PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 42, NO. 2, PP. 251-262 (1998)

ARCHITECTURAL PROBLEMS OF THE HYBRID POSITION AND FORCE CONTROL SYSTEM OF ROBOTS[1]

Gábor TEVESZ

Department of Automation
Technical University of Budapest
H-1521 Budapest, Hungary
Phone: (+36-1) 463-2870, Fax: (+36-1) 463-2871
E-mail: tevesz@aut.bme.hu

Received: Sept. 17, 1998

Abstract

An experimental robot control system has been developed at the Budapest Technical University, providing access for research and development in terms of modern robot control algorithms. The multiprocessor robot control system is connected to a NOKIA-PUMA 560 humanoid robot arm. The purpose of this study is to discuss the architectural problems of the mentioned multiprocessor control system. The guiding principles of the modern hybrid position-force control are explained and the structure of the multiprocessor control system is presented. The tasks of the components and the features of the communication channels are also discussed. Finally, the study gives the summary of the architectural and communication requirements of a hybrid position and force control system in the above environment.

Keywords: robot control, hybrid position and force control, multiprocessor systems, IBM PC, CAN, DSP.

1. Introduction

Numerous advanced robot control algorithms, such as computed torque method (non-linear decoupling and decentralised PID controllers), resolved motion acceleration control, hybrid position and force control (operational space formulation), model reference adaptive control and fuzzy control are currently subject of research and experiment. Such algorithms increase the demand for the workplaces applied.

Advanced robot control algorithms are based on the non-linear dynamic model of the robot arm with a driving torque of τ:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}_{\mathrm{ccfg}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}_{\mathrm{ccf}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{h}_g(\mathbf{q}) = \tau\ , \qquad (1)$$

---

[1] The research work was supported by the Hungarian Research Fund (OTKA) under the terms of grant No. T 016855.

where $\mathbf{H}$ is the inertia matrix, $\mathbf{h}_{\mathrm{ccf}}$ represents the centripetal, Coriolis and friction effects, while $\mathbf{h}_g$ is the gravitational part and q stands for the joint coordinates. A well known overriding problem in the implementation of such algorithms is that the driving torque needs to be computed within 1–10 msec. This computation requires remarkable computing power, so multiprocessor architecture and non-recursive algorithms are favoured if parallel computing is possible (LANTOS, 1997 and SOMLÓ et al, 1997).

2. Principle of the Hybrid Position and Force Control

The kinetics of the robot manipulator (1) in the operating space is given by

$$\mathbf{H}^*(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{h}^*_{\mathrm{ccf}}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{h}^*_g(\mathbf{x}) = \tau \, , \tag{2a}$$

where

$$\begin{aligned} \mathbf{H}^* &= \mathbf{J}^{-T}\mathbf{H}\mathbf{J}^{-1} \, , & (2b)\\ \mathbf{h}^*_{\mathrm{ccf}} &= \mathbf{J}^{-T}\mathbf{h}_{\mathrm{ccf}} - \mathbf{H}^*\dot{\mathbf{J}}\dot{\mathbf{q}} \, , & (2c)\\ \mathbf{h}^*_g &= \mathbf{J}^{-T}\mathbf{h}_g \, . & (2d) \end{aligned}$$

($\mathbf{J}$ is the Jacobian matrix of the robot). Let $\mathbf{F}$ be the force of the respective joint torque at the place of the manipulator. In this case

$$\tau = \mathbf{J}^T\mathbf{F} \, . \tag{3}$$

Non-linear dynamic decoupling is a powerful method for the movement control of large, dynamically non-linear and coupled systems (LANTOS, 1997). The form of this method in case of active force and torque control with restricted movement of the manipulator is as follows:

$$\mathbf{F} = \mathbf{F}_{\mathrm{motion}} + \mathbf{F}_{\mathrm{active}} + \mathbf{F}_{\mathrm{ccfg}} \, , \tag{4a}$$

where $\mathbf{F}_{\mathrm{motion}}$, $\mathbf{F}_{\mathrm{active}}$, $\mathbf{F}_{\mathrm{ccfg}}$ are the forces in the operating space representing the motion, the active force control, and the summarised effect of the centripetal force, Coriolis force, friction and gravity components, respectively. Further on

$$\begin{aligned} \mathbf{F}_{\mathrm{motion}} &= \hat{\mathbf{H}}^*(\mathbf{x})\mathbf{S}\mathbf{u}_{\mathrm{motion}} \, , & (4b)\\ \mathbf{F}_{\mathrm{active}} &= \tilde{\mathbf{S}}\mathbf{u}_{\mathrm{active}} + \hat{\mathbf{H}}^*(\mathbf{x})\tilde{\mathbf{S}}\mathbf{u}_{\mathrm{damp}} \, , & (4c)\\ \mathbf{F}_{\mathrm{ccfg}} &= \hat{\mathbf{h}}^*_{\mathrm{ccf}}(\mathbf{x},\dot{\mathbf{x}}) + \hat{\mathbf{h}}^*_g(\mathbf{x}) \, , & (4d) \end{aligned}$$

where $\mathbf{S}$ is the general task specification matrix, $\hat{\mathbf{H}}^*(\mathbf{x})$ is the inertia matrix, $\hat{\mathbf{h}}^*_{\mathrm{ccf}}(\mathbf{x},\dot{\mathbf{x}})$ is a convenient estimation of the effect of the centripetal force, Coriolis force and friction. Finally, $\hat{\mathrm{h}}_g^*(\mathrm{x})$ is a convenient estimation of the gravity. The adequate control signal components introduced in the above equations are

$$\begin{aligned}
\mathrm{u}_{\text{motion}} &= \ddot{\mathrm{x}}_d + \mathrm{K}_{Pm}(\mathrm{x}_d - \mathrm{x}) + \mathrm{K}_{Im}\int(\mathrm{x}_d - \mathrm{x})\,\mathrm{d}t + \\
&\quad + \mathrm{K}_{Vm}(\dot{\mathrm{x}}_d - \dot{\mathrm{x}})\,, && (5a)\\
\mathrm{u}_{\text{active}} &= \mathrm{F}_d + \mathrm{K}_{Pf}(\mathrm{F}_d - \mathrm{F}) + \mathrm{K}_{If}\int(\mathrm{F}_d - \mathrm{F})\,\mathrm{d}t\,, && (5b)\\
\mathrm{u}_{\text{damp}} &= -\mathrm{K}_{Vf}\dot{\mathrm{x}}\,, && (5c)
\end{aligned}$$

where K_{Pm}, K_{Im}, K_{Vm}, K_{Pf}, K_{If} and K_{Vf} are the control gain matrices. The joint torque concluded from the above relations:

$$\begin{aligned}
\tau = \mathrm{J}^T(\mathrm{q})\left\{\hat{\mathrm{H}}^*(\mathrm{q})\left[\mathrm{S}\mathrm{u}_{\text{motion}} + \tilde{\mathrm{S}}\mathrm{u}_{\text{damp}}\right] + \tilde{\mathrm{S}}\,\mathrm{u}_{\text{active}}\right\} + \\
+\hat{\mathrm{h}}_{\text{ccf}}(\mathrm{q}, \dot{\mathrm{q}}) + \hat{\mathrm{h}}_g(\mathrm{q}) - \mathrm{J}^T\hat{\mathrm{H}}^*(\mathrm{q})\dot{\mathrm{J}}\dot{\mathrm{q}}\,. \qquad (6)
\end{aligned}$$

The architecture of the robot control algorithm based on *Eq.* (6) is shown in *Fig. 1.*

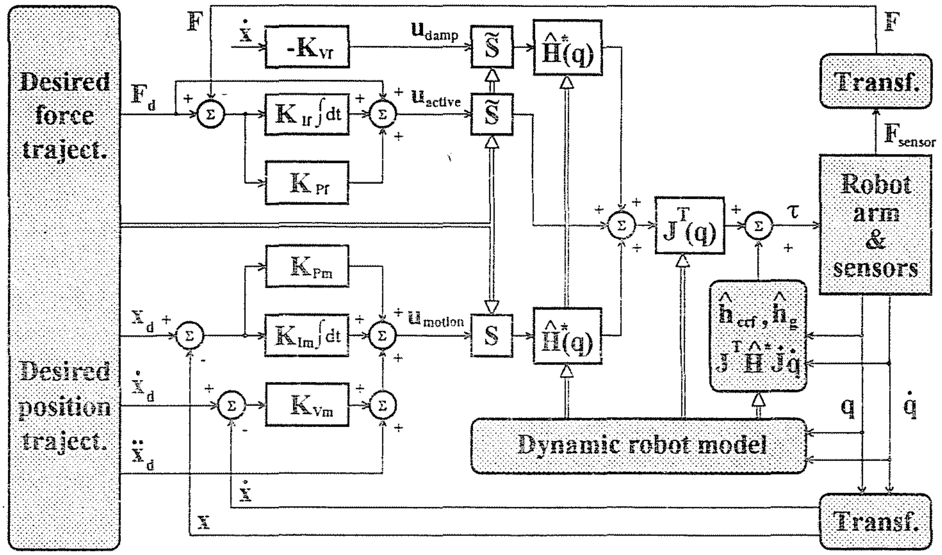

Fig. 1. Principle of the hybrid position and force control

3. Architecture of the Experimental Robot Controller

The experimental robot controller was initially developed for a six degree of freedom NOKIA-PUMA 560 robot arm. The requirements concerning the controller hardware were:

- modular architecture,
- expandability,
- possibility for easy system development,
- possibility for interfacing several sensor processors,
- possibility for changing the host computer.

As far as the operational specifications are concerned, the robot controller should:

- be an open system,
- have a modular (layered) structure that does not reduce efficiency,
- be based on such programming languages and methods which guarantee software portability.

Based on the above requirements a PC based robot controller has been developed at the Technical University of Budapest. Two controllers were built for the cooperating departments. One works at the Department of Automation while the other is used at the Department of Control Engineering and Information Technology. The multiprocessor architecture of the robot controller (BÉZI and TEVESZ, 1994; BÉZI and TEVESZ, 1996) is as presented in *Fig. 2*. The first two prototypes of the robot controller were put into operation in 1996.

The base computer of the system, called host, is an IBM compatible PC. Because of the large amount of expansion cards the system has been placed in two racks, which are connected with a special ISA bus extending card without any restrictions for the communication speed. The host and the additional intelligent parts of the environment (i.e. a computer vision system) are connected using a LAN.

The realisation of the robot control algorithm and the closed-loop control of the robot arm are performed by several home developed ARC (Advanced Robot Controller) cards (TEVESZ, 1995). Depending on the computational demands of.the robot control algorithm, three or four ARC cards form the core of a complete system. These elements communicate via the ISA bus of the host and the direct connection is realised by a high speed serial bus and a special parallel (so-called synchronisation) bus. Each ARC card is of a two processor system containing a communicational preprocessing unit (i386EX) and a signal processor (TMS320C31). Especially the second one deserves attention with its speed and computational capacity (16 MIPS, 32 MFLOPS). The ARC cards provide all the signals for the external DC servo amplifiers and receive the feedback signals from the incremental encoders built into the robot arm.

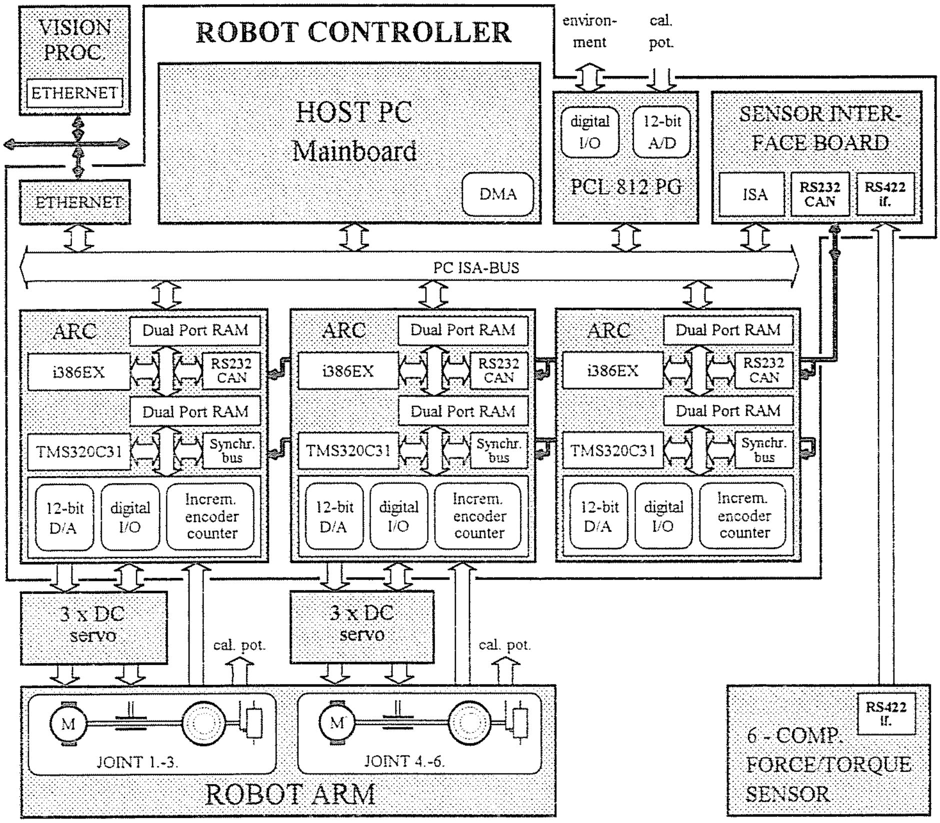

Fig. 2. Architecture of the experimental robot controller

The connection between the host and the passive environment is provided by a general purpose data acquisition card (PCL-812PG). The analogue signals from the calibrating potentiometers of the robot arm are connected here and there is a possibility to connect 16 digital input and 16 digital output signals for completing additional control tasks.

The development of an intelligent sensor interface card is currently in progress. This card has a task to provide connection towards the 6 component force-torque sensor necessary for implementing hybrid position-force algorithms. As a matter of fact this card provides optimal connection between the physical interface of the sensor (RS422 standard differential lines) and the preprocessing unit of the ARC cards (RS232 and/or CAN bus). Additionally, it has been designed to support the mathematically very demanding preprocessing (digital filtering and coordinate transformations) of signals coming from the force-torque sensor.

The required power for the direct control of the six joint robot arm is provided by a home developed servo amplifier, which has switching frequency over 18 kHz and is based on the Pulse Width Modulation (PWM) principle. These devices receive the motor current set points (as analogue signals) and high speed PI controllers form (and limit if required) the real current values to be released for control.

4. Communication between the System Components

During the hardware development one of the most important issues was the availability of the suitable communication channels of the multiprocessor system (TEVESZ et al, 1997). If a communication channel proved to be of low bandwidth, it would restrict the utilisation of the available capacity of the processing units. The overview of the communication channels is presented in *Fig. 3*.

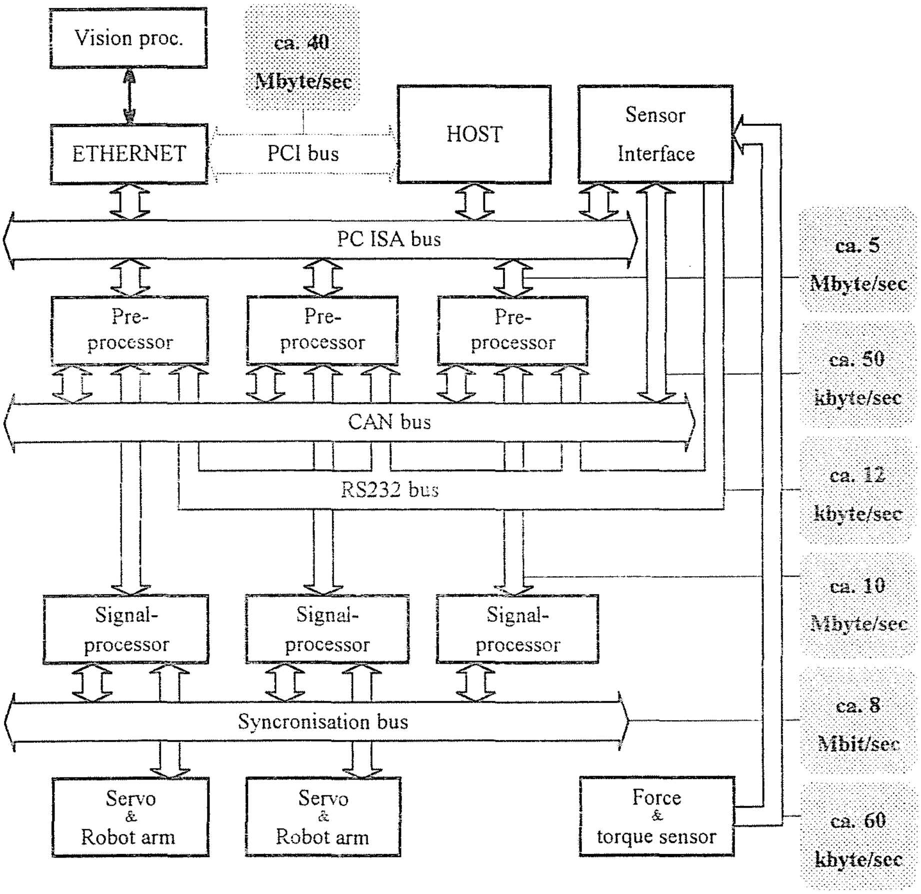

Fig. 3. Communication between the system components

The key units of the control system are the host accomplishing the supervisory tasks and the signal processors providing the direct control of the robot arm. The transmission of the highest dataflux is expected here, so a parallel connection with the highest possible speed has been established. The capacity of both high performance units would have been decreased by synchronised processor–processor communication and by conversions required by the different floating point formats. These tasks have been completed by a preprocessor applied between the host and the signal processor, which has the high speed connection provided by dual port RAMs towards both of the other two main processing units. For the arranging of the data transmission between the cards these preprocessors have been connected to each other and to the other parts of the system via high speed serial lines. Let us discuss these communication channels in details:

a) Connection toward the host

Each ARC card has an 8/16 bit connection toward the host via the ISA bus. The word size of the high speed dual port RAM is 16 bits, that is suited to the bus system of the preprocessor. Because of the restrictions of the PC ISA bus the default communicating scheme uses words of 8 bit and the user has to take care of enabling the 16 bit word data transmission setting the appropriate control bits in the control logic of the card. This enabling must take place for each data unit and in case of larger data packets (over 10 bytes) it is worth to pay attention to this double transmission speed (ca 5 Mbyte/sec) facility. The communication and the handling of the shared data area are supported by 8 hardware semaphores and possible interrupt requests in each direction.

b). Connection toward signal processors

The connection has been established by 16 bit word size dual port RAM, here as well. Although this is the half of the 32 bit signal processor word length but the built-in barrel shifters give a simple way to join the transmitted half words at an optimal speed. Because of the better architectural fitting the channel of the connection is at least twice as fast as in the previous direction, toward the host. There is a possibility to use the 8 hardware semaphores and the interrupt requests, as well.

As far as the transmission of floating point values is concerned, the digital signal processors use a format different from the IEEE-754 standard for representing these values, so the conversion has to take place in transmission time. The switch-over between the two formats can be achieved using a simple conversion algorithm and this function has been installed in the preprocessor for relieving the communicating partners.

c) Connection between preprocessing units

There are two high speed communication channels established between the preprocessors, each of them is configurable depending on the current task specifications.

The first possibility is the asynchronous serial line with a maximum speed of 115.2 kbaud. This is not a bus like link so the required communication should be configured by specific wiring. The main purpose of this connection is transmitting broadcast messages (several receiving units get the same data from a single transmitter). The master in this scheme can be an intelligent sensor interface for example, which provides the information received from the sensor and transformed for each other controller card. The disadvantage of this scheme is that the receivers and transmitters involved in the communication cannot handle data packets, so there is a need to process every byte sent or received by preprocessor interrupts. These interrupts occur with an average time period of ca 100 μsec.

The second possibility is to use the standardised CAN (Controller Area Network) based serial link with much higher speed than the previous one (1 Mbaud). This bus system allows of a non synchronised connection between several units. The CAN protocol uses a multi-master contention-based bus configuration for the transfer of 'communication objects' (messages) between nodes of the network. This multi-master bus is also referred to as CSMA/CR (Carrier Sense, Multiple Access, with Collision Resolution). In case of more than one initiator the arbitration is realised by the units themselves using 11 bit identifiers (standard mode) at the beginning of the packets. One message contains 0-8 data bytes and it is protected by a 15 bit hardware composed and checked CRC (Cyclical Redundancy Check). Defective transmissions are automatically repeated by the bus controllers. The presence of subsidiary information (arbitration identifier, data length, CRC, etc.) naturally decreases the amount of information forwarded within a second, this is summarised in *Table 1*.

Table 1. Data transfer rate in CAN communication

Number of data bytes	Transfer rate [messages/sec]	Transfer rate [data bytes/sec]
0	22 727	0
1	19 231	19 231
2	16 667	33 333
3	14 706	44 118
4	13 158	52 632
5	11 904	59 524
6	10 870	65 217
7	10 000	70 000
8	9 259	74 074

This communication channel plays very important role in connecting components of the robot control system. The main considerations are:

- The multi-master nature and the priority driven arbitration are freely configurable in run-time, which results a connection capable to handle the messages with different priorities.
- The packet size can be adjusted optimally to the communication tasks.
- The applied CAN controllers completely relieve the preprocessors of additional functions of communication. The complete message-handling management (collision handling, error recognition and correction, repetition) is carried out by the CAN controllers.

d) The synchronous connection of signal processors

The synchronous start and stop of robot axes has been assured by a specifically developed synchronisation bus. It contains only 3 lines allowing a wired OR (in inverse logic, AND) logical connection between the signal processors.

5. Architecture of a Hybrid Robot Control System

The purpose of the ongoing research and development activity is to realise the hybrid position-force control introduced in *Fig. 1* using the experimental robot control system. The task distribution and the complete architecture is presented in *Fig. 4*. The chosen concept eliminates the exchange of the elements of large dimensional matrices (H, J, J^{-1}) on CAN bus which would enormously increase the transmission time.

The key design criterion concerning the system architecture was given by the overall goal of the project, i.e. to establish a platform capable to realise most of the modern robot control algorithms. The sensor coupled robot programming language and the possibility to connect the system to other (future) parts of an expert system are to be implemented on the host. In this way this unit controls the whole system and performs the trajectory planning in world coordinates.

The task demanding most of the arithmetic capacity is supplied by signal processors of 1 or 2 ARC cards. The tasks of these units are the calculation of forward and inverse kinetics and high speed determination of the kinetic and dynamic attributes of the robot arm (J, H, h, etc.). With the help of the calculated force vectors (considering the influence of the movement, the centripetal, the Coriolis, the friction and gravity force) defined in the operating space the joint torque vector is computed as presented in *Eq.* (6). This work demands high computing power, so the cycle time of the calculation is estimated between 2..5 msec initially.

The signal processors of two additional ARC cards are responsible for the direct control of the joint servo systems taking care of 3 joints each. Its tasks are to receive the incremental encoder signals from the robot arm, the filtering, irrationality test, optional error correction of this information and the high precision computation of the joint coordinates for the robot arm.

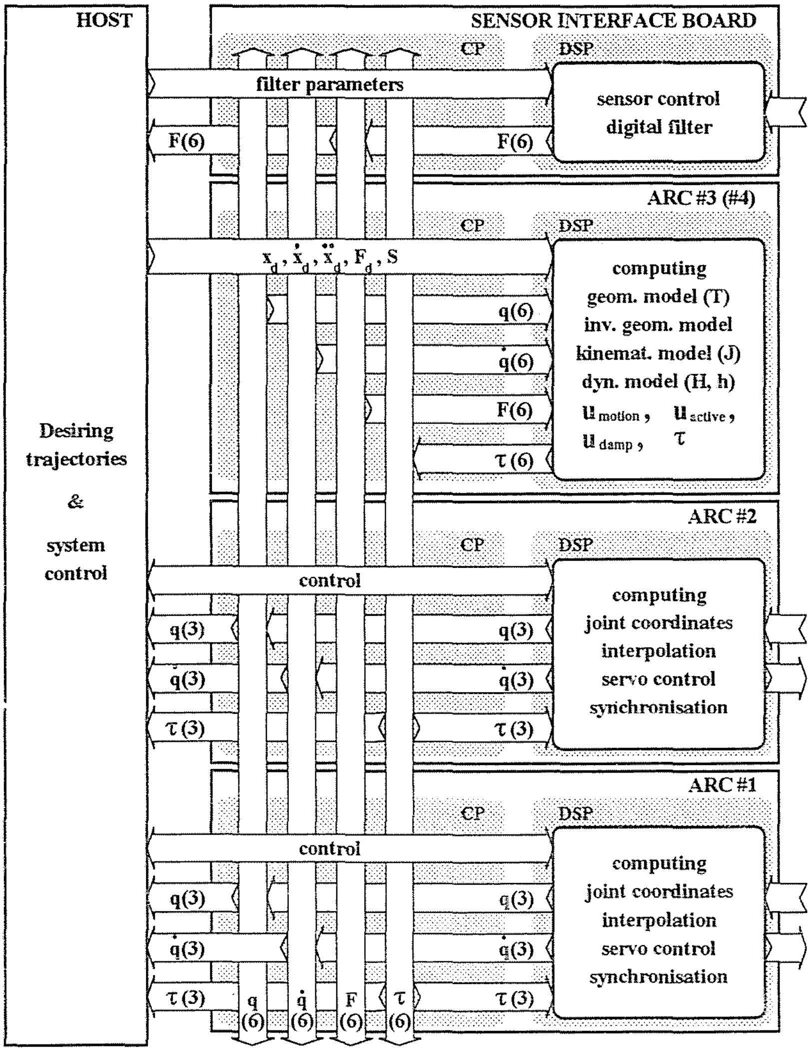

Fig. 4. Architecture of a hybrid robot control system

Based on the torque set points received from the supervisory control system these signal processors provide the current set points for the joint servo amplifiers. The torque vector calculation is a time demanding process and it is executed less frequently than the intervention, so the signal processors have to complete an additional interpolation (cycle time: 500 μsec − 1 msec).

The horizontal connection of host and signal processors and the vertical connection between ARC cards and the 6 component force-torque sensor are supplied by the communication preprocessing units (CP) using high speed dual port RAM and CAN bus. The tasks of these processors are:

- cyclic update of the dual port RAM content,
- the floating point format conversion between the host and signal processors,
- forward the data; optional transfer time data compression maintaining the optimal channel capacity.

One of the most important resources of the system is the CAN bus described in the previous section. The information forwarded on this bus is chosen taking into account that all of the (preprocessed) signals from sensors of the robot and the data necessary for execution have to be available continuously. The information to be forwarded between the system components is summarised in *Table 2*.

Table 2. Data transfer between preprocessors

Parameter	# of data bytes	Compressed size	Transmission method	Transmission time
q	6×4	16 byte	2×8 bytes	216 μsec
$\dot{q}$	6×4	16 byte	2×8 bytes	216 μsec
F	6×4	24 byte	3×8 bytes	324 μsec
τ	6×2	12 byte	2×6 bytes	184 μsec
			$\sum$:	940 μsec

The information update cycle of information must satisfy the future demands against the system without being altered, so the cycle time has been set to 1 msec. It was taken into consideration that from technical point of view some data contents can be compressed.

6. Conclusions

The experimental robot control system provides excellent opportunity for research and development activity in the field of advanced robot control algorithms as well as education of robot control. Based on the experiences gained so far the following research and development activity is in process currently:

- developing an intelligent interface for 6 component force-torque sensor,
- extending the BIOS of ARC cards, adding some new functionality,
- speeding up communication between host and ARC cards,
- shaping up the final form of CAN bus communication,
- optionally installing an operating system subset (QNX kernel and network manager) on the preprocessing units.

The software system based on the above architecture is being developed, the final realisation will be shaped with an expected support of OTKA starting in 1999. The experience using the exposed experimental robot control system has provided many achievements that are to be used for the further development of the system. The most important results are the new communication principle between host PC and ARC control boards, new basic software components, advanced control algorithms using self tuning adaptive control, neural control and fuzzy experts (KLATSMÁNYI, 1996) developed at the Department of Control Engineering and Information Technology.

References

[1] BÉZI, I. – TEVESZ, G. (1994): PC Based Robot Controller for Advanced Control Algorithms. *Proc. of the Joint Hungarian – British International Mechatronics Conference*, Budapest, September 21-23, 1994, pp. 743–748.

[2] BÉZI, I. – TEVESZ, G. (1996): Komplett aus Standard – Komponenten. *Elektronik*, 1/1996, pp. 44–48.

[3] KLATSMÁNYI, P. (1996): Design and Implementation of a New Communication Concept and its Application in the New Version of the Control System of a PUMA Robot. OTKA T 16855/3 Research Report at the Department of Process Control, Technical University of Budapest.

[4] LANTOS, B. (1997): Robotok irányítása, (Robot control). 2nd edition. Akadémiai Kiadó, Budapest (in Hungarian).

[5] SOMLÓ, J. – LANTOS, B. – CAT, P. T. (1997): Advanced Robot Control. Hungarian Academic Press, Budapest.

[6] TEVESZ, G. (1995): ARC Robot vezérlõ kártya Felhasználói leírás (ARC Robot Controller Board User's Guide). Internal Document at the Department of Automation, Technical University of Budapest (in Hungarian).

[7] TEVESZ, G. – BÉZI, I. – OLÁH, I. (1997): A Low-cost Robot Controller and its Software Problems. *Periodica Polytechnica El. Eng.* (to appear).