslására. Az 8.1. ábrán látható, hogy az egyes információs rendszerek által rögzített naplókban különböző folyamatbányászati eszközökkel nyerhetjük ki azt a modellt, amit később elemezni tudunk. Alapvetően három ilyen eszközt különböztetünk



8.1. ábra. A folyamatbányászat magas szintű modell diagramja

meg. Azt az eszközt, amely kizárólag az eseménynaplóban talált adatokra épül **discovery** (feltérképezés) pluginnak nevezzük. **Conformance** (megfelelőség) pluginnak nevezzük azt, amely összehasonlítja a rendszer működését az eseménynaplóban lévő adatok és a fejlesztett modellben előírt viselkedés alapján. Végezetül az **extension** (kibővítés) plugin segítségével egy modelltől és a kapcsolódó naplófájlból olyan információkat nyerhetünk ki, amelyekkel tovább lehet fejleszteni a modellt [14].

8.3. Eseménynapló - log

Tekintsünk egy **vizsgálati folyamatot**, pl. lehet ez valamilyen gyógyszer hatékonyság vizsgálata. Ez a komplett vizsgálat több szakaszra bontható, nevezzük ezeket részvizsgálatoknak. Ezek a részvizsgálatok több eseményből, tevékenységből állnak. Egy részvizsgálat több személy esetében is elvégzésre kerülhet.

A vizsgálat során a műszerek egyrészt rögzítik a mért értékeket és a hozzájuk tartozó időpontokat, másrészt az is rögzíthető, hogy ki vagy mi milyen beavatkozást végzett el a vizsgálat közben. Ha ezeket az információkat megfelelő sorrendbe állítjuk, akkor a vizsgálati folyamatnak mintegy a "történetét", vagyis a lefolyását vagy menetét kapjuk meg. Egy konkrét részvizsgálathoz tartozó információk rögzítésével egy **trace-t** kapunk, több hasonló részvizsgálat sorozatával a trace-ek egy halmazát állíthatjuk elő, amelyre **log-ként** (naplófájl) hivatkozhatunk.

A folyamatbányászat alapvető elemei azon fájlok, amelyekben egyes eseményekkel kapcsolatos megtörtént eseteket regisztrálhatunk. A naplófájl (8.2. ábra) az alapja minden folyamatbányászati eszköznek, ami segítségünkre van egy rendszer feltárásában. Az információs rendszerek mindegyike másképpen naplózza az eseményeket, ennek következtében a folyamatkezelő információs rendszerek és a folyamatbányászati alkalmazások

```

Fájl Szerkesztés Beállítások Kiközlés Súlyg 1%
<?xml version="1.0" encoding="UTF-8" ?>
<!-- MXML version 1.0 -->
<!-- Created by Fluxicon Nitro (http://fluxicon.com/nitro/ -->
<!-- (c) 2010 Fluxicon Process Laboratories / http://fluxicon.com/ -->
<WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://
<Source program="Fluxicon Nitro"/>
<Process id="Final_log4.xml.gz" description="Converted to MXML by Fluxicon Nitro">
  <ProcessInstance id="Case139">
    <AuditTrailEntry>
      <Data>
        <Attribute name="Via">other</Attribute>
        <Attribute name="CustomerID">Customer2</Attribute>
        <Attribute name="Complexity">complex</Attribute>
        <Attribute name="ServiceType">client</Attribute>
        <Attribute name="Originator">ASA</Attribute>
      </Data>
      <WorkflowModelElement>Incomingtask</WorkflowModelElement>
      <EventType>complete</EventType>
      <Originator>07-11-2011 18:07:55</Originator>
    </AuditTrailEntry>
    <AuditTrailEntry>
      <Data>
        <Attribute name="Via">other</Attribute>
        <Attribute name="CustomerID">Customer2</Attribute>
        <Attribute name="Complexity">complex</Attribute>
        <Attribute name="ServiceType">client</Attribute>
        <Attribute name="Originator">KR</Attribute>
      </Data>
      <WorkflowModelElement>Handlingtask</WorkflowModelElement>
      <EventType>complete</EventType>
      <Originator>09-11-2011 18:14:54</Originator>
    </AuditTrailEntry>
    <AuditTrailEntry>
      <Data>
        <Attribute name="Via">other</Attribute>
        <Attribute name="CustomerID">Customer2</Attribute>
        <Attribute name="Complexity">complex</Attribute>
        <Attribute name="ServiceType">client</Attribute>

```

8.2. ábra. Egy log fájl felépítése - részlet

között meg kell találni a megfelelő kapcsolatot. Elengedhetetlen, hogy helyesen térképezzük fel és értelmezzük a naplózást, amit a különböző rendszerek más-más módon rögzítettek. Készült egy **meta modell** (8.3. ábra) a naplófájlok számára. A meta modellben megadásra kerültek az alapvető követelmények arra vonatkozóan, hogy milyen adatoknak kell egy logban feltétlenül rögzítve lenniük. A meta modellhez egy, az XML nyelv speciális változatát használják, az úgynevezett Mining XML, MXML formátumot. ([15])

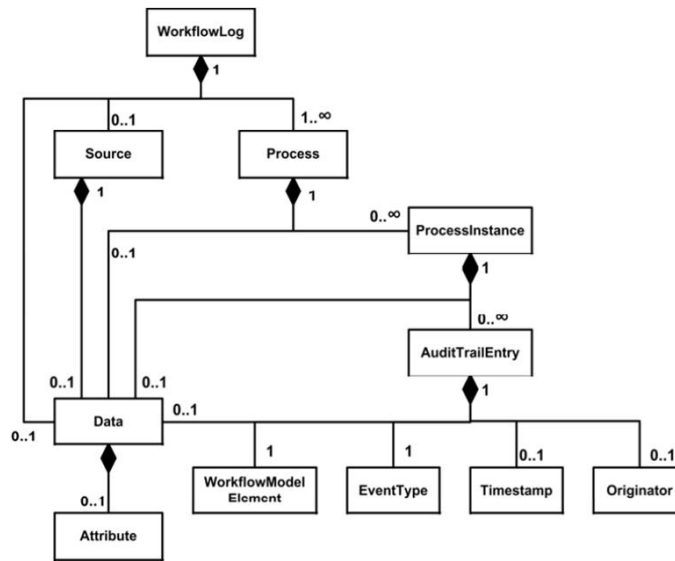
8.4. Modellezési módszerek - Petri háló

A folyamatbányászatban nagyon fontos szerepet játszó eszközök a modellezési technikák. Ahhoz, hogy egy naplófájlból kinyert struktúrát értelmezni tudjunk, olyan eszközre van szükség, ami átláthatóvá, egyértelművé teszi a bonyolult folyamatok lezajlását. Ilyen modellezési módszer például a Petri háló. A Petri hálókkal már korábban megismerkedtünk. [9]

A Petri hálókat a következő fejezetben ismertetésre kerülő ProM rendszer több bányászati és analízis módszernél is alkalmazza.

8.5. Folyamatbányászati lehetőségek a diagnosztikában

Mivel a vizsgálati folyamat szakaszokra bontható, így amikor például egy anyag valamilyen szempontú tesztelését követjük nyomon, a részvizsgálat közben rögzített adatok és a



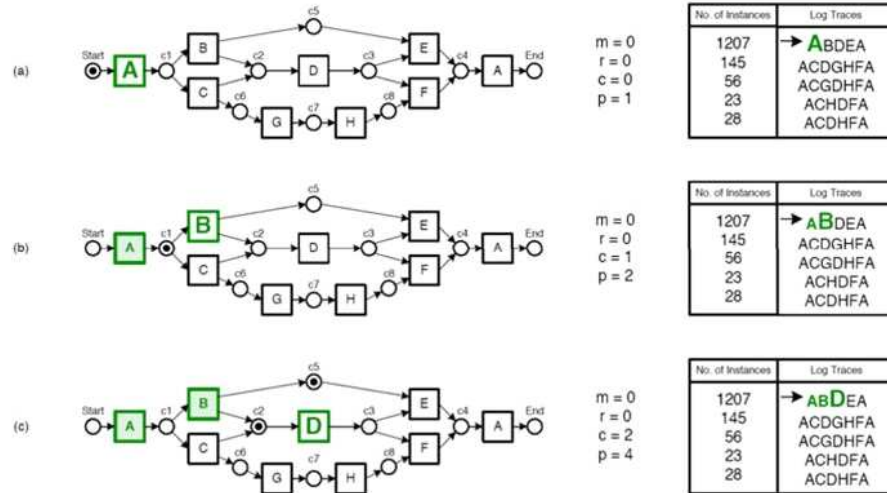
8.3. ábra. Meta modell egy log felépítéséhez

vizsgálat eredményessége között a legtöbb esetben kapcsolat van. Ha ilyen részvizsgálatot többször elvégzünk, akkor az adatsorok elemzésével tendenciák is kimutathatóak, és így bizonyos megállapítások tehetők, esetlegesen hibásan elvégzett műveletek felderíthetőek, beazonosíthatóak.

Az egyik lehetőség az analízisre az, hogy először elkészítjük a részfolyamat működését leíró modellt, a modell a részfolyamat vizsgálata során lejátszódó eseményeket és azok előfeltételeit, illetve következményeit emeli ki a leíráshoz. Ezután a részfolyamat többszöri végrehajtása után kapott logot összehasonlítjuk ezzel a modellel. Megvizsgálhatjuk egyrészt, hogy a modell mennyire illeszkedik a logokhoz, azaz annak alapján a logok milyen hányadát lehetne előállítani, másrészt a modell a szerkezetét és struktúráját tekintve mennyire felel meg a logoknak.

Egy másik lehetséges módszer lehet az, hogy a trace-ek összehasonlításával például láthatóvá tehetjük a folyamatokban lévő események közötti feladatok különbségeit. Kimutatható, hogy két folyamat mennyire azonos, illetve különböző. Két folyamat összehasonlítása például azért fontos, mert ezáltal látható, hogy melyik folyamat melyik esemény megtartásával vagy elhagyásával válhat költséghatékonyabbá a másikkal. [19], [20], [13]

Nézzük meg részletesebben a következő vizsgálati lehetőséget: Ha rendelkezésünkre áll egy folyamat modellje és valamilyen információs rendszer felhasználásával lehetővé vált a valós folyamat loggolása, amiből rendelkezésre áll egy napló fájl is, lehetőségünk nyílna arra, hogy összehasonlítsuk a modellt és a logot. Az összehasonlítás lehetővé teszi a tervezett folyamat és a valóságban végbement folyamat közötti eltérések kiemelését, mérhetjük a hasonlóságot. A következő ábrákon végig követhetjük egy modell és log összehasonlítását. Mint ahogy azt a 8.4. és 8.5. ábrán láthatjuk, a modellünk egy speciális Petri háló (Workflow-net). A logban összesen 5 trace helyezkedik el, ami azt mutatja, hogy az egyes események egymásután 5-ször hajtották végre a rögzített sorrendben. Az



8.4. ábra. A modell és a log 1. trace-nek összehasonlítása - első 3 lépés

m (missing) értéke a hiányzó, az r (remaining) értéke pedig a megmaradó műveletek számát mutatja meg. A c (consumed) változó megmutatja, hogy hány helyről vettük el a tokeneket, míg a p (produced) jelzi, hogy hány helyre került vagy éppen kerül token a Petri hálóban.

Amennyiben egy adott trace-hez tartozó egymást követő események végig követhetőek a modellen, az m és r értéke 0 marad.

A 8.6. és 8.7. ábrán a 4. trace-t hasonlítjuk össze a modellel. Mivel a trace eseményei és a Petri háló egy útvonalán elhelyezkedő elvárt események között eltérés van, ezért az m értéke változni fog, ez látható a 8.6. ábra 3. lépésétől kezdve.

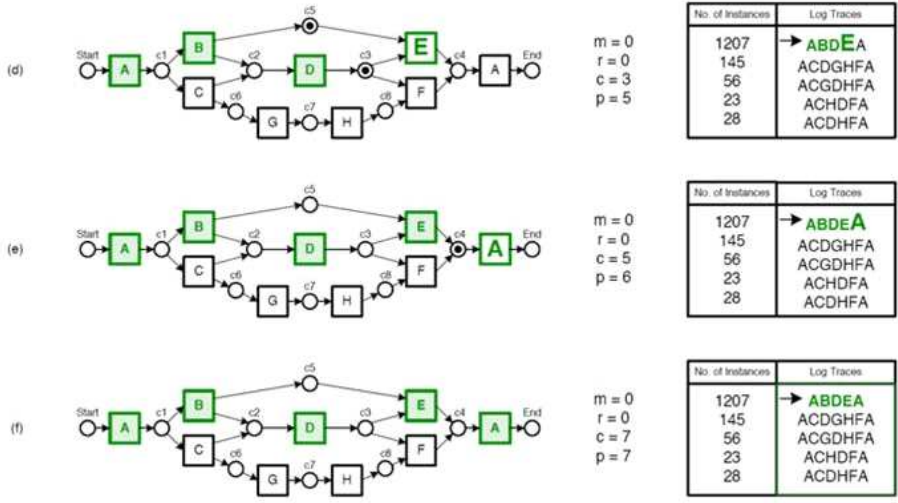
Ebből az következik, hogy az események a valóságban nem a tervezett sorrendnek megfelelően hajtottak végre, valószínű valamilyen hiba következett be a rendszerben (lásd 8.8. ábra). Mint ahogy azt korábban említettük, lehetőségünk van a hasonlóság mérésére, amihez be kell vezetnünk egy ún. **fitness függvényt** (jelöljük f -fel), amely figyelembe veszi a logban szereplő trace-ek számát (k) és az események számát (n), valamint a már korábban bevezetett m , r , c és p értékeket. A hasonlóság mértéke kiszámítható az alábbi összefüggés felhasználásával:

$$f = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i m_i}{\sum_{i=1}^k n_i c_i} \right) + \frac{1}{2} \left(1 - \frac{\sum_{i=1}^k n_i r_i}{\sum_{i=1}^k n_i p_i} \right)$$

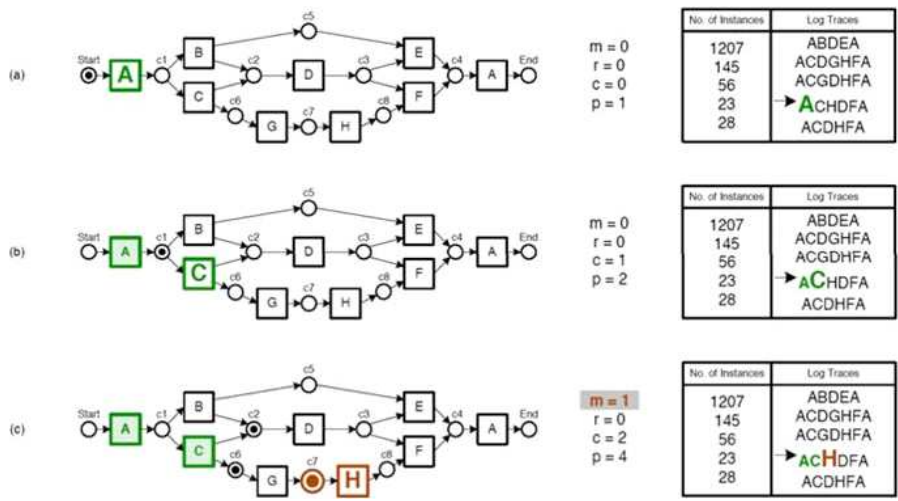
Egy konkrét számítás eredménye:

$$f(M1, L2) = \frac{1}{2} \left(1 - \frac{51}{10666} \right) + \frac{1}{2} \left(1 - \frac{51}{10666} \right) \approx 0.995$$

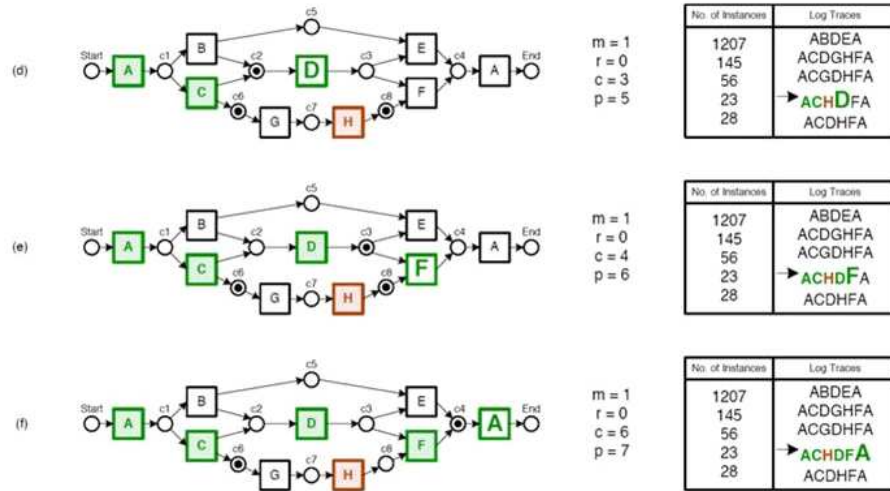
A talált hiba helyét a 8.9. ábra mutatja. A 8.10. ábrán 3 különböző trace-nek a modellel való összehasonlítási eredménye látható, amely jól tükrözi a loggoláskor rögzített információk különbözőségeit és a valós folyamatoknak a tervezett végrehajtási folyamatától való eltéréseinek mértékét.



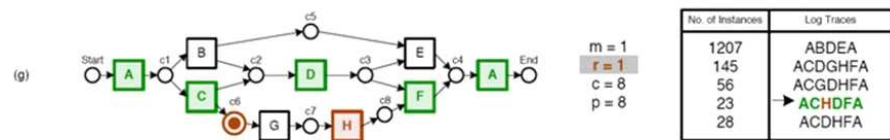
8.5. ábra. A modell és a log 1. trace-nek összehasonlítása - következő 3 lépés



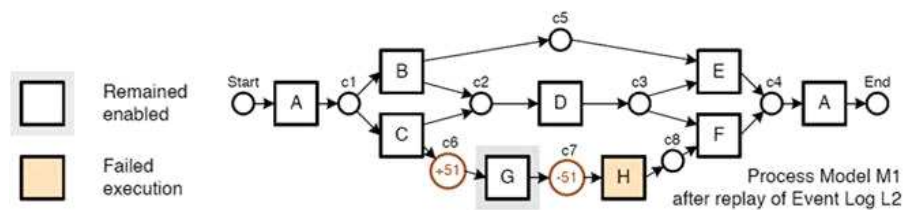
8.6. ábra. A modell és a log 4. trace-nek összehasonlítása - első 3 lépés



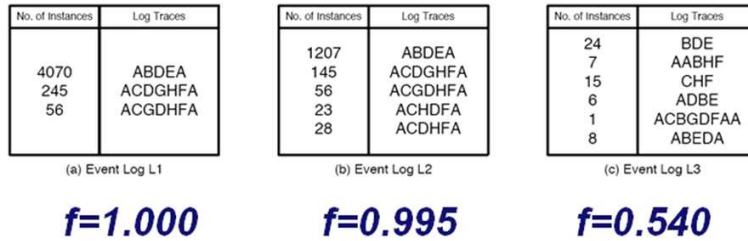
8.7. ábra. A modell és a log 4. trace-nek összehasonlítása - következő 3 lépés



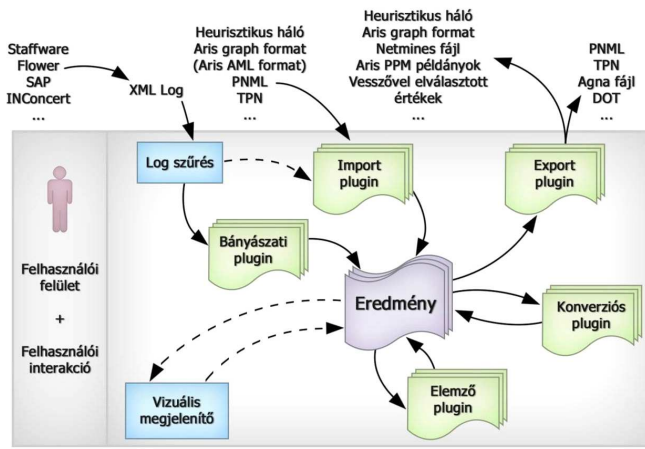
8.8. ábra. A modell és a log 4. trace-nek összehasonlításának eredménye



8.9. ábra. A sikertelen végrehajtás helye a modellben



8.10. ábra. A kiindulási modell és 3 log összehasonlítása



8.11. ábra. A ProM keretrendszer architektúrája

8.6. Információszerzés folyamatbányászati módszerekkel a ProM 6.0 keretrendszer segítségével

Ebben a fejezetben konkrét példákon keresztül adunk segítséget a folyamatbányászattal ismerkedőknek a ProM 6.0 keretrendszer (lásd 8.11. ábra) használatához. A keretrendszer telepítésével és használatával kapcsolatos legfontosabb tudnivalók a 8.6.4-ben olvashatók.

8.6.1. Példa log elemzése

Ebben a fejezetben példa log-ok segítségével mutatjuk be, hogy hogyan kell használni a ProM-et, és választ adunk néhány, a vállalati vezetők által gyakran felmerülő kérdésre. Ezek a kérdések általában az alábbiak szoktak lenni:

- Mi az átlagos/minimum/maximum átviteli ideje az egyes eseteknek?

- Melyik út/utak tartanak túl sokáig? Mik az ezekhez tartozó kritikus alutak?
- Mi az átlagos szolgálati idő az egyes feladatoknál?
- Mennyi idő telik el két feladat végrehajtása között?
- Valójában mi történik az esetekkel, hogyan vannak végrehajtva?
- Hogy vannak a szabályok valójában teljesítve?
- Hány ember kapcsolódik egy esethez?
- Mi a kommunikációs struktúra és függőségi viszony az emberek között?
- Kik a fontos szereplők a kommunikációs áramlásban?
- Ki kinek dolgozik be?
- Kik dolgoznak azonos feladaton?

Ezek általános jellegű kérdések, melyek mind egy adott folyamat hatékonyabbá tételét szolgálják. Megválaszolásukkal többet tudhatunk meg egy adott folyamatról és minőségi, időbeli, finansiális előnyökhöz juthatunk.

8.6.2. A repairExamples.xes log elemzése

Az első dolog, amit egy log vizsgálatakor tennünk kell, az az, hogy betöltésre kerül a ProM környezetbe. Az első tesztfeladat során a *repairExamples.xes* log-ot használjuk.

A feladat a következő: tekintsük egy hibás telefonokat javító vállalat működési folyamatát. A vállalat 3 különböző telefont képes javítani: $T1$, $T2$ és $T3$. A folyamat az ügyfelek által küldött hibás készülékek regisztrálásával kezdődik. Ezután az eszköz a probléma felderítését végző osztályra kerül, ahol megvizsgálják és kategorizálják az eszközt a talált hiba alapján. Későbbi megjavítása alapján, az előre definiált összesen 10 különböző hibakategória egyikébe kerül a telefon. Miután a hiba behatárolása megtörtént, az eszközt tovább küldik a javítási részlegre, miközben az ügyfél is kiértesítésre kerül. A javítási részlegnek két csoportja van: az egyik az egyszerű problémákat képes megoldani, a másik pedig a komolyabbakat. Vannak azonban olyan hibák is, melyek megoldását mind a két csoport el tudja látni. A hiba elhárítása után a telefon a minőségbiztosítási részlegre kerül. Itt ismét ellenőrzik, hogy a hiba valóban elhárításra került-e. Amennyiben további probléma áll fent, az eszköz újabb vizsgálatra visszakerül a javítási részlegre. Ha pedig a telefon valóban hibamentes, akkor elküldik az ügyfélnek és az esetet lezárják. Azért, hogy időt takarítsanak meg, a vállalat csak korlátozott számban próbál megjavítani egy telefont. Ha a javítás nem sikerül, akkor az ügyfélnek küldenek egy teljesen új telefont, és az esetet lezárják.

Mielőtt elkezdenénk a példa folyamatbányászati ismertetését, rendkívül fontos tisztázni még azt is, milyen adatokra, információkra van szükségünk egy esemény log-jaiból. Ennek fő oka az, hogy csak akkor válaszolhatunk konkrét kérdésekre, ha a válaszhoz szükséges információt a log tartalmazza is. Például abszolút nem lehet számolni egy eset átfutási idejével, ha a hozzá tartozó log nem tartalmaz információt az egyes feladatok elvégzésének

The screenshot shows the ProM 6 software interface. The main window displays a 'Log Summary' for a file named 'repairExample.zip'. The interface includes a sidebar with 'Dashboard', 'Inspector', and 'Summary' options. The 'Log Summary' table is as follows:

| Class | Occurrences (absolute) | Occurrences (relative) |
|---------------------------|------------------------|------------------------|
| Test Repair+complete | 1508 | 12,72% |
| Test Repair+start | 1508 | 12,72% |
| Register+complete | 1104 | 9,313% |
| Analyze Defect+complete | 1104 | 9,313% |
| Analyze Defect+start | 1104 | 9,313% |
| Inform User+complete | 1102 | 9,296% |
| Archive Repair+complete | 1000 | 8,435% |
| Repair (Simple)+complete | 785 | 6,622% |
| Repair (Simple)+start | 785 | 6,622% |
| Repair (Complex)+start | 725 | 6,116% |
| Repair (Complex)+complete | 724 | 6,107% |
| Restart Repair+complete | 406 | 3,425% |

8.12. ábra. A hibadetektálási log adatai

idejéről. Valamint az is nagyon fontos, hogy képesek legyünk a szükségtelen információkat kiszűrni, és "megtisztítani" a log-ot a felesleges adatoktól.

Például amikor egy konkrét esetben csak az elvégzett feladatokra vagyunk kíváncsiak; értelemes, hogy ebben az esetben szükségtelen a folyamatban lévő feladatok pontos számának ismerete. Ennek egy további vetülete, hogy azt is tudnunk kell, hogyan kell kezelni ezeket a megtisztított log-okat, hogy elkerülhessük a felesleges munkát. Viszont a példához, meghatároztunk néhány általános érvényű kérdést, amikre a log elemzésével majd választ tudunk adni:

- Hány eset/trace van a log-ban?
- Hány feladat/művelet van a log-ban?
- Mennyi erőforrás van a log-ban?
- Melyik erőforrás melyik feladathoz van rendelve?

Erre az öt kérdésre a log összefoglalójának megtekintése után kaphatunk választ. Ehhez a **Workspace** ablakon belül a **Visualize** menü megnyitására van szükség, ahol a **Summary** alatt válik láthatóvá a számunkra fontos összegzés. Az **MXML Legacy Classifier / End events** tartalmazza az első négy kérdésre a választ, hiszen ebből a részből világosan látszik, hogy jelenleg 1000 lezárt eset és 1104 folyamatban lévő van a vállalatnál. A 8.12. ábra mutatja a vállalat egyes feladatrészeinek összegzését.

Az ötödik kérdésre a választ az **Event name AND Resource** rész adja meg: ebből jól látszik, hogy a SolverC kezdetű azonosítóval rendelkező emberek foglalkoznak a bonyolult feladatokkal, míg a SolverS kezdetű azonosítóval ellátott emberek pedig a könnyebb

problémákkal. Az is megfigyelhető, hogy mindkét csoportban 3 – 3 ember dolgozik. Az *Inspector* menüpontban az egyes esetek részletesen elemezhetőek, kiértékelhetőek.

8.6.3. Esettanulmány - Operátori eljárással irányított garázska

A továbbiakban egy operátori eljárással irányított garázska működését leíró modelleket és logokat használjuk fel a folyamatbányászat bizonyos lehetőségeinek a bemutatására. A rendszer két fő részből áll, az egyik maga az automata, a másik pedig az operátor, azaz az autóvezető. Miután az autós beáll a garázsba a csukott sorompó elé, meg kell nyomnia egy gombot a jegykiadó gépen, amit az automata érzékel, és egy tálcára kiad egy parkolójegyet. A jegy elvételének hatására a sorompó felnyílik, az autós behajthat és leparkolhat. A sorompó lecsukódik, és az automata várja a következő leparkolni kívánó járművet.

A folyamatot leíró modellt meg fogjuk vizsgálni normál körülmények között, azaz amikor hibamentes esetet ír le a modellünk, illetve úgy is, amikor hibák vannak beiktatva a működésbe.

Hibamentes működésről akkor beszélünk, amikor az egész folyamat pontosan úgy viselkedik, ahogy azt előre megjósolnánk. Nem lép fel hibás működés sem az automata részéről, sem pedig az operátor felőli oldalról. A Petri háló a HPSim nevezetű szimulációs programmal készült, ezáltal lehetőség nyílt a modellt úgy megvizsgálni, ahogy a tokenek ténylegesen mozognak, valamint azt is könnyebben ki lehetett elemezni, hogy az átmeneteknek milyen előfeltételei vannak. A 8.13. ábrán látható a hibamentes folyamat Petri hálója. A bal oldali részben találjuk meg az autósra vonatkozó leírást, míg a jobb oldaliban az automatához tartozó akciókat.

Bizonyos átmeneteket kiemelve bemutatjuk azok leírását, mivel az eseménynapló is ezen logika szerint épül fel:

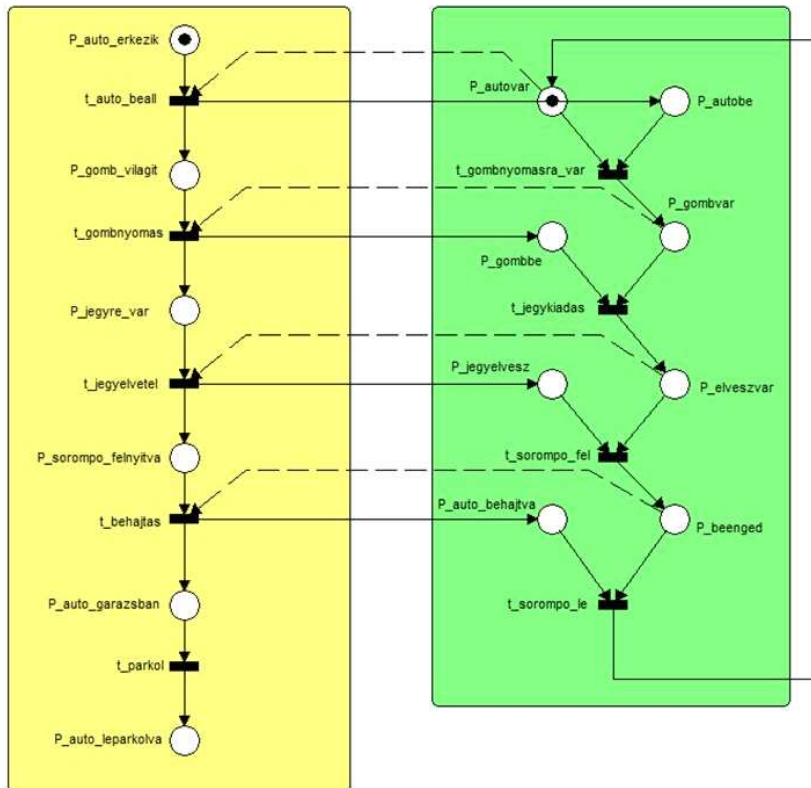
t_{auto_beall}: Ennek az átmenetnek az előfeltétele, hogy a *P_{auto_erkezik}* helyen legyen token (azaz van autó, amely parkolni szeretne), valamint az automata oldalán a *P_{autovar}* helyén is kell lennie egy tokennek, ami azt jelenti, hogy az automata képes autót fogadni.

t_{gombnyomasra_var}: Ahhoz, hogy engedélyezve legyen ez az átmenet, a *P_{autovar}* és a *P_{autobe}* helyeken kell lenniük tokeneknek. Ekkor az automata a gombnyomásra váró állapotba kerül.

A 8.13. ábra alapján a többi átmenet leírása hasonlóan megadható. Amennyiben rendelkezésre áll a folyamat modellje és elő tudjuk állítani a folyamat egy eseménynaplóját, akkor lehetőségünk van a modellünk helyességének ellenőrzésére.

Ha rendelkezésre áll a folyamat eseménynaplója, betöltjük a logot a ProM keretrendszerbe és ellenőrizhetjük, hogy az α -*algoritmus* kimeneti modellje megegyezik-e a kiindulási modellünkkel.

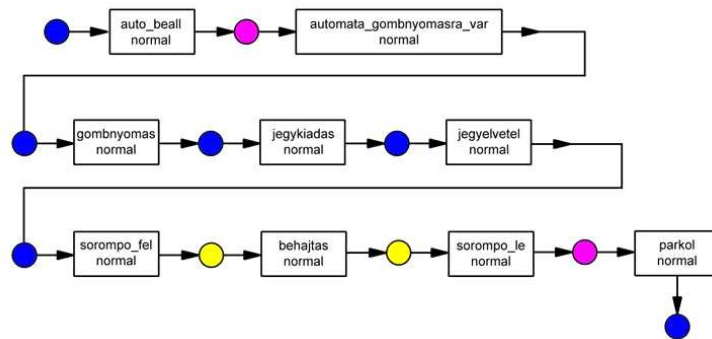
A vezérjelek mozgását is tudjuk szimulálni a generált hálóban, ehhez rendelkezésre áll egy *Fuzzy Miner* algoritmus. Az animációs modellből kiderül, hogy az eseménynaplónk biztosan helyes, hiszen pontosan úgy viselkedik, mint amit a kiindulási modellünknel tapasztaltunk, amikor a HPSim program szimulációs környezetében vizsgáltuk. Egy átfogó lényegi összegzést kaphatunk az *Open log with classic dialog* plugin használatával. Az *Originator by task Matrix* plugin segítségével megvizsgálhatjuk, hogy



8.13. ábra. Garázskapu rendszer hibamentes modellje

| originator | auto_beall | automata_go... | behajtas | gombnyomas | jegyvetel | jegykiadas | parkolas | sorompo_fel | sorompo_le |
|------------|------------|----------------|----------|------------|-----------|------------|----------|-------------|------------|
| Auto | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| Automata | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

8.14. ábra. Hibamentes működés, Originator by Task Matrix plugin

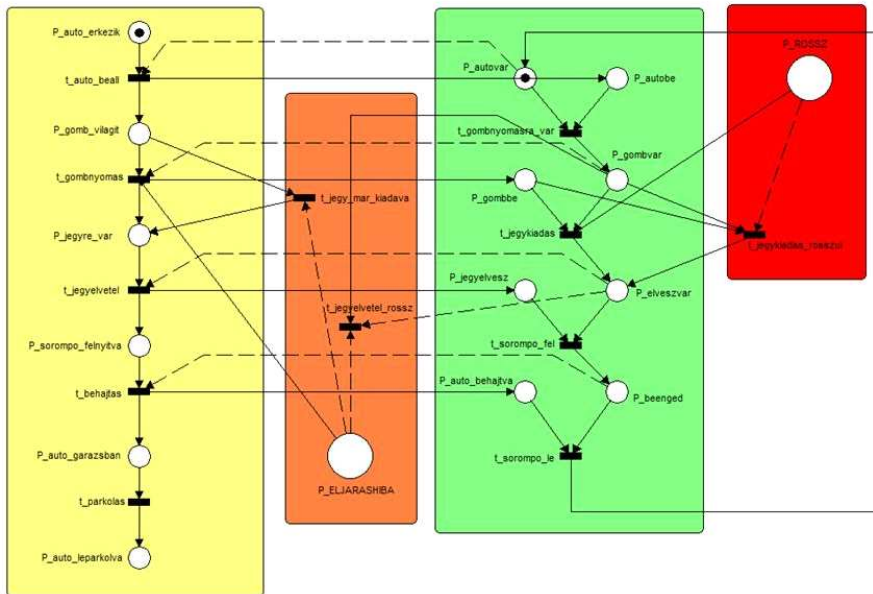


8.15. ábra. Hibamentes működés, Teljesítményelemzés plugin

milyen események tartoznak az egyes résztvevő tagokhoz (lásd 8.14. ábra).

Az eseménynapló vizsgálatával lehetőségünk nyílt időtorlódásra vonatkozó elemzések elvégzésére. Ehhez szükségünk van egy referencia modellre, amit az α -algorithm kimeneti Petri hálójá biztosíthat számunkra. A *Performance Analysis with Petri Net* plugin segítségével végezhetjük el az ilyen jellegű vizsgálatainkat a folyamatokra vonatkozóan. Lehetőségünk van az időegységre vonatkozó mértékegység és a tizedes jegyek számának a beállítására. A torlódásra vonatkozóan háromféle küszöbértéket adhatunk meg, amelyekkel szabályozhatjuk, hogy az egyes események időbeli lefutása milyen szintnek (alacsony, közepes, magas) feleljen meg. Vagyis, ha az alacsony szinthez tartozó kék színnel jelölt értékhez 10 másodpercet állítunk be, akkor a modellen azon helyek színe (a Petri hálóban ezek a körök), ahol egy esemény maximum 10 másodpercig tartott, kék színű lesz. Hasonlóan működik a közepes szinthez tartozó sárga színű érték is, amelyhez 30 másodpercet állítunk be. A magenta színű érték, mely a magas szintet képviseli, automatikusan 30 másodpercnél nagyobb értékekre fog vonatkozni (lásd 8.15. ábra). A színek segítségével a modelltől azonnal leolvashatjuk, hogy mely események fognak a leghosszabb ideig tartani, melyek a 10 és 30 másodperc közöttiek, illetve melyek 10 vagy ennél kevesebb idő alatt. Láthatjuk, hogy míg az autós eljut az automatáig, ami érzékeli őt és gombnyomásra aktívává teszi a jegykiadó gombot, több mint 30 másodperc telik el. Jelen esetben ez 35 másodperc, amit torlódáspontként is felfoghatunk. A hatékonyabb működéshez a való életben itt gyorsításra lenne szükség.

A parkoló rendszer hibás működése annyiban fog eltérni az alap modellünktől, hogy kétféle hibaeseményt tartalmaz. Ezt úgy valósítjuk meg az eseménynaplóban, hogy három különböző folyamati esetet hozunk létre. Az első részfolyamat a hibamentes esetet tartalmazza, a második és harmadik részfolyamat pedig a hibás eseteket, amelyek mint



8.16. ábra. Garázskapu rendszer hibákat észlelő modellje

majd láthatjuk, egymással kapcsolatban állnak. A 8.16. ábra a kiindulási modellünket ábrázolja. Az utolsó doboz azt a hibát észlelni, amikor a vezető megnyomja a gombot, ám az automata egy jegy helyett kettőt fog kinyomtatni.

Ekkor az autós elveszi az egyiket, majd a folyamat folytatódik tovább. Egy jegy azonban ott marad a $P_{elveszvar}$ helyen, és ez azt jelenti, hogy a következő autós számára már eleve ott lesz egy parkolójegy a tálcán. Persze megnyomhatja újra a gombot, ebben az esetben a kiadott jegyek száma mindig növekedni fog. Ez új átmenet lesz, a $t_{jegykiadas_rosszul}$.

A modellben a második doboz fogja reprezentálni azt a hibát, amikor az egyik lépést teljesen kiiktatjuk ($t_{gombnyomas}$). Ez azt jelenti, hogy az autós előbb veszi el a jegyet, mielőtt még megnyomná a gombot. Természetesen ez csak abban az esetben lehetséges, ha az automata már eleve hibásan működött. Ekkor az autós, az aktívvá vált gomb után annak megnyomása nélkül, az előzőleg kiadott jegyet veszi el. Ez a hiba csak addig állhat fenn, amíg a $p_{elveszvar}$ helyen el nem fogy a token (azaz el nem fogynak a már hibásan kiadott jegyek). Két új átmenet fog megjelenni a modellben, a $t_{jegy_mar_kiadva}$ illetve a $t_{jegyvetel_rossz}$.

A folyamattal kapcsolatos összegző adatok lekérdezésére, mely adatokat a logból nyerjük, az *Open log with slicker dialog* plugint is használhatjuk. Ennél a modellnél is hasonló lehetőségeink vannak az elemzésre, mint a hibátlan működési modellnél. Ezen kívül megvizsgálhatjuk, hogy egy valódi rendszerhez kapcsolódó log milyen összhangban van a készített modelljeinkkel, hol - ebben van segítségünkre a *Trace comparison* plugin - és mikor - itt használhatjuk a *Conformance checker* plugint - jelentkezhettek a hibák.

8.6.4. A keretrendszer ismertetése

Ahhoz, hogy egy eljárást hasznosítani tudjunk, nélkülözhetetlen megvalósítási környezetének ismerete. A ProM 6.0 szabadon letölthető és telepíthető program, mely magában foglalja a ProM 6 Package Manager-t és a XESame1 kiegészítést is. Szükség szerint rendelkezésre áll egy fórum és felhasználói levelező lista is a további információk szerzésére vagy problémák elhárítására. ([16])

A program a <http://www.promtools.org/prom6/> oldalról tölthető le. Például Windows XP platform esetében a hozzá tartozó telepítő fájl letöltésével és futtatásával válik elérhetővé a program. Ennek befejezése után számos további kiegészítő csomag telepítése is szükséges. A ProM6-ban igen sok előre elkészített funkcionális csomag érhető el 3 okból:

1. Licenz: a ProM6 az L-GPL licenz alatt érhető el, amely egy open source licenz.
2. Letöltési méret: néhány kiegészítés platform, azaz operációs rendszer-függő, így csak a számunkra szükségesek letöltése lényeges, ezáltal hely spórolható.
3. Nem használt kiegészítők: az egyéni felhasználók más és más kiegészítő csomagokat használhatnak egyéni igényeik szerint, így nincs szükség minden, a programhoz tartozó kiegészítés telepítésére, megtartására; a nem használt kiegészítések törölhetőek.

A kiegészítő csomagok telepítéséhez a program futtatása szükséges. Az első futtatása során automatikusan telepíti a legszükségesebb kiegészítéseket, ehhez természetesen működőképes internet kapcsolat szükséges. Lehetséges alternatíva lehet a kiegészítések telepítésére a Package Manager futtatása. Kezdetben nincs semmilyen csomag telepítve, és ezt manuálisan tehetjük meg a "Not installed" gomb megnyomásával és a kiválasztott BPM2010 csomag telepítésével. Telepítése után elérhetővé válik a kiegészítés.

A ProM6 Package Manager minden opcionálisan telepített csomagot eltárol a számítógép merevlemezén az adott ".ProM" végződésű könyvtárban. A *packages.xml* fájl tárolja az összes elérhető és telepített csomagot. Bármilyen észlelt probléma esetén célszerű ennek a fájlknak a letörlése, majd a BPM2010 kiegészítő csomag ismételt telepítése. A ProM6 használatát 3 különböző funkcionális nézet határozza meg: a "**Workspace**" *fül*, amely a munkafelület nézetet jeleníti meg, az "**Action**" *fül*, amely a különböző műveleteket jeleníti meg, és a "**View**" *fül*, ami által az elkészített munka tekinthető meg. Minden nézethez más és más funkciók társulnak, annak előre definiált szerepe szerint.

| Perspektíva | ProM plugin |
|--------------------------------|---|
| Ellenőrzött folyam perspektíva | Alpha algorithm Multi-phase Macro plugin Heuristic miner Genetic algorithm plugin |
| Szervezeti perspektíva | Social Network Miner Organizational Miner Analyze Social Network Originator by Task Matrix |
| Eset szerinti perspektíva | Performance Sequence Diagram Analysis |
| Egyéb | Semantic LTL Checker Conformance Checker Performance Analysis with Petri net Decision Point Analysis |

8.17. ábra. A folyamatbányászat perspektívái és a hozzájuk tartozó plugin-ek

Irodalomjegyzék

- [1] Bartolozzi, V., Castiglione, L., Piciotto, A., Galuzzo, M. (2000) Qualitative models of equipment units and their use in automatic HAZOP analysis, *Reliability Engineering and Systems Safety*, **70**, pages 49 - 57.
- [2] Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M. (2006) *Diagnosis and Fault-tolerant Control*, Springer-Verlag.
- [3] Cassandras, C.G., Lafortune, S. (1999) *Introduction to Discrete Event Systems*, Kluwer Academic Publishers.
- [4] Faltings, B., Struss, P. (1992) *Recent Advances in Qualitative Physics*, The MIT Press, Cambridge, MA.
- [5] Fanti, M.P., Seatzu, C. (2008) Fault diagnosis and identification of discrete event systems using Petri nets, 9th International Workshop on Discrete Event Systems, WODES 2008, 432 - 435.
- [6] Forbus, K.D. (1984), Qualitative Process Theory, *Artificial Intelligence*, **24**, pp. 85–168.
- [7] Hangos, K.M., Cameron, I.T. (2001) *Process Modelling and Model Analysis*. Academic Press, London.
- [8] Lakner R., Hangos K., Gerzson, M. (2011) *Intelligens irányító rendszerek*. Typotex, Budapest, pp. 1-119.
- [9] Hangos, K.M., Lakner, R., Gerzson, M. (2001) *Intelligent Control Systems: An Introduction with Examples*. Kluwer Academic Publisher, New York.
- [10] Jensen, K. (1997) *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag.
- [11] Kuipers, B. (1986). Qualitative Simulation. *Artificial Intelligence*, **29**, pp. 289–388.
- [12] Palmer, C., Chung, P.W.H. (2009) An automated system for batch hazard and operability studies, *Reliability Engineering and Systems Safety*, **94**, pp. 1095 - 1106.

- [13] Starkné Werner Á., Dulai T. (2011) Folyamatbányászati eszközök felhasználása irányítási folyamatok elemzéséhez, *Kitekintés-Perspective Magyar-román-szlovák periodika*, XV. évfolyam különszám, 189-198, ISSN 1454-9921.
- [14] van der Aalst, W.M.P. at al. (2007) Business Process Mining: An Industrial Application. *Information Systems*, 32(5), pp. 713-732.
- [15] van der Aalst, W.M.P., Weijters, A.J.M.M. (2004) Process mining: a research agenda. *Computers in Industry*, 53(3), pp. 231-244.
- [16] van Dongen, B.F., at al. (2007) Process Mining with ProM. *Proceedings of the 19th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC)*.
- [17] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. (2003) A review of process fault detection and diagnosis Part II: Qualitative models and search strategies, *Computers and Chemical Engineering*, **27**, pp. 313 - 326.
- [18] Weld, D. S., de Kleer, J. (Eds.) (1990) *Readings in Qualitative Reasoning about Physical Systems*, The Morgan Kaufman.
- [19] Werner-Stark, Á., Németh, E., Gerzson, M., Hangos, K.M. (2011) Model-Based Fault Detection and Isolation using Process Mining, World Academy of Science, *Engineering and Technology*, year 7, issue 73 January 2011, pp. 851-856.
- [20] Werner-Stark, Á.; Gerzson, M.; Hangos, K.M. (2011) Discrete event model structure identification using process mining, Proceedings of the IASTED International Conference Modelling, Identification, and Control, Innsbruck, Austria, February 14-16, 2011, pp. 228-233, ISBN 978-0-88986-863-2.