Acta Agraria Kaposváriensis (2003) Vol 7 No 3, 69-86
Kaposvári Egyetem, Állattudományi Kar, Kaposvár
University of Kaposvár, Faculty of Animal Science, Kaposvár

Hisztorikus folyamat-adatok szegmentálása fuzzy csoportosítási algoritmus segítségével

Feil B., Abonyi J., Németh S., Árva P.
Veszprémi Egyetem, Folyamatmérnöki Tanszék, Veszprém, 8200 Pf. 158.

ÖSSZEFOGLALÁS

A modern folyamatirányító számítógépek által rögzített hatalmas adathalmaz a folyamat modellezésében, és a modellen keresztül a folyamat irányításában és fejlesztésében is hasznos lehet. E komplex kérdéskörön belül, az ipari technológiák állandósult üzemeltetéséhez tartozó állapotváltozók közti kapcsolatok elemzéséhez elengedhetetlenül szükséges eszköz fejlesztésével foglalkozunk. Egy adekvát stacioner összefüggéseket leíró modell felállításához szükség van az adatsorok azon értékeinek kiválasztására, amelyekről biztosan állíthatjuk, hogy ok-okozati kapcsolatban vannak egymással, pl. melyek alapján meg lehet határozni, hogy a termékminőséget milyen állapotváltozó-értékek eredményezték. Mivel a működő technológiai rendszerekben sok mért állapotváltozó nyomonkövetése szükséges, olyan eszköz kell, amely lehetővé teszi sok párhuzamos idősor szimultán elemzését és szegmentálását. E feladat megoldására egy fuzzy csoportosítási algoritmust alkalmazó módszert javaslunk, mely zajjal szemben nem érzékeny és képes a csoportok számát automatikusan meghatározni. A megoldás alapgondolata, hogy a vizsgálandó idősor adott pontja helyett a pontot transzformált változók segítségével a környezetével közösen jellemezzük, majd ezek között a transzformált változók között keresünk hasonlóságot az említett csoportosítási algoritmus segítségével. Az általunk kifejlesztett eszközt egy nagysűrűségű polietilén-gyár (TVK Rt.) mért állapotváltozóinak elemzésében alkalmaztuk.
(Kulcsszavak: adatelemzés, csoportosítás, folyamat irányítás, fuzzy modellek, idősorok)

ABSTRACT

Segmentation of historical process data based on fuzzy clustering algorithm

B. Feil, J. Abonyi, S. Németh, P. Árva
University of Veszprém, Department of Process Engineering, Veszprém, H-8201 P.O.Box 158.

The segmentation of historical process and business data is an important data-mining task, as the resulted segments are used for building predictive models, effective storing and querying of historical databases, and rule-searching. In this paper a tool is presented which can analyse the data collected during the operation of technologies and can determine the time periods in which the behaviors of the analysed variables are similar. This tool is based on fuzzy clustering. The effectiveness of the proposed algorithm is presented in the case studies based on real data sets from the polyethylene factory of the TVK Ltd. The results prove well that this tool can be applied to distinguish the typical operational periods i.e. to qualify the operation of the technology posteriorly based on the measured data.
(Keywords: data analyse, clustering , process engeneering, fuzzy models, time periods)

BEVEZETÉS

A gyorsabb és megbízhatóbb számítógépes rendszerek megjelenése forradalmasította azokat a módszereket, melyeket vegyipari technológiákban használnak, gondoljunk csak az adatnaplózó és tároló egységekre. Ezek a számítógépes rendszerek már képesek kifinomult szabályozási stratégiák véghezvitelére, szimulációra és optimalizációra annak érdekében, hogy javítsák a műveleteket és így csökkentsék a költségeket (*Doymaz*, 2001; *Kosanovich*, 1997; *Lane*, 2001). Ezek az előnyök azt is eredményezték, hogy hatalmas mennyiségű adatot grafikusan és szövegesen is megjelenítsenek, elérhetővé tegyenek az operátorok számára. Ezzel együtt az operátoroktól azt is megkövetelik, hogy rutinszerűen valósítsák meg az előírt műveleteket, optimalizálják a teljesítményt és reagáljanak az esetleges vészjelzésekre (*Huang*, 1995). A nagymennyiségű adat interpretálása és analizálása azonban rendkívül komplex feladat, mivel az operátoroknak nincs sem idejük, sem szakértelmük a folyamat monitorozásához (*Lindheim*, 1997; M*ishitani*, 1996; *Wang*, 1999).

Fontos célunk tehát, hogy ebből a hatalmas adathalmazból megfelelő információkat szerezzünk, amely mind a folyamat modellezésében, jobb megértésében, mind – akár a modellen keresztül – a folyamat irányításában is hasznos lehet (*Lakshminarayanan*, 2000; *MacGregor*, 1995). Ehhez használható és hatékony eszközt kell találnunk. Ez különösen fontos számos gyakorlati alkalmazásban, amelyekben fehérdoboz-modellek felállítása nem lehetséges a rendszer komplexitása miatt.

A komplex kérdéskörön belül a dolgozat egy, az ipari technológiák állandósult üzemeltetéséhez tartozó állapotváltozók közti kapcsolatok elemzéséhez elengedhetetlenül szükséges eszköz fejlesztésével foglalkozik. Tegyük fel, hogy egy már működő technológiai rendszerben nem ismerjük a számos állapotváltozó közül azokat, amelyek befolyásolják vagy lényegesen meghatározzák az előállított termék minőségét. Ez természetesen a szabályozásban is gondot okoz, és a rossz minőségű termék előállítása pedig költségnövekedést eredményez. Ezért végső célunk egy modell identifikálása, egy adekvát modell felállításához azonban szükség van az adatsorok azon értékeinek kiválasztására, amelyekről biztosan állíthatjuk, hogy ok-okozati kapcsolatban vannak egymással és be lehet azonosítani, hogy – konkrétan – az adott termékminőséget milyen állapotváltozó-értékek eredményezték. Ehhez a rendelkezésre álló adatsorokból meg kell állapítanunk a stacioner szakaszokat, illetve fel kell ismernünk a tranziens állapotokat.

Mivel működő technológiai rendszerekben sok mért állapotváltozó van, ezért olyan eszközre van szükség, amely lehetővé teszi több párhuzamos idősor szimultán elemzését és szegmentálását. Ennek megoldására egy számítási intelligencián alapuló fuzzy csoportosítást alkalmazó módszert javaslunk.

Legyen $x(t)$ egy időben változó állapotváltozó. Célunk ezt az idősort szakaszokra bontani, szegmentálni, vagyis megállapítani azon t_i időpontokat, amelyekre teljesül, hogy a t_i és t_{i+1} közötti időtartamban az $x(t)$ állapotváltozó jellemző homogén viselkedést mutat. Ilyen jellemző homogén viselkedésminták lehetnek az alábbiak.

- Az epizódok (*Stephanopoulos*, 1995; *Kivikunnas*, 1998), melyek a jel, az első és a második deriváltjának előjelét veszik figyelembe, vagyis azt, hogy a jel pozitív-, vagy negatív-e, csökken, nő, nem változik, illetve milyen az alakja. Ennek alapján hét primitív epizód különböztethető meg.
- Lineáris szegmensek (*Keogh*, 2001), melyben az adott szakaszt egyenessel közelítjük. Ebben az esetben a szakaszt az egyenesek paramétereivel lehet jellemezni.
- Cél lehet az abnormális viselkedésmód megállapítása és annak kezelése is (*Wong*, 1998), ebben az esetben a normális, átmeneti és abnormális állapotokat, viselkedéseket érdemes megkülönböztetni.

- A mi esetünkben az állandósult üzemeltetési állapotokat keressük, tehát a stacioner és tranziens állapotokat kell megkülönböztetnünk, illetve alapvető fontosságú még a stacioner állapotok egymástól való megkülönböztetése is.

A homogén viselkedésminták megkülönböztetése természetesen nagyban függ attól, hogy milyen célt akarunk elérni ezekkel. Célunk lehet:

- becslés (előrejelzés, illesztés) (*Keogh*, 2001; *Boekhoudt*, 1995; *Rossiter*, 1998),
- hatékony tárolás (*Keogh*, 2001),
- elemzés, kiértékelés,
- bizonyos szegmensek keresése,
- szabályok megtalálása …

A szegmentálási algoritmusok (többek között) a következő módszerekkel dolgozhatnak.

- Csúszó ablak: egy szegmenst addig növelünk, amíg pl.: egyenest illesztve a vizsgált szakaszra egy bizonyos hibahatárt át nem lépünk (a folyamatot attól a ponttól ismételjük, amelyik már nem tartozik bele a lezárt szegmensbe).
- Top-down: az idősort rekurzív partícionáljuk, amíg bizonyos leállási feltételt el nem érünk.
- Bottom-up: a lehető legfinomabb felosztással kezdünk, majd a szegmenseket egyesítjük bizonyos leállási feltétel eléréséig.
- Inflexiós pontok keresése, mely az epizód-elemzésnél fontos, mivel a primitív epizódok két inflexiós pont között helyezkednek el (*Fujiwara*, 1994).

Ezeket a módszereket sokan és sokféle módon ötvözték fuzzy logikával (*Wong*, 1998; *Last*, 2000; *Rossiter*; 1998) és fuzzy csoportosítással is (*Boekhoudt*, 1995; *Goutte*, 1998).

Az általunk vizsgált esetben rendkívül nagy mennyiségű adat elemzésére volt szükség, így igényként merült fel, hogy az algoritmus gyors és viszonylag egyszerű legyen, és kezelni tudjon párhuzamos idősorokat szimultán módon, valamint legyen robusztus. Ennek érdekében a fuzzy szubsztraktív csoportosítási algoritmust használtuk fel, mely zajjal szemben nem érzékeny, mivel fuzzy logikát alkalmaz, és képes a csoportok számát automatikusan meghatározni.

Ennek a munkának az alapgondolata az volt, hogy a vizsgálandó idősor adott pontja helyett a pontot a környezetével együtt elemezzük, majd ezek között a transzformált változók között keresünk hasonlóságot, melyhez az említett csoportosítási algoritmust alkalmaztuk. A módszer tehát paraméterként tartalmazza annak a tartománynak a méretét, amelyet a ponttal együtt vizsgálunk, illetve azokat a jellemzőket, amelyek a környezet jellemzésére szolgálnak, indikálják tehát azt, hogy a pont milyen tulajdonságú más pontokkal van körülvéve. Az általunk kifejlesztett eszközt valós ipari adatokon fogjuk bemutatni a rendelkezésünkre bocsátott nagysűrűségű polietilén-gyár (TVK Rt.) mért állapotváltozói alapján.

E fuzzy technikák folyamatmérnöki problémák megoldásában történő alkalmazhatóságát már számos rendszeridentifikációval kapcsolatos munkánk igazolja (*Feil*, 2001; *Abonyi*, 2002; *Abonyi*, 2003).

IDŐSOROK SZEGMENTÁLÁSA CSOPORTOSÍTÁSI ALGORITMUSSAL

A probléma megfogalmazása

Célunk n párhuzamos idősor vizsgálata és ezek szegmentálása, vagyis olyan szakaszokra való bontása, melyekben hasonló tulajdonságú pontok helyezkednek el. Adott tehát x_{ki} ($k=1,...,n$), ($i=1,...,n$) állapotváltozó $N \cdot \Delta t$ időtartam feletti értéke, ahol Δt a

mintavételezési idő. Az N megfigyelés halmazát jelöljük **X**-szel és ezeket egy $N \times n$-es mátrixban tároljuk:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \Lambda & x_{1n} \\ x_{21} & x_{22} & \Lambda & x_{2n} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ x_{N1} & x_{N2} & \Lambda & x_{Nn} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \mathrm{M} \\ \mathbf{x}_N^T \end{bmatrix}, \tag{1}$$

melyben tehát egy-egy sor egy-egy mintának, egy-egy oszlop pedig egy-egy idősornak felel meg. A szegmentálás célja, hogy az **X** adatmátrixot úgy „tördeljük szét", hogy az egyes egységekben az állapotváltozók egymástól eltérő jellegű időbeli viselkedést mutassanak:

$$\mathbf{X} = \left[\begin{array}{ccc} x_{11} & \Lambda & x_{1n} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ x_{i1} & \Lambda & x_{in} \\ \hline \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \hline \mathrm{M} & \mathrm{O} & \mathrm{M} \\ x_{N1} & \Lambda & x_{Nn} \end{array}\right]. \tag{2}$$

Egy-egy szaggatott vonal közötti szakaszban hasonló tulajdonságú pontok találhatók, például az egyik egységben az állapotváltozó növekszik, míg egy másikban nem változik. Jelöljük **b**-vel azt a vektort, mely a töréspontokat tartalmazza:

$$\mathbf{b}=[b_1\ b_2\ \ldots\ b_{N_b}]^T, \tag{3}$$

ahol $b_1=1$, $b_{N_b}=N$ és a b_i az i-edik szegmens kezdőpontját tartalmazza. A technológia állandósult állapotainak elemzéséhez, a végső célunk ezen szegmentálás eredményéből a **b** vektor azon értékeinek megtalálása, melyek között az állapotváltozók nem vagy csak megengedhetően kis mértékben változnak. E feladat megoldására a következő fejezetben bemutatott algoritmust fejlesztettük ki.

Az algoritmus leírása

Mivel az állapotváltozók dinamikus viselkedését elemezzük, ezért nem azok konkrét időpontban vett értéke az algoritmus alapja, hanem egy adott időhorizonton detektált változása. Ezt az időhorizontot a következőkben ablaknak fogjuk nevezni. Ennek méretét állandó értéknek vettük, hasonlóan az irodalomban bemutatott algoritmushoz (*Wong*, 1998). A szegmentálás első lépésében kiválasztjuk azokat a módszereket, amelyek az állapotváltozó környezetét indikálják. A változók vizsgált ablakban történő változása alapján fogjuk az elemzett egységek között levő hasonlóságot megkeresni. E dinamikus viselkedést transzformált változókkal írjuk le, melyek vektorát jelöljük a következő módon:

$$\mathbf{z}_{kj} = f(x_{ki},\ldots,x_{(k+N_W)i}), \tag{4}$$

amely az k-adik mintára $\mathbf{x}_k$, vonatkozó j-edik indikátor sorvektorát jelenti, az N_W pedig ablak méretét jelöli. Általában ezek skalárok, pl.: az átlag, a szórás, de lehetnek pl.: egy polinom paraméterei is, melyek esetén már vektort kapunk. Ezen indikátorokat bővebben a következő alfejezetben tárgyaljuk.

A **Z** mátrixban tároljuk ezeket az értékeket:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_{11} & \mathbf{z}_{12} & \Lambda & \mathbf{z}_{1n_Z} \\ \mathbf{z}_{21} & \mathbf{z}_{22} & \Lambda & \mathbf{z}_{2n_Z} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ \mathbf{z}_{N_Z 1} & \mathbf{z}_{N_Z 2} & \Lambda & \mathbf{z}_{N_Z n_Z} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \mathrm{M} \\ \mathbf{z}_{N_Z}^T \end{bmatrix}. \tag{5}$$

Mivel a megkövetelt ablakméret miatt bizonyos pontokat nem tudunk transzformálni, másrészt elegendően nagy ablakméret esetén a számítási igény csökkentése végett nem feltétlenül kell minden egymást követő adatpontot transzformálni, hanem bizonyos számút „át is ugorhatunk", ezért N_Z jóval kisebb lehet N-nél. Az n_Z dimenzió nagysága a választott indikátor változók számától és minőségétől függ.

Az algoritmus következő lépésében a **Z**-ben tárolt egyes ablakokat minősítő változók közötti hasonlóságot keressük meg fuzzy csoportosítási algoritmus segítségével. A csoportosítás során az adatokat olyan halmazokba soroljuk, melyekben hasonló tulajdonságú pontok helyezkednek el. A hasonlóság alapja általában valamilyen típusú távolságmérés. Ezen halmazok azon pontját, mely leginkább jellemző az adott halmazra, nevezzük csoportközéppontnak, prototípusnak. Az általunk választott algoritmus automatikusan határozza meg a csoportok számát (N_C), illetve az egyes csoportok középpontjait is ($\mathbf{z}^*_j$, j=1, ..., N_C), melyeket a meglevő adatpontokból választ ki.

A csoportközéppontok meghatározása után minden adatpontot besorolunk valamelyik csoportba. Ehhez a $\mathbf{z}_i$ pont adott $\mathbf{z}^*_j$ csoportközépponttól való távolságát vesszük alapul, és a következő függvényt definiáljuk, *Boekhoudt*, 1995-höz hasonlóan:

$$A_j(\mathbf{z}_i) = \exp\left(-\frac{\left\| \mathbf{z}_j^* - \mathbf{z}_i \right\|}{r_A^2} \right), \tag{6}$$

mely kifejezés tehát megadja a $\mathbf{z}_i$ pont j-edik csoportba tartozásának mértékét (az $\boldsymbol{r}_a$ értéke az algoritmus egyik paramétere lesz, értéke összefügg a csoportosítási algoritmussal is. Ezt követően meg kell állapítanunk, hogy a $\mathbf{z}_i$ pont melyik csoportba tartozik. Az adott pontot abba a csoportba fogjuk sorolni, melyre a fenti $A_j(\mathbf{z}_i)$ értéke a legnagyobb (itt tudatosan hibát vétünk, hiszen csak a csoportközéppontok tartoznak 100%-os súllyal egy bizonyos csoporthoz). Az eredményt a **h** vektorban tároljuk, melynek elemeit a következőképpen számoljuk:

$$h_i = \arg_j \max A_j(\mathbf{z}_i). \tag{7}$$

E lépésre azért van szükség, mert egy adott csoportba tartozó egységek nem feltétlen követik egymást az időben, hanem néha több különálló hosszabb-rövidebb azonos jellegű szakaszokat definiálnak. Ezt az információt a (3) egyenlettel megadott **b** vektor foglalja magába, melynek elemeit ebben a lépésben azonosítjuk be:

$$b_k = i \,|\, i : h_i \neq h_{i+1}. \tag{8}$$

Miután a transzformált változók tartományában elvégeztük az adatpontok csoportosítását, az eredeti idősorainkban is beazonosíthatjuk az egyes tartományokat. Az így megállapított időtartományokba eső adatok kiválasztott indikátorainak értékeit újraszámoljuk, és ezekkel az értékekkel fogjuk jellemezni az adott szegmenst.

A szubsztraktív csoportosítás

A csoportosításhoz az adatokat normálni kell úgy, hogy az adatok egy egységnyi hiperkockában helyezkedjenek el.

A következő lépésben alkotjuk meg az M potenciálfüggvényt:

$$M^{(1)}(\mathbf{z}_i) = \sum_{k=1}^{N_z} \exp(-\alpha d(\mathbf{z}_i, \mathbf{z}_k)), \tag{9}$$

ahol α egy pozitív konstans, $d(\mathbf{z}_i, \mathbf{z}_k)$ a lehetséges csoportközéppont $\mathbf{z}_i$ és a $\mathbf{z}_k$ adatpont távolsága és M indexe az iteratív lépés első elemére utal. Az adatpontok távolságát leggyakrabban az euklideszi értelembe vett távolságként definiálják:

$$d(\mathbf{z}_i, \mathbf{z}_k) = \sqrt{(\mathbf{z}_i - \mathbf{z}_k)^T (\mathbf{z}_i - \mathbf{z}_k)} \tag{10}$$

A fentiek alapján a potenciálfüggvény annak a valószínűségét fejezi ki, hogy az aktuális pont milyen mértékben lehet csoportközéppont. Minél közelebb van egy adatpont az adatok egy csoportjához, e csoport annál nagyobb mértékben járul hozzá ezen *i*-edik pont $M(\mathbf{z}_i)$ potenciáljához. Ez annyit jelent, hogy minél nagyobb az $M(\mathbf{z}_i)$ értéke, annál valószínűbb, hogy a $\mathbf{z}_i$ pont csoportközéppont. Az α paraméter értékét általában a következőképpen adják meg: $\alpha = 4/r_a^2$, ahol r_a a csoportközéppont befolyásának a mértéke.

A csoportosítási algoritmus utolsó lépésében felhasználjuk a potenciálfüggvényt a csoportközéppontok generálásához. Legyen $\mathbf{z}^*_1$ az első csoportközéppont, melynél a potenciálfüggvény a legnagyobb: $M^*_1 = \max_i M^{(1)}(\mathbf{z}_i) = M^{(1)}(\mathbf{z}^*_1)$. Ahhoz, hogy a sorrendben a következő csoportközéppontot megtaláljuk, minden lépésben eliminálnunk kell az éppen meghatározott csoportközéppont hatását, mivel ezt a pontot általában ugyancsak nagy potenciállal rendelkező pontok veszik körül. Ennek érdekében módosítanunk kell a korábbi potenciálfüggvényt. Ez azt követeli meg, hogy minden pont potenciáljából vonjunk le egy, az éppen meghatározott csoportközépponttól való távolsággal fordítottan arányos értéket, valamint ez az érték legyen arányos az éppen identifikált csoportközéppont potenciáljával. Ennélfogva a lehetséges csoportközéppontok potenciálját kifejező módosított $M^{(j+1)}$ függvényt a következőképpen fejezzük ki:

$$M^{(j+1)}(\mathbf{z}_i) = M^{(j)}(\mathbf{z}_i) - M^*_j \exp(-\beta d(\mathbf{z}^*_j, \mathbf{z}_i)), \tag{11}$$

ahol β egy pozitív konstans, a $\mathbf{z}^*_j$ pedig a *j*-edik iterációs lépésben meghatározott csoportközéppontot jelöli. A β paraméter értékét általában $\beta = 4/r_b^2$ -ként adják meg, ahol r_b-t szomszédossági rádiusznak is nevezhetjük. Elkerülendő, hogy két csoportközéppont túl közel kerülhessen egymáshoz, r_b értékét valamivel nagyobbra választjuk r_a-nál. A (*Boekhoudt*, 1995) szerint jó választás az $r_b = \frac{3}{2} r_a$.

Fontos, hogy mivel $\mathbf{z}^*_j$ volt az előbbiekben kiválasztott csoportközéppont és M^*_j a potenciálja, ezért azt kapjuk, hogy $M^{(j+1)}\mathbf{z}^*_j = 0$. E korrigált potenciálfüggvényt használjuk fel a következő csoportközéppont megtalálására pontosan azon elv szerint, ahogy azt az eredeti potenciálfüggvénynél tettük.

Az előzőekben vázolt folyamatot iteratívan folytathatjuk, így a teljes algoritmus a következőképpen néz ki:

0. Inicializálás: normálás.
1. Megtalálni $M^*_j = \max_i M^{(j)}(\mathbf{z}_i) = M^{(j)}(\mathbf{z}^*_j)$.
2. Ha $M^*_j < \varepsilon M^*_1$ teljesül, akkor megállítjuk a keresést.
3. Amennyiben az előző feltétel nem teljesül, akkor elfogadjuk $\mathbf{z}^*_j$-ot csoportközéppontnak.
4. Korrigáljuk az $M^{(j)}(\mathbf{z}^*_i)$ potenciálfüggvényt:

$$M^{(j+1)}(\mathbf{z}_i) = M^{(j)}(\mathbf{z}_i) - M^*_j \exp(-\beta d(\mathbf{z}^*_j, \mathbf{z}_i)), \tag{12}$$

és vissza 1-re.

Világos, hogy ez az iteratív folyamat véges sok lépésből áll és a 2. lépésben áll le, miután a N_c-edik csoportközéppontot meghatároztuk, melyre igaz az, hogy $\frac{M^*_{N_C+1}}{M^*_1}$ kisebb lesz, mint az előre definiált ε tolerancia.

Az általunk alkalmazott algoritmus a fenn ismertetettől annyiban tér el, hogy pontosabban figyelembe veszi a csoportközéppontok közötti távolságot. Az eredeti algoritmusban ez a szándék abban jutott kifejezésre, hogy a potenciálfüggvényt módosítottuk minden iterációs lépésben úgy, hogy két csoportközéppont ne kerülhessen túl közel egymáshoz. Ezen felül az általunk alkalmazott algoritmus az α, r_b/r_a és ε paramétereken túl egy harmadikat is bevezet, jelöljük ezt δ-val. Míg az ε-t elfogadhatósági aránynak nevezhetjük, addig δ-t elutasítási aránynak. A módosított algoritmus úgy fog működni, hogy amennyiben az $M^*_j \geq \varepsilon M^*_1$ fennáll, akkor az adott $\mathbf{z}^*_j$ pontot fenntartások nélkül elfogadjuk csoportközéppontnak, majd folytatjuk az algoritmust az eredeti lépések szerint. Azonban ha $M^*_j < \varepsilon M^*_1$, de $M^*_j \geq \delta M^*_1$, akkor további vizsgálatokat végzünk. Megkeressük az éppen vizsgált $\mathbf{z}^*_j$ ponthoz euklideszi értelemben vett legközelebbi már meglevő csoportközéppontot, jelöljük ezt $\mathbf{z}^*_k$-gal, majd a következő távolságtényezőt számoljuk:

$$dist = \frac{d(\mathbf{z}^*_j, \mathbf{z}^*_k)}{r_a} \cdot \tag{13}$$

Ezután a következő kritériumot vizsgáljuk:

$$\frac{M^*_j}{M^*_1} + dist \geq 1 \cdot \tag{14}$$

Amennyiben ez a feltétel teljesül, akkor elfogadjuk a vizsgált pontot csoportközéppontként, mivel elegendően távol van a legközelebbi, már meglevő csoportközépponttól is. Ezt követően a potenciálfüggvényt az eredeti módon módosítjuk. Amennyiben ez a kritérium sem teljesül, akkor a pontot nem fogadjuk el csoportközéppontnak, mivel egy, már meglevő csoportközéppont vonzáskörzetébe tartozik. Ekkor a potenciálfüggvényt úgy módosítjuk, hogy minden pont megtartja az eredeti potenciálját, csak az ebben a ciklusban elutasított pont potenciálját változtatjuk meg, nullával tesszük egyenlővé, hiszen vizsgálataink szerint ez már nem lehet csoportközéppont: $M^{(j+1)}(\mathbf{z}_i) = M^{(j)}(\mathbf{z}_i)$, $i = 1, \ldots, N_z$ kivéve $M^{(j+1)}(\mathbf{z}^*_i) = 0$. Az algoritmus akkor áll le, ha az aktuális lépésben a legnagyobb potenciállal rendelkező $\mathbf{z}^*_j$ pont már a $M^*_j \geq \delta M^*_1$ feltételt sem tudja teljesíteni.

A csoportközéppontokat és az r_a értékét a tagsági függvény karakterisztikájának első becsléseként is használhatjuk valamilyen finomhangolás kiindulási lépésében (*Boekhoudt*, 1995), ha modell-identifikáció a célunk.

Információs kritériumok: a változók transzformálása

Az előző fejezet mutatta be, hogy egy adott egységet minősítő változókat hogyan lehet csoportosítani. Az egész dolgozat gondolatának zárttá tételéhez már csak azokat a módszereket kell ismertetni, melyek a vizsgálandó pontot a környezetével együtt kezelik, tehát a megfigyelt értéken túl figyelembe veszik az állapotváltozó környezetét is. Erre egy jó eszköz, ha az adatpont helyett a pontot és a környezetét is jellemző változókat vizsgáljuk. Azt a tartományt, melyet figyelembe veszünk egy-egy pont értékelésénél, nevezzük ablaknak. Az ablak mérete, hosszúsága is fontos jellemző, hiszen ha egységnyi hosszúságú, tehát csak az eredeti adatpontot tartalmazza, akkor tökéletesen homogénnek

tekinthető, tehát nem nyújt semmi plusz információt. Ha túl nagy, akkor az növeli a számítási igényt, illetve az adatsor túl hosszú szakaszát sűríti magába, mely információvesztéshez vezethet.

Az ablakban levő változók többfajta jellemzőjét is felhasználhatjuk a pont jellemzésére. Az idősorok hasonló tendenciájú (stagnáló, emelkedő, rohamosan emelkedő …) részei más-más állapotváltozó-értékeknél is előfordulhatnak, így alapvető fontosságú az ablak átlagának figyelembevétele, mely az *i*-edik változóra N_W méretű ablakkal a következőképpen néz ki:

$$mean_{ki} = \frac{\sum_{j=k}^{k+N_W} x_{ji}}{N_W}. \tag{15}$$

Fontos információt hordoz az átlagtól való eltérés mértéke is, melyet az ablakban levő változók szórásaként vehetünk figyelembe:

$$std_{ki} = \sqrt{\frac{\sum_{j=k}^{k+N_W} \left(x_{ji} - mean_{ji}\right)^2}{N_W}}. \tag{16}$$

A szóráson kívül az ablak jellegéről további információkat hordoz a trendnek nevezett mennyiség is, mely két különböző időállandójú exponenciális szűrő által nyújtott eredmény különbségeként formulázható, és zárt alakban a következőképpen néz ki (*Gertler*, 1988):

$$trend_{ki} = (1-a_2)\sum_{j=k-N_W}^{k} a_2^{j-k} x_{(k-j)i} - (1-a_1)\sum_{j=k-N_W}^{k} a_1^{j-k} x_{(k-j)i}. \tag{17}$$

Ennek a mennyiségnek két paramétere van: a két exponenciális szűrő időállandója. A következő formulák az általunk vizsgált esetekben jól működtek:

$$a_1 = \frac{N_W - 1}{N_W + 1}, \tag{18}$$

$$a_2 = \frac{2N_W - 1}{2N_W + 1}. \tag{19}$$

Egy következő lehetőség az úgynevezett polinomiális indikátorok felhasználása. Ennek lényege, hogy az ablakban levő idősorra polinomot illesztünk, majd a polinom paramétereit használjuk az ablak jellemzésére. Hasonló ötletre épít (*Stephanopoulos*, 1995) az esemény-szemléletmódnak nevezett módszer, mely hét primitív epizódot különböztet meg. Ezek a függvény előjeléből, valamint első és második deriváltjából állnak. Ez annyit jelent, hogy a primitív epizódok információt hordoznak arról, hogy a függvény pozitív vagy negatív-e, csökken, nő, nem változik, illetve az alakjáról, konvexitásáról. Ezek az első és/vagy a második deriváltjukban különböznek egymástól, így a kitüntetett pontok közöttük helyezkednek el. Ezek az információk elégségesek a mi esetünkben is, ezért az illesztett polinom rendjét kettőnek választottuk. Mivel az ablak átlagát már figyelembe vettük, ezért a polinomban szereplő konstans tagot a csoportosításnál nem vesszük figyelembe. Meg kell jegyezni, hogy az eredeti módszer zajjal terhelt adatok esetén gyengén teljesít, másrészt a mi idősorunkat akkor tudnánk primitív szakaszokra felbontani, ha az ablak méretét automatikusan határoznánk meg. Nem így dolgozunk ugyan, azonban az illesztett polinomok paraméterei használható információforrásnak tűnnek.

A szegmentálás során a csoportosítási algoritmus paraméterei és az ablak mérete természetesen azonos volt az összehasonlíthatóság végett. Az átlag és a szórás alapján való szegmentálás viszonylag kevés szakaszra bontotta az idősorokat (5 csoport és 9 szakasz, *1. ábra*), ezt a számot a trend figyelembevétele sem tudta jelentősen módosítani (8 csoport és 13 szakasz, *2. ábra*). Az átlag és a szórás mellett a polinomiális indikátorok figyelembevétele azonban jelentős hatást gyakorolt a csoportok (12) és a szakaszok számára (20) is (*3. ábra*). Ezt az információtöbbletet ezeken kívül a trend számításbavétele sem tudta növelni: a négy indikátor esetén az algoritmus 13 csoportot és 22 szakaszt talált (*4. ábra*). Ezek alapján azt a következtetést vonhatjuk le, hogy az átlag és a szórás mellett érdemes figyelembe venni a polinomiális indikátorokat is, de ezeken kívül a trenddel való kiegészítés már feleslegesnek tűnik.

Az *1., 2. ábra* az indikátorok hatását, információtartalmát szemlélteti a TVK Rt. polietilént előállító reaktorán mért öt állapotváltozó adatai alapján. (Az egyes változók értékei titkosak, ezért a diagramokon a minimális értéket 0-val, a maximálist 1-gyel jelöltük.)

1. ábra

Az átlag és szórás alapján való szegmentálás

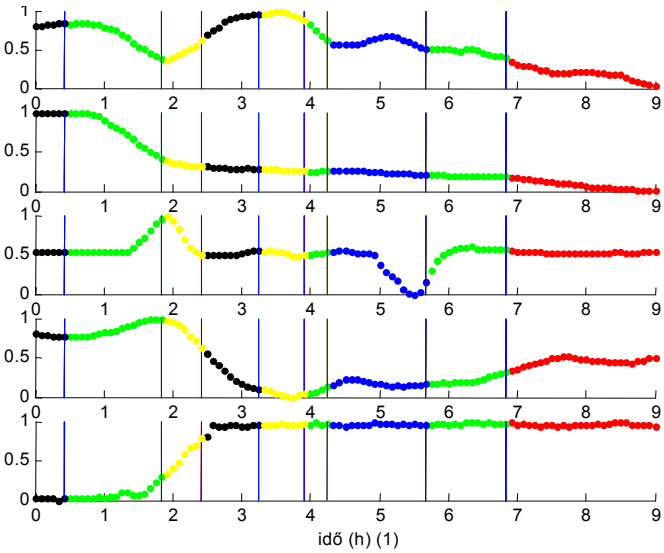

Figure 1: The results of the segmentation based on the mean and the standard deviation

Time (hour)(1)

A szegmentálás során a csoportosítási algoritmus paraméterei és az ablak mérete természetesen azonos volt az összehasonlíthatóság miatt. Az információs kritériumok vizsgálata során arra a következtetésre jutottunk, hogy az átlag és a szórás mellett érdemes figyelembe venni a polinomiális indikátorokat is, de ezeken kívül a trenddel való kiegészítés már feleslegesnek tűnik.

2. ábra

Az átlag, a szórás és a trend figyelembevétele

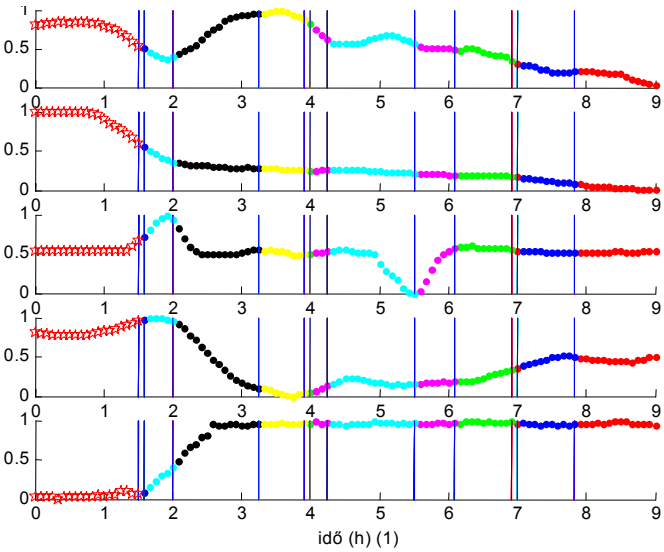

Figure 2: The segmentation based on the mean, the standard deviation and the trend

Time (hour)(1)

ESETTANULMÁNY

A technológia bemutatása

A TVK Rt.-nél a közepes-, illetve nagysűrűségű polietilént (MDPE, HDPE) a Phillips Petroleum Co. folyamatos zagyfázisú polimerizációs technológiája alapján állítják elő (*Kalafszki*, 1999; *Brandrup*, 1975; *Redman*, 1991). A reakció 40-42 bar nyomáson, 85-110°C-on játszódik le egy hurokreaktorban (*Ayres*, 1986; *Meketta*, 1992). A különböző termékekhez használt krómtartalmú, vagy fémorganikus titán-magnézium tartalmú katalizátort termikus aktiválás után az izobután oldószerben szuszpendáltatva vezetik a reaktorba. Az etilén, a komonomernek használt hexén, a hidrogén és az izobután a katalizátormérgektől való tisztítás után kerül a reaktorba.

Mivel a HDPE olvadáspontja 135°C körül van, a polimer szilárd fázisban keletkezik. A zagy folyamatos cirkulációját a reaktorba beépített zagykeringető szivattyú biztosítja. A hurokreaktor négy hosszú, egy méter átmérőjű függőleges csőszakaszból áll, melyeket rövid (5 m) vízszintes szakaszok kötnek össze. A cirkuláció gyors, 10-12 m/s sebességű azért, hogy a zagy falra való kiülepedését és ezzel a reaktor bedugulását megakadályozzák. Az inert oldószer a hő disszipálására szolgál, mivel a reakció nagyban exoterm. A reaktor hőmérsékletének pontos beállítását köpenyoldali hűtéssel, illetve fűtéssel szabályozzák. A polimer koncentráció a zagyban 25-35 tömeg%, a termék kinyerésére szolgáló ülepítő lábakban 60-70 tömeg% körüli. Az oldószert flash tartályban választják el a polimertermektől, a desztillációval visszanyert tiszta izobutánt és hexént visszavezetik a reaktorba. A polietilénport szárítás után pneumatikus úton portároló silókba szállítják, majd extrudálják.

3. ábra

A polimerizációs technológia sémája

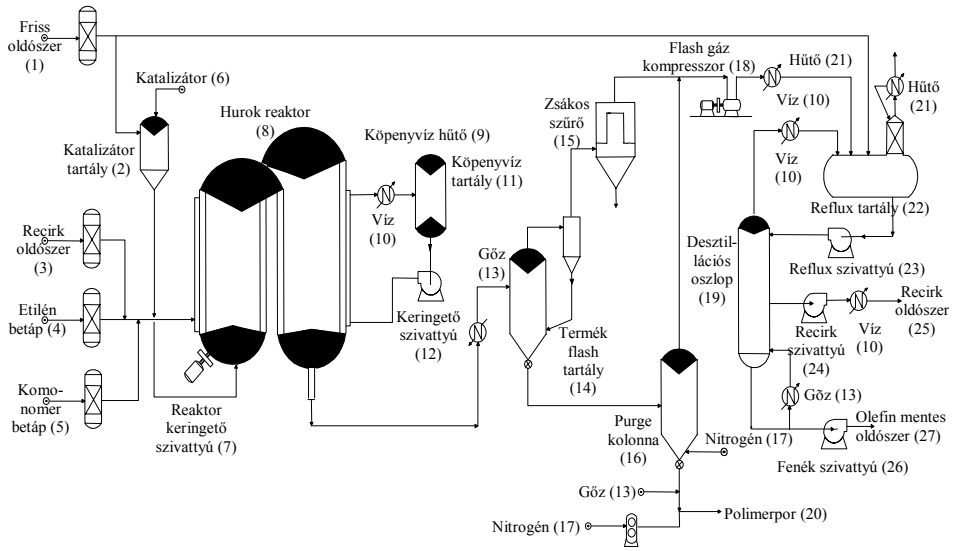

Figure 3: Scheme of the Phillips loop reactor process

Fresh solvent(1), Catalyst tank(2), Recycling solvent(3), Ethylene feed(4), Comonomer feed(5), Catalyst(6), Reactor circulating pump(7), Loop reactor(8), Jacket water cooler(9), Water(10), Jacket water tank(11), Circulating pump(12), Steam(13), Productflash tank(14), Bag filter(15), Purge column(16), Nitrogen(17), Flash gas compressor(18), Distillation column(19), Polymer powder(20), Cooler(21), Reflux tank(22), Reflux pump(23), Recycling pump(24), Recycling solvent(25), Bottom pump(26), Olefin free solvent(27)

A hidrogén a termékek molekulatömegének, a hexén komonomer a sűrűségének szabályozását biztosítja. A HDPE átlagos molekulatömegét a katalizátoraktiválás hőmérséklete szabja meg. Az etilén konverziója igen nagy (95-98%), ezért nincs szükség az etilén visszanyerésre.

A polimer termékeket a sűrűséggel és a folyásindexszel lehet jellemezni. A dolgozatban a folyásindexek (Melt Index - MI) értékeit használtuk fel a termékminőség jellemzésére, melyet kétóránként az üzemhez tartozó laboratóriumban mérnek. Annak érdekében, hogy a reaktor állapotát jellemző változókhoz tartozó MI-mérések értékeit válogassuk ki, figyelembe kell venni a reaktor után elhelyezkedő egységeket és a hozzájuk tartozó átlagos tartózkodási időket is. A folyásindexet háromféle súllyal mérik attól függően, hogy az előállított termék milyen tulajdonságú (MFI, MLMI, HLMI). A továbbiakban csak egyfajta súllyal mért folyásindexet használhatunk fel, mivel ezek az értékek nem összehasonlíthatók, és annak érdekében, hogy megfelelő számú adat álljon rendelkezésünkre, azokat az MI-méréseket választottuk ki, melyekből a legtöbbet mérték (MFI). (Kézenfekvő megoldás lett volna, ha a különböző súllyal mért MI-értékeket át tudjuk számolni egy adott súlyra, azonban megfelelő pontosságú összefüggés nem áll a

rendelkezésünkre, melyet a laboreredmények is alátámasztanak.) A rendelkezésünkre álló adatbázisban 120 gyártás adatsora található meg, melyekből 10 állapotváltozó időbeli értékeinek alakulását használtuk fel a további vizsgálatokhoz. A változók között vannak mért, illetve a folyamatirányító számítógép által számított adatok is. A folyamatirányító rendszer 15 sec-os mintavételezési idővel dolgozik. A vizsgált változók a következők: a folyamatirányító rendszer által számított etilén-, hexén- és hidrogénkoncentráció, a zagysűrűség, a reaktorban uralkodó hőmérséklet, a polietilén-elvétel, a hexén-, az etilén-, a hidrogén- és az összes olefin-mentesített izobután-betáplálás.

A szegmentálás bemutatása

A *4. ábra* mutatja egy adott gyártás tíz állapotváltozójának időbeli változását, illetve a szegmentálás eredményét.

4. ábra

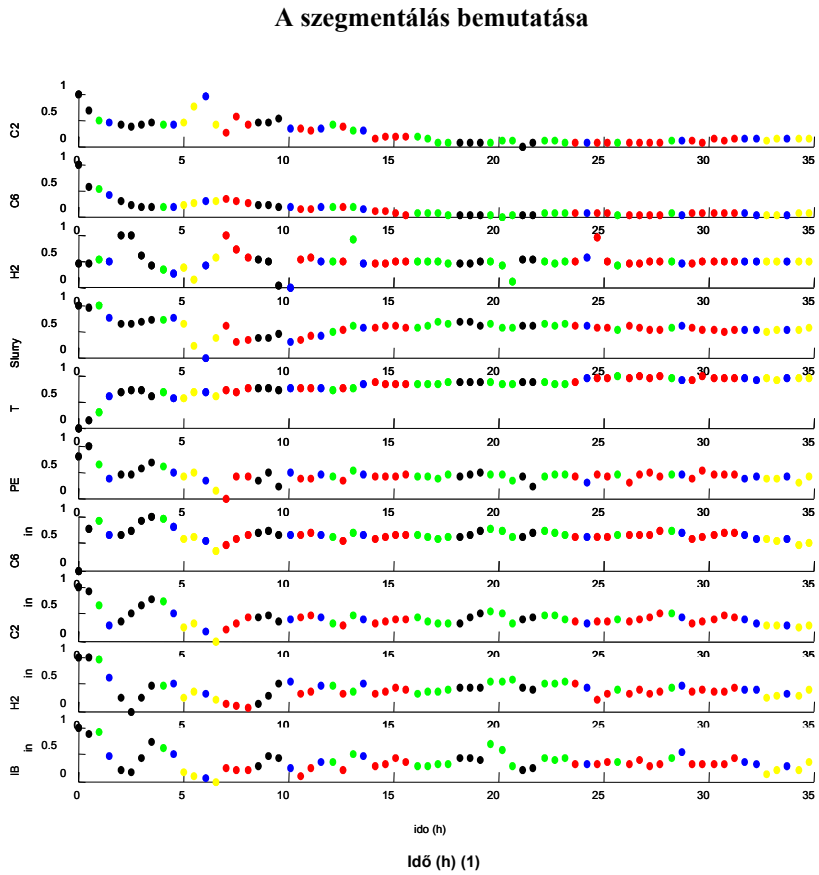

Figure 4: The results of the segmentation

Time (hour)(1)

Megállapíthatjuk, hogy egy gyártás sem tökéletesen homogén, hanem szakaszokra osztható, szegmentálható. Mivel a reaktor folyamatos üzemű, ezért egy adott termék gyártásánál az üzemeltetők arra törekednek, hogy olyan állandósult állapotot tartsanak, amely a gyártáshoz a leginkább megfelelő, azonban láthatóan dinamikus szegmenseket is találunk. Ennek oka zavarásokban (pl.: katalizátortartályok közötti váltás), szabályozási pontatlanságokban és egy termékről más termékre való átállásban keresendő. Az algoritmus 5 csoportot és 38 szakaszt talált. A szakaszok száma azért ilyen nagy, mert – a diagramról is láthatóan – viszonylag dinamikus változásoknál már az egymást követő adatpontok is más-más csoportba tartoznak. A dinamikus elemek elemzése is fontos információt nyújthat a reaktorban zajló folyamatokról (pl.: termék átállási stratégiák elemzése és fejlesztése). A szegmens stacioner vagy átmeneti, változó voltáról, dinamikus jellegéről a szakasz szórása informál. Minél dinamikusabb az adott szegmens, annál nagyobb a szórása. Annak érdekében, hogy a szegmenst egy szórással jellemezzük és ne annyival, ahány állapotváltozónk van, összehasonlíthatóvá kellett tennünk a különböző változók szórásait. Ennek érdekében az adott változó szórását a csoportátlaggal osztottuk el, majd ezeket az értékeket átlagoltuk. A *5. ábra* az adatbázisban található összes, megfelelő MI-mérésekkel rendelkező gyártások szegmentálását foglalja össze: a szegmensek szórásainak gyakoriságát, eloszlását ábrázolja.

5. ábra

A szegmensek szórásainak eloszlása

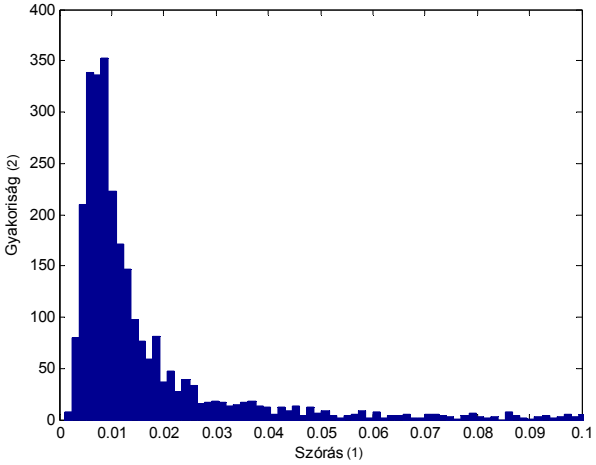

Figure 5: The distribution of the standard deviation of the segments

Standard deviation(1), Rate of occurence(2)

A szórások gyakoriságát vizsgálva megállapítható, hogy dinamikus, nagy szórással rendelkező szakaszok viszonylag ritkán fordulnak elő. Ennek oka az, hogy ezek elsősorban a termékek közötti váltásokhoz kapcsolódnak, és a váltások (néhány órás) időtartama a gyártások (több napos) időtartamához viszonyítva kicsi. A fentihez hasonló típusú diagram egy adott gyártás, illetve egy gyártási időszak minősítésére is használható.

Stacioner összefüggések elemzéséhez alkalmas adatok nyerése

A folyamat stacioner modellje alkalmas arra, hogy beállt állapot esetén az állapotváltozók mért, illetve számított értékei alapján becsüljük meg a termékminőséget jellemző változókat (vizsgálatunkban a folyásindexet). A modell alkalmazható arra is, hogy egy kívánt minőségű termék előállításához behatároljuk az üzemeltetési paramétereket. Mivel a modell stacioner, nem alkalmas a termékek közötti váltás és egyéb dinamikus folyamatok modellezésére, azonban a szegmentálási algoritmust – a változásokat tartalmazó szakaszokat felhasználva – a dinamikus modell felállításához is használhatjuk. A stacioner modellhez természetesen összetartozó stacioner értékek kigyűjtésére van szükség. A bemutatott algoritmust arra használtuk, hogy a termékminőséget befolyásoló állapotváltozók idősorait szegmentáljuk, majd a megfelelően beállt állapotot tartalmazó szegmenseket jellemző értékeket meghatározzuk. Ezeket az értékeket az állapotváltozók átlagai tartalmazzák, melyeket a szegmentálást követő lépésben kell újraszámolni az előző alfejezetben említett szórással együtt. Elképzelhető, hogy egy adott szegmenshez nem tartozik folyásindex-mérés, mivel az állapotváltozókat 15 sec-onként mérik, illetve számolják, míg a folyásindexet kétóránként laboratóriumi körülmények között vizsgálják. A vizsgálathoz azokat a szegmenseket válogattuk ki, melyekhez – a reaktort követő berendezéseken való áthaladási időt is figyelembe véve – tartozik folyásindex-mérés.

A stacioner szegmensek kigyűjtésével az a célunk, hogy adekvát stacioner modellt állíthassunk fel. A kiválogatással kapcsolatban azonban két további hatást is mérlegelni kell: pontos adatok gyűjtéséhez csak a legjobban beállt szakaszokat válogathatjuk ki (melyekben a változók értékei csak elhanyagolható mértékben változnak), azonban ezek kis száma miatt a stacioner modell pontossága leromlik, mivel elképzelhető, hogy ezek nem fedik le a teljes működési tartományt, így a modellünk – megfelelő adatok hiánya következtében – „tévedni fog". Nem szabad tehát túl kevés adattal dolgozni, azonban az adatok számát nem növelhetjük nagymértékben sem, mivel ekkor túl sok, dinamikus elemet is tartalmazó, nagyobb szórással rendelkező szakasz adatait is fel kellene használnunk, melyek értelemszerűen kevésbé alkalmasak stacioner modell felállításához. Az előzőek alapján fontos megfelelő küszöbértéket alkalmazni, melynél kisebb szórású szegmenseket felhasználunk a modell felállítása során, a nagyobbakat pedig elhagyjuk. Ezen megfontolások alapján a küszöbérték függvényében minimumot várunk a modellhibára.

A küszöbérték kiválasztása a modell jóságán, azaz a modell hibán kell, hogy alapuljon. A bemutatott szegmentálási eszköz nem korlátozza a lehetséges modellek halmazát. Illusztrációként a dolgozatban e modell-identifikáció céljára az alkalmazott fuzzy csoportosításhoz hasonlóan szintén egy számítási intelligencián alapuló eszközt használtunk fel.

A számtalan lehetséges modell közül (*Pal*, 1999) az önszervező háló (Self-Organizing Map, SOM) sokféle célra is rendkívül jól használható. A SOM algoritmus alkalmas arra, hogy sokdimenziós adatokat kétdimenziós, neuronokból álló hálóra képezzen le úgy, hogy az adatpontok közötti relatív távolság megmaradjon. A háló közelíti az adatok sűrűségfüggvényét, így ez az eszköz csoportosításra is felhasználható (*Kohonen*, 1990). Megvan a képessége az általánosításra is, azaz a hálózat interpolálni tud a bemenetek között. Mivel a SOM egy speciális csoportosítási eszköz, mely az adatok eloszlásának kompakt reprezentációját is képes nyújtani, ezért széles körben használatos sokdimenziós adatok vizualizációjára is (*Kohonen*, 1990). A SOM elősegíti a folyamatok vizualizáción alapuló megértését, így különböző változók és ezek kapcsolatai is szimultán vizsgálhatók. Például *Kassalin* a SOM algoritmust egy

transzformátor állapotának monitorozására használta, mely jelezte, ha a folyamat nem kívánt állapotba került, mely a térkép ismeretlen területének felelt meg (*Kassalin*, 1992). *Tryba és Goser*, 1991 egy desztillációs folyamat vizsgálatához használta a SOM algoritmust és ezzel is bizonyította az eszköz vegyiparban való használhatóságát. *Alander*, 1991 és *Harris*, 1993 a SOM algoritmust hibadetektálásra alkalmazta. Mivel a modellt normál állapotokhoz tartozó mérési vektorokkal alkották meg, ezért egy hibás állapotot a kvantálási hiba (a bemeneti vektor és a legjobban illeszkedő egység közötti távolság) megfigyelésén keresztül lehetett detektálni, mivel a nagy mértékű hiba jelezte azt, hogy a folyamat a normál működési tartományt elhagyta. A SOM algoritmus regresszióra is felhasználható, amelyben a háló részei a tér lokális lineáris modelljét adják meg. Ezt a partíciót a neuronok Voronoi-diagramja juttatja érvényre. A Voronoi-diagram ilyen irányú alkalmazását idősorok predikciójára már felhasználták (*Principe*, 1998).

Az általunk vizsgált esetben a SOM algoritmus nemlineáris regresszióra való alkalmasságát használtuk ki. A küszöbérték függvényében felhasznált adatokkal identifikált modell hibájának változását az *1. táblázat* tartalmazza.

1. táblázat

A küszöbérték kiválasztása

Maximális szórás (1)	Figyelembe vett adatok aránya (2)	Kvantálási hiba (3)	Topográfiai hiba (4)	Lineáris modell hibája (5)	Nemlineáris modell hibája (6)
0.1	0.9653	0.1344	0.0185	0.0552	0.0537
0.06	0.9537	0.1340	0.0463	0.0542	0.0538
0.03	0.9012	0.1318	0.0175	0.0503	0.0505
0.02	0.8391	**0.1296**	0.0238	0.0490	0.0489
0.015	0.7424	0.1346	**0.0127**	**0.0479**	**0.0471**
0.01	0.5563	0.1579	0.0321	0.0839	0.0797
0.0075	0.3302	0.2066	0.0032	0.0851	0.0846

Table 1: The selection of the threshold value

Maximal deviation(1), Rate of the considered data(2), Quantization error(3), Topology error(4), Error of the linear mode(5), Error of the non-linear mode(6)

Az optimális küszöbérték esetén az adatokat a *8. ábrá*n látható módon jeleníthetjük meg. Az ábrán látható térképek hatszögletű elemekből épülnek fel, melyek mindegyike egy-egy csoportnak, működési pontnak felel meg. A színek érzékeltetik azt, hogy az adott tartományban milyen az állapotváltozó nagysága. Ezek alapján viszonylag egyszerűen, akár vizuális módon is találhatunk összefüggéseket a változók között. Könnyen észrevehető, hogy a polietilén-termelési sebesség és az etilén-betáplálás nagysága (érthető okokból eredően) nagyban összefüggenek, térképük szinte azonos. Hasonló következtetéseket vonhatunk le a komonomer hexén-betáplálás és a reaktorban uralkodó hexén-koncentráció esetén is. A termékminőséget jellemző MI-t több tényező is befolyásolja (az operátorok tapasztalatai alapján elsősorban a hőmérséklet és az etilén-koncentráció).

6. ábra

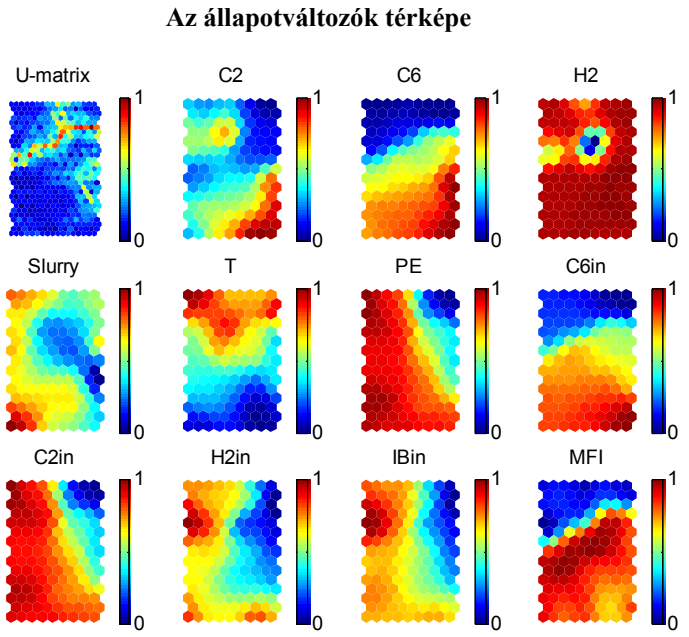

Figure 6: Component planes of the polyethylene production map

Összefoglalva, a vegyipari technológiákban a gyorsabb és megbízhatóbb számítógépes rendszerek megjelenésének hatására, a technológia fejlesztés és termelési költség csökkentés érdekében, lehetőség nyílik kifinomult szabályozási stratégiák véghezvitelére, szimulációra és optimalizációra. Ezen eszközök kialakításához fontos, hogy a tárolt hatalmas adathalmazból megfelelő információkat tudjunk szerezni, amely mind a folyamat modellezésében, jobb megértésében, mind – akár a modellen keresztül – a folyamat irányításában is hasznos lehet.

E komplex kérdéskörön belül a cikk egy, az ipari technológiák állandósult üzemeltetéséhez tartozó állapotváltozók közti kapcsolatok elemzéséhez elengedhetetlenül szükséges eszköz fejlesztésével foglalkozott, mely sok párhuzamos idősor szimultán módon történő szegmentálásán alapul.

Az irodalomban fellelhető szegmentálási algoritmusok nagyban különböznek attól függően, hogy milyen célt szolgálnak. Az ismertetésre került eszköz elsődleges célja az idősorok megfelelően beállt szakaszainak megkeresése volt. A szegmentálás során fuzzy csoportosítási algoritmust alkalmaztunk, mely a csoportok számát automatikusan határozza meg, és zajjal szemben nem érzékeny, valamint megfelelően robusztus.

Az algoritmus működését a TVK Rt. által rendelkezésünkre bocsátott valós ipari adatok elemzésén mutattuk be. Az eredmények rendkívül jól illusztrálják, hogy a termelés során keletkező adatokból miképpen rekonstruálhatók az üzemeltetés különböző szakaszai, illetve azt, hogy az állapotváltozók összefüggésének elemzésére alkalmas modellek identifikációját mennyire befolyásolja a felhasznált adatok minősége. Mivel a bemutatott szegmentálási algoritmus állandósult üzemeltetési tartományok

feltárására is alkalmazható, a létrehozott eszköz alkalmas a termelés során keletkező adatok közül e szempontból a relevánsak kiválasztására.

KÖSZÖNETNYILVÁNÍTÁS

A szerzők ezúton szeretnék kifejezni köszönetüket a Vegyészmérnöki Intézet Koordinációs Kutatási Központjának (KKK-II.-1A project), az Oktatási Minisztériumnak (FKFP-0073/2001), és az OTKA-nak (No. T037600) a támogatásért. Dr. Abonyi János munkáját a Magyar Tudományos Akadémia Bolyai János Kutatói Ösztöndíja is támogatta. Köszönet illeti ipari partnerünket a TVK Rt.-t, különösen Németh Miklós, Bálint Lóránt és dr. Nagy Gábor Urakat.

IRODALOM

Abonyi J., Feil B., Szeifert, F. (2002). 7th Online World Conference on Soft Computing in Industrial Applications. Determining the Model Order of Nonlinear Input – Output System by Fuzzy Clustering, (*http://wsc7.ugr.es*)

Abonyi J., Babuska, R., Feil, B. (2003). Structure Selection for Nonlinear Input–Output Models Based on Fuzzy Cluster Analysis. The IEEE International Conference on Fuzzy Systems, St. Louis, MO, USA.

Alander, J.T., Frisk, M., Holmstöm, L., Hämäläinen, A., Tuominen, J. (1991). Process error detection using self-organizing feature maps, In Artificial Neural Networks, II., 1229-1232. North-Holland.

Ayres C.A. (1986). Loop reactor setting leg system for preparation…, US 4. 613. 484.

Baldwin, J.F., Martin T.P., Rossiter, J.M. (1998.) Time Series Modelling and Prediction using Fuzzy Trend Information. Proceedings of Fifth International Conference on Soft Computing and Information/Intelligent Systems, 499-502.

Brandrup, J., Imergut, E.H. (1975). Polymer Handbook (Second edition) Joch Wiley & Sons Inc. Canada.

Doymaz, F., Chen, J., Romagnoli, J.A., Palayoglu, A. (2001). A Robust Strategy for Real-Time Process Monitoring. Journal of Process Control, 11. 343-359.

Feil, B. (2001). Nemlineáris bemenet-kimenet modellek rendűségének meghatározása csoportosítási algoritmus segítségével. Veszprémi Egyetem, Intézményi TDK.

Fujiwara T., and Nishitani, H. (1994). Abstraction of Operating Data on the Episode Map. Proceedings of the 1st Asian Control Conference (ASCC94), 725-728.

Gertler, J. (1988). Survey of Model-Based Failure Detection and Isolation in Complex Plants, IEEE Control Systems Magazine, December.

Goser, T.K. (1991). Self - Organizing Feature maps for process control in chemistry. In Artificial Neural Networks, 847-852. North-Holland.

Goutte, C., Toft, P., Rostup, E., Nielsen F.Å., Hansen, L.K. (1998). On Clustering fMRI Time Series, NeuroImage, 3. 298-310.

Harris, T., Kohonen, T. (1993). SOM based machine health monitoring systems which enables diagnosis of faults not seen in the training set. In Proc. of the Int. Conf. On Neural Networks (IJCNN`93), Nagoya, Japan, I., 947-950.

Huang, S-H., Qian, S-H., Shao, H-H. (1995). Human-Machine Cooperative Control for Ethylene Production, Artificial Intelligence in Engineering, 9. 203-209.

Kalafszki, L., Budai, G. (1999). A polietilén II., Magyar Kémikusok Lapja, 2. 70-81.

Kassalin, M., Kangas, J., Simula, O. (1992). Process state monitoring using self-organizing maps, In Artificial Neural Networks, II., 1531-1534. North-Holland.

Keogh, E., Chu, S., Hart D., Pazzani, M. (2001). An Online Algorithm for Segmenting Time Series, IEEE International Conference on Data Mining.

Kivikunnas, S. (1998). Overview of Process Trend Analysis Methods and Applications. ERUDIT Workshop on Applications in Pulp and Paper Industry. (http://www.erudit.de/erudit/events/tc-a/erudit-tca-2-Kivikunnas-13050.pdf)

Kohonen, T. (1990). The Self-Organizing Map, Proceedings of the IEEE, 78. 1464-1480.

Kosanovich K.A., and Piosovo, M.J. (1997). A Dynamical Supervisor Strategy for Multi-Product Processes, Computers & Chemical Engineering, 21. 149-154.

Lakshminarayanan, S., Fujii, H., Grosman, B., Dassau, E., Lewin, D.R. (2000). New product design via analysis of historical databases. Computers and Chemical Engineering, 24. 671-676.

Lane, S., Martin, E.B., Kooijmans, R., Morris, A.J. (2001). Performance Monitoring of a Multi-Product Semi-batch Process. Journal of Process Control, 11. 1-11.

Last, M., Klein Y., Kandel, A. (2000). Knowledge Discovery in Time Series Databases, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 1. 160-169.

Lindheim C., and Kristian Lien, M. (1997). Operator Support Systems for New Kinds of Process Operation Work. Computers & Chemical Engineering, 21. S113-S118.

MacGregor, J.F., and Kourti, T. (1995). Statistical process control of multivariate processes. Control Eng. Practice, 3. 403-414.

Meketta, J.J., (1992). Encyclopedia of Chemical Processing & Design, Marcel Delker Inc., New York.

Mishitani, H. (1996). Human-Computer Interaction in the New Process Technology. Journal of Process Control, 6. 111-117.

Pal, N.R. (1999). Soft computing for feature analysis, Fuzzy Sets and Systems, 103. 201-221.

Principe J.C., Wang L., Motter, M.A. (1998). Local Dynamic Modleing with Self-Organizing Maps and Applications to Nonlinear System Identification and Control. Proceedings of the IEEE, 86. 2241-2258.

Redman, J. (1991). Polyethylene, The Chemical Engineer, Sep. 26. 26-29.

Stephanopoulos G., and Han, C. (1995). Intelligent Systems in Process Engineering: A Review, May 11.

Wang, X.Z. (1999). Data Mining and Knowledge Discovery for Process Monitoring and Control, Springer.

Wong, J.C., McDonald J.C.K., Palazoglu, A. (1998). Classification of process trends based on fuzzified symbolic representation and hidden Markov models, J of Process Control, 8. 395-408.

Levelezési cím (*Corresponding author*):

Abonyi János
Veszprémi Egyetem, Folyamatmérnöki Tanszék
H-8200 Veszprém, Pf. 158.
University of Veszprem, Department of Process Engineering
H-8201Veszprém, P.O.Box 158.
Tel: +36-88-422-022/4209, Fax :+36-88-429-073
e mail:www.fmt.vein.hu/softcomp, abonyij@fmt.vein.hu