Vukobratović Miomir,
Kirćanski Nenad,
Stokić Dragan,
Kirćanski Manja,
Karan Branko
"Mihailo Pupin" Institute, Beograd

General-purpose controller for industrial manipulators

ABSTRACT - The general-purpose controller for industrial robots of arbitrary type is described in the paper. The controller enables control of the robots powered by DC electromotors or hydraulic actuators. The controller includes programs for communication with operators, robot program language, program for on-line robot kinematics, program for direct digital servosystems for tracking trajectories including on-line dynamics of the robot. The controller is implemented on two microcomputers based on INTEL 8086 microprocessors. All parts of the controller hardware and software are briefly described in the paper. Implementation of on-line kinematics and on-line dynamics and the choice of the control law are considered in more details.

1. INTRODUCTION

Rapidly growing mass-production of industrial manipulators asks for high performance controllers ensuring excelent tracking of desired trajectories. Usually for each type of manipulator robot special controller is developed. This approach is obviously not convenient for mass-production of industrial robots. Some general purpose controllers which can be applied for various types of industrial manipulators have been already announced on the market. However, those controllers do not ensure perfect tracking of fast trajectories, since they do not compensate for dynamics of the robot. On the other hand, their adjustment to various types of robots usually is not simple, so that they could not be adjusted by customers in industry. These are the reasons which inspare the development of general-purpose controller for industrial manipulators in the Institute "Mihailo Pupin", Beograd. This controller is designed to meet two main advantages over existing controllers for robots:

a) dynamic control of robots ensuring tracking of fast trajectories [1, 2]

b) easy maintaince and adjustment to arbitrary type of nonredundant robots (up to 6 degreed of freedom), powered either by DC electro motors or by hydraulic actuators.

The controller is implemented by two microcomputers based on powerfull 16-bit microprocessors INTEL 8086, which enables application of on-line kinematics and dynamic direct digital servosystems including on-line computation of robots dynamics. Also, communication with operator and teaching of the robot by robot program language and teaching box are elaborated.

Hardware and software support of this controller (called model UCS-1) are briefly presented in the paper. However, the key problem in implementation of this controller is accomplishing of on-line kinematics and on-line dynamics which is included in direct digital servosystems. Thus, in the text to follow we shall pay our attention to these problems.

2. DESCRIPTION OF HARDWARE SUPPORT OF THE CONTROLLER

Complete robot control system with controller UCS-1 is presented in Fig. 1. Model UCS-1 ensures interfaces to: a general-purpose computer, video terminal/ /printer, teaching box, and sensory system. The interfaces to actuators (DC motors or hydraulic) are realized by means of direct analog current or voltage signals from D/A converters. (Obvously appropriate amplifiers have to be implemented). Interfaces to a general-purpose computer, videoterminal/printer and teaching box are realized by means of serial lines, and to sensory system by means of digital and analog lines.

Model UCS-1 contains:

- 2 microcomputers based on INTEL 8086 microprocessors and 8087 coprocessors for processing floating-point operations and calculating various functions,
- memory (RAM and EPROM) in modul-s of 128 Kbytes each,
- real-time clock,
- battery back-up system,
- A/D 12-bit conversion modules,
- D/A 12-bit conversion modules,
- digital input/output modules,
- serial input/output modules, and
- modules for amplification/attenuation of current and voltage signals.

3. DESCRIPTION OF SOFTWARE SUPPORT OF THE CONTROLLER

Controller UCS-1 contains a powerfull software support which consists of several interconnected modules:

- software for communication with operator,
- software for initial specification of robot and sensor parameters, i.e. for adjustment of the controller to the particular robot, actuators and sensors system,
- robot program language,
- program for communication with the teaching box,
- software for on-line robot kinematics, and
- software for direct digital servosystems for tracking trajectories including on-line dynamics.

UCS-1 also contains a number of system routines for servicing discettes/ /magnetic tape conversion modules, "hardware" - tests, programs for checking of the controller's reliability and reactions in emergency situations, etc. General software organization is presented in Fig. 2.

In the text to follow we shall briefly describe some of the modules mentioned above.

3.1. Software for communication with operator

This module enables communication with the operating system. It enables the operator to call various programs which are included in the software of the controller, such as: programs for initial set of mechanism parameters, actuators, sensors, programs for teaching the robot, programs of robot's tasks, programs for hardware tests, program for adjustment of the system's parameters.

3.2. Robot programming language

Robot programming language RL is a high level textual language which is implemented in UCS-1 controller in order to simplify communications between the user and the system, i.e. to supply the user with a tool for simple defining paticular robot tasks. RL supports use of program variables, evaluation of logical and arithmetic expressions, use of conditional and repeatitional control structures and user-written subroutines. A set of system library subroutines incorporated in the language enables the user to specify manipulator motion, gripper operation, sensor control, synchronization etc.

Robot language programs editing, testing and execution is achieved via robot language processor RLP. Robot language processor is a component of UCS-1 software and can be executed in one of the two modes:

- Editor/compiler. In this mode of operation RLP enables interactive creating and editing a RL program via terminal keyboard, as well as including sequences of RL instructions generated by the use of the teaching box. During the operation RLP parses input instructions and prepares the RL program for execution. The syntax analysis is performed and diagnostic messages are displayed on the terminal screen immediatelly after instructions typing.
- Interpreter. In this mode RLP accepts and executes a RL program previously translated in internal form. RLP permanently monitores operation of controller and manipulator equipment and informs the user if any hardware or software error detected during the execution.

3.3. Teaching box

Teaching box represents "hand programmer" for quick and efficient programming of the robot. This teaching box is implemented by 8-bit microprocessor in CMOS technology, so that small sizes and low weight of the box are achieved. These advantages of the teaching box enables programming of the robot in its close surroundings. The teaching box is connected to controller UCS-1 by a long cable and it consists of two alfanumeric display lines, keyboard for selecting the execution mode, keyboard for choosing the execution speed and keyboard for teaching the robot. Keyboard for teaching the robot includes keys for starting and stopping of some procedures, for emergency stop of the robot, for memorizing of the robot's coordinates, defining of the coordinate system in which coordinates of the robot are specified, key for setting grasping of the working object, for defining duration of some actions, for defining the speed of motion, etc.

In defining some control tasks the operator can set some parts of the task using robot language, and some using teaching box. By this, maximal flexibility in control task specification is provided, and minimum time for teaching the robot is ensured.

3.4. Software for initial specification of parameters of robot

Program for initial generation of the system (PIGS) provides flexibility of the controller in the sense of its application to arbitrary configuration of the robot (with up to 6 degrees of freedom), to various types of actuators and sensors. This is why this program is realized as interactive program with simple communication with the user. It consists of several modules: program for initial specifications of parameters of the mechanism, actuators, and sensors, program for kinematic model generation, program for dynamic model generation, program for generating models of the actuators and calculation of the digital servosystems parameters.

Using simple dialogue between the user and the controller, all above mentioned data are imposed by the operator for specific manipulation robot, and can be easily changed. The user can also specify types of the sensors which are included in the system, together with numerical data on sensors. The controller UCS-1 supports the following types of sensors: binary sensors, multibits sensors, analog sensors, sensors which have to be elaborated by host-computer (e.g. camera), etc.

The programs for generation of kinematic and dynamic model of the robot automatically form the robot's models in the machine language of the microcomputers. They form analytic models for specific manipulation type, according to the parameters of the mechanism which have been set by the user.

The program for computing the model of the actuators and computation of the digital servosystems parameters has to synthesize gains of the digital servosystems on the basis of tolerances imposed by the user.

All this modules do not depend on the type of the particular robot, and thus the controller UCS-1 is general and can be easely "adjusted" from one to another robot type.

3.5. Program for robot's kinematics

The program for robot's kinematics realizes the commands of the robot program language or teaching box which are concerned with the motion of the gripper. This program enables on-line generation of the trajectories of joint coordinates of the robot according to the user's commands, or according to trajectories imposed by the host-computer. This on-line generation of the joint trajectories if external (gripper) trajectories are given, is achieved by off-line prepared kinematic model of the robot. This model has been generated by the PIGS. This problem is discussed in more details in Sect. 4. Kinematic program is designed to enable generation of joint trajectories for various types of external coordinates, taking into account all constraints in the working space of the robot (obstacles) and/or constraints upon the robot's joint drives. This program enables implementation of continual and "point-to-point" trajectories with linear and polynomial interpolation.

3.6. Program for direct digital servosystems (DDS)

Controller UCS-1 includes direct digital servosystems for control of DC electromotors or hydraulic actuators in the manipulator's joints. Thus, for robot control analog servosystems are not needed. On the basis of data on actuators and mechanism and on the basis of given tolerances which the robot must satisfy, PIGS computes feedback gains for DDS for all joints of the manipulator. Implementation of DDS requires feedback by position from the potentiometer in the manipulator's joints and velocity feedback from tachogenerator (Fig. 3). The tachogenerator might be ommitted, but this would result in worse tracking of fast trajectories. DDS also includes on-line computation of dynamics of the robot's mechanism. In this way very precise tracking of fast trajectories is achieved, since DDS compensates for dynamic effects (inertia and gravity) during the tracking. Thus, the performance of our DDS are improved in comparison to classical solutions of digital or analog servosystems. This matter will be discussed in more detail in Section 6. On-line computation of dynamic model of the mechanism is achieved by program for automatic generation of analytical model of the mechanism's dynamics on the basis of initially set data on the robot. This problem is considered in Sect. 5.

DDS are designed to withstand variations of parameters of the mechanism and actuators to some extent (variations with respect to initially specified values). However, the user can adjust servo-gains himself through corresponding commands and programs.

3.7. Software for execution checks

Beside the standard routines for software errors detection, the controller UCS-1 is supplied with library of routines for continuous check of the controller performance, during programming of the robot and during task execution. These programs detect all irregularities in the execution of the controller and the robot and send appropriate messages. The controller is also supplied with special routines for fast responces in the emergency situations. By this realibility of the controller is ensured and the system is protected from all irregularities in the controller execution which could cause demage on the robot or in working space of the robot. By this the use of the controller is made safe and simple, since the controller is protected from eventual irregular handle and programming.

4. ON-LINE KINEMATICS CALCULATION

As already stated in Sect. 3.4, on line trajectory synthesis in the space of joint coordinates, given the motion of the end-effector, is realized by the use of the analytical form of kinematic variables, obtained by the (PIGS). These variables involve the Jacobian matrix J, manipulator tip position and the transformation matrix A_n^0 between the coordinate frame attached to the last link and the reference coordinate system (necessary for Euler angles calculation). The use of the analytical form of the kinematic model is of essential importance for the microcomputer implementation of the controller. Namely, if the numeric model was used instead of the analytical model, the computational time would be too long, so that microcomputer implementation would not be feasible.

The computational complexity of the kinematic model depends on kinematical structure of the mechanism and the number of degrees of freedom (d.o.f.). Table T.1 presents a comparison of the number of floating-point multiplications and additions, necessary for kinematic model calculation, for different robot manipulators. The following kinematic structures are considered in the Table:

1. RTT - cylindrical robot - CL with 3 d.o.f. (UMS2, see [1])
2. RRR - arthropoid robot - AR with 3 d.o.f. (PUMA)
3. RRR - anthropomorphic robot - AN with 3 d.o.f. (UMS1, see [1])
4. RTTRRR - cylindrical-anthropomorphic CL-AN with 6 d.o.f. (UMS2 - [1])
5. RRTRRR - semiarthropoid-anthropomorphic - sAR-AN with 6 d.o.f. (UMS3B).

It can be seen that the number of floating point operations increases with the number of degrees of freedom, and the number of revolute joints.

Besides this number of operations, the on-line trajectory synthesis involves the Jacobian matrix inversion, using some of the standard algorithms, e.g. Gauss method. The number of multiplications and additions needed is $\frac{1}{3}(n^3-n)+n^2$ and $\frac{1}{3}(n^3-n)+\frac{1}{2}n(n-1)$, respectively, where n is the number of degrees of freedom. For 3 d.o.f. it amounts to 17 multiplications and 11 additions, while for 6 d.o.f. the total sum is 106 multiplications and 85 additions. Concerning the fact that a pair multiplication - addition takes about 100μs on microcomputer INTEL 8086/8087, one can conclude that total number of floating-point operations allows a sampling period up to 20ms, which is quite satisfactory for trajectory calculation.

Robot / Number of operations	CL	AR	AN	CL-AN	sAR-AN
n_M	2	20	26	58	86
n_A	2	8	9	26	46

T.1. Number of floating-point multiplications and additions necessary for kinematic model calculation

5. ON-LINE DYNAMICS CALCULATION

As pointed out in Sections 3.6, analytical dynamic robot model is generated automatically by PIGS. The following data should be submitted to initiate the model generation:

- number of links and type of joints (revolute or sliding),
- geometrical parameters of the links, disposition of joints, and
- dynamical parameters (masses and link moments of inertia).

The submission of these data is facilitated by the interactive program for data manipulation. Thereafter the automatic model construction can be started by an appropriate command. This program in some manner "imitates" the human action in analytical model derivation. The output of the program is analytical dynamic robot model which could be also obtained by a human action after a hard work (for 6 d.o.f. robots is misht be impossible due to complexity of trigonometric expressions and high probability of mistakes). Program for automatic model generation internally manipulate with analytical expressions, where parameters are treated numerically and functions (sines and cosines) symbolically [3]. The optimization of number of floating-point multiplications and additions/subtractions is carried out in any step of model generation. Thus, the obtained model contains less floating-point operations then many other modelling procedures.

After the model is generated, which can take a few hours for more complex structures of robots, the output program code is automatically stored on peripherial memory volume (Winchester disk or floppy discette). In such a way the dynamic model generation should be proceded only once for a given robot. After the generation the model is stored in UCS-1 library.

In order to illustrate the efficiency of the generated models, the number of floating-point operations necessary for model equations calculation is presented in Table 2. The dynamic effects which are taken into account are inertial and gravitational forces. The models are generated for 5 industrial robots which are mentioned in the preceding section. The results show the these models can be computed in real-time by only one microcomputer INTEL 8086 with data co-processor 8087.

Robot / Number of operations	CL	AR	AN	CL-AN	sAR-AN
n_M	2	21	23	81	283
n_A	2	11	14	39	166

T.2. Number of floating-point multiplications (n_M) and additions/subtractions (n_A) for dynamic model calculation (inertial and gravitational forces included)

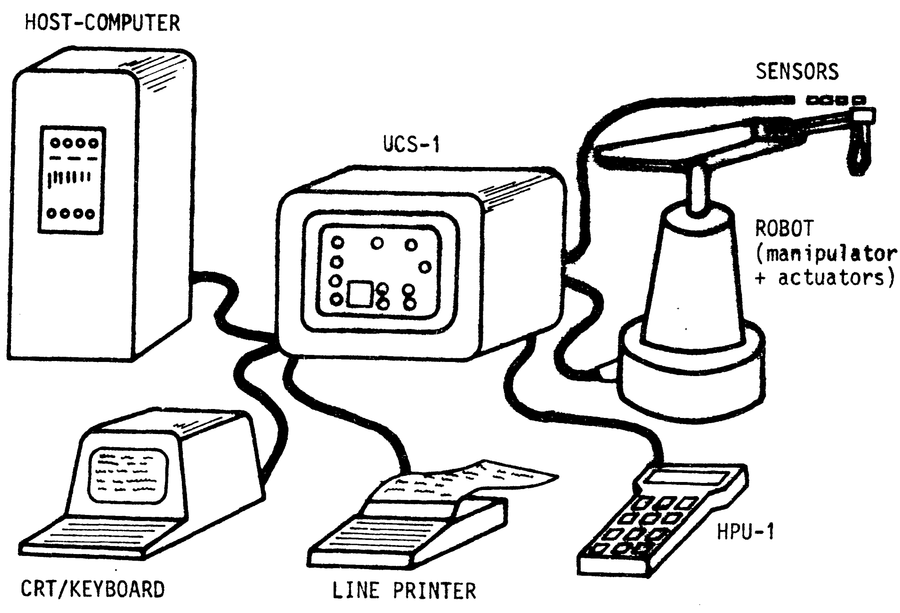

Fig. 1. Configuration of general robotic controller

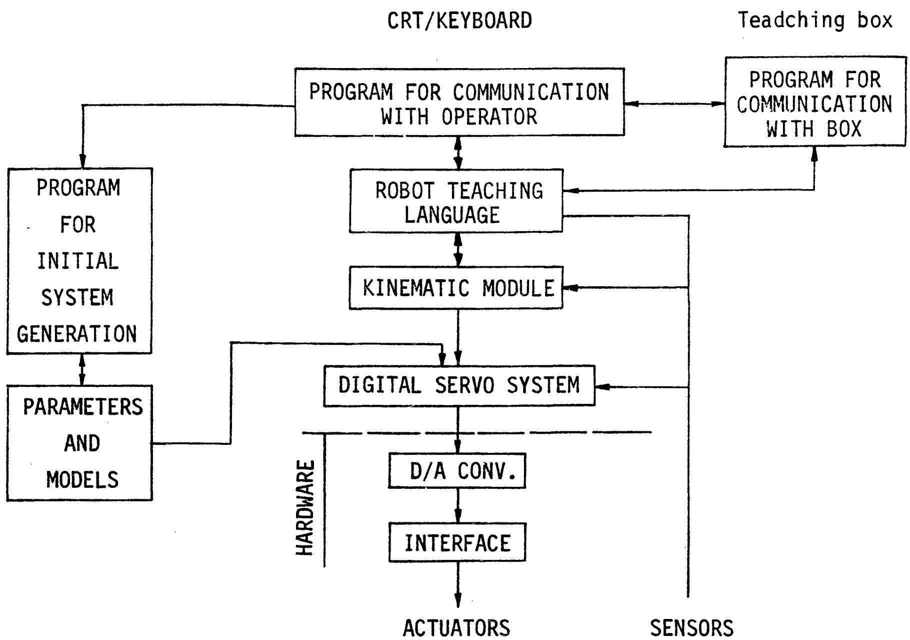

Fig. 2. General software organization

6. CHOICE OF CONTROL LAW

The key problem in design of general-purpose controllers for robotic systems lies in the choice of adequate control law. In order to ensure satisfactory tracking of fast trajectories with arbitrary type of manipulation structures (with which coupling among degrees of freedom might be very strong) the control must compensate for dynamic effects of the robot. On the other hand, as can be seen from the Sect. 5., the computation of the dynamic model of some types of robot structures may be time consuming. Thus, if we want to achieve sampling period which is compatible with the robot's dynamics we might be forced to implement powerfull and complex multiprocessor system which would lead to high price of the controller. This is the reason why we have carefully studied the choice of the control law. Our aim was to choose the simplest control law which could ensure precise tracking of fast trajectories for arbitrary robots structure we are analysed various robot's structures with various control laws, using our software packege for computer-aided synthesis of control for manipulation robots [7]. Our package is designed as to enable the user to choose the simplest control law for particular robot and particular control task. Here we shall bri-

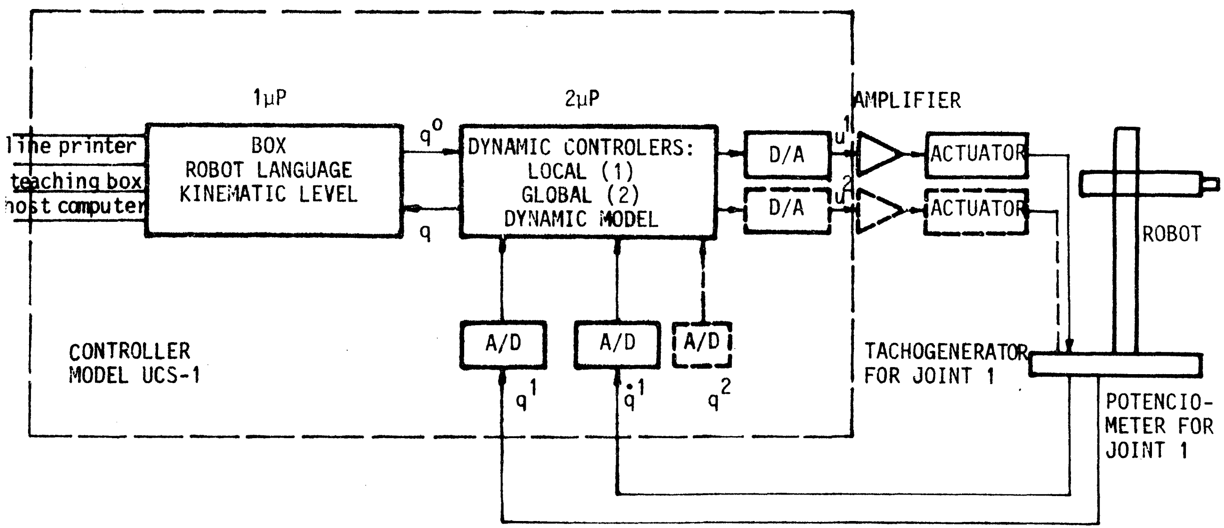

Fig. 3. Control scheme of general-purpose controller

efly summarize the conclusions that we have adopted on the basis of these extensive analysis.

According to our previous investigations in the field of robot control [1, 2, 4-7], we have adopted decentralized control structure since it is the simplest from the standpoint of control implementation. The decentralized control is synthesized in the following form:

$$u_L^i = -k_p^i(q^i-q^{io})-k_v^i(\dot{q}^i-\dot{q}^{io}) - k_I^i \int_0^t (q^i-q^{io})dt, \quad i=1,2,\dots,n \qquad (1)$$

where u_L^i is local control for the i-th actuator (which drives the i-th degree of freedom), q^i is th angle (displacement) of the i-th joint, q^{io} is the nominal trajectory of the i-th joint (which is calculated in the program with kinematic model), k_p^i is positional feedback gain, k_v^i is velocity feedback gain, k_I^i is integral feedback gain, n is the number of freedom of the particular robot. The feedback gains are synthesized using decoupled models of robots, i.e. using the models of actuators [1, 4, 5], in which coupling among degrees of freedom (actuators) is neglected. The decentralized control (1) can stabilize the robotic system if coupling among the robot's degrees of freedom is weak. In general case the influence of coupling among the robot's d.o.f. is very strong and thus the control (1) cannot ensure satisfactory tracking of fast trajectories [1]. Thus, global control which should compensate for destabilizing influence of coupling among d.o.f., i.e. for dynamics of the mechanism which is not included in the local control (1). The global control is adopted in the form [4 - 7]:

$$u_G^i = -k_G^i \cdot P_i \qquad (2)$$

where u_G^i is the global control in the i-th joint, k_G^i is the i-th global gain and P_i denotes coupling which acts upon the i-th actuator, i.e. the driving torque in the i-th d.o.f. of the robot. The driving torques P_i are nonlinear functions of angles q^i, velocities $\dot{q}^i$ and accelerations $\ddot{q}^i$ of all joints of the manipulators. The computation of P_i according to the complete dynamic model of the robot is very complex as pointed out in Sect. 6. However, it has been shown that it is not necessary to compute P_i using complete model of the robot's dynamics, but that we can adopt some approximative model in which some dynamic effects are neglected. In Fig. 4 the number of multiplications that have to be performed to compute local control (1) and global control (2) if various approximative dynamic analytical models are adopted, is presented for various manipulation structures. According to our investigations [1] Coriolis and centrifugal forces can be neglected without loosing in the robot's performance. Thus we can adopt approximative model of the robot's dynamics which include inertial and gravity forces (see Table 2). According to Fig. 4, local and global control with such approximative model (using analytical model) can be computed within 20ms by only one microprocessor INTEL 8086 with data coprocessor 8087. As we have presented in our previous papers [8], the sampling period of 20ms is acceptable for the robotic systems even with very fast trajectories.

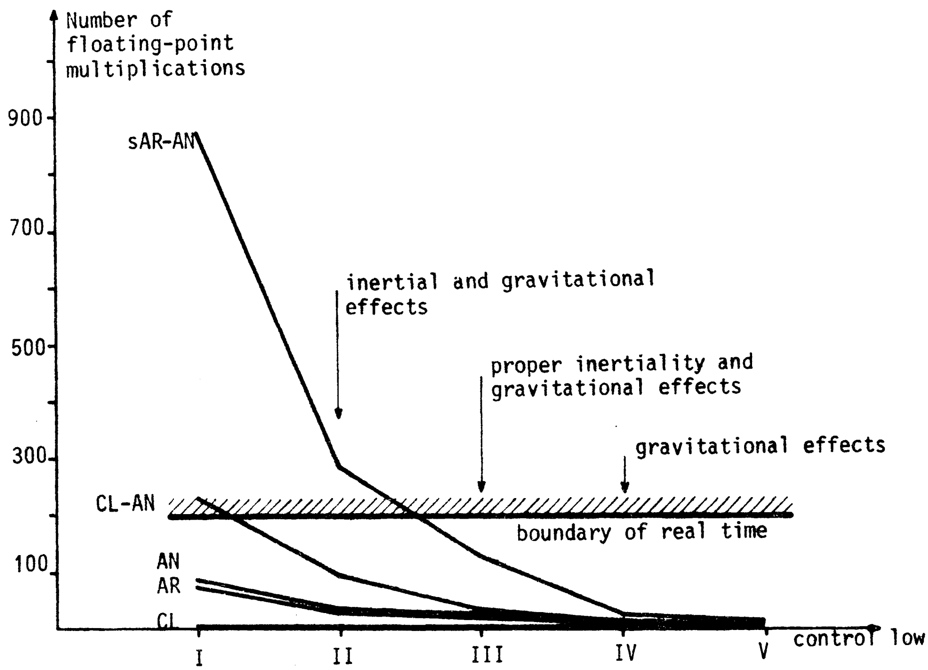

Fig. 4. Number of floating-point multiplications in decentralized control structures

Thus, the control (1), (2) with approximative model which includes inertial and gravity forces can be computed on-line by one microprocessor INTEL 8086. Actually, we have adopted the following solution: in order to improve tracking by local digital servosystems (1) we implement it with sampling period of 10ms, while global control (2) with on-line computation of inertial and gravity forces is somputed with sampling period of 20ms. According to our analysis this is the most suitable control law, which can be implemented by one microprocessor avaible on the market at moderate price, and when can accommodate sufficiently fast trajectories for arbitrary robot's structure with up to six degrees of freedom.

Thus, our control scheme, as partially described in Sect. 3, includes two microcomputers INTEL 8086 (see Fig. 3). The first microcomputer is used to implement communication with the user through terminal (robotic language) and/or teaching box in off-line regime; in on-line regime, during the execution of the robotic task, this microprocessor is used to implement kinematic level, i.e. computation of nominal trajectories of internal angles of the robot according to the operator's requirements (or data sent from the host computer if it is also included in the control of robot); thus, this microcomputer implements kinematic model of the robot and computes inverse of Jacobian matrix as described in Sect. 4. The computation of nominal trajectories on the kinematic level is performed with sampling period of 20ms. The second microcomputer is dedicated to implement local (1) and global (2) control as described above. PIGS is implemented also by the first microcomputer in off-line regime.

7. CONCLUSION

The new general-purpose controller for robotic systems is briefly described in the paper. The main problem in design of this controller was to ensure on-line computation of nominal trajectories (kinematic model) and dynamic digital control (i.e. direct digital controllers which include dynamic model of the robot), but to avoid too expensive and complex multiprocessor system. Using extensive analysis of dynamic performance of various robot's structures with variour control laws we have found out the most appropriate control law which can accomodate all requirements upon the controller for arbitrary robot's structure. Using analytical kinematic and dynamic models of robots (which are generated automatically in the phase of parameter initialization) the above stated problems of on-line computation of dynamic control are successfully solved. Thus, we have found the solution which enables implementation of the dynamic control for arbitrary type of robot by two microcomputers INTEL 8086. This solution is quite acceptable from the standpoint of the equipment complexity and price. It enables to achieve sufficiently good performanse of the robot's controller and to make it quite independent from the robot's structure.

8. REFERENCES

[1] Vukobratović M., Stokić D., Control of Manipulation Robots, Springer-Verlag, 1982.

[2] Vukobratović M., Stokić D., "One Engineering Concept of Dynamic Control of Manipulators", Journal of Dynamic Systems, Measurement, and Control, Vol. 103, No. 2, pp. 108-118, 1981.

[3] Vukobratović M., Kirćanski N., "Computer Assisted Generation of Robot Dynamic Models in Analytical Form", Acta Applicandae Mathematicae, International Journal of Applying Mathematics and Mathematical Applications, Vol. 2, No 2, 1984.

[4] Vukobratović M., Stokić D., "Contribution to the Decoupled Control of Large-Scale Mechanical Systems", Automatica, Vol. 16, No. 1, 1980.

[5] Vukobratović M., Stokić D., "Is Dynamic Control Needed in Robotic Systems and if so, to what Extent", International Journal of Robotic Research, No. 2, 1983.

[6] Vukobratović M., Stokić D., Kirćanski N., "Non-Adaptive and Adaptive Control of Industrial Robots", IEEE Transaction on Automatic Control, 1984.

[7] Vukobratović M., Stokić D., Kirćanski N., Kirćanski M., "Software Package for Computer-Aided Synthesis of Control for Manipulation Robots", IX World Congress of the International Federation of Automatic Control, Budimpešta, 1984.

[8] Vukobratović M., Stokić D., Timotijević M., "Multiprocessors Implementation of Dynamic Control Algorithms for Robots", Proc. of the First Soviet Union-Yugoslav Symposium for Applied Robotics, Moscow, Februar, 1983.