ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1994
Seria: AUTOMATYKA z.115 Nr kol. 1251

Krzysztof ROMANOWSKI, Waldemar WRÓBLEWSKI
Katedra Automatyki, Robotyki i Informatyki, Politechnika Poznańska

SERWER TRANSPUTEROWY NA POTRZEBY STEROWANIA ROBOTA ZEBRA ZERO

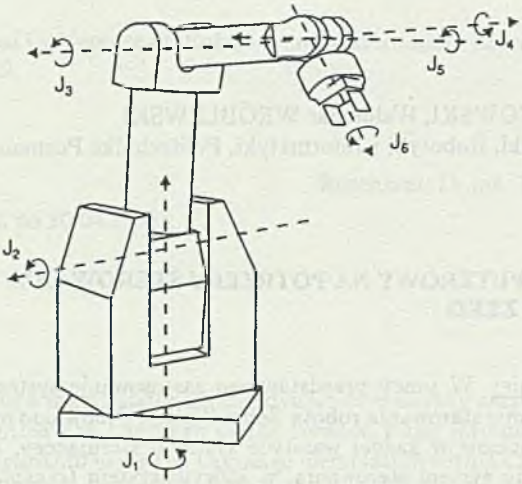
Streszczenie: W pracy przedstawiono zastosowanie systemu transputerowego Fast9 w systemie sterowania robota Zebra ZERO. Omówiono możliwości wykorzystania transputerów w każdej warstwie systemu sterującego, a następnie opisano zmodyfikowany system sterowania, w którym system transputerowy rezyduje na magistrali komputera IBM PC. Szczególną uwagę poświęcono serwerowi transputerowemu i bibliotece funkcji robota.

TRANSPUTER SERVER FOR THE ZEBRA ZERO ROBOT CONTROL SYSTEM

Summary: The paper presents an application of the Fast9 transputer system for the Zebra ZERO robot control system. Methods of introducing the transputers at each layer of the control system are discussed. The modified control system, in which the transputer system resides on the PC bus, is described. Special emphasis is given to the transputer server program, and the parallel C wrapper function library.

TRANSPUTERSERVER FÜR DAS SYSTEM STEUERUNGS DES ZEBRA ZERO ROBOTERS

Zusammenfassung: In diesem Bericht wird die Anwendung des Fast9-Transputersystems in dem Steuerungssystem des Zebra ZERO Roboters, präsentiert. Die Möglichkeiten der Anwendung von Transputern in jeder Lage des Steuerungssystem wird diskutiert. Dann wird das modifizierte Steuerungssystem, in dem das Transputersystem an der PC-Bus residiert, beschrieben. Eine spezielle Aufmerksamkeit ist auf das Serverprogramm des Transputers und die Instruktionenbibliothek des Roboters gelenkt.



Rys. 1. Robot Zebra ZERO
Fig. 1. Zebra ZERO robot

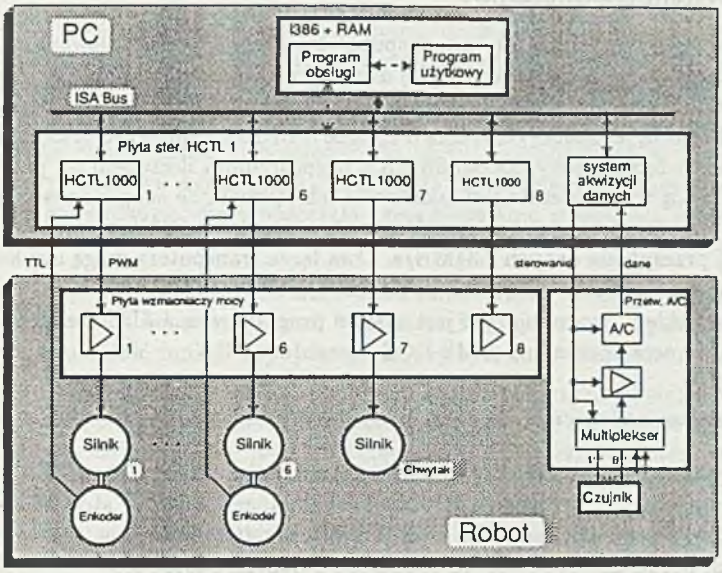
1. Wstęp

W niniejszej pracy przedstawiono zmodyfikowany system sterowania robota Zebra ZERO oparty na transputerach. Koncepcja implementacji transputerów w systemie sterowania robota wykorzystuje zalety zarówno systemu transputerowego – wieloprocesorowego, o dużej mocy obliczeniowej i łatwo rekonfigurowalnego, dającego potencjalne możliwości wykonania wielu obliczeń w zewnętrznej pętli sterowania – jak i istniejącego sterownika robota Zebra ZERO, cechującego się zintegrowaniem większości funkcji obsługi silników, enkoderów i czujnika siły/momentu na jednej płycie oraz uniwersalnym interfejsem sprzętowo-programowym od strony komputera nadrzędnego.

2. System wyjściowy

2.1. Robot Zebra ZERO

Robot Zebra ZERO, przedstawiony na rys.1, jest małym robotem o sześciu rotacyjnych stopniach swobody [3]. Charakteryzuje się udźwigiem 1kg, zasięgiem liczonym od środka ramienia do palców chwytaka 0.64m i maksymalnej prędkości obrotowej w węzłach 150deg/s. Każdy węzeł wyposażony jest w silnik prądu stałego i enkoder optyczny. Chwytak jest otwierany i zamykany za pomocą silnika prądu stałego. W nadgarstku robota umieszczony jest czujnik siły/momentu o sześciu stopniach swobody.



Rys. 2. Hierarchiczna struktura systemu sterowania robota Zebra ZERO
Fig. 2. Hierarchical structure of the Zebra ZERO robot control system

System sterowania robota składa się z trzech hierarchicznych warstw (rys.2):

1. Warstwa najniższa, bezpośrednio kontaktująca się z manipulatorem, zawiera wzmacniacze mocy oraz system akwizycji danych sensorycznych, składający się z multipleksera analogowego wybierającego jeden z sześciu sygnałów analogowych czujnika siły/momentu, wzmacniacza pomiarowego i 10-bitowego przetwornika analogowo-cyfrowego. Wszystkie te zespoły zamontowane są w obudowie manipulatora.

2. Warstwę pośrednią stanowi sterownik HCTL 1 zrealizowany na płycie rozszerzającej do komputera typu PC. Zasadnicze funkcje sterowania wykonywane są tu przez osiem układów scalonych HP HCTL 1000. Ponadto znajduje się tu układ sterujący systemem akwizycji danych sensorycznych.

3. Warstwę najwyższą stanowi komputer typu PC/AT-386, w którym umieszczono sterownik HCTL 1 wraz z programem obsługi komunikującym się z jednej strony z rejestrami sterownika, z drugiej zaś z operatorem (użytkownikiem) lub programem nadrzędnym na poziomie poleceń określających żądane ruchy poszczególnych osi lub wyższym. Środowiskiem programowym jest system programowania Microsoft C, działający w systemie operacyjnym MS-DOS.

2.2. System transputerowy

W pracach wykorzystano płytę transputerową Fast9 [1] zainstalowaną w komputerze typu PC/AT-386. Znajduje się na niej dziewięć procesorów INMOS T805-25MHz, każdy z pamięcią lokalną DRAM 4MB. Procesor T805 stanowi układ scalony VLSI zawierający m.in. 32-bitową jednostkę centralną o zbiorze instrukcji typu RISC oraz cztery interfejsy szeregowych łącz międzyprocesorowych z bezpośrednim dostępem do pamięci (DMA). Istotną cechą transputerów jest możliwość ich łączenia ze sobą poprzez łącza zapewniające transmisję szeregową niebuforowaną o prędkości 20Mbity/s, dające efektywną prędkość przesyłania danych 2.34MB/s. Dwa łącza transputera mogą być konfigurowane programowo; pozostałe połączone są na stałe.

Środowiskiem programowym jest system programowania 3L Parallel C współpracujący z systemem operacyjnym MS-DOS komputera PC.

3. Zastosowanie transputerów w systemie sterowania robota

Wykorzystanie transputerów w systemie sterowania robota Zebra ZERO może polegać na przekazaniu im pewnych funkcji oryginalnego systemu sterowania lub funkcji nowych, wyższego poziomu [2]. Rozważmy użycie transputerów w kolejnych hierarchicznych warstwach systemu sterowania omówionych poprzednio:

1. W najniższej warstwie systemu sterowania wymagane są proste operacje na sygnałach analogowych, co wyklucza zarówno możliwość,jak i celowość zastosowania transputerów.

2. W pośredniej warstwie systemu sterowania interfejs od strony warstwy niższej wykorzystuje sygnały o zmiennym współczynniku wypełnienia (PWM), wymagające specjalnych układów. Istniejące układy HCTL 1000 są tu najlepszym rozwiązaniem; realizują ponadto sterowanie niskiego poziomu. Wprowadzenie transputerów do tej warstwy może więc polegać na zastąpieniu płyty sterownika HCTL 1 specjalnie w tym celu skonstruowanym sterownikiem, w którym układy HCTL 1000 kontaktują się z jednej strony z warstwą niższą, z drugiej zaś strony, z umieszczonymi dodatkowo na płycie sterownika adapterami łącz transputerowych typu INMOS C011 lub C012. Adaptery te z kolei umożliwiają połączenie płyty sterownika poprzez łącza transputerowe z płytą transputerów - bez pośrednictwa magistrali komputera PC. Połączenie takie może być:

   (a) osobne dla każdego kanału sterowania (odpowiadającego osi manipulatora), co wymaga osobnego adaptera łącz dla każdego kanału i przynajmniej tylu samo dostępnych łącz na płycie transputerów; płytą tą musi być wobec tego płyta Fast9;

   (b) wspólne dla wszystkich kanałów, multipleksowane na płycie sterownika, co wymaga tylko jednego (wspólnego) adaptera łącz i jednego dostępnego łącza na płycie transputerów; płytą tą może więc być również płyta Fast1XL.

3. Najwyższa warstwa systemu sterowania komunikuje się z warstwą pośrednią przez zapis i odczyt danych w przestrzeni adresowej komputera PC. Dostęp transputerów do tych danych odbywa się poprzez adapter łącz na płycie transputerów i magistralę zewnętrzną komputera. Rozwiązanie to nie wymaga przeróbek sprzętowych systemu sterowania (poza zainstalowaniem płyty transputerowej w komputerze; może to być zarówno płyta Fast9, jak i Fast1XL). System transputerowy może przy tym komunikować się z pozostałą częścią systemu sterowania na jednym z dwóch poziomów:

   (a) zapisując i odczytując rejestry sterownika HCTL 1 poprzez dostęp do odpowiednich adresów pamięci komputera PC; system transputerowy przejmuje tu całość funkcji najwyższej warstwy systemu sterowania;

   (b) wywołując odpowiednie funkcje oryginalnego programu obsługi, wykonującego się na komputerze PC; system transputerowy stanowi tu dodatkową warstwę, nadbudowaną na oryginalnej najwyższej warstwie systemu sterowania.

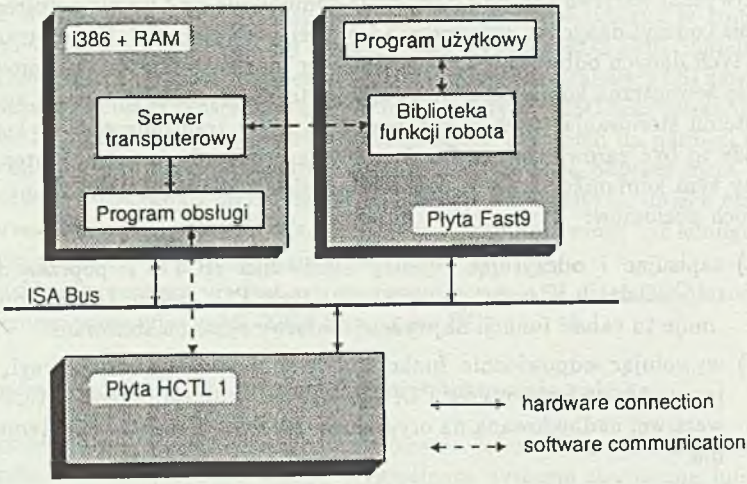
   Dostęp programowy transputerów do zasobów komputera PC można oprzeć na istniejących w systemach programowania 3L mechanizmach; można też zastąpić standardowy program-serwer transputerowy `afserver`, wykonujący się na komputerze PC i pośredniczący między komputerem a transputerami, wersją opracowaną specjalnie dla danego zadania.

Zaletą wariantu 2 jest ominięcie magistrali komputera i wprowadzanego przez nią opóźnienia oraz co ważniejsze, w przypadku 2a, kilkukrotne zwiększenie przepustowości między układami HCTL 1000 a transputerami. Istotną wadą jest natomiast konieczność budowy nowego sterownika w miejsce istniejącego HCTL 1.

Z kolei wariant 3 nie wymaga żadnych przeróbek sprzętowych. Mniejsza przepustowość nie wydaje się tu istotnym problemem, wziąwszy pod uwagę parametry czasowe sterowniką robota, który potrzebuje przynajmniej 30μs na dokonanie pojedynczej konwersji analogowo-cyfrowej i 150μs na przełączenie kanałów czujnika; czas pomiędzy kolejnymi odczytami powinien być wystarczający zarówno do obliczeń, jak i transmisji danych. Ten właśnie wariant przedstawiono poniżej.

4. Zmodyfikowany system sterowania

Zasadnicze elementy zmodyfikowanego systemu sterowania przedstawiono na rys.3. Zrealizowane zostały warianty 3a i 3b omówione wyżej; odpowiadają im dwa tryby dostępu do układów HCTL: pośredni i bezpośredni. W początkowej implementacji systemu zastosowano wariant 3b – tryb pośredniego dostępu do HCTL, z uwagi na to, iż pozwala on wykorzystać oryginalny program obsługi HCTL. Następnie wprowadzono wariant 3a – tryb bezpośredniego dostępu do HCTL, umożliwiający zastąpienie sekwencyjnych obliczeń kinematyki i trajektorii wykonywanych na procesorze komputera PC ich wersją równoległą wykonywaną na transputerach.



Rys. 3. Zmodyfikowany system sterowania
Fig. 3. Modified control system

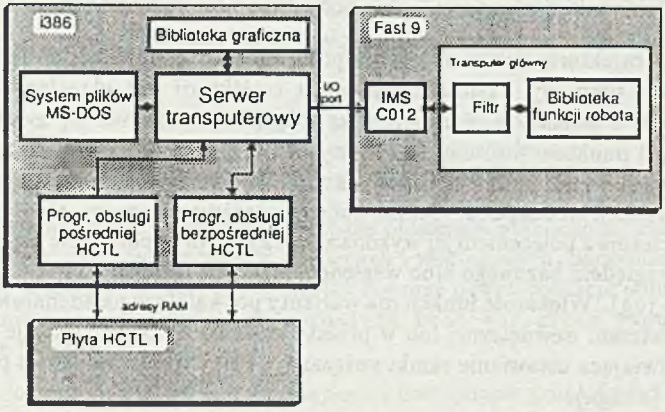
4.1. Programy obsługi HCTL

Dostęp do płyty HCTL 1 odbywa się poprzez wywołanie funkcji programu obsługi HCTL, wykonującego się na procesorze komputera PC. W trybie pośredniego dostępu do HCTL wywoływane są funkcje oryginalnego programu obsługi. Z kolei w trybie bezpośredniego dostępu do HCTL wykorzystywana jest zmodyfikowana wersja programu obsługi. Program ten otrzymuje polecenia dla płyty HCTL 1 odnoszące się do położeń, prędkości i przyspieszeń w poszczególnych węzłach manipulatora i dokonuje odpowiednich ich konwersji - skalowania i operacji na poszczególnych bajtach i bitach - do postaci wymaganej przez układy HCTL 1000.

4.2. Serwer transputerowy

Serwer transputerowy jest programem realizującym interfejsy ze środowiskiem zewnętrznym dla programów aplikacyjnych wykonujących się na transputerach. Komunikuje się on z czterema podsystemami (rys.4). Są to:

1. System transputerowy. Interfejs ten oparty jest na typowym połączeniu między transputerem a komputerem PC za pomocą adaptera łącza INMOS C012, zainstalowanego na płycie transputerowej, podłączonego do łącza 0 transputera głównego i odwzorowanego w przestrzeń adresową portów wejścia/wyjścia komputera PC od adresu 150H. Komunikaty między serwerem a transputerem głównym (konkretnie zadaniem `filter` na tym transputerze, które ma dostęp do łącza 0) są przesyłane przy użyciu przez serwer funkcji dostępu do portów wejścia/wyjścia



Rys. 4. Interfejsy serwera transputerowego
Fig. 4. Transputer server program interfaces

(inp and outp).

2. Płyta HCTL 1. W zależności od aktualnego trybu dostępu do HCTL, komunikacja realizowana jest w trybie dostępu pośredniego poprzez wywołania funkcji oryginalnego programu obsługi lub w trybie dostępu bezpośredniego poprzez wywołania funkcji zmodyfikowanego programu obsługi.

3. Konsola komputera PC – monitor i klawiatura, służące do komunikacji z użytkownikiem: wprowadzania poleceń i wyprowadzania informacji o stanie systemu. Interfejs ten jest zrealizowany przy użyciu wywołań funkcji z biblioteki graficznej kompilatora, co pozwala na rozszerzenie komunikacji o graficzną prezentację wartości zmiennych kinematycznych i dynamicznych.

4. System plików MS-DOS. Dostęp do nich realizowany jest za pomocą standardowych funkcji języka C. Pliki wykorzystywane są do zapisywania i odczytywania konfiguracji robota oraz do rejestrowania przebiegu pracy.

4.3. Transputerowa biblioteka funkcji robota

Biblioteka ta, złożona z funkcji wykonujących się na transputerach, pośredniczy w korzystaniu przez aplikacje transputerowe z funkcji dostępu do sterownika robota, wykonujących się na komputerze PC. W przypadku pracy w trybie pośredniego dostępu do HCTL biblioteka udostępnia programom aplikacyjnym te same możliwości, co oryginalny program obsługi na komputerze PC. Przy pracy w trybie bezpośredniego dostępu do HCTL umożliwia ona dostęp do funkcji niskiego poziomu układów HCTL, pozostawiając obliczenia wyższego poziomu programom aplikacyjnym, które – jako wykonujące się na transputerach – mogą przeprowadzać je w sposób równoległy.

Funkcje biblioteki dostępne w trybie pośrednim obejmują następujące kategorie:

1. Ruch. Trajektorię można określić poprzez zadanie jej punktów pośrednich (funkcja `addvia`), przy czym początkowy punkt trajektorii jest określony przez bieżące położenie i orientację manipulatora, a końcowy pokrywa się z ostatnio zdefiniowanym punktem pośrednim. Punkty te służą za podstawę wyznaczenia gładkiej trajektorii, którą można wykonać za pomocą funkcji `runpath` lub usunąć za pomocą funkcji `clearpath`. Punkt końcowy trajektorii można również zdefiniować jednocześnie z poleceniem jej wykonania, przy czym współrzędne jego mogą być podane względem bazowego albo względem lokalnego układu współrzędnych (funkcje `move` i `jog`). Większość funkcji ma warianty pozwalające na zdefiniowanie punktów w przestrzeni wewnętrznej lub w przestrzeni kartezjańskiej. Istnieje także funkcja umożliwiająca ustawienie ramki związanej z chwytakiem zgodnie z podanym wektorem (`align`).

2. Siły. Funkcja `push` powoduje wywieranie przez chwytak zadanej siły lub momentu przez zadany czas.

3. Chwytak. Jest to funkcja chwytania z określoną wielkością otwarcia chwytaka i prędkością ruchu (`gripper`).

4. Specjalne tryby ruchu. Możliwe jest włączanie i wyłączanie specjalnych trybów ruchu. W trybie pracy z ograniczeniami sił ruch robota zostaje zatrzymany, gdy określona składowa siły przekroczy zadaną wartość (funkcje `forcethresh` i `threshoff`). W trybie sterowania sztywnością manipulatora możliwe jest utrzymywanie zadanej sztywności manipulatora w czasie jego ruchu (funkcje `stiffness` i `stiffoff`). W trybie pracy z oscylacjami na zadaną trajektorię ruchu trzech złączy nadgarstka nałożone są oscylacje o częstotliwości 4Hz (funkcje `wobble` i `wobbleoff`). W trybie pracy bez napędu silniki są wyłączone, natomiast położenie jest na bieżąco odczytywane. W momencie opuszczenia tego trybu silniki zostają włączone dla utrzymania ramienia w bieżącym położeniu (funkcje `float` i `freeze`). W trybie uczenia napędy są sterowane tak, by siła wywierana przez chwytak na otoczenie była zerowa (funkcja `sfloat`).

5. Kinematyka. Są to funkcje obliczania rozwiązań zagadnień kinematyki – prostego (funkcja `kin`) i odwrotnego (funkcja `ikin`).

6. Odczyt stanu. Funkcje te odczytują bieżące położenie i orientację chwytaka oraz działające na niego siły i momenty (`where`, `readforce` i `adc`).

7. Parametry. Możliwe jest zadawanie parametrów wpływających na właściwości kinematyczne i dynamiczne robota, w tym maksymalnej prędkości i przyspieszenia poszczególnych złącz manipulatora (funkcje `setspeed` i `setacc`), minimalnego czasu ruchu pomiędzy punktami pośrednimi trajektorii (funkcja `segtime`), współczynnika tłumienia dla trybów sterowania siłą i sztywnością (funkcja

setdamp), wartości odniesienia siły (funkcja zeroforce), wartości odniesienia dla enkoderów (funkcja zeropos) oraz bieżącej ramki związanej z narzędziem (funkcja settool).

8. Inicjowanie. Są to funkcje inicjujące system sterowania, ramię robota i chwytak oraz pozycjonujące robot w położeniach wyjściowych *home* i *ready* (funkcje startup, robotreset, robotinit, gripinit, homerobot i gobackhome).

9. Funkcje pomocnicze. Wykonują one różne obliczenia pomocnicze: modułów wektorów, norm, obrotów, iloczynów skalarnych i wektorowych, iloczynów wektorów i macierzy oraz odwrotności macierzy, jak również pewne operacje pomocnicze, np. opóźnienie czasowe (funkcje mag, normalize, vrot, dot, cross, fvmult, ffmult, finvert i delay).

W trybie dostępu bezpośredniego dostępne są następujące grupy funkcji:

1. Ruch. Funkcje te służą do zadawania położenia, położenia końcowego, prędkości, prędkości maksymalnej i przyspieszenia odpowiednio dla aktualnego trybu pracy układu HCTL: sterowania położeniem, i różnych trybów sterowania prędkością (cmdpos, cmdposfin, cmdvel, cmdvelmax i cmdacc).

2. Odczyt stanu. Funkcje te przeznaczone są do odczytywania bieżących położeń (actpos) i prędkości (actvel). Istnieje również funkcja obsługująca dodatkowy kanał na płycie HCTL 1, połączony z czujnikiem sił i momentów w nadgarstku, która pozwala odczytywać jego wskazania (adc).

3. Parametry. Funkcje te służą do ustalania lub odczytywania wartości parametrów wpływających na działanie układów HCTL 1000: rejestrów flag i stanu, okresu próbkowania pętli sterowania oraz parametrów filtra (flag, status, sample, filtgain, filtzero i filtpole).

4. Inicjowanie. Są to funkcje inicjujące układy HCTL 1000 i ustalające żądany tryb sterowania (hctlreset, hctlinit i hctlmode).

Używając funkcji bezpośredniego dostępu do HCTL w połączeniu z obliczeniami wykonywanymi na transputerach można wykonywać te same zadania, co w trybie pośredniego dostępu do HCTL, jednocześnie zwiększając efektywność poprzez zrównoleglenie większej liczby operacji.

5. Podsumowanie

Przedstawiony wyżej zmodyfikowany system sterowania zrealizowano w laboratorium robotyki KARiI Politechniki Poznańskiej. Elastyczna, otwarta struktura warstwy programowej systemu czyni z niego dogodną platformę badania równoległych algorytmów sterowania o architekturze MIMD (Muliple Instruction stream, Multiple Data stream), jak również uniwersalne narzędzie dydaktyczne. W chwili obecnej system służy za podstawę projektów studenckich w zakresie programowania robotów i planowania trajektorii.

LITERATURA

[1] The Fast9 Transputer Board for the IBM PC. User Manual, Quintek Ltd., 1988.

[2] Nigam R., Lee C.S.G.: A Multiprocessor-Based Controller for the Control of Mechanical Manipulators, IEEE Journal of Robotics and Automation, RA-1, nr 4, 1985, s. 173-182.

[3] Zebra ZERO User's Manual. Zebra Robotics Inc., 1990.

Recenzent: Prof.dr hab.inż. Leszek Trybus

Wpłynęło do Redakcji do 30.04.1994 r.

Abstract

The concept of implementation of transputers for robot control systems, presented in this paper, makes use of advantages of the transputer system as well as of the existing Zebra ZERO robot controller. The main features of the robot controller are: integration of most service functions of the motors, encoders, and force/torque sensor on a single board, as well as universal hardware-software interface on the side of the host computer. The multiprocessor transputer system Fast9, consisting of nine INMOS T805-25MHz processors, has great throughput, is easy reconfigurable and gives possibilities of various computations in the external feedback loop.

The original control and transputer systems are first presented. Both systems are hosted by a PC/386 computer and operate under the MS-DOS operating system. Special emphasis is given to the hierarchical structure of the control system, which incorporates several hardware and software layers. This is followed by design considerations concerning possible methods of integration of the transputers into the control system at each hierarchical layer. The variants adopted are described in greater detail. The main software components of the modified control system are presented. These include the transputer server program, that runs on the host computer and interfaces to the transputer system, the HCTL drivers, that also run on the host and interface to the HCTL chips on the robot controller board, and the parallel C wrapper function library, that runs on the transputer system and gives parallel programs access to the robot controller. The functions provided by the library are described.