COMPLEX WALKING ROBOT KINEMATICS ANALYSIS AND PLC MULTI-TASKING CONTROL

LUIGE VLĂDĂREANU [1], ADRIAN CURAJ[2], RADU I. MUNTEANU[3]

Key words: Kinematics walking robot, Walking robot control, Multi-microprocessor system, Robot dynamic control, Static and dynamic stability.

The paper presents a dynamic control system for the walking robot which is a complex structure of a hexapod walking robot, having six degrees of freedom for each leg of the three degrees of movement for positioning and three degrees of movement to paw foot orientation. Issues for direct and inverse kinematics of the walking robots structure are analyzed, determining the joints coordinates of the robot leg and of the orientation mechanism. In the end, a multi-microprocessor architecture with multi-tasking control is designed, tested through a virtual projection method and confirmed through computer simulation, which allows a fast reply in feedback loop for robot real time control, with improved stability and flexibility performance.

1. INTRODUCTION

The movement activity of walking robots, especially walking, is characterized as a movement with a high degree of automation. The mechanical system must be equipped with a large number of degrees of mobility, in order to form complex synergies, and to control leg movements in real-time by adapting to the robot environment. Various control methods are known for robot movement trajectory with general application or in regard to the mechanical structure. A simple method in which the contact force is used to modify the reference position trajectory of robot end-effector, called "adjustment position" was proposed by Whitney (in 1977). A solution to the robot interaction with the environment is solved by Raibert, Craig (1981) and Manson (1980) which provides the force and position control by breaking them into "position sub-space" and "force sub-space". These two subspaces correspond to the direction of movement of the robot, namely moving freely or constrained by the environment. Separate processing and processing by different laws for position control and force control requires

---

[1] Romanian Academy, Institute of Solid Mechanics; E-mail: luigiv@arexim.ro

[2] "Politehnica" University of Bucharest

[3] Technical University of Cluj-Napoca

Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **57**, *1*, p. 90–99, Bucarest, 2012

significant preparation of the tasks procedures and changing control loops in the implementation; in addition, this method may lead to instability problems especially during transition between free and constrained motion (Zhang and Paul – 1989, An and Hollerbach – 1989, Fisher and Mujtaba – 1992).

Compliant movement control, which is essentially the default force control based on position was suggested by Lawrence, Stoughton (1987) and Kazerooni, Waibel, Kim (1990). Salisbury (1980) presented a method for active control of the end-effector apparent stiffness of the robot in Cartesian space. In this method the reference position is used to control the contact force and no force reference points are used. The Pieper and Khalil method [9] and the Paul method stand out among them, for the advantages they offer. The first one allows inverse kinematics problems to be solved regardless of the values of robot geometric characteristics, for robots with six degrees of mobility which have three rotational kinematic couplings on concurrent axis or three translation kinematic couplings. Because of the flexibility and that it has a solution for the inverse kinematics problem, this "decoupled" structure with three rotation couplings and concurrent axis is found in most robot models on the market. The position of the three axes intersection point is uniquely determined only by the q_1, q_2, q_3 variables. Another advantage of the decoupled structure is allowing splitting and separate negotiating of the positioning and orientation. Paul's method as well as Lee and Elgazaar’s treat each case separately without generalizations.

The paper presents the direct and inverse kinematics modelling of the walking robot which is a complex structure of hexapod with three degrees of movement for positioning and three degrees of movement for orientation of the foot paw. By kinematic "decoupling" of movement, the positioning control is practically separated from orientation in robot modelling. Following the performed studies there are developed the walking patterns, dynamic control phases of walking robot and a multi-microprocessor architecture with multi-tasking control that allows a fast reply in the feedback loop for real time robot control.

2. DIRECT AND INVERSE KINEMATIC CONTROL OF THE WALKING ROBOT

A robot can be considered as a serial link manipulator where the links sequence is connected by an actuated joints mathematical relation which ensures coordinate transformation from one axis to the other. Figure 1 illustrates a robot position control based on the **D**enavit-**H**artenberg transformation. The robot joint angles, θ_c, are transformed in X_c – Cartesian coordinates with the ***D-H*** transformation. Considering that a ${}^{j}P_{j+1}$ point in ***j*** quadrant and ${}^{j+1}P$ in ***j*+1** quadrant, it can be determined through the equation ${}^{j}P_{j+1} = {}^{j}T_{j+1} \times {}^{j+1}P$, where ${}^{j}T_{j+1}$ is the ***D-H*** transformation matrix. For the walking robot having the kinematics structure of a leg shown in Fig. 2, a θ_j initial with 0 degrees ensured, with D-H parameters in the Table 1.

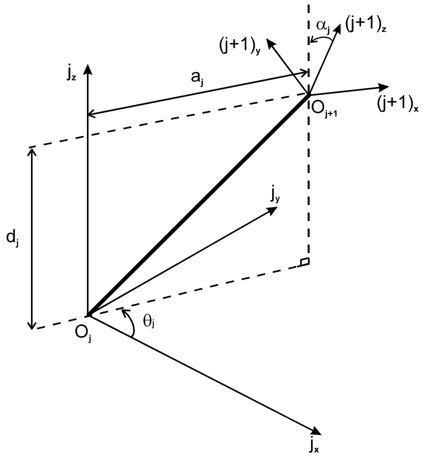
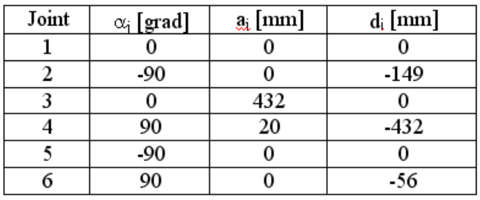


Fig. 1 – The robot position control based on the Denavit-Hartenberg transformation.

Table 1

D-H parameters for the walking robot



Joint	α_j [grad]	a_j [mm]	d_j [mm]
1	0	0	0
2	-90	0	-149
3	0	432	0
4	90	20	-432
5	-90	0	0
6	90	0	-56

In the end, by direct kinematics modelling, using homogeneous transformation matrices, we have obtained the coordinates of a positioning element point for the hexapod walking robot according to the ***D-H*** transformation. For $j = 3$, reported to quadrant 4, the ***D-H*** transformation is given in relation (1).

Inverse kinematics modelling. Inverse kinematics modelling allows generating joint coordinates in the desired position of the robot's point of support expressed in the operational coordinates (environment). For certain points in robot space, in which the equations for inverse kinematics modelling don’t have a solution, called singularities, which happen quite frequently, we turn to numerical methods. The most common numerical method is the Newton-Raphson method whose main drawback is the large amount of calculations. We say that a robot has a solution to the problem of inverse kinematics if we can calculate all configurations for achieving a given position [1]. Not all jointed mechanisms satisfy this condition. As Roth said, for robots with less than six degrees of freedom there always exists a solution to the problem of inverse kinematics. Robots that have six degrees of freedom have solutions to the problem of inverse kinematics, if they have either of the following characteristics: have three joints of translation; have three rotation joints with competing axis; has a coaxial rotation and a coaxial translation joint; have two pairs of rotation joints with concurrent axis.

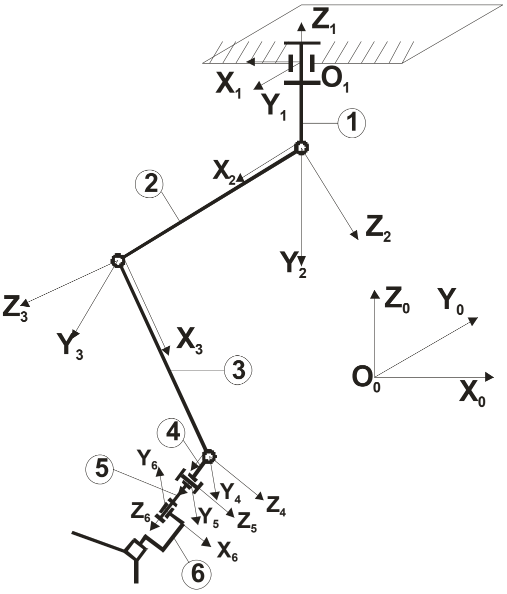


Fig. 2 – Kinematics structure of a moving element of the walking robot.

For the kinematics problem, for the robot class with six degrees of freedom, having three kinematic rotation couplings with the competing axis, the maximum number of solutions is 32. The number of solutions depends on robot architecture and decreases when the parameters take certain unique values. The number of solutions also depends on the size of joint distance between the edges. For the walking robot, by ***D-H*** transformation we obtained:

$$ {}^{3}P_{4} = {}^{3}T_{4}\begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \begin{bmatrix} C\theta_4 & -S\theta_4 & 0 & d_4 \\ C\alpha_4 S\theta_4 & C\alpha_4 C\theta_4 & -S\alpha_4 & -r_4 S\alpha_4 \\ S\alpha_4 S_4 & S\alpha_4 C\theta_4 & C\alpha_4 & r_4 C\alpha_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \begin{bmatrix} d_4 \\ -r_4 S\alpha_4 \\ r_4 C\alpha_4 \\ 1 \end{bmatrix}, \tag{1} $$

where “*s*” and “*c*” are abbreviations for sinus and cosinus, d_4- offset distance, r_4-link length. Noting with fi the equations in robot coordinates from quadrant 2, for a certain position of the leg support point, one achieves the relation:

$$ {}^{2}P_{4} = {}^{2}T_{3}\begin{bmatrix} d_4 \\ -r_4 S\alpha_4 \\ r_4 C\alpha_4 \\ 1 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ 1 \end{bmatrix}. \tag{2} $$

Using the general form of ${}^{2}T_3$, equation solutions f_i can be determined:

$$\begin{aligned} f_1(\theta_3) &= C_3 d_4 + S_3 S\alpha_4 r_4 + d_3, \\ f_2(q_3) &= C\alpha_3(S_3 d_4 - C_3 S\alpha_4 r_4) - S\alpha_3(C\alpha_4 r_4 + r_3), \\ f_3(q_3) &= S\alpha_3(S_3 d_4 - C_3 S\alpha_4 r_4) + C\alpha_3(C\alpha_4 r_4 + r_3). \end{aligned} \tag{3}$$

Similarly, noting with g_i the equations for the point in the first quadrant, expressed in robot coordinates, we obtain:

$$\begin{aligned} g_1(q_2, q_3) &= F_1(\theta_3, q_3) + d_2, \\ g_2(q_2, q_3) &= C\alpha_2 F_2(q_2, q_3) - S\alpha_2 F_3(r_2, q_3), \\ g_3(q_2, q_3) &= S\alpha_2 F_2(q_2, q_3) + C\alpha_2 F_3(r_2, q_3), \end{aligned} \tag{4}$$

where:

$$\begin{aligned} F_1(\theta_3, q_3) &= C_2 f_1 - S_2 f_2, \\ F_2(q_2, q_3) &= S_2 f_1 + C_2 f_2, \\ F_3(r_2, q_3) &= f_3 + r_2. \end{aligned} \tag{5}$$

Multiplying to the left with ${}^0T_1$, we obtain:

$$ {}^0P_4 = {}^0T_1\,{}^1P_4 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 1 \end{bmatrix}. \tag{6}$$

But as ${}^0P_4$ is given by relation (7), resulting equations (8) for robot angular coordinates according to a certain position of the support point, expressed in coordinates of the foot environment.

$$ {}^0P_4 = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \tag{7}$$

$$\begin{aligned} P_x &= C_1 g_1 - S_1 g_2 \\ P_y &= S_1 g_1 + C_1 g_2 \; . \\ P_z &= g_3 + r_1 \end{aligned} \tag{8}$$

Starting from equations (3)–(8) in chapter 3, a new method for processing the support point coordinates in robot environment was development and was implemented in a real-time and multi-tasking multiprocessor system control, which allows control of the walking robot motion trajectory, avoiding the laborious calculation of equations $x_1 = r(\theta_1)$, $y_1 = s(\theta_1)$, ..., $z_3 = n(\theta_3)$ from Fig. 3, for six ($k = 1$–6) elements of the walking robot leg mechanisms.

3. MULTI-TASKING ARCHITECTURE FOR WALKING ROBOT CONTROL

Based on modeling presented in previous chapters a real-time control system of the robot mechanisms motion trajectory can be achieved [3–6]. The multiprocessor system architecture PLC (SM-PLC) we developed is shown in Figure 3, from where the implementation of the data flow for single leg robot controls results. The multiprocessor system PLC (SM-PLC) and PLC0 have the purpose to provide in real time, through fast communication bus ARCNET, angular reference positions for the PID position regulator, software implemented on the PLC. PLC0 generates the current values θ^{k}_{ci}, (I = 1–3, k = 1–6) caused by incremental sensor interface and transmitted through the ARCNET network to the multiprocessor system that consists of PLC_1-PLC_5. Through our studies were identified five processes in the robot motion control program. In the active topology [2, 8, 11] in the process (1), each PLC generates a data stream increasing from PLC5 to PLC0.

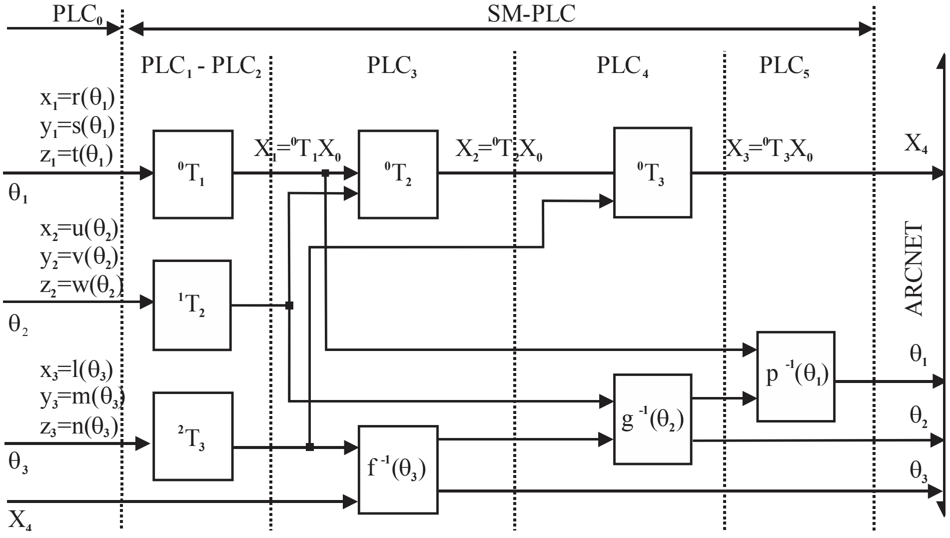


Fig. 3 – Data flow implementation with multiprocessor PLC systems.

By calculating the transformation matrix ${}^{i}T_{j}$ for leg k (k = 1–6), from the axis "i" to the axis "j" to obtain the matrix of coordinates on "j" axis, ultimately resulting coordinates in robot environment $X^{k}_{\mathrm{C}} = {}^{1}T^{k}_{3}\, X^{k}_{0} = {}^{0}T^{k}_{1} \times {}^{1}T^{k}_{2} \times {}^{2}T^{k}_{3} \times X^{k}_{0}$. In the active topology for the process (2) are stored at each PLC, ${}^{i\text{-}1}T^{k}_{\mathrm{i}}$ matrices, determining the Cartesian coordinates X^{k}_{i} for the axis "i" by multiplication with ${}^{1}T^{k}_{j\text{-}1}$ and calculating the change in position $\delta X^{k}_{\mathrm{C}}$.

Generating the inverse matrix $f^{-1}(\theta_1)$, $g^{-1}(\theta_2)$, $p^{-1}(\theta_3)$, is done in the processes (3)–(5) through an upward data flux in conjunction with the process (1). As the calculations of matrix ${}^{1}T^{k}_{j}$, PLC$_3$ is assigned to matrix $f^{1}(\theta_1)$ and PLC$_5$ to the matrix $p^{-1}(\theta_3)$. For every PLC has been allocated a column of the transformation matrix, the data flow being from the PLC$_0$ to PLC$_5$, both for direct and inverse kinematics. For the orientation mechanism, built from three kinematic coupling with common axis, was applied the same concept of data flow with appropriate allocation of the PLC$_1$-PLC$_5$ systems. Having the mathematical relationships from the discussed mathematical model, the execution program was developed for PLC$_0$-PLC$_5$ in which each central unit is the Master station with data stream communication via the ARCNET network.

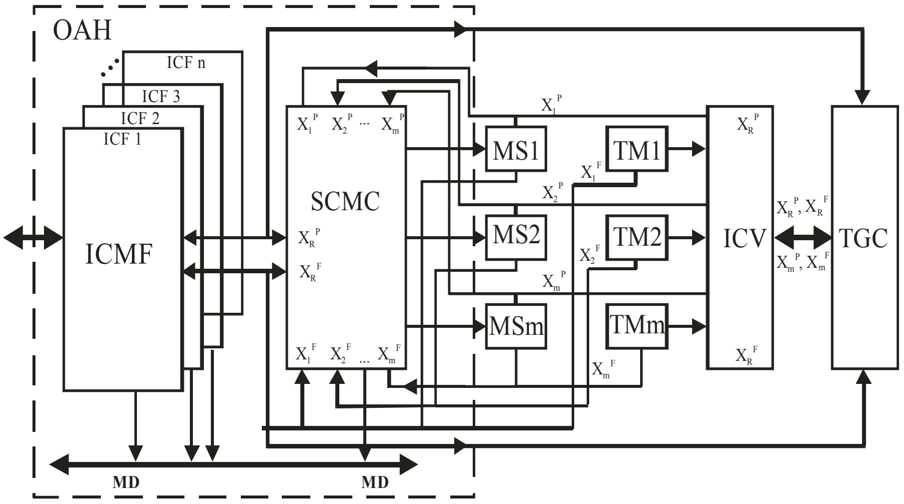


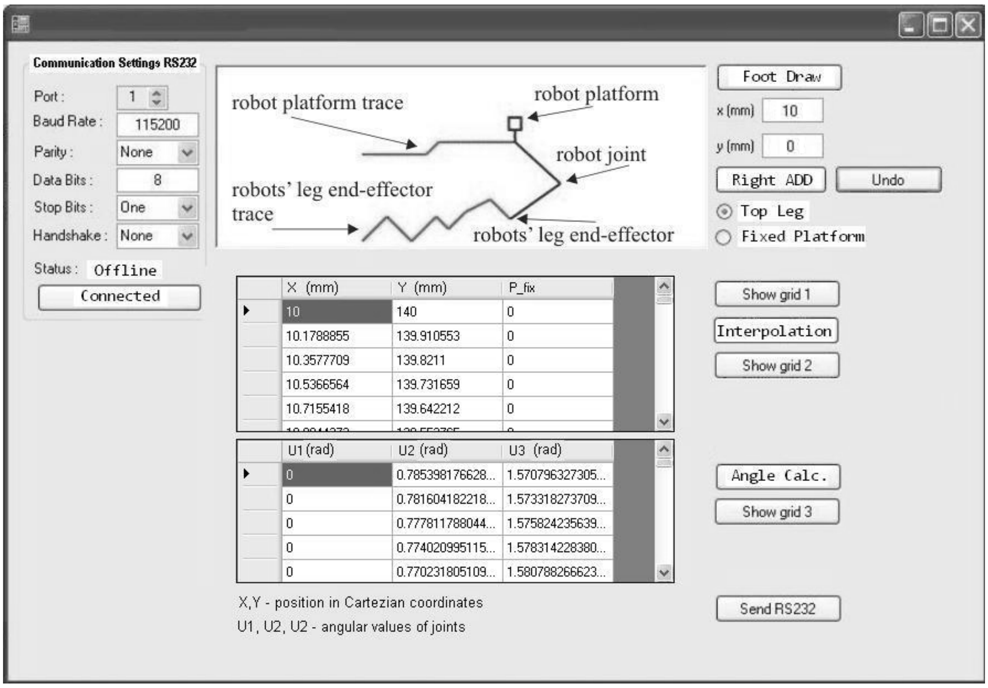
Fig. 4 – The experimental control system architecture.

The architecture of the walking robots' control system [5,12-EU patent], employed in experiments, is presented in figure 4, in which SCMC is the control system developed by us, directly coupled with the servo-actuators MS1, MSm, where m is the number of the robots degrees of freedom, and receives signals of the TM1-TMm measurement transducers. These signals are transmitted to an ICV virtual control interface, which by processing generates the signals required for the graphical representation in 3D, or for a projection in 2D on a TGC graphic module. A number of n control interfaces functions ICF1-ICFn ensure the development of a control system with open architecture by adding a number of *n* control functions, supplementary to those provide by the classical SCMC mechatronic control, with the aim of allowing the implementation of the control methods presented, to which there can be added: contour-following functions, the tripod walk of a walking robot, gravity center and orientation control, by means of image processing. The ICMF multifunctional control mode of interfacing ensures the real-time control, the control of priorities, and the management of information exchange between the *n* interfaces of ICF control functions, interconnected through a high-speed data bus.

4. **EXPERIMENTAL RESULTS AND CONCLUSIONS**.

In this tackling of the walking robot, a complete structure of a hexapod walking robot, with 6 degrees of freedom for each leg, of which 3 degrees of freedom for positioning and 3 degrees of freedom for the orientation of the foot and some elements useful for their operative system building, are developed. Starting from the premise that the position and orientation of the robot are given by the highest level of control, at the operative level the method was developed for calculating the coordinates in the robot system for a robot leg support point and the control strategy of the walking robot, with implementation in a multiprocessor control system with real time multi-tasking execution (Fig. 3). The algorithms take particular forms depending on the configuration of the robot parameters [2, 10]. The gait quality is improved if the control is assisted by sensors (force / torque, inertial), providing accommodation to sloping land, with obstacles and bumps. The robot's walking types are issued from 3 block-programs located in ICF1-ICF4 control interfaces functions of the OAH system (Fig. 4) with Open Architecture [2, 7] which contains the modules: the block of walking shaping ICF1, which determine the succession and the way of moving the legs, the block of static stability control ICF2, granting the robot shift so that the system centre of gravity projection may remain inside the convex polygon formed by the leg's, the block of platform control ICF3, maintaining the prescribed height and the horizontal position of the platform.

A distributed control system architecture was integrated into the architecture so that it can be controlled with high efficiency and high performance. In order to determine the performance of the control system, the virtual projection method was applied [2, 5, 12]. The theoretical results obtained in the control strategy of the robot were confirmed by computer simulation through implementation in a TGC graphic module of the graphical representation in Fig. 5, in which two situations were taken into account: (1) the platform is moving discreetly on movement intervals, also having station points, (2) the platform is moving with a constant speed throughout the time the legs follow their trajectories. The graphic representation of a walking cycle outlining the successive leg configurations for a number of *q* cycles, where at the end of a walking cycle the leg moves into the configuration specific to the initial position, displaced with *d*, the length of a step.



Port : 1
Baud Rate : 115200
Parity : None
Data Bits : 8
Stop Bits : One
Handshake : None
Status : Offline

x (mm) 10
y (mm) 0

● Top Leg
○ Fixed Platform

X (mm)	Y (mm)	P_fix
10	140	0
10.1788855	139.910553	0
10.3577709	139.8211	0
10.5366564	139.731659	0
10.7155418	139.642212	0

U1 (rad)	U2 (rad)	U3 (rad)
0	0.785398176628...	1.570796327305...
0	0.781604182218...	1.573318273709...
0	0.777811788044...	1.575824235639...
0	0.774020995115...	1.578314228380...
0	0.770231805109...	1.580788266623...

X,Y - position in Cartezian coordinates
U1, U2, U2 - angular values of joints

Fig. 5 – Control of the movement of walking robots, visualized in a 2D projection.

The results show that the time necessary to perform program for the walking robot position control in Cartesian coordinates is 40% shorter by applying the new modelling method presented in Chapter 4 in which the functions f_i and g_i are used, as opposed to the classical method of inverted kinematics. Moreover, the short time execution will ensure a faster feedback, allowing other programs to be performed in real time as well, like the prehension force control, objects recognition, making it possible that the control system have a human flexible and friendly interface.

Received on 5 July 2010

ACKNOWLEDGEMENTS

The authors wish to express their gratitude to the Research and Educational Ministry (MEC) and to the Romanian Academy for their support of the program of work reported herein. This work was supported by CNCSIS –UEFISCSU, project number PNII – IDEI 005/2007.

REFERENCES

1. Chircor M., Curaj A., *Elements of kinematics and dynamic trajectory planning of industry robots*, Edit. Academiei Române, Bucharest, 2001.
2. Vlădăreanu L., Curaj A., Ion I., Velea L.M., Munteanu M., Mitroi D., Gal A., Vasile A., *Fundamentale and Aplicative Researche for Hybride Force-Position Control of Modular Walking Robots* (phase 2/2008 and 3/2009), Research Project no. 263/2007-2010 in the framework of Grants of CNCSIS Program IDEAS, PN II (National Program for Scientific Research and Innovation Technologies).
3. Jung-Yup Kim, Ill-Woo Park, Jun-Ho Oh, *Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor*, Springer Science, J. Intell Robot Syst, **48**, pp. 457-484, 2007; DOI: 10.1007/s10846-006-9107-8.
4. Rummel J., A. Seyfarth, *Stable Running with Segmented Legs*, The International Journal of Robotics Research, **27**, p. 919, 2008; DOI: 10.1177/027836490895136.
5. Vlădăreanu L., Ion I., Velea L.M., Mitroi D., Gal A., *The Real Time Control of Modular Walking Robot Stability*, Recent Advances in Electrical Engineering, A Series of Reference Books and Textbooks, WSEAS Press; Proceedings of the 8th International Conference on Applications of Electrical Engineering (AEE ’09), Houston, USA, pp. 179-186.
6. Chin Pei Tang and Venkat N. Krovi, *Manipulability-based configuration evaluation of cooperative payload transport by mobile manipulator collectives*, Robotica, **25**, pp. 29-42, 2006, Cambridge University Press; Doi: 10.1017/S0263574706002979.
7. Vlădăreanu L., *Open Architecture Systems for the Compliance Robots Control*, WSEAS Transactions on Systems, **5**, *9*, pp. 2243-2249, 2006.
8. Vlădăreanu L., Tont G., Ion I., Vladareanu V., Mitroi D., *Modeling and Hybrid Position-Force Control of Walking Modular Robots*, ISI Proceedings, Recent Advances in Applied Mathematics, Harvard University, Cambridge, USA, 2010, pp. 510-518.
9. Khalil T.R., Levinson D.A., *The use of Kane’s dynamic equations in robotics*, International Journal of Robotic Research, **2**, 1983.
10. Ion I., Vladareanu L., Muntanu R. A., Munteanu M.S., *The Improvement of Structural and Real Time Control Performances for MERO Modular*, in: *Advances in Climbing and Walking* Robots, Ed. Ming Xie, S. Dubowsky, Word Scientific Publishing, British Library Cataloguing, 2007, pp. 252-263.
11. Vlădăreanu L., Munteanu R.I., Curaj A., Ion Ion, *Open Architecture Systems for MERO Walking Robots Control*, Proceedings of the European Computing Conference, Volume 2, Series: Lecture Notes in Electrical Engineering, XXIV, **28**, 2009, Part VI – Advances in computer science and applications, Springer Verlag, pp. 437-444.
12. Vlădăreanu L., L. M. Velea, R. Munteanu, A. Curaj, S. Cononovici, T. Sireteanu, L. Capitanu, MS Munteanu, *Real time control method and device for robot in virtual projection*, Patented Invention EU, no. EPO-09464001, 2009.