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

LUIGE VLADAREANU[1], ION ION[2], L. MARIUS VELEA[3], DANIEL MITROI[4]

[1] Institute of Solid Mechanics of Romanian Academy, C-tin Mille 15, Bucharest 1
[2] Politehnica University of Bucharest, Splaiul Independentei 125, Bucharest 5
[3] Industrial Engineering and Technology VTC, Burla Vasile 24, Bucharest 6
[4] 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 multi-microprocessor 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 dominant coordinate 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+1 then ${}^{j}P$ can be determined in relation to ${}^{j+1}P$ through the equation :

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

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, θ_c, are transformed in X_c - Cartesian coordinates with D-H transformation, where a matrix results from (1) with θ_j -joint angle, d_j -offset distance, a_j - link length, α_j - twist.

Position and orientation of the end-effectors with respect to the base coordinate frame is given by X_C :

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

Position error ΔX 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 ΔX with continuous measurement of the current position $\theta_{1,2,\dots,6}$.

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

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

where θ 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 $\delta\theta_{1,2,\dots,6}$ represents the differential change of the set of joint angles [5, 8, 14]. J (θ) 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 θ .

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

$$ \delta\theta_{1,2,\dots6} = J^{-1}(\theta) \cdot \delta\, {}^{6}X_6 , \qquad (6) $$

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 sucession 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 $\mathbf{S_1}$, there is the possibility of obtaining the driving matrix, from which the command matrix is then resolved with the help of the operator $\mathbf{S_2}$. 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.

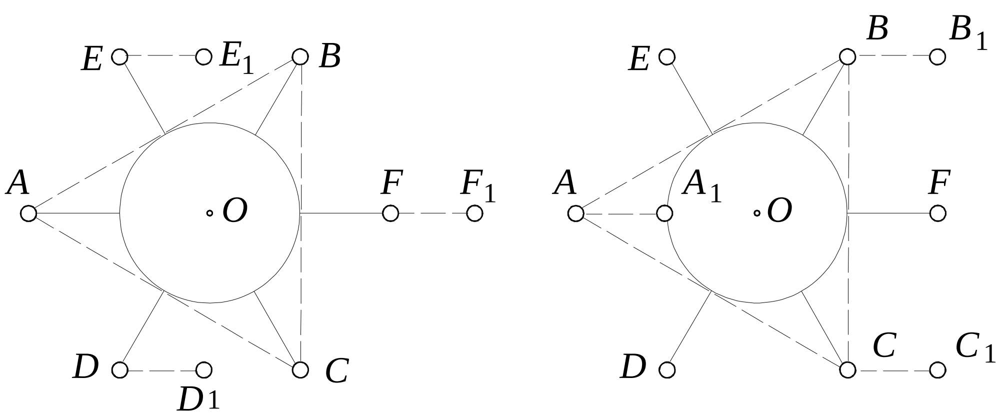

Fig. 1 The motion module of a hexapod robot

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