

Sterownik robotów wyposażonych w różnorodne czujniki

Cezary Zieliński

Politechnika Warszawska, Instytut Automatyki,
ul. Nowowiejska 15/19, 00-665 Warszawa

Streszczenie

W referacie omówiono strukturę sprzętową i programową sterownika dla robotów wyposażonych w różnorodne czujniki. Ponieważ *a priori* nie można określić jakich czujników będą wymagały zadania postawione przed robotem, a ponadto w jaki sposób będą wykorzystywane dane otrzymywane z czujników, zdefiniowano jedynie ogólną strukturę sterownika oraz opracowano bibliotekę procedur i funkcji, z których tworzy się sterownik. Używając tych procedur oraz ewentualnie dodając nowe wypełnia się zdefiniowaną strukturę – szkielet. W ten sposób powstaje sterownik dokładnie dostosowany do potrzeb wykonania zadania, które postawiono przed robotem.

1 Wprowadzenie

Producenci robotów przemysłowych starają się rozszerzyć możliwości "uczenia" poprzez wyposażenie programatorów w specjalizowane języki programowania robotów, w ten sposób czyniąc sterowniki coraz bardziej uniwersalnymi. Z drugiej strony, tworzone są specjalizowane sterowniki do realizacji pojedynczego, acz skomplikowanego, zadania (np. robot grający w ping-ponga [1]). Wiele wysiłku wkłada się w opracowanie specjalizowanego sprzętu przyspieszającego obliczenia charakterystyczne dla robotyki [2]. Zaproponowano szereg architektur, w skład których wchodził specjalizowany sprzęt [3, 4]. Architektury zawierające jedynie sprzęt uniwersalny również były wykorzystywane [5]. Zajmowano się również takim oprogramowaniem systemów sterujących, aby dołączanie czujników było proste [6, 7, 8]. Analiza literatury oraz doświadczenia autora wskazują, że różnorodność potencjalnych zadań dla robota do wykonania, czyni sterowniki uniwersalne zbyt kosztownymi oraz pomimo ich uniwersalności zbyt mało elastycznymi. Dlatego stworzony został rekonfigurowalny sterownik, który może być dostosowywany dokładnie do zadania, które ma być wykonane przez robota [9, 10, 11, 12, 13]. Ponieważ w momencie wytwarzania robota nie można określić zadania, które będzie on wykonywał, to również nie można przewidzieć w jakie czujniki będzie wyposażony oraz jak będą musiały być przetwarzane i wykorzystane dane otrzymywane z tych czujników. W związku z tym określono ogólnie strukturę sterownika – szkielet – oraz stworzono bibliotekę modułów programowych, które mogą zostać wykorzystane do wypełnienia tego szkieletu. Jeżeli zadanie wymaga stworzenia nowych lub modyfikacji już istniejących modułów, to projektant sterownika (programista) może to zrobić, jedynie musi stosować się do pewnych reguł, aby nie zniszczyć szkieletu. Całość oprogramowania tworzona jest w języku C.

2 Struktura sprzętowa i programowa systemu

Sterownik wykorzystuje komputer IBM-PC/486 połączony specjalizowanym równoległym interfejsem MIAT z pakietami sterowników osi MA-70 oraz pakietami we/wy MC-42 (rys. 1). Procesor oraz pamięć (ML-16 i MM-16) oryginalnego sterownika robota IRp-6 przechodzą w stan zawieszenia, gdy sterowanie przejmowane jest przez IBM PC/486. Moduł procesów współbieżnych czyni warstwę sprzętową systemu przezroczystą dla programisty (rys. 2). Moduł ten umożliwia tworzenie i mieszanie procesów współbieżnych, ich synchronizację za pomocą semaforów, przesyłanie danych między procesami za pomocą potoków (ang. pipes) oraz kolejkowanie zadań. Ponieważ w obecnej wersji systemu wszystkie procesy wykorzystują jeden procesor fizyczny, muszą one pracować z podziałem czasu.

System, oprócz własnych procesów modułu współbieżności, składa się z następujących procesów (rys. 3) – stanowią one szkielet systemu:

- Sterującego Procesami Przetwarzającymi Dane (SPD),
- Wypisywania Odpowiedzi (WO),
- Sterującego Robotem (SR),
- i z zera lub więcej Procesów Agregacji Danych z czujników rzeczywistych (PA).

Programista może tworzyć i modyfikować procesy SR i PA zgodnie z pewnymi zasadami. Zazwyczaj wystarcza wypełnienie szkieletu wywołaniami funkcji i procedur bibliotecznych oraz dodanie instrukcji inicjalizujących zmienne. W bardziej skomplikowanych przypadkach można dopisać w języku C własne funkcje. Po wypełnieniu szkieletu, całość jest kompilowana i konsolidowana z kodem procesów niemodyfikowalnych, tzn. modułem współbieżności oraz procesami SPD i WO.

2.1 Procesy współpracujące z operatorem: SPD i WO

Kontakt operatora z systemem odbywa się za pośrednictwem procesów SPD i WO. Operator wydaje polecenia systemowi poprzez proces SPD korzystając z menu. Polecenia dotyczą: inicjalizacji robota (sterowanie ręczne, synchronizacja); wykonania, zawieszenia i odwieszenia Procesu Sterującego Robotem SR oraz zakończenia pracy systemu. Wszystkie procesy przekazują informacje operatorowi za pośrednictwem procesu WO. Proces WO otrzymuje od innych procesów wiadomości potokami danych, formatuje je, i wyświetla w adekwatnych okienkach na ekranie monitora. Każdy z procesów posiada swoje własne okienka na ekranie.

2.2 Procesy Agregacji Danych: PA

Dane otrzymywane z czujników mogą stanowić, w najprostszyc przypadkach, pojedyncze bity, natomiast w najbardziej złożonych przypadkach składają się z wielowymiarowych struktur bitowych (np. obrazy z kamer CCD). Ponadto dane otrzymywane z pojedynczego czujnika rzeczywistego mogą nie nieść w sobie pełnej informacji niezbędnej do sterowania. W związku z tym opracowano metodę dołączania czujników rzeczywistych do systemu, niezależną od złożoności i wielości czujników. Rys. 4 przedstawia sieć działań Procesu Agregacji Danych z czujników rzeczywistych (PA). Zadaniem takiego procesu (system może zawierać ich wiele) jest odczytanie wskazań czujników rzeczywistych, dokonanie agregacji odczytanych danych oraz przesłanie zagregowanych danych potokiem do Procesu Sterującego Robotem SR. Agregacja może polegać zarówno na wyodrębnieniu

istotnych informacji z otrzymanych danych, jak i scaleniu informacji z wielu prostych źródeł informacyjnych. Otrzymany agregat informacyjny nazywany jest odczytem czujnika wirtualnego. Dołączenie nowych czujników do systemu sprowadza się do ich mechanicznego i elektrycznego dołączenia oraz napisania (według wzorca) nowego procesu PA.

2.3 Proces Sterujący Robotem: SR

Proces Sterujący Robotem jest jedynym procesem bezpośrednio oddziałującym na manipulator. Jego głównym zadaniem jest obliczenie wartości zadanych dla 5 serwo-mechanizmów poruszających ramię. Jediną możliwością wpływania na dynamikę systemu jest odpowiednie kształtowanie tych wartości zadanych. W bibliotece istnieją funkcje obliczające powyższe wartości zadane na podstawie: współrzędnych końca narzędzia wyrażonych w przestrzeni kartezjańsko-eulerowskiej, wzajemnych położenia *kątowych* poszczególnych członów manipulatora oraz przyrostów położenia wałów silników. Założono, że manipulator wyposażony jest w narzędzie osiowo-symetryczne, gdyż ramię ma 5 stopni swobody. W trakcie obliczania kolejnej wartości zadanej brane są pod uwagę dane otrzymywane od procesów PA. Rys.5 przedstawia ogólną postać procedur poruszających manipulatorem. Postać ta nie ulega zmianie niezależnie od sposobu obliczania wartości zadanej. Do generacji trajektorii zastosowano funkcje sklepane trzeciego stopnia. W ten sposób robot może łagodnie przyspieszać na początku realizowanej trajektorii oraz łagodnie wyhamowywać na jej końcu. Możliwy jest również ruch z maksymalnym przyspieszeniem na początku i końcu trajektorii przy zachowaniu stałej prędkości wzdłuż pozostałej jej części. Dodanie innych sposobów generacji trajektorii nie następuje żadnych kłopotów. Ponadto istnieje zestaw procedur oczekujących na zaistnienie zdarzenia. Zajście zdarzenia sygnalizowane jest przez czujniki wirtualne. Procedury przemieszczające manipulator oraz procedury oczekiwania tworzą i niszczą procesy PA w zależności od swych potrzeb. Ogólna struktura procesu SR uwidoczniła jest na rys. 6.

3 Wnioski

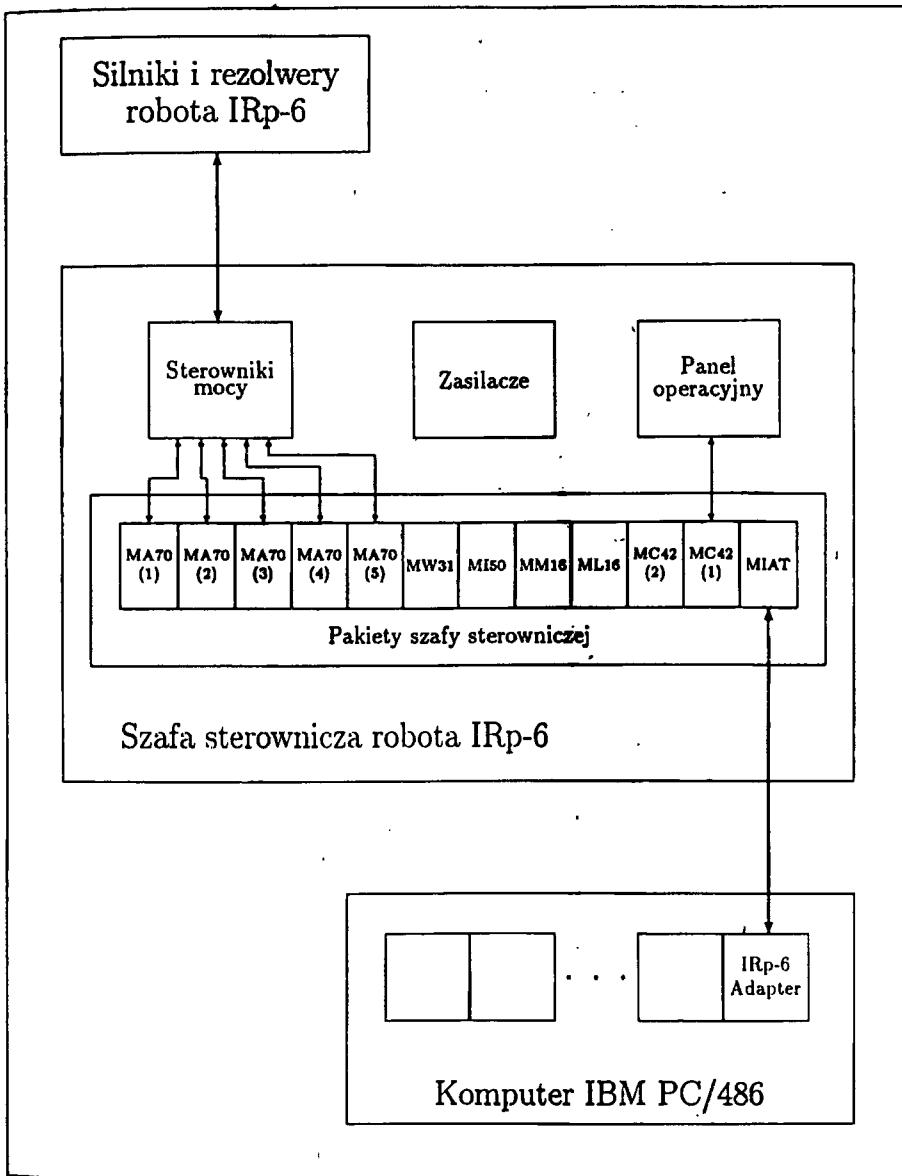
Celem realizacji opisanego sterownika było zbadanie różnych metod agregacji danych z czujników rzeczywistych oraz generacji i modyfikacji trajektorii ruchu ramienia na podstawie tych danych. Przeprowadzono eksperymenty z prostymi pięciobitowymi czujnikami dotykowymi, sześciosiowym czujnikiem sił i momentów sił oraz kamerą CCD. W przypadku czujników dotykowych oraz kamery zadanie polegało na znalezieniu drogi do celu w labiryncie, natomiast zadaniem robota wyposażonego w czujnik siły było poruszenie się wywierając stałą siłę na powierzchnię poruszającego się obiektu. W eksperymencie z czujnikami dotykowymi ich zadaniem było dostarczenie wiedzy lokalnej o labiryncie. Na podstawie zdobytej wiedzy sterownik szukał wyjścia z labiryntu. Natomiast w eksperymencie z kamerą dostarczała ona obraz całego labiryntu. Na podstawie tego obrazu wyznaczano najkrótszą trajektorię do celu. Eksperymenty potwierdziły przydatność sterownika do badań ruchu robotów wyposażonych w czujniki.

Podziękowania

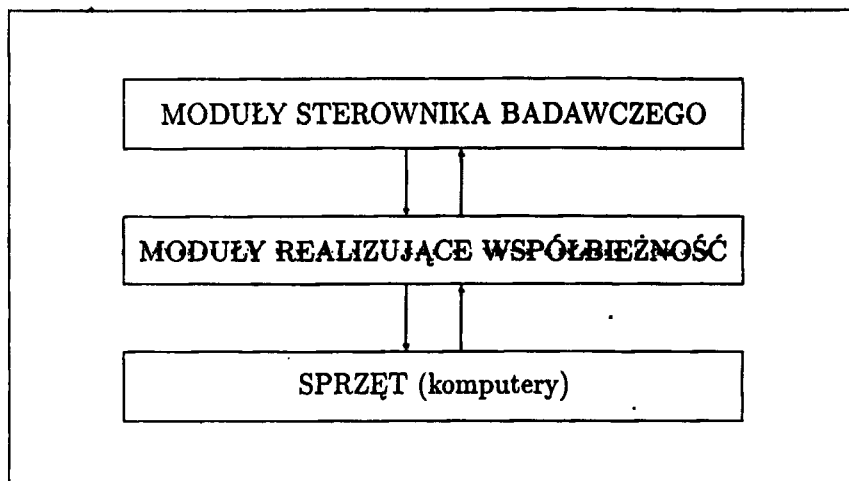
Autor pragnie podziękować następującym osobom biorącym udział w implementacji sterownika: A. Grodecki, U. Kręglewska, J. Sobczyk, A. Śluzek, T. Zielińska. Praca była finansowana przez Komitet Badań Naukowych – grant nr 7 1083 91 01.

Literatura

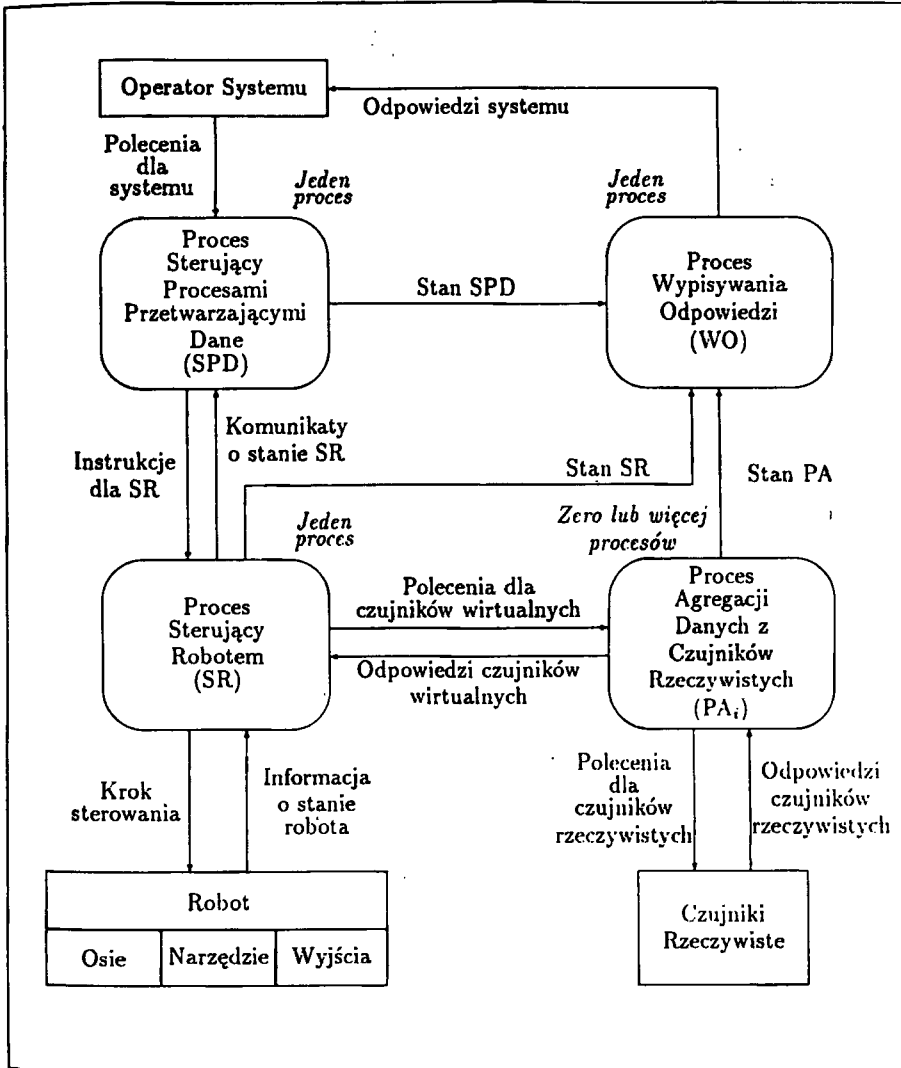
- [1] R. L. Andersson; *A Robot Ping-Pong Player*. The MIT Press, 1988.
- [2] P. Sasayappan, Y-L. C. Ling, K. W. Olson, D. E. Orin; *A Restructurable VLSI Robotics Vector Processor Architecture for Real-Time Control*. IEEE Transactions on Robotics and Automation, Vol.5 No.5, October 1989.
- [3] Y. Wang, S. E. Butner; *RIPS: A Platform for Experimental Real-Time Sensory-Based Robot Control*. IEEE Transactions on Systems, Man, and Cybernetics, Vol.19 No.4, July/August 1989.
- [4] J. H. Graham; *Special Computer Architectures for Robotics: Tutorial and Survey*. IEEE Transactions on Robotics and Automation, Vol.5 No.5, October 1989.
- [5] S. Narasimhan, D. M. Siegel, J. M. Hollerbach; *CONDOR: An Architecture for Controlling the Utah-MIT Dexterous Hand*. IEEE Transactions on Robotics and Automation, Vol.5 No.5, October 1989.
- [6] Y. F. Zheng; *Integration of Multiple Sensors into a Robotic System and its Performance Evaluation*. IEEE Transactions on Robotics and Automation, Vol.5 No.5, October 1989.
- [7] R. C. Luo, M-H. Lin, R. S. Scherp; *Dynamic Multi-Sensor Data Fusion System for Intelligent Robots*. IEEE Journal of Robotics and Automation, Vol.4 No.4, August 1988.
- [8] R. C. Luo, M. G. Kay; *Multisensor Integration and Fusion in Intelligent Systems*. IEEE Transactions on Systems, Man, and Cybernetics, Vol.19 No.4, July/August 1989.
- [9] C. Zieliński, A. Grodecki, U. Kręglewska, A. Śluzek, T. Zielińska; *Koncepcja sterownika robotów przeznaczonego do celów badawczych*. III Krajowa Konferencja Robotyki, Wydawnictwo Politechniki Wrocławskiej, Wrocław 1990,
- [10] C. Zieliński; *Flexible Controller for Robots Equipped with Sensors*. Proceedings of the 9-th CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators, Ro.Man.Sy'92, 1-4 September 1992, Udine, Italy.
- [11] C. Zieliński; *Controller Structure for Robots with Sensors*. Mechatronics Vol.3 No.5, Pergamon Press, 1993.
- [12] C. Zieliński, A. Grodecki, U. Kręglewska, J. Sobczyk, A. Śluzek, T. Zielińska; *Sterownik robotów przeznaczony do celów badawczych*. Opracowanie dla KBN, Inst. Aut. P.W., Warszawa, grudzień 1992.
- [13] C. Zieliński, A. Grodecki, U. Kręglewska, J. Sobczyk, A. Śluzek, T. Zielińska; *Speyfikacja sterownika robotów przeznaczonego do celów badawczych*. Opracowanie dla KBN, Inst. Aut. P.W., Warszawa, grudzień 1992.



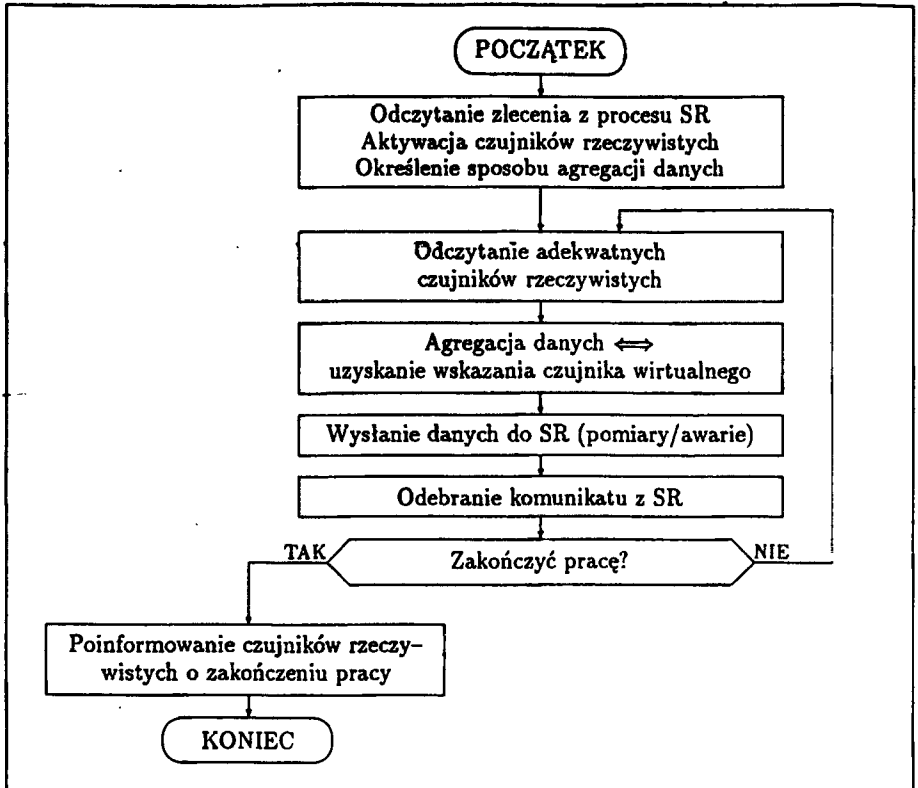
Rys. 1: Struktura sprzętowa systemu



Rys. 2: Warstwowa struktura systemu

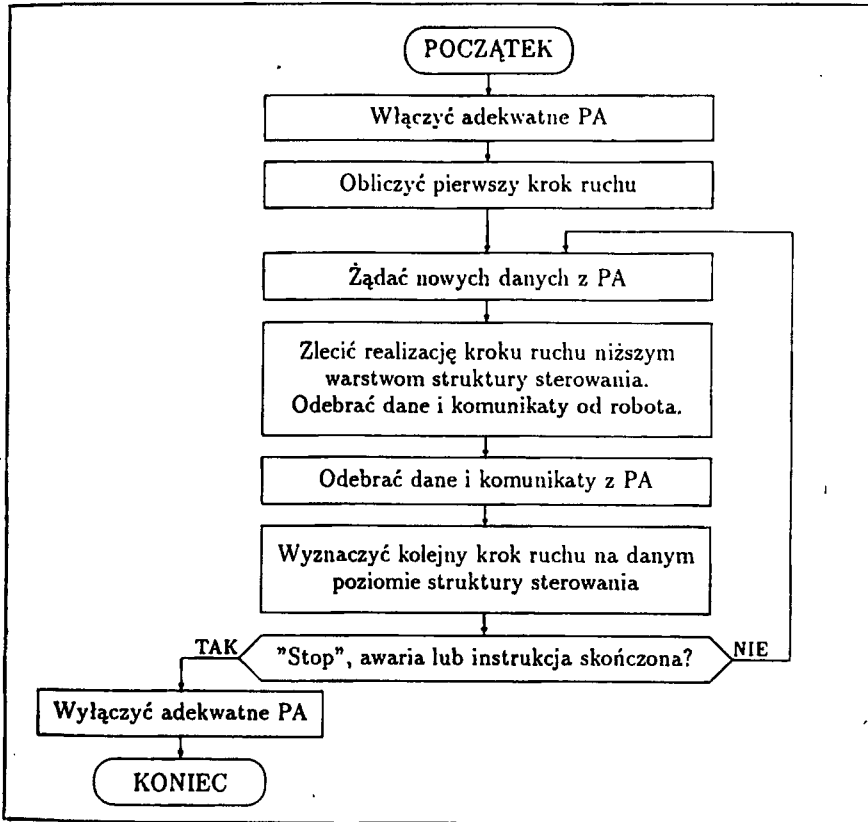


Rys. 3: Struktura programowa systemu

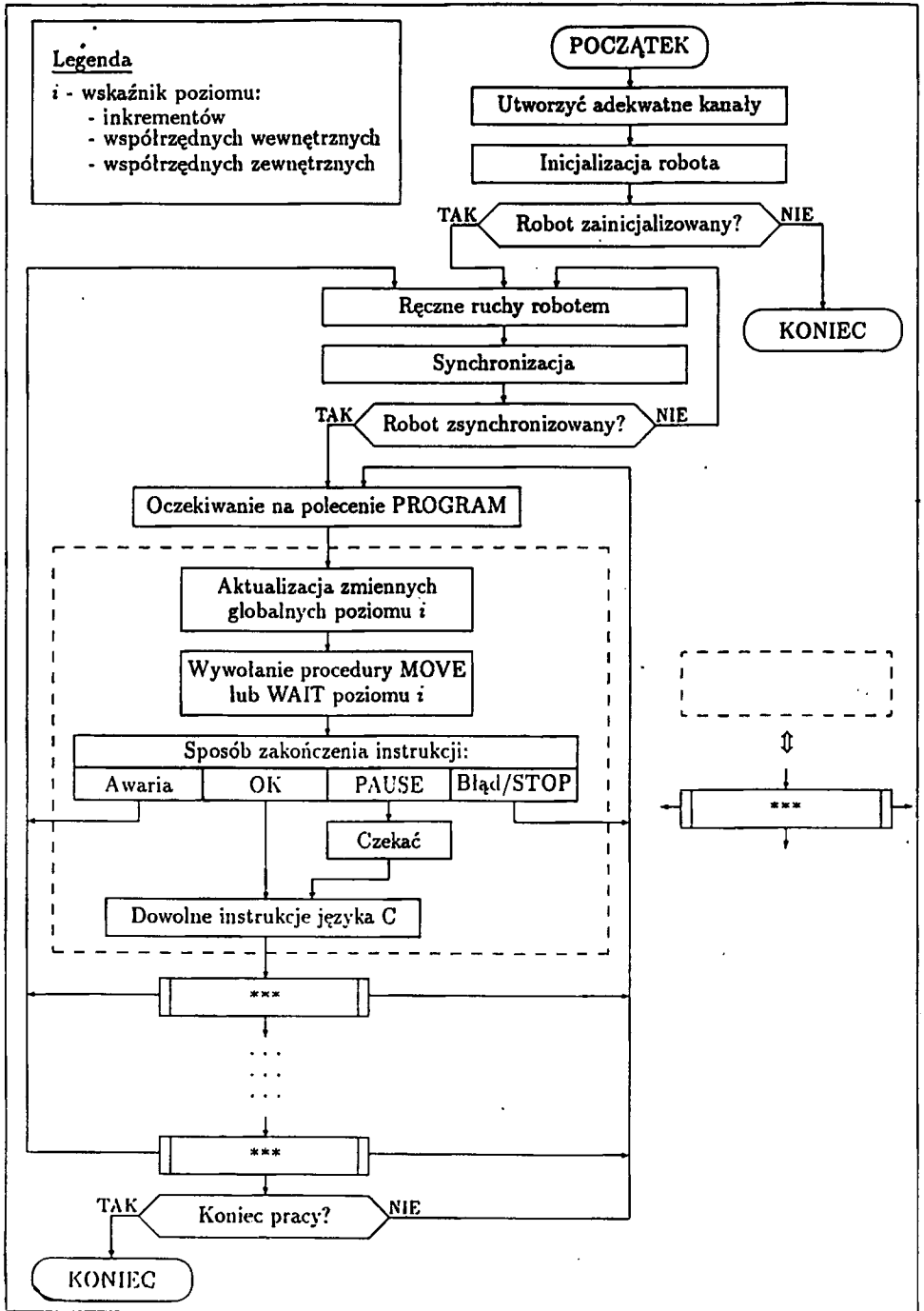


Rys. 4:

Ogólna postać procesu agregacji danych z czujników rzeczywistych (PA)



Rys. 5: Ogólna postać procedury ruchu



Rys. 6: Sposób konstruowania procesu SR