ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ
Seria: AUTOMATYKA z.96 Nr kol.972

Jarosław Warczyński, Antoni Woźniak
Politechnika Poznańska

ZASTOSOWANIE GRAFIKI KOMPUTEROWEJ W PROCESIE ROBOTYZACJI GNIAZDA OBRÓBCZEGO

Streszczenie. W referacie przedstawiono system graficznej symulacji robotów przemysłowych o dowolnej strukturze kinematycznej oraz jego zastosowanie do programowania robotów w trybie off-line.

1. Wstęp

We współczesnych systemach produkcyjnych rośnie zapotrzebowanie na gniazda obróbcze złożone z obrabiarek, urządzeń peryferyjnych i robotów, które stanowiłoby zintegrowaną jednostkę funkcjonalną. Jednostki takie stanowią bowiem podstawę tworzenia systemów komputerowo zintegrowanego wytwarzania CIM /ang. Computer Integrated Manufacturing/ oraz elastycznych systemów produkcyjnych. W ramach systemów tego typu gniazda obróbkowe są planowane i programowane w kontekście jednolitego modelu danych, odnoszącego się do całego systemu wytwórczego, w którym istnieje określona organizacja przepływu i przetwarzania informacji i danych. Warunkiem takiej organizacji jest scalenie procesów CAD/CAM oraz sterowania robotów [1,2,3]. Jednym z istotnych elementów tej struktury jest system interaktywnego planowania i programowania wspomnianych jednostek funkcjonalnych. Obejmuje on utworzenie modelu gniazda obróbczego, który umożliwiałby emulację planowanych operacji wytwórczych za pomocą komputera. Pozwala to obserwować przebieg planowanego procesu bez potrzeby eksperymentowania z urządzeniami fizycznymi. Można w ten sposób łatwo zweryfikować różne warianty przebiegu procesu w różnych konfiguracjach gniazda. Można tą drogą ustalić również potencjalne zakłócenia, kolizje, wąskie gardła oraz niektóre wielkości, jak np. spowolnienia /opóźnienia/ lub przyspieszenia cyklu wytwórczego.

W pracy przedstawiono analogiczny system programowania robotów przemysłowych o strukturze otwartych łańcuchów kinematycznych o ogniwach sztywnych, przeznaczony do redagowania i weryfikacji programów działań robota w trybie off-line, tj. bez udziału robota fizycznego, z zastosowaniem graficznej symulacji gniazda obróbkowego.

2. Struktura systemu

Prezentowany system stanowi integralną część znacznie większego systemu przeznaczonego również do symulacji robota z uwzględnieniem zjawisk dynamiki ruchu, działania sensorów i układów wykonawczych oraz sterowania robota [4]. Prace nad takim systemem są prowadzone w Instytucie Automatyki Politechniki Poznańskiej w ramach CPBR 7.1 "Roboty przemysłowe". System ten będzie miał budowę zintegrowanego pakietu programów, w którym można wyróżnić sześć następujących modułów programowych :

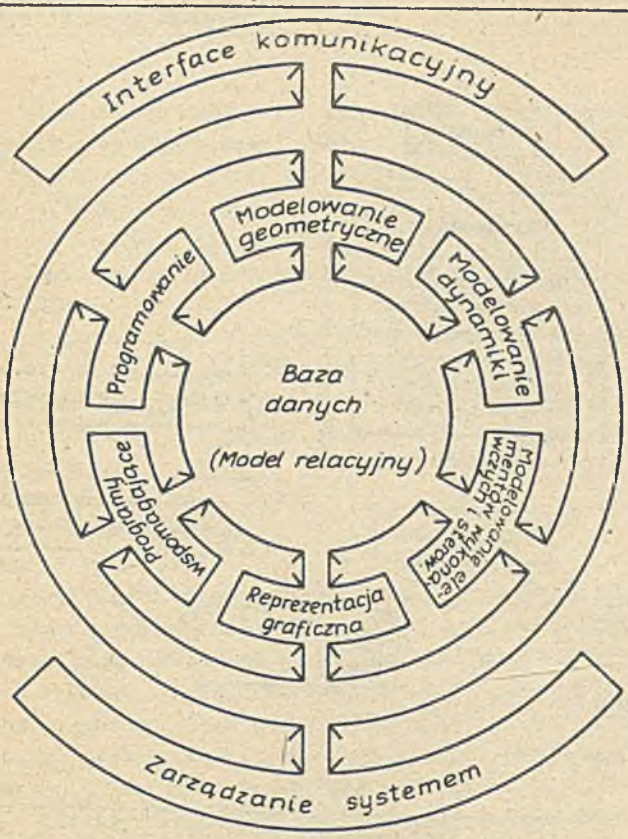
- moduł programowania,
- edytor graficzny,
- moduł prezentacji graficznej,
- moduł programów wspomagających,
- moduł modelowania elementów wykonawczych, sensorów i układów sterowania,
- moduł generacji modelu dynamiki.

Moduły te korzystają ze wspólnej, systemowej bazy danych oraz komunikują się z użytkownikiem poprzez interfejs komunikacyjny pod kontrolą programu zarządzającego systemem /rys.1/.

Zastosowanie koncepcji modularności oraz bazy danych zapewnia otwartość systemu dzięki uzyskanej w ten sposób niezależności danych, co oznacza, że system może być rozwijany bez konieczności przebudowy powstałych wcześniej modułów nawet w przypadku zmiany struktury pamięci lub strategii dostępu do danych. Z drugiej strony, możliwe jest również wykorzystywanie tylko niektórych modułów systemu, tworzących pewną całość z punktu widzenia rozwiązywanego problemu. Zwróćmy uwagę, że pierwsze cztery spośród wymienionych modułów stanowią właśnie taki podsystem, który może być wykorzystany do symulacji na poziomie gniazda obróbkowego, umożliwiając graficzną weryfikację organizacji gniazda oraz zachodzących w nim procesów typu przepływ materiałów, operacje załadunkowe i wyładunkowe. Pozwala on również na uruchamianie programu działań robota w trybie off-line. Dopiero dekompozycja zadań tego poziomu wymaga uwzględnienia zagadnień dynamiki ruchu manipulatora, sterowania napędów itp., które są rozpatrywane na poziomie ramienia robota.

3. Moduł programowania

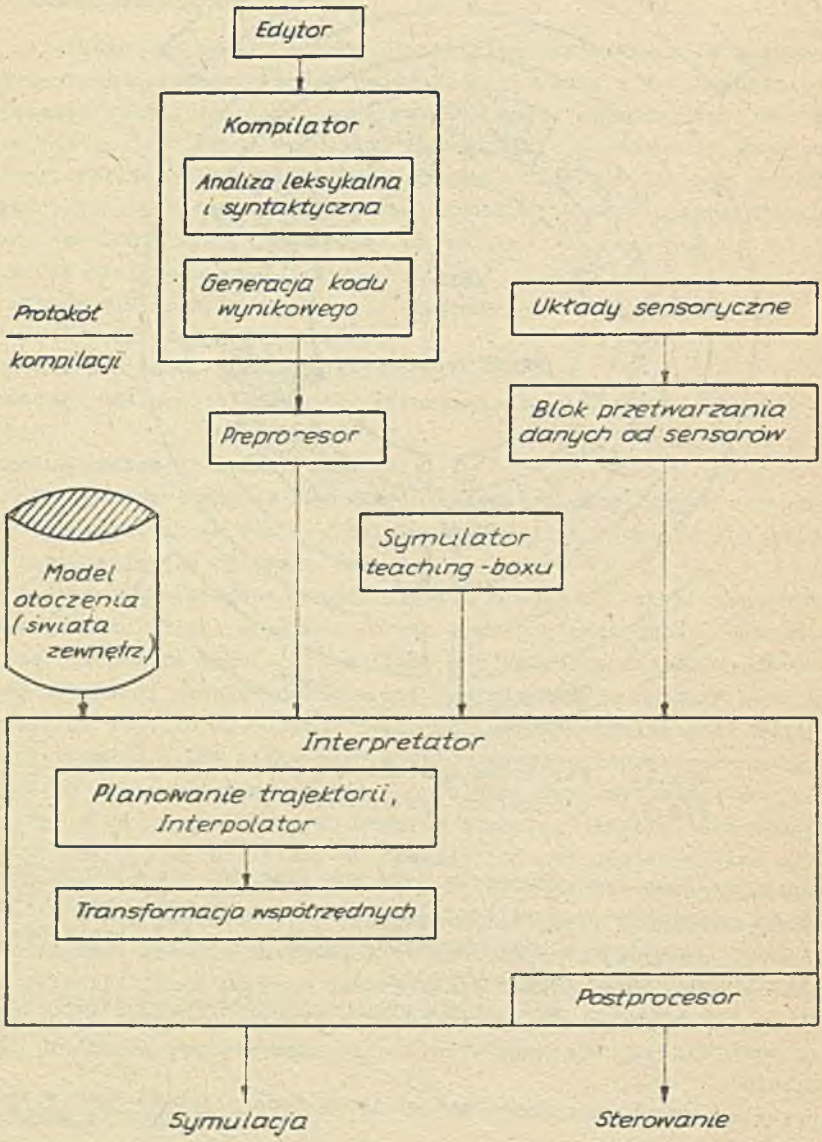
Ma on umożliwiać programowanie robota w sposób tekstowy, tj. poprzez zredagowanie programu jego działań w języku wysokiego poziomu lub w ramach metody teach-in poprzez zadanie sekwencji rozkazów, wprowadzanych za pomocą symulatora teaching-boxu. Program może być również zestawiony w wyniku połączenia obu tych sposobów programowania, co niekiedy znacznie upraszcza jego zadanie. Moduł programowania łączy w sobie dwa sys-



Rys.1. Struktura systemu symulacji
Fig.1. Simulation system structure

temy, mianowicie język programowania oraz run-time system, który interpretuje kody wygenerowane przez kompilator języka i steruje bądź inicjuje wykonanie zawartych w nich rozkazów. Moduł ten składa się z następujących bloków funkcjonalych /rys.2/ :

- edytora umożliwiającego redagowanie programu źródłowego w języku wysokiego poziomu,
- kompilatora,
- preprocesora, przetwarzającego kod wynikowy danego kompilatora w kod pośredni,
- interpretatora,
- interpolatora,
- bloku transformacji współrzędnych zewnętrznych /współrzędnych przestrzeni roboczej/ we współrzędne konfiguracyjne,
- symulatora teaching-boxu oraz
- bloku przetwarzania danych od sensorów.



Rys.2 Moduł programowania
Fig.2. Programming module

Preprocesor umieszczony na wyjściu kompilatora języka wysokiego poziomu przetwarza kod wynikowy kompilatora w kod pośredni, wzorowany na kodzie IRDATA [5]. Podejście takie uczyni omawiany system programowania i symulacji uniwersalnym. Oznacza to, że system będzie mógł być wykorzystany do programowania w dowolnym języku oraz że ten sam język będzie mógł być wykorzystany do programowania różnych robotów. Właśnie te zalety oferuje IRDATA.

Wspomniany kod pośredni przedstawia zbiór rekordów, z których każdy charakteryzuje się określoną liczbą atrybutów, zależną między innymi od typu danego rekordu. Występują tutaj typy przeznaczone do :

- opisu robota, narzędzi, informacji od sensorów, przestrzeni roboczej, położeń i orientacji,
- określenia sposobu wykonania ruchu,
- sterowania wykonaniem programu,
- sterowania operacjami we/wy.

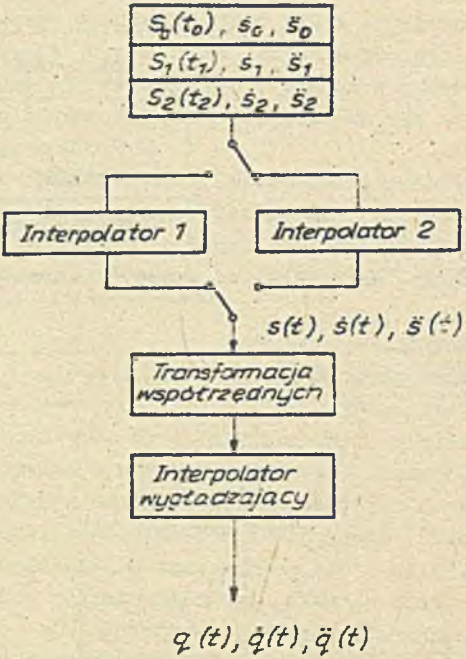
Każdy rekord składa się z ciągu wartości jego atrybutów A1,A2,...,An rozdzielonych separatorami. Wartościami atrybutu A1 są zawsze numery porządkowe poszczególnych rekordów. A2 jest zawsze atrybutem głównym, tj. identyfikatorem rekordu, którego wartość określa między innymi typ rekordu. Wartości atrybutów A1-An podają dane dopełniające atrybutu głównego. Przykładowo, rekord określający sposób zmiany orientacji efektora końcowego ma następującą strukturę syntaktyczną :

⟨Nr rekordu⟩ , ⟨Identyfikator rekordu⟩ , ⟨Rodzaj zmiany orientacji⟩

lub

⟨A1⟩ , ⟨A2⟩ , ⟨A3⟩

Interpretator kodu pośredniego dzieli się funkcjonalnie na dwie części : część wykonującą zadania niezależne od konkretnej implementacji robota oraz część do wykonania zadań bezpośrednio od niej zależnych. Ta ostatnia wymaga obecności postprocesora współpracującego ze sterownikiem robota. W ramach zadań niezależnych interpretator inicjuje interpolację trajektorii. Jest ona dokonywana za pomocą wielomianów interpolacyjnych na podstawie zadanych punktów węzłowych trajektorii. Ponieważ ze wzrostem liczby zadanych punktów węzłowych trajektorii gwałtownie rośnie czas obliczeń, można zastosować tutaj podejście polegające na tym, że trajektorię interpoluje się tylko pomiędzy dwoma sąsiednimi punktami, a następnie łączy się uzyskane odcinki trajektorii. Można tego dokonać stosując dwa współpracujące interpolatory; podczas gdy jeden oblicza kolejny odcinek trajektorii, drugi wyprowadza aktualne dane dla trajektorii. Po transformacji uzyskanych punktów trajektorii możliwe jest ostateczne jej wygładzenie już we współrzędnych konfiguracyjnych poprzez zastosowanie jeszcze jednego interpolatora. Proces ten ilustruje rys.3.



Rys.3 Interpolacja trajektorii
Fig.3. Trajectory interpolation

Blok transformacji współrzędnych stanowią procedury rozwiązania odwrotnego zadania kinematyki manipulatora, tj. procedury wyznaczania wzajemnego przesunięcia lub obrotu dla wszystkich półpar kinematycznych w łańcuchu kinematycznym manipulatora, odpowiadających osiągnięciu przez efektor końcowy robota zadanego punktu trajektorii jego ruchu, określonego we współrzędnych przestrzeni zadaniowej /roboczej/.

4. Edytor graficzny

Programowanie robota w trybie off-line oraz planowanie organizacji gniazda obróbkowego z wykorzystaniem symulacji wymaga istnienia odpowiednich modeli, uwzględniających wszystkie te aspekty modelowanych obiektów, które są istotne z punktu widzenia zadań symulacji. Jednocześnie możliwa jest znacząca redukcja w modelu tych elementów rzeczywistości, które nie wpływają na wynik postawionego zadania. Fakt ten ma is-

totne znaczenie z punktu widzenia modelowania, gdyż pozwala posługiwać się modelami informacyjnymi wyabstrahowanymi ze złożonych obiektów fizycznych z zastosowaniem jedynie cech interesujących dla konkretnego zastosowania.

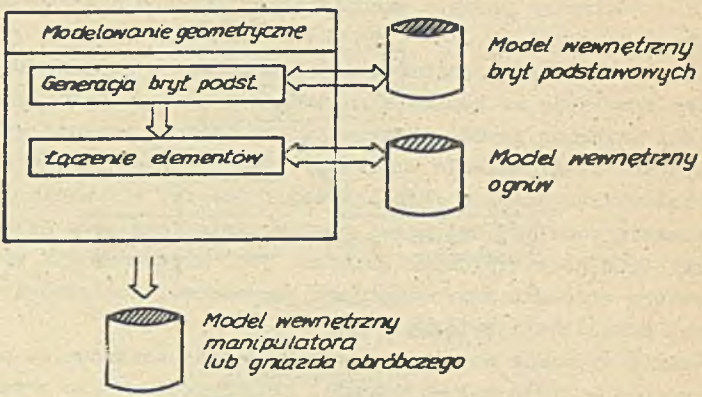
Dokonując weryfikacji programów działań robota w zaplanowanym otoczeniu poprzez symulację na poziomie gniazda obróbkowego można posługiwać się modelami informacyjnymi odwzorowującymi jedynie aspekty kinematyczne i geometryczne. Uwzględnienie struktury kinematycznej jest niezbędne w celu przedstawienia położeń /konfiguracji/ robota, w trakcie jego działania /animacja modelu/, natomiast odwzorowanie jego cech geometrycznych, jak również cech geometrycznych gniazda, jest niezbędne dla zobrazowania ich wzajemnego stosunku przestrzennego, co stanowi podstawową przesłankę ostatecznej organizacji gniazda.

W omawianym systemie modele geometryczne będą tworzone za pomocą specjalnego modułu - edytora graficznego. Proces modelowania robota i gniazda przebiega tutaj w sposób interaktywny i może być podzielony na dwie zasadnicze fazy. W fazie pierwszej użytkownik tworzy modele graficzne brył niezbędnych do zamodelowania poszczególnych ogniw robota. Są to modele 2,5 wymiarowe, które obrazują jedynie krawędzie reprezentowanych obiektów /ang. wire frame model/. Bryły te tworzone są na bazie wielokątów, co pozwala utworzyć wielościany. Pewien ograniczony zbiór podstawowych obiektów geometrycznych /ang. primitives/ jest podany do stałej dyspozycji użytkownika. Ich wymiary są sparametryzowane, co pozwala na ich dowolne ustalanie. "Sklejając" odpowiednio dobrane bryły, użytkownik może wygenerować modele poszczególnych ogniw manipulatora. Dane dotyczące tych modeli zostaną zapamiętane w odpowiednich strukturach listowych, których zawartością będą współrzędne charakterystycznych punktów, tj. takich, które po połączeniu odcinkami prostych, zgodnie z zapamiętaną topologią, dadzą w rezultacie obraz utworzonego modelu ogniwa. Dane geometryczne obiektów odnoszą się do ich lokalnych układów współrzędnych. Manipulowanie obiektem sprowadza się więc do manipulowania jego układem współrzędnych.

Druga faza modelowania polega na połączeniu modeli ogniw w jeden model manipulatora. Szeregowe łączenie poszczególnych ogniw zachodzi w procesie definiowania zmiennych opisujących wzajemny ruch ogniw. Łączenie odbywa się od chwytaka do podstawy. Rys.4 przedstawia strukturę edytora graficznego.

5. Moduł prezentacji graficznej

Moduł ten zapewnia właściwą prezentację graficzną informacji wyjściowej systemu. Do ważniejszych funkcji tego modułu należą te, które wyni-



Rys.4. Struktura modułu modelowania geometrycznego
Fig.4. Graphical editor structure

kają z zadania trójwymiarowej /ściśle 2,5-wymiarowej/ prezentacji modelu geometrycznego robota i jego otoczenia. Należą do nich :
- transformacje układów współrzędnych,
- przetwarzanie hierarchii połączeń elementów,
- aktualizacja,brył, płaszczyzn, linii i punktów /animacja/,
- clipping,
- lokalizacja okienek /windowing/,
- usuwanie ukrytych linii,
- inicjalizacja procedur graficznych.

Omówimy tutaj w skrócie jedynie animację modelu geometrycznego. Koncepcja tego procesu jest następująca : Struktury danych graficznego modelu robota składają się z list, których pola zawierają współrzędne charakterystycznych punktów elementów modelu /analogicznie do struktur danych stosowanych w systemach CAD/. Pola danych odpowiadające liczbie oraz kolejności ogniw robota są ponumerowane we wzrastającym porządku. W celu animacji modelu, zawartości tych pól zostaną poddane transformacji, po czym nowe współrzędne zostaną zapisane w miejsce starych. Istotny jest tutaj porządek dokonywania transformacji - powinny się one dokonywać w kolejności malejących numerów pól. Oznacza to, że najpierw zostanie dokonana transformacja współrzędnych efektora końcowego, a na końcu transformacja współrzędnych podstawy. Przedstawienie ruchu chwytaka

względem kolejnych ogniw /o coraz niższych numerach/ następuje w wyniku transformacji jego współrzędnych względem układów współrzędnych poszczególnych ogniw. Procedura ta powtarza się aż do podstawy manipulatora.

Moduł prezentacji graficznej współpracuje ściśle z edytorem graficznym.

6. Baza danych

Koncepcja bazy danych została oparta na nieznormalizowanym modelu relacyjnym, tj. modelu, w którym atrybuty relacji mogą z kolei być relacjami. W takim modelu danych dziedziną atrybutu może być zarówno atomowy typ danych /nierozkładalny/, jak np. liczbowy bądź znakowy, jak również strukturalny - rekord, lista, relacja. Rekord odpowiada rekordowi w znormalizowanym modelu relacyjnym. Lista jest uporządkowanym zbiorem elementów, które z kolei mogą być strukturalne lub atomowe. Natomiast relacje pełniące rolę wartości atrybutu mogą ponownie być nieznormalizowane, tj. również mogą zawierać listy i relacje jako wartości swoich atrybutów. Jednocześnie wykorzystywana jest koncepcja tzw. abstrakcyjnych typów danych - ADT-/ang. abstract data types/. Abstrakcyjne typy danych służą jako środek strukturalizacji danych dając możliwość zamknięcia pewnego zbioru danych w określonym pojęciu, którym można potem manipulować nie odwołując się już do zawartych w nim detali. Abstrakcyjny typ danych powinien być równoprawny z typami danych przyjętymi w modelu bazy, co oznacza, że jest on zdefiniowany następująco :

$$\langle \text{adt identifier} \rangle = \langle \text{relation type} \rangle \,|\, \langle \text{list type} \rangle \,|\, \langle \text{tuple type} \rangle$$

Zalety takiej definicji ADT jest to, że może on być użyty wszędzie tam, gdzie stosowane są podstawowe typy danych. W szczególności może więc być zastosowany do definicji danych. Np. relacja ROBOT składa się z atomowego atrybutu identyfikującego, który zawiera nazwę i opis typu robota oraz z atrybutów RAMIĘ i EFEKTOR KOŃCOWY. Atrybut EFEKTOR KOŃCOWY jest sam jednocześnie typu relacji, gdyż robot, z reguły, może być wyposażony w wiele efektorów końcowych. Ramię robota jest odwzorowane w relacji ROBOT przez listę rekordów oddającą uporządkowanie kolejnych ogniw robota. Każdy rekord tej listy składa się z atrybutów GEOMETRIA, KINEMATYKA i DYNAMIKA, których wartości charakteryzują poszczególne ogniwa robota.

LITERATURA

[1] Rembold U.,Blume Ch,Dilmann R. : Computer - Integrated Manufacturing Technology and Systems, M.Dekker Inc. New York, Basel 1985.

[2] Smith R.M. : Computer-Aided Fully Generative Process Planning, Manufacturing Engineering, May 1981.

[3] Woźniak A. : Roboty współczesnym środkiem automatyzacji produkcji, Wykład Inauguracyjny Polit.Poznańskiej /w/ Informator i skład osobowy na rok akademicki 1987/88, Wydawn.Pol.Poz. Poznań. 1987.

[4] Warczyński J. : Założenia dla komputerowego systemu symulacji kinematyki i dynamiki robotów, Raport Inst.Automatyki Polit. Poznańskiej, . Poznań 1987.

[5] DIN/VDT 2863 : IRDATA - programming of numerically controled handling devices, general structure, types of records and data transfer, Part 1 ISO/TC 184N 37 document, 1984.

Recenzent: Prof.dr inż.H.Kowalowski

Wpłynęło do Redakcji do 1988-04-30.

ПРИМЕНЕНИЕ МАШИННОЙ ГРАФИКИ В РОБОТИЗИРОВАННОЙ ОБРАБАТЫВАЮЩЕЙ ЯЧЕЙКЕ

Р е з ю м е

В статье предоставлена автономная система для программирования и эмуляции роботизированной обрабатывающей ячейки. Она состоит из базы данных и четырёх программных модулей содействующих друг с другом под управлением программы управления системы.

Система даёт возможность независимых отладок и проверки запрограммированных действий робота в обрабатывающей ячейке а также разметки самой ячейки без привлечения какого-либо реального оборудования . Для этого применяется эмуляция , использующая движимые графические модели , изображающие аксонометрически контуры имитируемых объектов, составляемые при помощи специальной программы графического редактора.

COMPUTER GRAPHICS APPLICATION FOR ROBOT BASED MANUFACTURING CELLS

S u m m a r y

The paper describes a robot off-line programming and simulation system for robot based manufacturing cells. It consists in a data base and four program modules cooperating with each other under the control of the systems management program.

The System enables debugging and veryfying the programmed robot actions within a manufacturing cell as well as the layout of a cell itself. For this purpose the simulation process using the animated cell- and robot wire-frame graphical models, composed with the aid of a special graphical editor instead of any real hardware can be applied.