

1. Wstęp

Problemy związane z komputerowym sterowaniem procesami biologicznymi można generalnie podzielić na sprzętowe i programowe. Problemy sprzętowe wiążą się najczęściej z określoną, szczególną realizacją procesu, programowe natomiast mają charakter bardziej ogólny i mogą być rozważane w oderwaniu od konkretnej realizacji.

Znany informatyk szwajcarski Niklaus Wirth – twórca języka PASCAL – powiedział kiedyś, że (9):
PROGRAM = DANE + ALGORYTM.

W stwierdzeniu tym słowo „program” należy rozumieć w sensie ogólnym; w zastosowaniach praktycznych słowo „program” oznacza z reguły zakodowany algorytm, możliwy do wykonania przez komputer. Przeglądając publikacje na temat sterowania procesami, zwłaszcza podręcznikowe i pisane przez autorów ze środowisk związanych z inżynierią chemiczną, biologią czy biochemią, a nie z informatyką, można odnieść wrażenie, że w powyższym równaniu istnieje tylko drugi człon – **algorytm** (1,2,3,4,5,6). Wynika to prawdopodobnie ze szczególnej roli modelowania matematycznego w biotechnologii – wszak na modelach procesu i sterowania oparte są algorytmy sterowania.

Również w najnowszych doniesieniach można się spotkać z podobnym podejściem, jakkolwiek duży nacisk kładziony jest także na problemy identyfikacji parametrów modelu, a te z kolei wiążą się w sposób zasadniczy z danymi o sterowanym procesie (16,17,18,19,20,21). Można też zauważyć wzrost nacisku na analizę dokładności z jaką dane odzwierciedlają proces; zarówno w kontekście prawidłowej oceny przebiegu badanego zjawiska (30), jak i stabilności samego algorytmu sterowania (22). Czynione są też próby opracowania algorytmów sterowania nie wymagających kompletnej informacji o modelu sterowanego zjawiska, np. algorytmów adaptacyjnych (23,25,26) lub algorytmów bazujących na koncepcji linearyzowanego modelu lokalnego, identyfikowanego na bieżąco (24).

Z praktycznego punktu widzenia, efektem prac nad komputerowym sterowaniem procesami biologicznymi, jest **program**, a zatem nie umniejszając w niczym roli **modelu matematycznego i algorytmu**, należy jednakże przyrzeć się bliżej pozostałemu składnikowi programu, czyli **danym**. Z praktyki wynika, że niewłaściwe postawienie zagadnienia danych może w pierwszym rzędzie doprowadzić do trudności z realizacją komputerowego sterowania, na ogół znacznie wcześniej nim będzie można stwierdzić, że to „zbyt mała moc obliczeniowa komputera” przeszkadza w osiągnięciu celu. Pierwsze kłopoty pojawiają się wtedy, gdy dane nie mieszczą się na nośniku, lub ich zawiła struktura przeszkadza w ergonomicznej prezentacji (9,12); może się też okazać, że niewłaściwa struktura danych ma tak duży wpływ na spowolnienie ich przetwarzania, że zagraża to wykonaniu podstawowego zadania, tj. sterowania tym procesem.

2. Cechy zbiorów danych

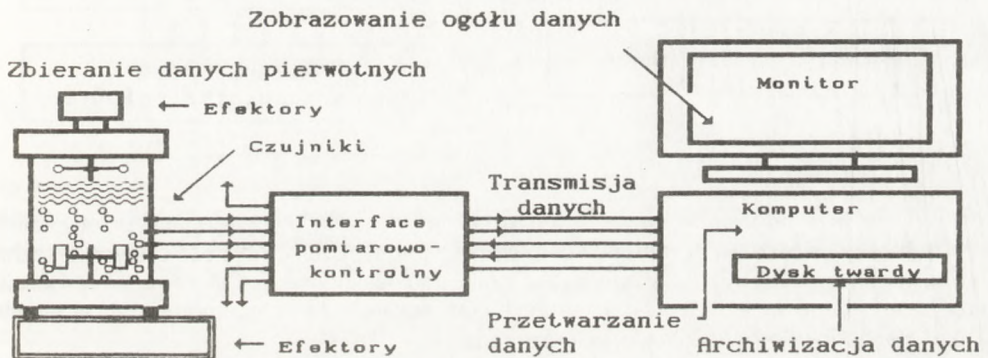
Podstawowe cechy zbiorów danych (każdych !) to:

- struktura,
- hierarchia,
- objętość.

Cechy te są bardzo silnie związane ze sobą i zmiana którejkolwiek z nich może mieć niebagatelny wpływ na pozostałe. Intuicyjnie wyczuwalne są zależności pomiędzy hierarchią i strukturą danych (8); nieco mniej oczywiste są już związki pomiędzy strukturą a ich objętością, na co zostanie w dalszym ciągu zwrócona szczególna uwaga.

W dalszym omówieniu termin: zbiory danych zostanie zastąpiony krótkim określeniem – dane.

Od razu należy zaznaczyć, że metody matematycznego poszukiwania w danych ich cech strukturalnych są na ogół dosyć złożone (7,8) i znacznie wykraczają poza ramy niniejszego opracowania. Dalej zostanie zaprezentowane praktyczne, intuicyjne podejście do tego zagadnienia, oparte o podstawowy zasób doświadczeń, jakie każdy biotechnolog może zdobyć (pochodząc w sposób planowy do swych eksperymentów). Warto dodać, że podejście tzw. cybernetyczne, a zatem kładące silny nacisk na strukturę i wewnętrzne powiązania podmiotu badań, jest również wykorzystywane w modelowaniu procesów biologicznych (13,14,15), jednakże w nieco innym kontekście niż to ma miejsce w niniejszym opracowaniu.



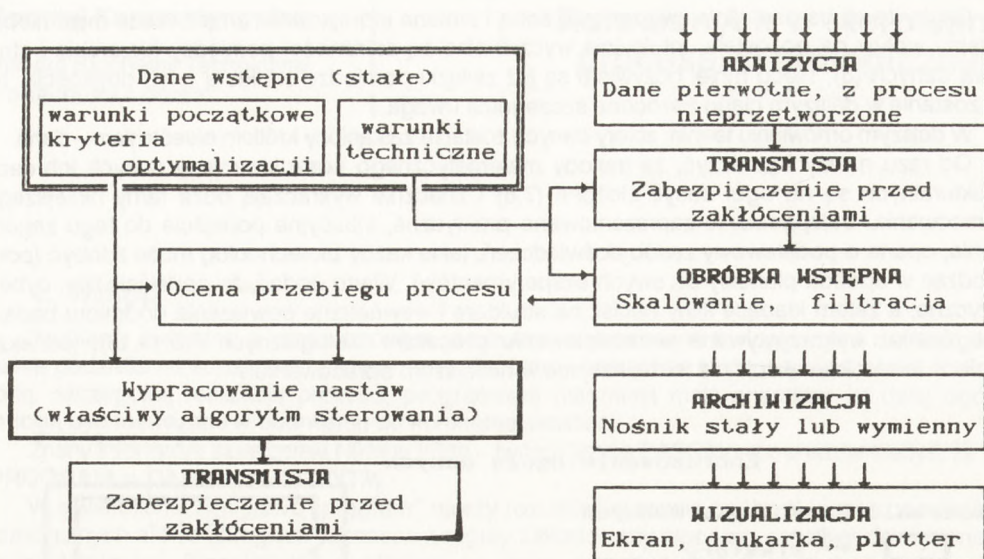
Rys.1. Manipulacje z danymi w typowym systemie sterowania komputerowego procesem w fermentorze.

Podstawowe manipulacje jakim poddaje się dane w typowym systemie komputerowego sterowania hodowlą drobnoustrojów w fermentorze pokazano na rys 1. W zależności od ich typu różna będzie optymalna struktura danych. Tak np. dla archiwizacji danych wskazana będzie struktura zapewniająca ich minimalną objętość; z kolei dla zobrazowania danych – niezbędnego dla operatora systemu – optymalna będzie struktura silnie zhierarchizowana, umożliwiającą kolejne przeglądanie bloków danych związanych z różnymi aspektami sterowanego procesu.

Hierarchia manipulacji danymi ma bezpośredni wpływ na hierarchię danych wynikającą z porządku ich przetwarzania. Należy zwrócić uwagę na brak określonego przyporządkowania dla bloków wizualizacji i archiwizacji danych. Dla bloku wizualizacji danych wynika to z konieczności zapewnienia **przeglądu danych każdego typu (!)** z czego z kolei wynika hierarchia danych związana z ich związkami skutkowymi–przyczynowo i wagą dla prawidłowego przebiegu procesu. Z tego punktu widzenia dane o stanach alarmowych powinny mieć bezwzględne pierwszeństwo w wizualizacji w dowolnym stanie pracy. Aby to osiągnąć stosunkowo prostym nakładem środków programistycznych, najlepiej wykorzystać tzw. system okien nakładanych, z bezwzględnym pierwszeństwem dla okienka z sygnałami alarmowymi.

Powyższe uwagi dotyczące danych mają oczywisty wpływ na strukturę całego programu, jego możliwości i sposób posługiwania się nim, zostaną omówione w dalszej części artykułu.

Wspomniano już, że z archiwizacją danych najsilniej związana jest taka ich cecha, jak objętość. Już intuicyjne podejście do analizy struktury i hierarchii danych przedstawionych na



Rys.2. Hierarchia przetwarzania danych.

rys. 2 wykazuje, że **informacja** zawarta w danych na poszczególnych etapach przetwarzania **może dublować się**, dzięki czemu można ją pominąć, zmniejszając objętość danych przeznaczonych do archiwizacji. Aby to uczynić (bez strat ogólnej informacji o sterowanym procesie) należy szczegółowo przeanalizować informację jaka dopływa do komputera w trakcie trwania procesu, np. hodowli drobnoustrojów w fermentorze.

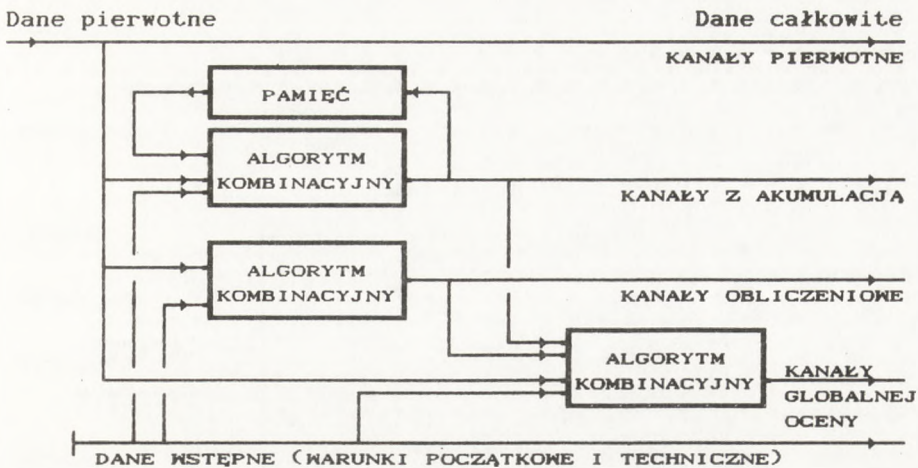
3. Kanaly danych

3.1. Idea metody kompresji informacji

Podczas przetwarzania informacji za pomocą komputera, szczególnie na styku człowiek – maszyna, bardzo często istnieje potrzeba kompresji informacji, czyli takiego jej przedstawienia – **bez strat** (!) – aby można było ją pamiętać na stosunkowo mało pojemnym nośniku, np. dyskiecie.

Podstawowa zasada kompresji danych polega na wyeliminowaniu z danych tzw. redundancji informacji, czyli informacji nadmiarowej (8). Aby tego dokonać należy znaleźć ewentualne wzajemne powiązania pomiędzy poszczególnymi elementami (jednostkami) danych, co pozwala je sprowadzić do innej postaci. Za przykład może posłużyć operacja zapamiętania narysowanego na ekranie odcinka, będącego jego elementem. Zapamiętanie całego odcinka bez kompresji jest równoważne zarejestrowaniu dla każdego elementu fragmentu ekranu – informacji o tym, czy jest on zapalony, czy zgaszony, jak to przedstawiono na rys. 3. Można jednak zapamiętać dla tego, szczególnego odcinka, jego parametry i algorytm wykreślenia. Oczywiście dla każdego następnego odcinka należy zapamiętać tylko 4 bajty współrzędnych, bo algorytm pozostaje ten sam. Oszczędność jest znaczna, szczególnie dla dużych rysunków (fragmentów ekranu) słabo wypełnionych odcinkami.

Proponowany sposób jest szeroko stosowany w komputerowych technikach kompresji obrazów, szczególnie w programach projektowych typu CAD (ang. *Computer Aided Design*).



Rys.3. Istota kompresji informacji przez zapamiętanie danych początkowych i algorytmu rozwinięcia.

Jeżeli potraktować proces przebiegający w czasie jako pewien obraz (system danych), to kompresja związanej z nim informacji polega na wyodrębnieniu struktury i hierarchii tych danych, określeniu algorytmów przetwarzania, określeniu danych pierwotnych i zapamiętaniu ich wraz z algorytmami – zamiast całej ich struktury.

Powyższy sposób kompresji informacji jest jednym z wielu. Wynika on wprost z pewnych metod stosowanych do tzw. rozpoznawania obrazów (ang. *pattern recognition*), umożliwiających sztuczemu systemowi, będącemu w istocie bardzo skomplikowanym programem, orientowanie się w otaczającym go środowisku (7,8). Dla danych występujących w sterowaniu procesami biologicznymi jest on szczególnie wygodny, ze względu na wyraźnie wyodrębniony etap strukturyzacji, pomocny w późniejszym konstruowaniu programu sterującego. Możliwe są inne sposoby kompresji informacji, oparte o takie pojęcia, jak: kodowanie lub analizę syntaktyczną informacji (7).

3.2. Abstrakcyjna struktura danych w procesie sterowania

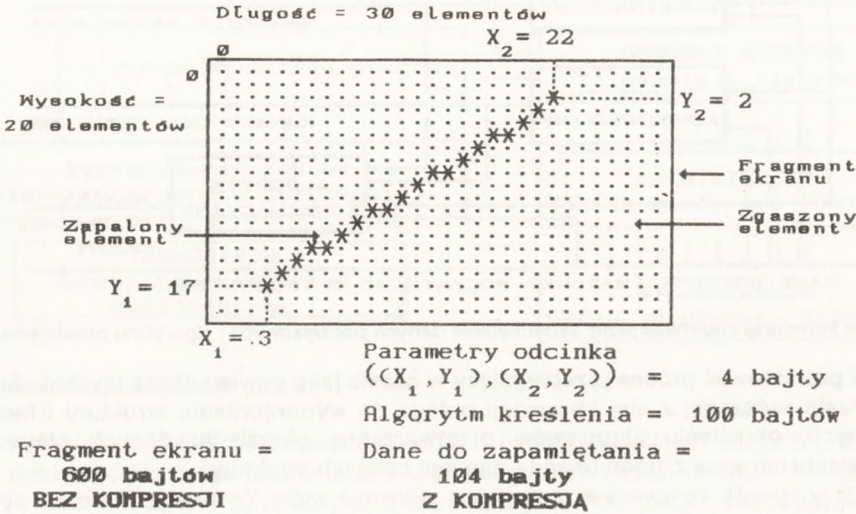
Proponowany schemat struktury danych występujących podczas komputerowego sterowania procesem przedstawiono na rys. 4. Poszczególne typy danych nazwano kanałami, aby podkreślić fakt przepływu ich strumienia w czasie. Schemat ten jest ogólny i kompletny, dzięki czemu jest adekwatny dla każdego możliwego procesu. Uwzględniono w nim przetwarzanie z uzależnieniami czasowymi (kanały z akumulacją) i przetwarzanie bezpośrednio (kanały obliczeniowe). Wyodrębniono osobno kanały globalnej oceny procesu, w których przepływają dane obrazujące aktualny stan procesu względem zadanego kryterium sterowania; odpowiadają one blokowi „ocena stanu procesu” na rys. 2.

Ze wzajemnej odpowiedniości bloków z rys. 2 i kanałów z rys. 4 wynika immanentny związek pomiędzy strukturą danych i algorytmami ich przetwarzania, sygnalizowany we wstępie. W dalszym ciągu zostaną omówione poszczególne jego elementy wraz z przykładami praktycznymi.

Stosując zasady kompresji do danych zawartych w rys. 4 (wyrażone w poprzednim rozdziale), można wskazać elementy struktury niezbędne do zapamiętania i ujmujące całość informacji o procesie:

- dane pierwotne,
- dane wstępne,
- wszystkie algorytmy kombinacyjne.

Z powyższego zestawienia wynika, że algorytm też może pełnić rolę danych!



Rys.4. Abstrakcyjna, ogólna struktura danych występujących w procesie sterowania komputerowego.

4. Przykład struktury danych

Poniżej przedstawiono hierarchiczną strukturę danych dla prostego procesu hodowli drobnoustrojów w fermentorze. Należy podkreślić, że jest to przykład jak najbardziej realny; oprzyrządowanie do realizacji takiej hodowli jest do dyspozycji w Instytucie Biochemii Technicznej Politechniki Łódzkiej. W szczególności dotyczy to bardzo nowoczesnego sterownika silnika mieszańca, wykonanego w ramach jednej z prac programu CPBP 04.11 nad biosyntezą α -amylazy.

Umożliwia on sterowanie silnikiem wraz z uzyskaniem zwrotnie sygnałów reprezentujących napięcie i prąd twornika. Po ich przemnożeniu (w komputerze) uzyskuje się niezbędną moc do utrzymania zadanych obrotów, a ze wzoru definicyjnego:

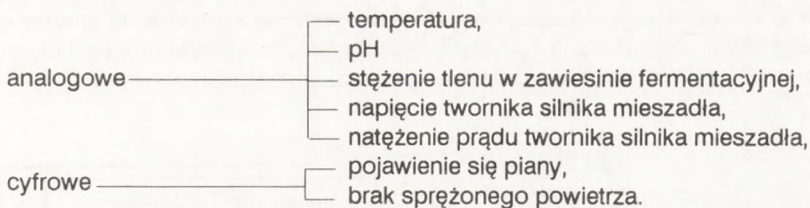
$$M = \frac{P}{2 \cdot \pi \cdot n}$$

gdzie: P – moc [W],
 n – prędkość obrotowa [s^{-1}]

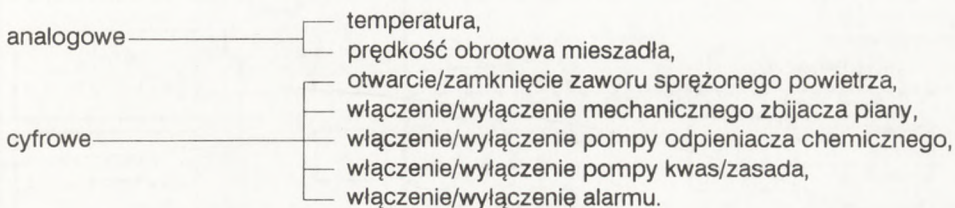
można uzyskać moment napędowy, w stanie ustalonym równy momentowi hamującemu. Znając moment hamujący mieszańca w zawieszynie można wyznaczyć lepkość, jakkolwiek wymaga to dodatkowych badań modelowych (1,2,3).

Struktura danych zgodna z ideą przedstawioną na rys. 4 – wygląda następująco:

Kanały pierwotne wejściowe:



Kanały pierwotne wyjściowe (nastawy):



Kanały obliczeniowe:

- moc mieszania,
- moment hamujący mieszadło,
- lepkość zawiesziny fermentacyjnej,
- stężenie biomasy,
- czas opóźnienia transportu tlenu w fermentorze,
- stała czasowa dyfuzji tlenu w zawieszynie $k_L \cdot a$

Kanały z akumulacją (z pamięcią):

- szybkość wzrostu drobnoustrojów,
- energia zużyta na stabilizację temperatury,
- energia zużyta na mieszanie zawiesziny,
- energia zużyta na napowietrzanie zawiesziny,
- ilość zużytego odpieniacza chemicznego,
- ilość zużytego kwasu/zasady do utrzymania pH.

Kanały globalnej oceny:

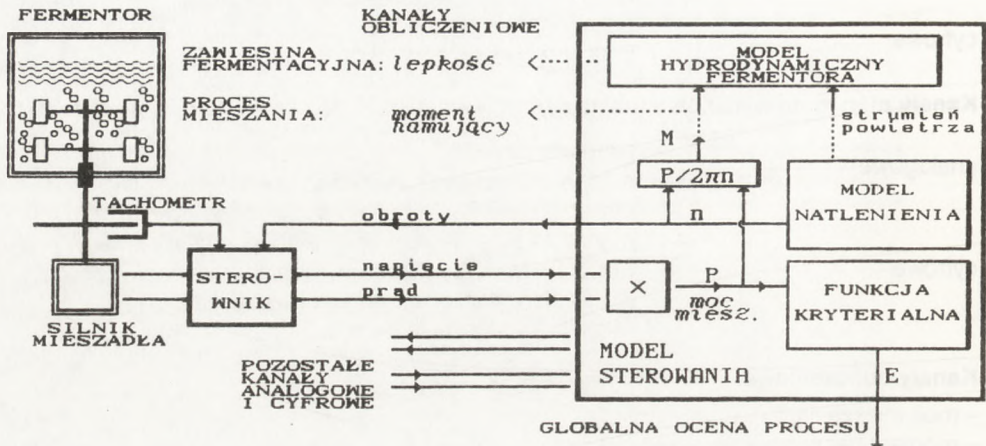
- współczynnik oceny energetycznej,
- wartość funkcji kryterialnej (oceny) procesu.

Dane wstępne:

- minimalne dozwolone obroty,
- maksymalne dozwolone obroty,
- okres próbkowania danych w kanałach pierwotnych,
- okres wykonywania pomiaru $k_L \cdot a$ (np. metodą dynamiczną (1,3)),
- minimalne stężenie tlenu podczas pomiaru $k_L \cdot a$,
- zezwolenie na odpienianie mechaniczne,
- zezwolenie na odpienianie chemiczne,
- zezwolenie na odpienianie przez zmniejszenie obrotów,
- współczynniki do funkcji kryterialnej (konkretna ich liczba zależy od wybranej funkcji kryterialnej).

W zestawieniu dane w kanałach pierwotnych podlegają transmisji od i do fermentora, a ściśle jego urządzeń wykonawczych.

W danych wstępnych nie uwzględniono parametrów technicznych (fermentora, akwizycji i transmisji sygnałów, itp.), wymienionych na rys. 2, ze względu na ich całkowitą zależność od konkretnej realizacji technicznej. Dane te są zawsze takie same przy każdym procesie realizowanym na tym samym sprzęcie i nie wchodzi w zakres prowadzonych dalej rozważań.



Rys.5. Ilustracja koncepcji różnych typów kanałów na przykładzie pomiaru pośredniego lepkości zawiesiny fermentacyjnej, z jednoczesnym obliczaniem elementu funkcji kryterialnej w postaci energii zużytej na mieszanie.

Fragment wzajemnych zależności pomiędzy kanałami obejmujący przede wszystkim proces mieszania i związany z nim bardzo ważny parametr jakim jest lepkość zawiesiny fermentacyjnej przedstawiono na rys. 5. Ostateczne jego wyliczenie wymaga udziału co najmniej dwóch dodatkowych algorytmów opartych na modelu hydrodynamicznym fermentora i modelu rozchodzenia się tlenu w zawieszynie fermentacyjnej. Wskazuje to dobitnie na wzajemne, wielopłaszczyznowe zależności pomiędzy różnymi elementami struktury danych, a stąd – na konieczność wysoce uporządkowanego ich rozpatrywania.

5. Archiwizacja i kompresja danych

Poniżej przedstawiono oszacowanie objętości danych bez i z kompresją, dla typowej hodowli periodycznej (uwzględniono dane zawarte w poprzednim rozdziale).

Założenia:

- czas trwania hodowli 48 h ,
- okres próbkowania stanu procesu 1 min,

(częstotliwość przesyłania danych od i do fermentora w kanałach pierwotnych – wejściowym i wyjściowym).

Wybrany okres próbkowania jest odpowiedni dla fermentora o objętości kilkudziesięciu litrów, zgodnie z zasadą, że okres próbkowania powinien być przynajmniej 10 razy mniejszy od stałych czasowych procesów cząstkowych – np. mieszania cieczy, transportu masy itp. (11).

Z założeń wynika, że proces jest opisany za pomocą 2880 próbek. Założywszy dalej, że parametry procesu kwantowane są z dokładnością do 10 bitów (oznacza to znajomość wartości zmiennej z dokładnością ok. 0,1 %), każda jednostka zajmuje 2 bajty. Daje to dla 5 zmiennych

(analogowych) i 1 bajtu zmiennych cyfrowych (z przykładu z poprzedniego rozdziału) następującą objętość danych pierwotnych wejściowych:

$$N_{Pi} = (2 \cdot 5 + 1) \cdot 2880 = 31\ 680 \text{ bajtów}$$

Dla kanału danych pierwotnych wyjściowych mamy 2 zmienne analogowe i bajt zmiennych cyfrowych, co daje:

$$N_{Po} = (2 \cdot 2 + 1) \cdot 2880 = 14\ 400 \text{ bajtów}$$

Razem dane pierwotne – wejściowe i wyjściowe, mają objętość:

$$N_p = 31\ 680 + 14\ 400 = 46\ 080 \text{ bajtów.}$$

Przechowanie samych danych pierwotnych wraz z algorytmami wyznaczania:

- mocy silnika (proste mnożenie),
- momentu hamującego (według wzoru (1)),
- lepkości zawiesiny (wg badań modelowych),
- stężenia biomasy (wg badań modelowych),

wystarcza do całościowego opisanie procesu w aspekcie wzrostu biomasy, przedstawionego na rys. 5. Z punktu widzenia objętości danych jest to bardzo korzystne, gdyż wymienione algorytmy mogą być wspólne dla dużej liczby procesów. Szczególnie w badaniach optymalizacyjnych, np. składu podłoża, poszczególne hodowle realizowane są w bardzo zbliżonych warunkach, co pozwala stosować ten sam model wiążący moment, lepkość i stężenie biomasy dla wielu procesów (1,2,3).

Zakładając dalej, że pozostałe dane dotyczące przykładowego procesu są reprezentowane w komputerze w tzw. arytmetyce zmiennoprzecinkowej, czyli w najlepszym wypadku liczba zajmuje 4 bajty (tzw. pojedyncza precyzja); otrzymuje się dla kanałów:

- obliczeniowych

$$N_o = 6 \cdot 4 \cdot 2880 = 69\ 120 \text{ bajtów,}$$

- z akumulacją

$$N_A = 6 \cdot 4 \cdot 2880 = 69\ 120 \text{ bajtów,}$$

- globalnej oceny

$$N_Q = 2 \cdot 4 \cdot 2880 = 23\ 040 \text{ bajtów,}$$

$$N_D = N_o + N_A + N_Q = \frac{161\ 280 \text{ bajtów.}}$$

Tak zatem całkowita objętość danych reprezentujących proces wynosi:

$$N_C = N_p + N_D = 207\ 360 \text{ bajtów}$$

Jest to bardzo duża objętość danych. Jeśli jeszcze uwzględnić takie fakty jak:

- bardzo skromne założenia co do liczby kanałów pierwotnych,
- ewentualne zapamiętanie danych wstępnych, przynajmniej dotyczących procesu (techniczne są stałe),
- krótki czas trwania procesu,

to otrzymana liczba **ponad 200 Kb** informacji na proces jest w zasadzie **dolnym oszacowaniem** liczebności zbioru danych! Jeśli więc porównać ją jeszcze z pojemnością standardowej dyskietki wynoszącą **360 Kb**, to stanie się jasne, że aby archiwizować informacje dotyczące prowadzonych procesów na dyskietkach (bezpieczniej niż na twardym dysku), niezbędne są metody kompresji, choćby tak proste jak przedstawiona na rys. 3 – oparta na strukturalizacji danych.

6. Dane a struktura programu

Blizsza analiza danych zawartych na rys. 2 i 4 pozwala dojść do wniosku, że starannie zbudowana struktura jest już najprostszą wytyczną do opracowania budowy programu. W szczególności dotyczy to tak ważnej sprawy, jak kolejność przetwarzania danych (9).

Dobrze określona struktura danych pozwala również na zastosowanie tzw. zmiennych domyślnych (ang. *default variables*), których wartość jest w programie predefiniowana automatycznie. Uwalnia to użytkownika programu (biotechnologia!) od żmudnego ustalania wartości, czasami setek zmiennych, powtarzających się np. z hodowli na hodowlę, co ma duże znaczenie z punktu widzenia ergonomii obsługi programu.

Dotyczy to również ergonomii; ze strukturyzacji danych wynika możliwość przyporządkowania pewnym typom informacji stałych kolorów podczas wizualizacji na ekranie. Ma to niebagatelne znaczenie dla szybkiej oceny stanu procesu przez operatora, zwłaszcza w sytuacjach alarmowych. Wymaganie to, może niezbyt istotne w sterowaniu procesami na skalę laboratoryjną, staje się niezwykle ważne w przypadku zastosowań przemysłowych, gdzie tzw. czynnik ludzki gra niepoślednią i nie zawsze pozytywną rolę, a poziom czynników środowiskowych rozpraszających uwagę jest z reguły większy niż w „pracowniach uczonych”.

Z hierarchizacji i strukturyzacji danych wynika również możliwość ergonomicznego zaprojektowania systemu okien do ich prezentacji. W jednym określonym oknie zgrupowane powinny być dane jednego typu, bądź opisujące jeden aspekt procesu. Hierarchia danych pokazuje od razu sposób szeregowania okien, oraz sposób ich wywoływania i przesłaniania na ekranie monitora. O wielkiej wygodzie takiego sposobu przedstawiania informacji (zarówno dla programisty jak i użytkownika) świadczy opracowywanie przez wielkie firmy programistyczne gotowych programów narzędziowych do takiej właśnie prezentacji danych. Wydaje się, że najlepszy jest program MS-windows produkcji firmy Microsoft (USA).

Komputerem można sterować nie tylko realny proces (np. hodowlę drobnoustroju), ale i symulować cyfrowo przeprowadzanie procesu. O wadze właściwego uporządkowania danych w tym przypadku można się przekonać choćby z literatury (14,27,29). Symulacja taka może oddać nieocenione usługi przy projektowaniu pewnych procesów niejako „na sucho” (28,29), gdyż wówczas mogą ujawnić się błędy popełnione w założeniach do procesu... Nieoceniony wydaje się również walor dydaktyczny programów symulacyjnych, pozwalających w dowolnym momencie pokazać słuchaczom wykładu efekty omawianych zjawisk. W tym przypadku rola dobrego modelu matematycznego symulowanego zjawiska odgrywa równie wielką rolę, jak właściwe postawienie problemu struktury danych.

7. Uwagi końcowe

W artykule starano się pokazać nieco inne podejście do problemu konstrukcji oprogramowania w odniesieniu do sterowania procesami biologicznymi. Baczniejsze zwrócenie uwagi na pierwszy człon równania Niklausa Wirtha, **czyli na dane**, zrodziło się podczas współpracy autora z biotechnologami w rozwiązywaniu konkretnych problemów praktycznych. Okazało się wówczas, że programy dzielą się na teoretycznie, absolutnie doskonale i zarazem absolutnie nieużywane oraz na programy zbudowane na podstawie bardzo ścisłej współpracy programisty z użytkownikiem, przekazującym mu swą wiedzę praktyczną. Pierwszym krokiem w owej współpracy okazuje się nieodmiennie **określenie, co właściwie program ma przetwarzać, czyli jakie są do niego dane**. Dopiero za tym następuje określenie, **jak przetwarzać**.

Bardzo ważne, z punktu widzenia praktycznego, okazują się także kwestie wizualizacji i archiwizacji danych. Są one często niedoceniane lub wręcz pomijane i to nie tylko przez krajowych programistów. W efekcie otrzymuje się wyroby tak nieporęczne w użyciu, że ewentualny (mocno skonfundowany!) biotechnolog nigdy nie przechodzi do etapu swobodnego posługi-

wania się programem, twierdząc na wszelki wypadek, że cała ta komputeryzacja to wymysł co najmniej „określonych sił” ... A wystarczy zacząć *ab ovo* czyli od danych! Dopiero po przebrnięciu przez ten etap można przystąpić do systematycznego opracowania procesu technologicznego, przy czym wartość tego opracowania jest proporcjonalna do zgromadzonej na jego temat informacji, podlegającej następnie obróbce statystycznej, pozwalającej uchwycić występujące prawidłowości i albo wykorzystać zjawiska pozytywne, albo zapobiec zjawiskom szkodliwym.

Literatura

1. Aiba S., Humphrey A. E., Millis N. F., (1977) Inżynieria biochemiczna. WNT, Warszawa.
2. Kafarow W. W., Winarow A. Ju., Gordiejew L. S., (1983), Modelowanie reaktorów biochemicznych, WNT, Warszawa.
3. Wang D. C., Cooney C. L., Deeman A. L., Dunnill P., Humphrey A. E., Lilly M. D., (1979), Fermentation and Enzyme Technology, John Wiley and Sons.
4. Warfotomiejew S. D., (1987), Sowriemiennyje problemy biokinetiki, Izd. Moskowskowo Uniwiersiteta, Moskwa.
5. Warfotomiejew S. D., Zajcew S. W., (1982), Kineticzieskije mietody w biochimizieskich issliedowaniach, Moskwa.
6. Rubin A. B., Pyt'ewa N. F., Rizniczenko G. Ju., (1987), Kinetika biologiczeskich processow, Izd. Moskowskowo Uniw., Moskwa.
7. Fu K. S., (1974), Syntactic Methods in Pattern Recognition, Academic Press, New York-London.
8. Kulikowski J. L., (1972), Cybernetyczne układy rozpoznające, PWN, Warszawa.
9. Wirth N., (1989), Algorytmy + struktury danych = programy, WNT, Warszawa.
10. Kącki E., Woźniakowski M., (1979), Modelowanie analogowe, hybrydowe oraz cyfrowa symulacja maszyn analogowych, PWN, Warszawa.
11. Larminat P. de, Thomas Y., (1982), Automatyka – układy liniowe, WNT, Warszawa.
12. Tassel van D., (1982), Praktyka programowania, WNT, Warszawa.
13. Turner B. G., Ramakrishna D., Jansen N. B., (1988), Biotech. and Bioeng., 32, 1988, 46–56.
14. Turner B. G., Ramakrishna D., Jansen N. B., (1989), Biotech. and Bioeng., 34, 252–261.
15. Nielsen J., Emborg C., Halberg K., Villardsen J., (1989), Biotech. and Bioeng., 34, 478–486.
16. Pavlou S., (1987), Biotech. and Bioeng., 30, 413–419.
17. Ivanitskaya J. G., Petrikevitch S. B., Bazykin A. D., (1989), Biotech. and Bioeng., 33, 1162–1166.
18. Abulesz E.-M., Lyberatos G., (1987), Biotech. and Bioeng., 30, 1059–1065.
19. Modok J. M., Lim H. C., (1989), Biotech. and Bioeng., 33, 11–15.
20. San K. Y., Stephanopoulos G., (1989), Biotech. and Bioeng., 34, 72–78.
21. Skowlund C. T., Kirmse D. W., (1989), Biotech. and Bioeng., 33, 164–172.
22. Suarez-Cortez R., Alvarez-Gallegos J., Gonzalez-Mora E., (1989), Biotech. and Bioeng., 33, 377–385.
23. Harmon J., Svoronos S. A., Lyberatos G., (1987), Biotech. and Bioeng., 30, 335–344.
24. Modok J. M., Lim H. C., (1987), Biotech. and Bioeng., 30, 528–540.
25. Semones G. B., Lim H. C., (1989), Biotech. and Bioeng., 33, 16–25.
26. Shi Z., Shimizu K., Watanabe N., Kobayashi T., (1989), Biotech. and Bioeng., 33, 999–1009.
27. Fishman G. S., (1981), Symulacja komputerowa. Pojęcia i metody, PWE, Warszawa.
28. Chang Y. K., Lim H. C., (1989), Biotech. and Bioeng., 34, 577–591.
29. Kishimoto M., Alfara C. G., Nakajima M., Yoshida T., Taguchi H., (1989), Biotech. and Bioeng., 33, 191–196.
30. Liao J. C., (1989), Biotech. and Bioeng., 33, 613–622.

The importance of data in biological processes computer control

Summary

The analysis of computer control has been presented from the point of data related with biological process. The aim has been to stress the practical bearings of the development of the complex computer control systems. Data have been examined due to their basic features: hierarchy, structure and volume. The idea of putting structure and hierarchy onto data by use of the flow channels has been presented. Basing upon this idea, a comparison has been made among different ways of data visualization and economic storage for archival purposes. The interrelations among control program structure and functions due to processed data has been demonstrated.

Adres dla korespondencji:

Andrzej Konstantynowicz, ul. Nowa 20 m. 72, 90-030 Łódź.

NOWOŚCI

Wysokociśnieniowe enzymatyczne utlenianie cholesterolu

Nowe możliwości syntezy interesujących substancji, związane z rozwojem biokatalizy w rozpuszczalnikach organicznych, zostaną prawdopodobnie znacznie poszerzone dzięki zastosowaniu w charakterze środowiska dla reakcji enzymatycznych takich rozpuszczalników, jak np. dwutlenek węgla (mało toksyczny, niepalny, tani), w temperaturach i przy ciśnieniach nieco wyższych niż w punkcie krytycznym. Ostatnio skonstruowano bioreaktor z unieruchomioną oksydazą cholesterolu z *Gleocysticum chrysocreas*, w którym w środowisku dwutlenku węgla, w temperaturze 35°C i pod ciśnieniem 100 atmosfer przeprowadzono udaną reakcję utleniania cholesterolu tlenem do cholest-4-en-3-onu, ważnego prekursora niektórych farmaceutyków, np. estradiolu. Unieruchomiony enzym był w tych warunkach bardzo stabilny, zaś dla odpowiedniej wydajności reakcji konieczna była obecność w jej środowisku śladowych ilości wody. Atrakcyjność dwutlenku węgla i innych substancji, np. etanu, etylenu, tlenku azotu czy fluorometanu, które w warunkach nieco powyżej punktu krytycznego mogą stanowić dobre środowisko dla reakcji enzymatycznych, wynika m.in. z ich niskiej lepkości, wysokiej dyfuzyjności oraz dużej zdolności zmian stałej dielektrycznej, towarzyszących małym zmianom temperatury i ciśnienia. Uważa się, że te związki stosowane jako środowisko reakcji enzymatycznych pozwolą na lepszą kontrolę ich szybkości i selektywności, jak też ułatwią oddzielanie i enzymu i produktu ze środowiska.

M.T

Opracowano na podstawie: Randolph T. W., i in. (1988), *AIChE Journal*, 34, 1354-1360.