# The Robot Hybrid Position and Force Control in Multi-Microprocessor Systems

LUIGE VLADAREANU<sup>1</sup>, ION ION<sup>2</sup>, L. MARIUS VELEA<sup>3</sup>, DANIEL MITROI<sup>4</sup>

<sup>1</sup> Institute of Solid Mechanics of Romanian Academy, C-tin Mille 15, Bucharest 1

<sup>2</sup> Politehnica University of Bucharest, Splaiul Independentei 125, Bucharest 5

<sup>3</sup> Industrial Engineering and Technology VTC, Burla Vasile 24, Bucharest 6

<sup>4</sup> The National University Research Council, Schitu Magureanu 1, Bucharest 5, ROMANIA luigiv@imsar.bu.edu.ro www.acad.ro

*Abstract.* This paper shows a new robot hybrid position and force control of walking robots in multimicroprocessor system in order to obtain high performances. For this purpose kinematics and kinetostatics analysis are performed, and the mathematic model of the inverted kinematics is determined for controlling the main trajectory of the robot. The method of computation in real time of the inverse Jacobean matrix, topology of transducers networks and the data flux corresponding to implementation of the multi-microprocessor system for path control of industrial robots is presented. Related to this there is presented an Open Architecture system for the robot position control in Cartesian coordinates through real time processing of the Jacobean matrix obtained out of the forward kinematics using the Denevit-Hartenberg method and calculating the Jacobean inverted matrix for feedback. The obtained results prove a significant reduction of the execution time for the real time control of robot's position in Cartesian coordinates and increased flexibility.

*Key Words*: real-time digital processing, hybrid position-force control, compliance function, multi-microprocessor system.

## **1. Introduction**

Data acquisition systems for robot positions control in real time need flexibility, accuracy, high-speed processing and feedback control. The robots' flexibility can be improved if the target generated in the environment coordinates is calculated while moving from the previous point. This can be control for various positions and velocities in the environment coordinates but more processing is needed. Until recently these robots were controlled by a microprocessor system with a floating-point unit. This can control the position only in robot coordinates, while the desired position is recalculated and saved in memory to the previous point with the robot in a waiting state position. These microprocessors have led to an increasing complexity of control algorithms, but they were not sufficient for online control in a changing environment. The solution of this problem might be a parallel system extended to a high processing capacity with a large communication overhead.

Developing a complete mathematical model to study the walking robot movement presents great interest both in terms of developing the high quality system for robot control as well as in terms of verifying the simplifying principles and hypotheses upon which the algorithms used in building the command programs are based., An important part of the studied algorithm hypotheses can be highlighted and removed from the designing stage when such a model of the walking robot is used, so that using it in the computer simulation requires less time and labor [14.16]. The locomotory activity of these robots, and - mainly - walking, come under the category of motions with a high degree of automation. The mechanic system must be equipped with a large number of mobility degrees, in order to form complex synergies, and also to achieve coordinated movements of the legs. By using the robot as a means of transport, some of the parameters that characterize his dynamic features can be subjected to a wide adequate range of changes.

For example, when additional load appears this changes the weight, the position of center of gravity and moments of inertia of the robot platform. A number of environmental factors, such as wind or other forces, whose influence can hardly be anticipated can act against the walking robot. The action of such disturbances may be a cause of considerable deviations of the actual movements of the robot in relation to those prescribed [12,14,16].

Developing and using operational determination methods for causes of deviation from the desired movement, highlighting and avoiding these causes, represent an appropriate means to increase the effectiveness of walking robot control and the efficiency of energetic resources.

#### The direct and inverted mathematical model. A

robot can be considered as a mathematical relation of actuated joints which ensures coordinate transformation from one axis to the other connected as a serial link manipulator where the links sequence exists. Considering the case of revolute-geometry robot all joints are rotational around the freedom axis. In general having a six degrees of freedom manipulator the mathematical analysis becomes very complicated. There are two coordinate dominant systems: Cartesian coordinates and joints coordinates. Joint coordinates represent angles between links and link extensions. They form the coordinates where the robot links are moving with direct control by the actuators. The position and orientation of each segment of the linkage structure can be described using Denavit-Hartenberg [D-H] transformation [13]. Considering that a point in *j*, respectively j+1then  ${}^{j}P$  can be determined in relation to  ${}^{j+1}P$ through the equation :

$${}^{j}P = {}^{j}A_{j+1} \cdot {}^{j+1}P,$$
 (1)

where the transformation matrix  ${}^{j}A_{j+1}$  is defined by the robot's mechanical structure. The control using forward kinematics consists of transforming the actual joint coordinates, resulting from transducers, to Cartesian coordinates and comparing them with the desired Cartesian coordinates. The resulted error is a required position change, which must be obtained on every axis. Using the inverted Jacobean matrix it will manage to transform the change in joint coordinates that will generate angle errors for the motor axis control. The robot joint angles,  $\theta_c$ , are transformed in  $X_c$  - Cartesian coordinates with D-H transformation, where a matrix results from (1) with  $\theta_j$  -joint angle,  $d_j$  -offset distance,  $a_i$  - link length,  $\alpha_i$  - twist.

Position and orientation of the endeffectors with respect to the base coordinate frame is given by  $X_C$ :

$$X_C = A_1 \cdot A_2 \cdot A_3 \cdot \dots \cdot A_6 \qquad (2)$$

Position error  $\Delta X$  is obtained as a difference between desired and current position. There is difficulty in controlling the robot trajectory, if the desired conditions are specified using position difference  $\Delta X$  with continuous measurement of the current position  $\theta_{1,2,-6}$ .

The relation between the end-effector's position and orientation at a given time considered in Cartesian coordinates and the robot joint angles  $\theta_{1,2,-6}$ , is :

$$x_i = f_i(\theta) , \qquad (3)$$

where  $\theta$  is vector representing the degrees of freedom of robot. By differentiating we will have:

$$\delta^{6} X_{6} = J(\theta) \bullet \delta \theta_{1,2,\dots,6} , \qquad (4)$$

where  $\delta^{6}X_{6}$  represents differential linear and angular changes in the end-effectors at the currently values of  $X_{6}$  and  $\delta \theta_{1,2,...,6}$  represents the differential change of the set of joint angles [5, 8, 14]. J ( $\theta$ ) is the Jacobean matrix in which the elements  $a_{ij}$  satisfy the relation:

$$a_{ij} = \delta f_{i-1} / \delta \theta_{j-1} , \qquad (5)$$

where *i*, *j* are corresponding to the dimensions of *x* respectively  $\theta$ .

The inverse Jacobean transforms the Cartesian position  $\delta^6 X_6$  respectively  $\Delta X$  in joint angle error ( $\Delta \theta$ ):

$$\delta \theta_{1,2,\dots,6} = \mathbf{J}^{-1}(\theta) \cdot \delta^{-6} \mathbf{X}_{6} , \qquad (6)$$

The Jacobean computation consists in consecutive multiplication of manipulator A matrix. Gaussian elimination provides an efficient implementation of matrix inversion. The method consists of reducing the J matrix to the upper triangulate form and finding errors in  $\Delta \theta$  joint coordinates using back-substitution. The joint angle errors  $\Delta \theta$  can be used directly as control signals for robot motors.

The motion module of a hexapod robot. The walking robot is treated as an ensemble formed of solid rigid articulated bodies, which represent the plaftorm and leg elements. Increasing the number of legs complicates the driving and command system. On the other hand, as a result of an increase in the number of support points, the static and quasi-static motion becomes more stable [15,16].

As previously pointed out, the for-legged robot's motion is stable only under certain conditions, which are quite restrictive. The issue of static stability is solved by calculating the position of extremity for each leg in relation to the axial system annexed to the platform, with its origin in the center of gravity. The walking robots having the most statically stable configurations are those with more than four legs (six or eight), in which motion can be achieved using different walking types (for example, diagram 3-3, fig. 1). In this case, the centre of gravity G of the mechanical system is always inside the support hexagon ABCDEF. The succession of actions is the following: at first, the D, E and F extremities of the legs move to  $D_1$ ,  $E_1$  and  $F_1$  respectively, after which the extremities A, B and C move to  $A_1$ ,  $B_1$  and  $C_1$  etc.

The existence of different walking types brings about the necessity of organizing them into a system. Applying matrix theory, each type of motion can be described and the motion matrix can be constructed. Applying to the motion matrix an arbitrary operator  $S_1$ , there is the possibility of obtaining the driving matrix, from which the command matrix is then resolved with the help of the operator  $S_2$ . The transducer system being defined, there is the possibility of constructing the external environment matrixes. By analogy, the data matrixes are formed. These, together with the external environment matrixes, allow the control system to choose the corresponding control matrixes of the driving devices.



Fig. 1 The motion module of a hexapod robot

# **2.** Hybrid position and force control of robots

As part of the manufacturing process, especially with regards to the automation assemblers, compliance is necessary to avoid power impact forces, to correct position error of robots or of special mechanical processes devices, and to allow tolerant relaxation of component elements. The compliance can be fulfilled either through passive compliance, as in Remote Center of Compliance (RCC) [2, 3] or in many other of its versions [1, 7, 8, 9,10], or through force control active methods [4, 6]. In any case, there are fundamental problems in both tehniques, when these are implemented in industry. Passive compliance can lower robot position capacity. The active compliance can have problems with sensibility in a rigid enviroment. That is why, although many recent investigations regarding this research purpose have been reported [5, 11], a simple, economical and reliable method is still being sought.

By separating the force positions, the advantage of hybrid control is that information about the position and force are analyzed independently, allowing the application of the already known control laws separately for each loop, the motion errors being combined only at the final stage, when both had been transformed in joint torques.

In general, for each load configuration, a generalized area in a space of restriction with N degrees of freedom, with position constraints along the normal to the surface and force constraints along the tangents, can be defined. These 2 types of constraints, force and position, share the degrees of freedom of possible motions of robot end- effector in 2 orthogonal parts, which must be controlled according to different criteria. In order to specify the desired motions or the force configuration, model in load additional constraints, called artificial constraints, are introduced in accordance with these criteria. Thus, each time the user specifies a desired trajectory in position or force, an artificial constraint is being defined.

These constraints are been applied along the tangent and normal on generalized areas, but, unlike the natural constraints, artificial constraints of force are specified along the normal surface, and artificial constraints of position appear along the tangents, maintaining consistency with natural constraints.

Hybrid position and force control of industrial robots equipped with compliant joints must take into consideration the passive compliance of the system. The generalized area where a robot works can be defined in a constraint space with six degrees of freedom (DOF), with position constrains along the normal force of this area and force constrains along the tangents. On the basis of these two constrains there is described the general scheme of hybrid position and force control in figure 2. For simplification the coordonate transformations are not noted. Variables  $X_C$  and  $F_C$  represent the Cartesian position and the Cartesian force exerted onto the enviroment.

**The selection matrixes.** Considering  $X_c$  and  $F_c$  expressed in specific frame of coordinates, the selection matrixes  $S_x$  and  $S_f$  can be determined, which are diagonal matrixes with 0 and 1 diagonal elements, and which satisfy relation:

$$S_x + S_f = I_d . (7)$$

In approaches [5, 6, 11]  $S_x$  and  $S_f$  are methodically deduced from kinematics constrains imposed by the working environment. Let *A* and *B* be two matrixes with full column rank that satisfy the equation  $A^{t}B = 0$  and correspond to the twist and wrench spaces of constraint, then there can be determined  $S_{x}$  and  $S_{f}$  through the relations:

$$S_x = (A^t \Psi A)^{-1} A^t \Psi , \qquad (8)$$

$$S_f = (B^t \Psi^{-1} B)^{-1} B^t \Psi^{-1} , \qquad (9)$$

where usually  $\Psi$  is symmetrical matrix, positively defined.



Fig. 2. General structure of hybrid control.

Mathematical equations for the hybrid position-force control. A system of hybrid position-force control normally achieves the simultaneous position-force control. In order to determine the control relations in this situation,  $\Delta X_P$  – the measured deviation of Cartesian coordinate command system is split in two sets:  $\Delta X^{F}$ corresponds to force controlled component and  $\Delta X^{\mathcal{P}}$  corresponds to position control with axis actuating in accordance with the selected matrixes  $S_f$  and  $S_x$ . If there is considered only positional control on the directions established by the selection matrix  $S_x$  there can be determined the desired end-effectors differential motions that correspond to position control in the relation:

$$\Delta X_{P} = K_{P} \Delta X^{P} , \qquad (10)$$

where  $K_P$  is the gain matrix, respectively desired motion joint on position controlled axis:

$$\Delta \theta_{P} = \mathbf{J}^{-1}(\theta) \cdot \Delta X_{P} , \qquad (11)$$

Now taking into consideration the force control on the other directions left, the relation between the desired joint motion of end-effectors and the force error  $\Delta X_F$  is given by the relation:

$$\Delta \theta_{\rm F} = {\rm J}^{-1}(\theta) \cdot \Delta X_{\rm F} \quad , \qquad (12)$$

where the position error due to force  $\Delta X_F$  is the motion difference between  $\Delta X^F$  – current position deviation measured by the control system that generates position deviation for force controlled axis and  $\Delta X_D$  – position deviation because of desired residual force. Noting the given desired residual force as  $F_D$ and the physical rigidity  $K_W$  there is obtained the relation:

$$\Delta X_D = K_W^{-1} \cdot F_D \quad , \tag{13}$$

Thus,  $\Delta X_{\text{F}}$  can be calculated from the relation:

$$\Delta X_F = K_F \left( \Delta X^F - \Delta X_D \right) \quad , \tag{14}$$

where  $K_{F}$  is the dimensionless ratio of the stiffness matrix.

Finally, the motion variation on the robot axis matched to the motion variation of the end-effectors is obtained through the relation:

$$\Delta \theta = \mathbf{J}^{-1}(\theta) \,\Delta X_F + \mathbf{J}^{-1}(\theta) \,\Delta X_P \,, \qquad (15)$$

Regarding the construction of mathematical model, based on quasi- dynamic analysis, each leg is considered as a generator function, with limited accuracy for displacement systems. If the number of degrees of mobility is equal to n and if the interior limitations have the following form:

$$F_j(x_1, x_2, ..., x_n) = 0, \ j = 1, \ m, \ n \ge m,$$

then, in the differential equations structure :

$$\frac{\mathrm{d}x_j}{\mathrm{d}t} = f_i(x_1, x_2, ..., x_n, u_1, u_2, ..., u_s, t), \ i = \overline{1, n},$$

there are arbitrary  $u_i$  coefficients, which are used to obtain the "stepping" algorithm. For differential equations, the limitations imposed by the general (platform) base, where the legs are fixed, are applied in first case, and the limitations imposed on the supporting surface, secondly.

#### **3.** The transputers network.

The method control of robot position is not easy. The six multiplication in equation (2), six equations for the six freedom degrees for the calculation of the Jacobean matrix and the matrix inversion in equation (15) in real time need processing power and high-speed for microsystems. The topology of the transputer's network for the industrial robot position control is presented in figure 3, where the  $t_0$  transputer ensures the interface between host-computer and slave processors. Transputer  $t_1$  must be connected to I/O modules that generate the robot position by the hardware. Depending on the process a part of the links become active.

The active topology for process generating  $\theta_{1,...,6} \ll X_C$  supposes the realization of repeated multiplication corresponding to the following relation  $X_6 = A_1 * A_2 * A_3 * ... * A_6$ . Calculating **A** matrix requests three transputers  $t_2$ ,  $t_4$ ,  $t_6$  and a fourth one for data summing, which will be used alternatively and parallel by the  $t_3$ ,  $t_5$  transputers.

The active topology in position errors generating process, in  $\Delta X$  Cartesian coordinates can be made parallel using Jacobean matrix calculation [8,9]. Multiplication and running processing of the A matrix, using homogenous transformations obtain the active topology for the third process, respectively the Jacobean The active transputer matrix calculation. links will be the same as for matrix multiplication, choosing t3 as a centralizing transputer. The active topology for the fourth and the fifth processes, respectively a triangulate matrix realization for the Jacobean and the obtain of an inverted  $J^{-1}(\theta)$  matrix for a  $\Delta \Theta$ angular error, is realized in the pipeline way, in accordance with recurrent relation :

 $a(k)_{ij} = a(k-1)_{ij} - a(k-1)_{ik} * a(k)_{kj}$ , (16) where:

$$a(k)_{kj} = a(k-1)_{kj} / b(k-1)_{kk}$$
 (17)

The network becomes a processor area, where processors are communicating in a unidirectional way from  $t_0$  to  $t_6$  until the matrix become a triangulate matrix, and after data obtain, will start the back-substitution process and the data flow will be sent from  $t_6$  to  $t_0$  transputer.

The transputerised multiprocessor system must perform in real time 1026 arithmetical operations out of which 630 of multiplication/divisions and 396 additions/ subtractions [7,11]. All this will be completed with 24 evaluations of area elements to determine trigonometric functions and 15 inputs/outputs for interfacing with the robot.



Fig. 3. The topology of the transputer's network

The lines of the transputer provide the implementation of parallel process programming in an efficient way. Trough these communication lines the processor can perform: control of some data, memorizing information in a table-like form, data distribution and most frequently data transmission from one transputer to another. Taking into account that the data transmission rate on the transputers' lines it is 100-120 Mbytes/s, the delay coming from communications between processes is under 0,05 µs and may be neglected.

# 4. Open architecture Control System (OAH) for the Robot Control

The implementation of the OAH Open Architecture Control System for robots with compliant wrist allows for the control of the hybrid position and force in Cartezian coordonates through real time processing of the Jacobean matrix obtained out of the forward kinematics using the Denevit-Hartenberg method and calculating the Jacobean inverted matrix for control in closed loop. Using the joint rate control reprezentation, having  $J(\theta)$  as the position Jacobean matrix and  $\Delta X$  as the generalized position error vector, this process is analyzed in a simultaneous two-way fashion: the first to determine the  $\Delta X_F$  matrix corresponding to the force controled component and the second to determine the  $\Delta X_P$  matrix corresponding to the position controled component. The  $\Delta \theta_F$  joint error of the force component and the  $\Delta \theta_P$  position component error are insered to a fuzzy controller. The system architecture is presented in figure 4.

The programmable automate in decentralised and distributed structure (PLC0) ensures the control of the freedom axis and of the robot execution elements. The real position on each motion axis is read through a specialised module for counting impulses from a incremental transducer (IGR) or through the high counting speed specialised modules SA93 with self contained central unit and display on SAE MT65 intelligent terminal with an multifunctional keyboard. Six LENZE incremental transducers of 2,048 increments per rotation generate measurement impulses. Thus the counting of 2,048,000 impulses on each axis is ensured, sufficient compared to the maximum of 480,000 impulses, with a 0.001 mm resolution that can be processed on the free axis. The PLC ensures a maximum number of 62 digital inputs, 16 digital outputs and 80 other configurable inputs/outputs. The communication between the PC and the PLC is made through an

RS-232 serial interface using from PLC the DRUK and EMAS function block.



Fig. 4 Open architecture systems for the compliance robot control.

Mathematical processing modules on which the control system is built runs the following operations:

• forming the stereotypes of motion, with corresponding adjustments in accordance with the configuration of surface on which the displacement is made;

• determining the stability limits of walking and ensure the stability conditions of the robot, by appropriate positioning of the support points in relation to the platform;

stabilizing horizontal platform position depending on the configuration of the surface (where there are no other requirements);
maintaining the height of the platform above the surface level (where there are no other requirements);

• monitoring the operator functions.

Thus are transmitted from the PLC to the PC the current angle motion values in absolute value  $\sum \theta_{ci}$ . From the PC to the PLC are transmitted continuously the reference positions on each axis  $X_{DI}$ , in conformity with the technological program  $X_{pi}$  values. On the *manual* function the values for

the reference angle positions on each motion axis  $\theta_{ri}$  are transmitted, generated directly by the INTERPOLATOR. On the automat function, the values for the reference angle positions on each motion axis  $\partial \theta_{ri}$  are generated from the PLC multiprocessor in real time through the ARCNET communication network, based on the mathematical model, by applying the Denevit-Hartenberg method and determining the inverted Jacobean matrix. Input parameters are: from the PC, the technological position given by the  $X_{Pi}$ technological contouring program, processed and transformed by the INTERPOLATOR into reference position coordinates from the  $X_{DI}$  robot environment; and from the PLC0 the  $\theta_{ci}$  current angular values.

The PLC (SM-PLC) and PLC0 multiprocessor system is meant to send, in real time, through the ARCNET fast communication network, the angular reference positions for the position regulator type PIDT soft implemented on the PLC.

To achieve the walking robot control, the multiprocessor system PLC0 acquires data from

numerous transducers or measuring devices that are filtered, scaled, processed and transmits the data to the OPEN PC, SM-PLC and CP<sub>i</sub>. The main components from which PLC0 acquires data are:

- position transducers, which measures couples variables, namely the relative positions of cinematic elements adjacent to the leading couples;
- touch transducers, mount at the extremities feet, signaling their contact with the surface support;
- force transducers, which measured the forces of reaction between land and feet in the points located at their extremities
- verticality transducers that measure deviations from the platform horizontal position, if there are no other conditions;
- devices that measure the height of the platform over the surface of the land
- Devices that measure the phase of transfer the height of the extremities feet to the surface of the land

There were identified 11 main processes for robot control:

- process 1-current position determined by the matrix in X<sub>c</sub> Cartesian coordinates
- process 2-error position resulting in  $\delta X_C$ Cartesian coordinates
- process 3- Jacobean matrix setting;
- process 4-triangular Jacobean matrix obtaining
- process 5-processing of joint errors δθ 1,2, ...,6
   by inverse substitute

Process 6 determines the position deviation on the force-controlled axis to a  $F_D$  desired residual force. Processes 7–11 are identical as functions and control mode with processes 1–5 but correspond to force controlled axis.

With the help of the relations from the mathematical model the execution program for the PLC0-PLC11 has been conceived and executed, in which each central unit has the role of MASTER communication by data flux through the ARCNET network.

The **PC-OAH** system allows the introduction of new control functions on the basis of supplementary programs, which makes for an open system. Due to high processing speed with operation systems, which allow for programming in evolved programs, the basic functions can be implemented, using an ExTR real time multitask executive: interpolation, the operator interface main program (PP), the technological operating

system, as well as programs with functions of compliant control (CTRL C), pressure control (CTRL P) and/or image recognizing.

An important role of the PC-OAH has building specific types of walk of the robot with six feet. Walking type 3-3 is regarded as the basic type of walking hexapod robot, this work may be presented in two ways, namely: intermittent and continuous. Each sequence of the intermittent walk of the type 3-3 consists in three parts: *work, passive and helpful*; the first two stages are controlled by block formation of walking, and the third – by the control blocks of balance and horizontal position of the platform.

Initial position of the robot, considered the early stage of the feet, corresponding to a situation in which three feet, A, B, C or E, F and D are in a phase of support (fig.1), to obtain a sufficient reserve stability in horizontal position of the platform. Movement consists in movement in a specific direction, of three feet; to that end, in the blocks from the first command of the feet, are inserted cosine guiding values and speeds of movement, calculated in the block of walk building.

The end of leg movement is determined by one of the following conditions:

- the vertical projecting of the center of gravity exceeds the limit of safety of the polygon support;
- a variable of the walking mechanism's couple, has reached the limit scope of operation;
- has reached the limit of length of the step;
- it has been received the "contact" signal given by one of the touch sensors from the feet extremities.

Passive stage includes three phases:

- moving in the direction of the walk of the three legs that they are not in phase support;
- lowering these three legs to contact with the surface support;
- lifting legs that were at the previous stage in support of the movement.

In the process of lifting the feet, height control is achieved of extremities position in relation to the area of support. In case of a height is less than allowed, it is ordered the lifting of the foot until the required height is reached, after which the foot will still be moving in the direction of advancement. Stopping the movement is determined by one of the following reasons:

• the length step limit is reached;

- a variable of the walking mechanism's couple, has reached the limit of operation field;
- the "contact" signal given was received by one of the touch sensors from the feet extremities.

During help phase, restoring of the horizontal position and the prescribed height of the platform takes place, and - if necessary – moving the center of gravity in the field of stability.

The PLC (SM-PLC) and PLC0 multiprocessor system is meant to send, in real time, through the ARCNET fast communication network, the angular reference positions for the position regulator type PIDT soft implemented on the PLC. The proportionality range, the gain factor, the integration and derivation time are established in the PLC0 central unit program depending upon the answer constant of the mechanical system.

PLC0 generates the current values  $\theta_{ci}$  (i = 1-6), calculated through the ITI incremental transducer and transmitted through the ARCNET network to the PLC1–PLC5 multiprocessor system. In the active topology [7, 13] for process (1) each PLC generates an ascendant data flux from PLC0 to PLC5 by processing the transformation matrix  ${}^{i}A_{j}$  from i axis to j axis, to obtain the coordinate matrix in j axis, resulting in the coordinates for the robot environment  $X_{C} = {}^{1}A_{6} = A_{1} \cdot A_{2} \cdot A_{3} \cdot A_{4} \cdot A_{5} \cdot A_{6}$ . In the active topology for process (2) the  ${}^{i-1}A_{j}$  matrixes are stored for each PLC, the Cartesian coordinates  $X_{i}$  in i axis, by multiplying with  ${}^{1}A_{j-1}$ , are determined and the  $\delta X_{C}$  position variation is calculated.

#### 5. Experimental results

Experimental results have shown that the robot's control system in real time with Open Architecture (OAH) ensure flexibility, short time execution, the precision targets and repeatability of the moving programs, eliminating completely the closed systems with projects meant for specified applications. The OAH control system allows the insertion of new control functions based on supplementary programs, which makes it an open system. Supplementary developments in order to increase the performances or new functions adding are possible only by modifying the software relating to the control modules in PC-OAH for laborious computations, respectively in the PLC multiprocessor for complex real time control. This system can be integrated using field bus having the ADVANT OCS, ARCNET, PDnet,

MODBUS, PROFIBUS, RCOM network for increasing communication safety. This allows the increasing of the connected number modules, the system becoming more powerful, with only three conductors for communication. In developing the open architecture system for hybrid position and force control, fuzzy variables for input and output of the system and the membership function reflecting the deflection in Cartezian coodonates are studied. The results show that the approach by fuzzy control leads to a better control of robots with compliant functions with only a few steps to full insertion and jamming avoidance.

Moreover using intelligent PLC the system is ready for automations with I/O redundant and feasible modules, remote text display, interface with robots and different local modules, supervised in hierarchic structures. The obtained results prove a significant reduction of over 50% of the execution time for the control program of robot's position in Cartesian coordinates if compared with processing time resulted from other experiments. Owing to the great computation speed of microprocessor systems and serial connection links for data transmission, the time necessary for establishing the inverted matrix is short enough to allow the robot control in real time, with no influence in performing the other programs.

Acknowledgments. The authors wish to express their gratitude to the Research and Educational Ministry (MEC) and Romanian Academy for its support of the program of work reported herein. The work took place as part of the research project no. 263/2007-2010 in the framework of Grants of CNCSIS, IDEAS Exploratory Research Program, PN II (National Program for Scientific Research and Innovation Technologies).

### 5. REFERENCES

- 1. Jalili N., An Infinite Dimensional Distributed Base Controller for Regulation of Flexible Robot Arms, Journal of Dynamic Systems, Measurements and Control, 123(4), pg.712-719, 2001
- 2. Vladareanu L., Peterson T. New Concepts for the Real Time Control of Robots by Open Architecture Systems, Machine

Building, vol.55, ISSN 0573-7419 no.11, 2003

- 3. Yangsheng Xu, Paul R.P., Robot A., Compliant Wrist for Automated Assembly – Proceedings of the IEEE International Conference of Robotics and Automation, p.1750-1755, 1990
- 4. Yoshikawa T., Sugie T., Tanaka M., Dynamic Hybrid Position-Force Control of Robot Manipulators – Controller Desing and Experiment, IEEE Journal of Robotics and Automation, 1988, vol.4, no.6, p.699-705
- 5. Joly L.D., Andriot C., Hayward V., Mechanical Analogic in Hybrid Position/Force Control, IEEE Albuquerque, New Mexico, pg. 835-840, April 1997
- 6. Joly L.D., Andriot C., Imposing Motion Constraints to a Force Reflecting Telerobot through Real-Time Simulation of a Virtual Mechanism, International Conference on Advanced Robotics, Sant Feliu de Guixols, Spain, 1995
- Sidhu G.S. Scheduling algorithm for multiprocessor robot arm control, Proc. 19<sup>th</sup> Southeastern Symp., March, 1997
- 8. Vladareanu L, Vogelsang G. Real Time Control by INTER-CPM Acquisition System of the Five Axes MINI Machanical Centers -MT'2003 International AMSE Conference of Management and Technology, April 23-25 of 2003, Havana, Cuba.
- 9. Cononovici S.B., Racovita W., Nitu I., Fundamentals of a contact tracking control strategy for Industrial Robots, Machine Building, vol.55, ISSN 0573-7419 no.11, 2003.
- 10.Bratosin D., Non-linear implications in seismic isolation systems for buildings, 9th WSEAS Int. Conf. on Mathematics& Computers in Business and Economics (MCBE '08), Bucharest, Romania, June 24-26, 2008, pg.56-62, ISBN: 978-960-6766-76-3, ISSN 1790-5109
- 11.Yoshikawa T., Zheng X.Z. Coordinated Dynamic Hybrid Position/Force Control for Multiple Robot Manipulators Handling One Constrained Object, The International Journal of Robotics Research, Vol. 12, No. 3, June 1993, pp. 219-230.
- 12.Vladareanu L., Ion I., Sandru O., Lucian M.Velea, Munteanu Mihai S., The Actuators Control by Probabilistic Mathematical Modelling, WSEAS Transactions on Systems and Control, Volume3, 2008, pg. 547-557, pp.11, ISSN: 1991-8763

- 13. Denavit J., Hartenberg RB A kinematics notation for lower-pair mechanism based on matrixes. ASMEJ. Appl. Mechanics, vol.23June 1955, pg.215-221.
- 14. Vladareanu L., Open Architecture Systems for the Compliance Robots Control, WSEAS Transactions on Systems, issue 9, Volume 5, September 2006, ISSN 1109-2777, pg. 2243-2249
- 15. Capitanu L., Vladareanu L., Onisoru J., et.all., Mathematical Model and Artificial Knee Joints Wear Control, Journal of the Balkan Tribological Association, vol.14, no.1, 2008, pg.87-101, ISSN 1310-4772
- 16. Ion I., Vladareanu L., Simionescu I., Vasile A., The Movement of Modular Walking Robot MERO in the Obstacles' Area, WSEAS Transation on Systems, Vol.7, 2008, pg.843-856, pp. 14, ISSN 1109-2777