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

Aleksander STASZULONEK
Silesian Technical University

MULTIPROCESSOR, ADVANCED ROBOTICS CONTROLLER DESIGN USING MVME162

Summary: The advances in microprocessor technology and decreasing price of microcomputer components encourage works on the design and the implementation of open, flexible and universal controllers able to meet the demanding requirements of the wide range of applications, especially the real time applications. The system discussed in this work represents the attempt to design, build and test such an advanced, real time controller for tracking robot. It is assumed that the proposed control system should be able to track and acquire arbitrarily positioned and oriented moving objects.

PROJEKT WIELOPROCESOROWEGO, SYSTEMU STEROWANIA ZŁOŻONYM ROBOTEM PRZEMYSŁOWYM PRZY ZASTOSOWANIU ZINTEGROWANEGO KONTROLERA MVME162

Streszczenie: Postęp w dziedzinie technologii mikroprocesorowej i zmniejszające się ceny elementów komputerowych zachęcają do prac zmierzających do tworzenia otwartych, elastycznych i uniwersalnych sterowników spełniających wymagania szerokiego zakresu zastosowań, szczególnie w systemach czasu rzeczywistego. System rozważany w tej pracy stanowi próbę zaprojektowania, zbudowania i testowania takiego właśnie zaawansowanego kontrolera, do zastosowań w robotyce, pracującego w czasie rzeczywistym. Zakłada się że prezentowany system będzie umożliwiał chwytanie losowo położonych i zorientowanych poruszających się przedmiotów.

ENTWICKLUNG DER MEHRPROZESSORSTEUERUNG FÜR ROBOTIK MIT HILFE MVME162

Zusammenfassung: Die neusten Fortschritte der Mikroprozessortechnik und Rückgang der Mikrorechnerkomponenten regen zu der Projektierung und Implementierung der offenen Anwendungen, besonders in Echtzeit, an. In derAufsatz das System für die Projektierung, Aufbau und Testung Solches Types Echtzeitsteuerungen für Nachfuhr roboter. Die Möglichkeit der Nachfürurung der beliebig orientierten, bewegten Objekten durch entwickelten Steuerungssystem ist angekommen.

1. Introduction

There exist a large number of digital controllers suitable for robotics applications. These controllers are based on different microprocessor families and represent highly varying degree of sophistication. Many new controllers are still being now developed in advanced laboratories. A strong disadvantage of great majority of existing controllers is their limited or not possible at all, access to the servo controllers level. Also the hardware and software structures are usually rigid and hermetically closed which does not leave any space for expansions or changes very often required by the specific user's applications.

The control algorithms most commonly applied are different variations of PID controllers with additional feedbacks and feed forwards. The application of often redundant sensory systems, the increase of sophistication level of the robotics tasks and of the industrial environment create the demand for more and more computational power and the real time control.

The general objective of any control system is the successful achievement of commanded motion or task in face of unpredictable disturbances. A control system design becomes most attractive when it achieves very accurate tracking and rejects broad class of disturbances (including parameter variations) but also accomplishes these ends with minimal complexity and maximal reliability.

From the structural point of view, the presented system can be clearly divided into two subsystems. The goal of first of them is to decompose the task and to generate the desired trajectory. The second subsystem, in a way subordinate to the first one is the system whose task is the execution of specified trajectory. It consists usually from one or more computers

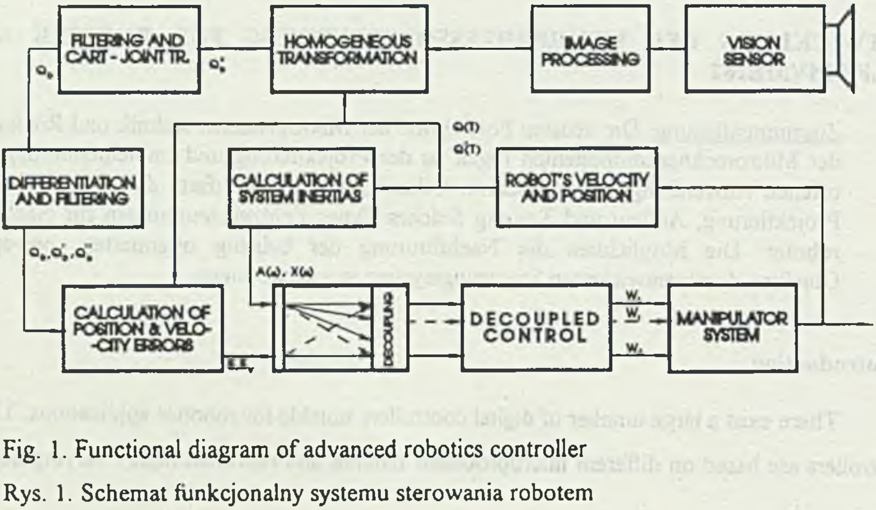

Fig. 1. Functional diagram of advanced robotics controller

Rys. 1. Schemat funkcjonalny systemu sterowania robotem controlling the servomechanisms of each degree of freedom present in the manipulator system. The functional structure of such a system is presented on the fig. 1.

An important precept in the design of control systems, which is rarely mentioned explicitly, is the internal model principle. It means that the control system reflects certain properties of kinematic and dynamic model as well as the type of task to be performed by the system for which it has been designed.

2.General requirements set on the system

The design of robot control system (RCS) is usually very difficult and complicated task. This is the result of the requirements set on RCS which have to be simultaneously satisfied. These requirements are following:

- the control system must be stable,
- high accuracy of positioning and/or tracking is expected,
- high speed of motion is usually necessary for the cost effective applications,
- repeatability of trajectories in the presence of various disturbances such as changes of load and moments of inertia should be guaranteed,
- simplicity of design and servicing,
- reliability,
- lack of overshoots during the transient process.

The last requirement is the critical one. It sets severe limitations on the quality of transient, thus limiting or even prohibiting the use of certain methods and controllers.

3. Computational load

The computational load can be qualitatively estimated from the analysis of the functional diagram of controller structure shown on fig. 1. As the position sensor a commercial vision system can be used. The vision system generates the information about position and orientation of the object in Cartesian co-ordinates. Since the intended application is of the real time type, the position and orientation data have to be obtained frequently enough to guarantee continuous or at least quasi continuous desired trajectory specification. Most of the vision systems available are capable to process one image every 200 ms which is not satisfying for the real time application. Therefore instead of accurate image processing an approximate analysis of the picture should be performed. This can give new data set at approximately 30 ms intervals. The desired trajectory data is generated as the vector expressed in Cartesian co-ordinates having the floating point data format. To refer this data to the global co-ordinates a homogenous matrix transformation should be performed. This is the operation involving floating point multiplication. Due to the fact that only an approximate image analysis can be applied obtained results should be filtered to smooth the desired trajectory and to eliminate noise introduced by the image simplification. This is again a floating point operation where additionally certain compromise has to be accepted between the quality of filtering, calculation time and system stability.

Wen the reliable trajectory data have been obtained, the inverse kinematic problem has to be solved. Several methods of solving this problem have been well described in [3], but since the speed of processing is essential in case of real time tracking control, none of them has been used. Instead, the Newton-Raphson iterative method for the set of non-linear equations has been applied. This is possible after some analytical pre-processing of the kinematic equations done manually. Despite the fact that this decreases the amount of calculations required, it still represents significant load exclusively consisting of floating point operations.

Since in case of tracking control the instantaneous values of desired velocity and acceleration are necessary, the trajectory data obtained from the inverse kinematic problem solution have to be differentiated twice. The desired acceleration value does not appear directly in the control law equation but is required when the necessary and sufficient conditions for existence of certain control regimes are to be checked. Again the trajectory differentiation has to be performed using the floating point arithmetic. At this point the calculations related to the desired trajectory specification are completed. These calculations represent the part of the computational load concerning the real time task specification.

Another part of the computational load is related to the execution of this task. The trajectory execution part can be split into the following sub tasks:

- collecting the present position and velocity data from appropriate sensors,
- calculation of position and velocity errors,
- inertial de coupling,
- control vector generation.

First of these tasks is executed simply by periodical reading the relevant interface devices like counters and analogue to digital converters. The obtained data is of integer format and is stored for further processing. Based on the present and desired position and velocity data the position and velocity errors for each degree of freedom should be calculated. This is again the operation performed on integer numbers. The errors calculation and the reading of present state of manipulator are the operations which do not introduce significant computational load unless the forward estimation of the desired trajectory is necessary to guarantee smooth motion. This may become the necessity, if sampling of the desired trajectory is not frequent enough due to the low speed of vision system.

The inertial de coupling is the sub task representing the heaviest computational load during the trajectory execution. During this operation, first the present position data has to be converted from the integer to floating point format. Then pseudo inertia matrix has to be calculated. When this is done, the pseudo inertia matrix has to be inverted and a matrix equation has to be solved to obtain the components of inertially de coupling matrix. All this operations are performed on the floating point format numbers. The results however have to be converted to the integer format. The detailed description of the inertially de coupling algorithm can be found in [3]. As the results of computer simulations show, it is always profitable to use the inertial de coupling and in some control algorithms its application becomes the necessary condition for feasibility of the whole system. Once the system has been de coupled the control vector can be calculated. This is an integer type operation which in some cases, like sliding mode control algorithms, has to be performed with relatively high frequency, minimum 3 to 5 kHz, therefore representing simple but heavy computational load.

The abundance and sophistication of computational problems results in the need for specific hardware and software structure of the controller. The design of hardware and software for a controller capable to meet the requirements and fulfil all the tasks mentioned above represent two separate but strongly interrelated problems. In case of software design the biggest problem is the optimal timing of each task separately and their synchronisation. These problems are discussed in the separate work and are only mentioned here when necessary. The hardware system able to perform all the tasks mentioned here is further presented here.

4. Adopted hardware solution

Having in mind the requirements set initially by the intended application it has been concluded that the universal controller has to be provided with:

- sufficient processing power,
- rich communication capabilities,

- functions specific for control applications,
- very good interrupt system,
- global system interface for multiprocessing capabilities.

All these functions are indeed available from the presented controller. Block diagram of the universal controller satisfying these requirements is given on Fig. 2.

The type, amount and timing of the computations performed by the controller almost undisputedly impose the general layout of the controller under consideration. First a clear division is made between the integer and floating point tasks.

Due to the fact that the amount of floating point calculations and required speed is very high, the use of arithmetic coprocessors is a must. Each task is clearly separated and dedicated to one function. Additionally all tasks have to be concurrently executed if the real time operation of the controller is to be achieved. Another fundamental feature is the difference in the frequency of execution of particular tasks. For example the solution of inverse kinematics problem can not be performed faster than once per image analysis cycle, which requires at least 30 ms. The frequency of performing the inertial de coupling should be optimised with respect

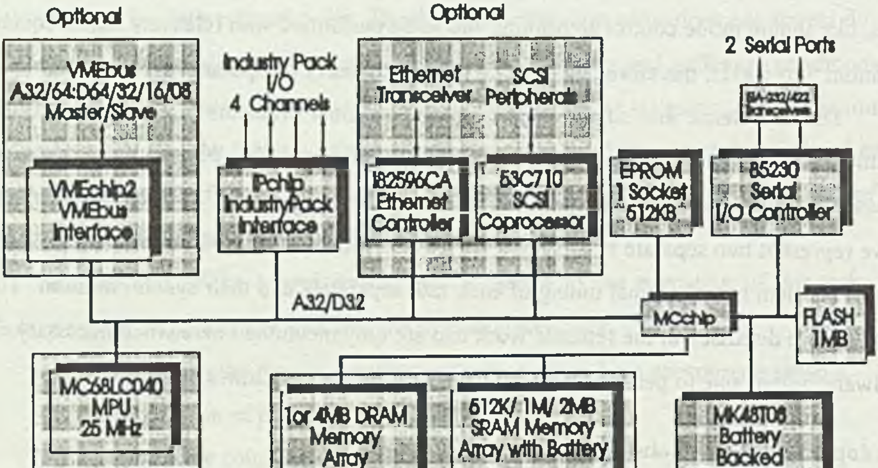

Fig. 2. Block diagram of MVME162

Rys. 2. Schemat blokowy sterownika MVME162 to the speed of manipulator motion. However for good control again frequent de coupling is desired. The position sampling and servo control algorithm have to be performed with very high frequency. Also the communication between servo controllers and system memory is equally frequent. This does not leave almost any time to perform lengthy floating point calculations.

All these remarks lead clearly to the conclusion that a bilevel structure is required. The block diagram of such a bilevel controller for robotics applications has been given on fig.3. It can be seen from fig. 3. that the system has been divided into two levels: higher level called further artificial intelligence level (AI), and lower level called further servo controller (SV) level. This division reflects what has been said about the computational load at the beginning of this point. The AI level consists of the single board computers (SBC) provided with the arithmetic coprocessors improving floating point operations. As a SBCs on the AI level and on the SV level as well the embedded controllers MVME162 based on the 32 bit Motorola 68040 microprocessors have been chosen. This microprocessor having powerful instruction set and flexible set of addressing modes has been chosen as the one most suitable for control applications. The choice of microprocessor automatically leaves no doubt that a VME bus

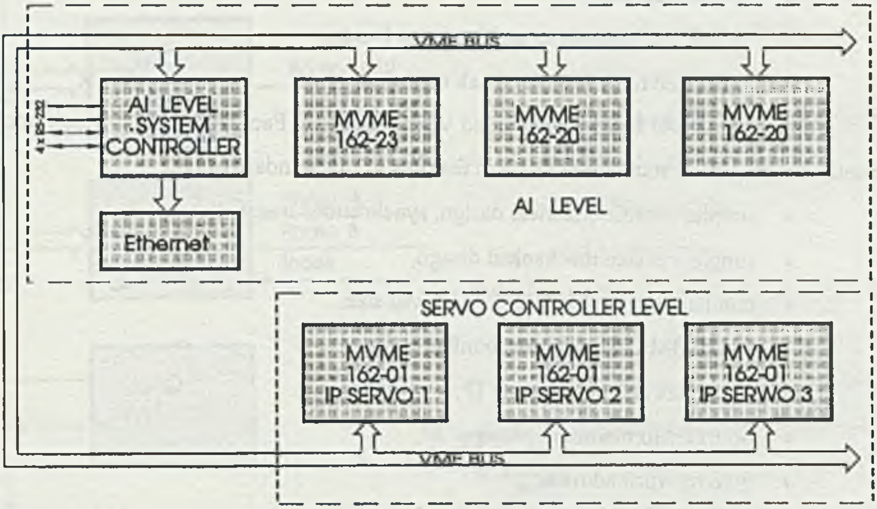

Fig. 3. Hardware structure of bilevel controller

Rys. 3. Struktura dwupoziomowego systemu sterowania should be chosen as the system interface bus. This represents powerful, open structure allowing to choose from the rich family of readily available modules. If due to the increased volume of sensory information there would be a need for more computational power it is enough to plug in one more SBC and update software without need to redesign the hardware. Between the most important features of the MVME162 one can mention:

- MC68040 or MC68LC040 microprocessor,
- 1 to 4 Mb of DRAM with parity protection,
- 512 Kb of SRAM with battery back-up,
- standard PLCC PROM socket,
- 1 Mb x 8 flash memory,
- 8 Kb x 8 battery backed RAM with time of a day clock,
- 6 32-bit timers for periodic interrupts,
- watchdog timer,
- 8 software interrupts,
- 2 serial ports,
- master-slave VME bus interface with system controller,
- Ethernet interface,
- SCSI interface,
- VME bus interrupter and interrupt handler.
- interface for 4 Industry Pack modules.

The industrial I/O has been provided via the Industry Pack™ interface which becomes currently a widely used standard. The basic features of this standard are:

- simple, reliable electrical design, synchronous transfer
- simple, reliable mechanical design,
- compatible with 3U VME, 6U VME size,
- ID PROM permiting autoconfiguration,
- 128 bytes of I/O space per IP,
- up to 8 Mb memory space per IP,
- byte or word addresing,
- two interrupts per IP,
- two DMA channels per IP,
- 8 Mbyte/second continuous data rate,

- 32-bit data width and 32 MHz clock options defined.

A block diagram of servo interface IP module has been shown on fig. 4. Each servo interface provides analogue input, analogue output and optical encoder interface. Two phases and index pulse are available.

The block diagram of the MVME162 embedded controller has been presented on the fig. 2. Such a powerful set of features gives virtually unlimited possibilities for the advanced controller design. In particular, wide variety of control outputs can be implemented. Among the types of control signals available the following can be listed:

- pulse amplitude modulation of the first amd second kind,
- pulse width modulation,
- pulse width and sign modulation,
- sign only modulation.

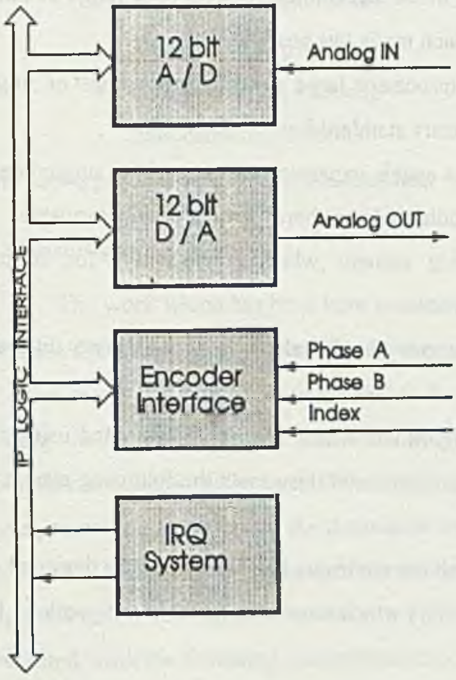

Fig. 4. IP servo interface diagram

Rys. 4. Schemat interfejsu serwo

Such flexiblity enables the implementation of almost any control algorithm. The following tasks are executed on AI level:

- transformation of the vision system data to robot base coordinates,
- data filtering,
- inverse kinematic problem solution,
- trajectory differentiation,
- inertial decoupling,
- data conversion.

These tasks are equally distributed between the SBCs available on the AI level.

The servo controller level is designed in the same way as the AI level, except that the SBCs on SV level do not have the ability for floating point processing. As far as the hardware design concerned the SBCs on the SV level are exactly the same as the ones used at the AI level. They differ only in configuration and software which is there executed. Additionally each of the CPU units at this level is provided with the dual servo interfaceand industrial I/O module connected via the standard Industry Pack interface.

5. Possible applications

During the entire process of the system design two things have been kept in mind:

- the primary purpose for which the controller had been designed, which is the sliding mode tracking controllers applicability investigation,
- since the sliding mode controllers, however good, represent only one special class of controllers, it would not be economical to design the controller solely for the sliding mode algorithms applications.

For these reasons the multiprocessor multilevel system for control engineering applications here developed, has been provided with open, flexible and powerful architecture enabling not only the implementation of sliding mode algorithms, but also wide range of other control algorithms and systems. The features which made this possible are:

- VME bus interface enable multiprocessor large systems applications or single computer expansions via the industry standard bus,
- standard Industry Pack interfaces enable expansion of each of the single board computers by use of variety of additional peripheral devices readily available,
- powerful timing and interrupting systems which are essential for control applications, usually interrupt driven.

To evaluate the range of possible applications one should take into account two different aspects.

The first one is the range of control algorithms which can be implemented using the single board controllers here presented. Taking into account this aspect the following classes of controllers can be implemented:

- sliding mode controllers for which the controller has been primarily designed,
- conventional P, PD, PID controllers which have been used in this work as the reference for SM controllers,
- bang-bang controllers,
- adaptive controllers,

- variety of combinations of the above controllers with the additional feedback and feedforward loops.

Three different types of control signal modulation are available:

- pulse width modulation,
- pulse amplitude modulation of the first kind,
- pulse amplitude modulation of the second kind.

The second aspect is which hardware applications can be developed using the controller developed during the course of this work. To answer this, one can mention the following applications:

- high speed, high accuracy position, velocity and acceleration controllers,
- robotic controllers,
- numerically controlled machine tools,
- plotters,
- factory floor automation,
- ata transmission systems,
- data processing systems,
- data collecting systems.

The above presented list is surely incomplete.

6. Conclusions

The work which has been here presented had been carried out during three consecutive phases. After the control problem has been specified, initial studies performed and the method of solution chosen, the theoretical solution has been obtained as the result of the first phase. During the second phase of this work the obtained solution has been verified using the computer simulation techniques. When the results obtained from the simulation experiments have proven the feasibility of the theoretical solutions, the third phase had begun. During this phase the single board computer for the control applications has been developed and experiments on real system carried out. Between the results of final experimental stage of presented work the following can be mentioned:

- taking into account the computational load related to all the tasks which have to be performed by the advanced tracking controller the general structure of such a controller has been proposed,
- the universal multiprocessor, multilevel computer for the control applications has been designed,
- software for the tracking controllers have been developed and tracking experiments have been carried out.

REFERENCES

[1] Hanselmann H.: Implementation of Digital Controllers - A Survey, Automatica, vol. 23, no. 23 pp. 7-32, Jan 1987.

[2] Motorola Corporation, VME Bus Specification Manual Rev. D, Aug. 1987.

[3] Staszulonek A., Van Brussel H.: Inertially Decoupled, Sliding Mode Controller Design For Trace and Pick-Up Robot, Proceedings of 16th ISIR, Bruxelles, Sep. 1986.

[4] Green Spring Computers, Industry Pack^{TM} Logic Specification, Rev. 0.7, 1992.

[5] Motorola Corporation, MVME162 Embeded Controller User's Manual, 1993.

[6] Motorola Corporation, M68040 Microprocessors User's Manual, 1993.

Recenzent: Prof. dr hab. inż Jerzy Cyklis

Wpłynęło do redakcji do 30.04.1994r.

Streszczenie

Wielką wadą istniejących systemów sterowania robotów jest ograniczony dostęp do struktury programowej oraz brak możliwości dokonywania rozszerzeń i modyfikacji systemu od strony urządzeniowej. Zastosowanie różnorodnych systemów sensorycznych oraz wzrost stopnia złożoności zadań wykonywanych przez roboty zwiększają wymagania dotyczące mocy obliczeniowej komputerów sterujących.

Zakłada się że omawiany system będzie wyposażony w system wizyjny, a struktura urządzeniowa i oprogramowanie umożliwią chwytanie losowo położonych i zorientowanych przedmiotów poruszających się na przenośniku taśmowym.

Prezentowany system można podzielić pod względem struktury na dwa podsystemy. Celem pierwszego z nich jest dekompozycja zadania i generowanie trajektorii zadanej. Drugim z tych podsystemów, w pewnym sensie podporządkowanym pierwszemu z nich, jest system realizacji trajektorii zadanej.

Wśród zadań realizowanych przez system nadrzędny wymienić można pobieranie i analizę obrazu, wyznaczanie położenia i orientacji obiektu manipulacji w przestrzeni kartezjańskiej, rozwiązanie odwrotnego zadania kinematycznego, wyznaczanie parametrów trajektorii zadanej w układzie współrzędnych naturalnych robota, wyznaczanie elementów macierzy odsprzęgającej oraz przesyłanie informacji o trajektorii zadanej do warstwy sterowania serwomechanizmami robota.

Różnorodność realizowanych zadań obliczeniowych oraz ich złożoność powodują, że system musi posiadać wystarczającą moc obliczeniową, szerokie możliwości komunikacyjne, funkcje specyficzne dla zastosowań do sterowania, rozbudowany system przerwań oraz magistralę systemową umożliwiającą pracę wieloprocesorową. Wymagania te spełnia system sterowania omawiany w pracy.

Spełnienie tych wymagań osiągnięto poprzez zastosowanie wieloprocesorowego systemu z magistralą VME oraz zintegrowanych kontrolerów zbudowanych w oparciu o mikroprocesory MC68040.