IMPROVEMENT OF DYNAMICAL STABILITY FOR THE REAL TIME WALKING ROBOT CONTROL PERO

LUIGE VLADAREANU¹, OVIDIU ILIE SANDRU², DANIEL MITROI¹,

¹ Institute of Solid Mechanics of the Romanian Academy, C-tin Mille 15, Bucharest 1, ROMANIA, ² “Politehnica” University of Bucharest, 313 Splaiul Independenți, 060042 Bucharest, ROMANIA,

luigiv@arexim.ro; www.acad.ro

Abstract. The paper presents strategies for improvement of real-time dynamical stability control for a complex structure of hexapod walking robots with six degrees of freedom for each leg of which, three degrees of freedom for positioning and three degrees of freedom for orientation of the foot. Issues for direct and inverse kinematics of the structure of walking robots are analyzed, by determining the first three coordinates of the robot leg joints and by determination of the joint coordinates for the mechanism of orientation. By kinematic "decoupling" of the movement, a separation of positioning control from the orientation control in robot modeling appears. Linear invariants are studied to calculate the position and orientation of the leg support point to determine robot transfer matrix. In terms of dynamic modeling for robot motion control, walking schemes and dynamic control phases are developed. The control system architecture for the dynamic walking robot is presented in correlation with the control strategy which contains many real time control loops. In the end, a multi-microprocessor architecture with multi-tasking control is designed that allows a fast feedback loop for real time robot control while improving stability and flexibility performance.

Key-Words: kinematic walking robot, walking robot control, multi-microprocessor system, robot dynamic control, static and dynamic stability.

1. Introduction

Locomotory activity of walking robots, and - mainly - walking, fall into the category of movements with a high degree of automation. The mechanical system must be equipped with a large number of degrees of freedom, to form complex synergies, namely to control leg movements in real time to adapt to the robot's environment.

Robust and reliable functioning of walking robots in contact with objects in their environment is the basic requirement for their tasks as given applications. A simple method in which the contact force is used to modify the reference position’s trajectory of final endeffector of the robot, called "position adjustment" was proposed by Whitney (1977). The interaction with the environment is solved by Raibert, Craig (1981) and Manson (1980), providing force and position control in the decomposition “sub-space position” and “sub-space force” [1, 2, 3]. These two subspaces correspond to the directions of movement of the robot, which moves freely or constrained by the environment. By this approach some of the final endeffector Cartesian coordinates of the robot are position controlled while others are explicit force control. Separate processing by different laws for position control and force control requires significant training to the treatment of tasks and an exchange control loop in implementation; in addition, this method may cause instability problems, especially during transition between free and constrained movements [4, 5, 6].

Motion control of resistance to bending, which is essentially the default power control based on position was suggested by Lawrence and Stoughton (1987) and Kazerooni, Waibel, and Kim (1990). Salisbury (1980) presented an active control method of robot’s final endeffector apparent stiffness in Cartesian space. In this method the reference position is used as command to control the contact force, and reference points are not used for the forces.

In a similar paper, Hogan (1985, Karen 1986) introduced "impedance control", which seeks to establish a desired dynamic relationship between the position of robot’s final endeffector and the contact force. This methodology can often be implemented with some off-line burden planning, provides resistance to the uncertainties and disturbances, and may provide a stable transition between free and constrained movements (Hogan 1988). But, using the methodology, the impedance control device for the movement, the size of the contact force depends on the trajectory of the endeffector reference position and the location and environment stiffness. Ideally, the environmental parameters (eg. the location and stiffness) are known exactly, the position reference trajectory can be done before to produce the desired contact force. A main disadvantage of this method is that in cases where environmental parameters are not known precisely, impedance control system tends to lead to reduced performances in force tracking.

There are a variety of methods to solve an inverse kinematics problem (Pieper 68, Paul 81, Lee 83, Elgazaa 85, Pieper and Khalil 88) . Among them
is remarkable, for the facilities it offers, the method of Khalil Pieper and Paul [9]. The first one allows the inverse kinematics problem solving regardless of the values of geometrical characteristics of the robot, for robots with six degrees of kinematic couplings possessing three rotational joints with competing axis or three cinematic translations. Due to the flexibility and the fact that it has the solution of inverse kinematics problem, this "disengaged" structure with three rotational joints and competing axis, is found in most models of robots sold. The position of intersection point of the three axes is uniquely determined only by the variables q1, q2, q3. Another advantage of the decoupled structure is that it allows dissociation and separate treatment of positioning and orientation. Paul's method and Lee's method Elgazaaar treat each case individually, without supporting generalizations.

The article presents a dynamic control system for the PERO-WALK robot, a complex structure of hexapod walking robot, with six degrees of freedom for each leg of which three degrees for positioning movement and three degrees of movement to guide the feet. There are analyzed problems of direct and inverse kinematics of the walking robot structure, establishing the first three joints of the foot robot coordinates and coordinates joint determination of orientation mechanism. Linear invariants are studied to calculate the position and orientation of robot's leg support section, determining the transfer matrix using Olinda-Rodrigues parameters. In terms of dynamic modeling of the PERO-WALK robot motion gait schemes and dynamic control phases are developed. As an application, a robot joint control method after a predetermined trajectory is presented. Finally, a multi-microprocessor architecture with multi-tasking control is designed that allows a rapid response in feedback loop for real time control of robot with improved performance stability and flexibility.

2. Direct and inverse kinematic control of the PERO-Walk 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. Considering the case of revolute-geometry robot all joints are rotational around the freedom axis [7, 8, 15]. In general having six degrees of freedom the manipulator mathematical analysis becomes very complicated.

There are two dominant coordinate systems: Cartesian coordinates and joint 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 [1].

To determine the D-H transformation matrix (fig.1) it is assumed that the Z-axis (which is the system’s axis in relation to the movement surface) is the axis of rotation in each frame, with the following notations:

- \( \theta_j \) - joint angle is the joint angle positive in the right hand sense about \( j Z \);
- \( a_j \) - link length is the length of the common normal, positive in the direction of \( (j+1)X \);
- \( \alpha_j \) - twist angle is the angle between \( j Z \) and \( (j+1)Z \), positive in the right hand sense about the common normal;
- \( d_j \) - offset distance is the value of \( j Z \) at which the common normal intersects \( j Z \); as well if \( jX \) and \( (j+1)X \) are parallel and in the same direction, then \( \theta_j = 0 \);
- \( (j+1)X \) is chosen to be collinear with the common normal between \( j Z \) and \( (j+1)Z \) [2, 6, 17].

**Figure 1 illustrates a robot position control based on the Denavit-Hartenberg transformation.**
respectively s and c are abbreviations for sinus and cosinus.

Fig. 2. The kinematics structure of an element of movement for robot steps PERO-WALK.

For the walking robot PERO-WALK with kinematic structure of a leg shown in figure 2, there is obtained an initial \( \theta_i \) of 0 degrees with the D-H parameters in Table 1.

<table>
<thead>
<tr>
<th>Joint</th>
<th>( \alpha_i ) [grad]</th>
<th>( a_i ) [grad]</th>
<th>( b_i ) [grad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-90</td>
<td>0</td>
<td>-0.149</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.432</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>0.02</td>
<td>-0.432</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
<td>0</td>
<td>-0.056</td>
</tr>
</tbody>
</table>

Tabel 1. D-H parameters for PERO-WALK walking robot

The coordinates of a point of positioning element, modeling direct kinematics using calculation of homogeneous transformation matrices for hexapod stepping robot Pero, are given by the following relations:

\[
X_4 = 0.432 \cos^3 \theta(t) + 0.149 \sin \theta(t) - 0.864 \cos^2 \theta(t) \cdot \sin \theta(t) - 0.02 \left[ \cos^3 \theta(t) - \cos \theta(t) \sin \theta(t) \right] \\
Y_4 = -0.149 \cos \theta(t) + 0.432 \cos \theta(t) \sin \theta(t) - 0.864 \cos \theta(t) \cdot \sin \theta(t) - 0.02 \left[ \cos^3 \theta(t) - \cos \theta(t) \sin \theta(t) \right] \\
Z_4 = -0.432 \sin \theta(t) + 0.04 \cos \theta(t) \sin \theta(t) + 0.432 \cdot \sin^2 \theta(t) + \cos^2 \theta(t) \\
\]

Modeling inverse kinematics. Modeling inverse kinematics [MGI] allows the generation of joint coordinates in a desired position of the robot’s point of support expressed in the operational details (environment). For certain points of the robot’s space that are no solutions by solving equations using the inverse kinematics model. In these cases, called singularities, which happen quite often, we resort to numerical methods[12, 13, 14].

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 inverse kinematics problem if we can compute all configurations which attain a given position. Not all articulated mechanisms satisfy this condition. After Roth, for robots with less than six degrees of freedom there is always an inverse kinematics solution to the problem. The robots with six degrees of freedom have a solution to the problem of inverse kinematics, if either of the following characteristics is met: possessing three translational joints, possessing three rotational joints with concurrent axes, have a rotation and a pair of coaxial translations, having two pairs of rotational joints with concurrent axes.

For the PERO walking robot, starting from the relationship 1 there is obtained:

\[
^3p_4 = ^3T_4 \\
\begin{bmatrix}
0 \\
0 \\
0 \\
1 \\
\end{bmatrix}
= \\
\begin{bmatrix}
C_{\theta_4} & -S_{\theta_4} & 0 & d_4 \\
C_{\alpha_4}S_{\theta_4} & C_{\alpha_4}C_{\theta_4} & -S_{\alpha_4} & -r_4S_{\alpha_4} \\
S_{\alpha_4} & S_{\alpha_4}C_{\theta_4} & C_{\alpha_4} & r_4C_{\alpha_4} \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix} 0 \end{bmatrix}
\]

Respectively:

\[
^3p_i = \begin{bmatrix} d_i \\
r_iC_{\alpha_i} \\
1 \end{bmatrix} \\
\]

Using the general form of \(^3T_3\), we can determine the solutions of the equation as:

\[
f_i(\theta_i) = C_{\alpha_i}d_i + S_{\alpha_i}S_{\alpha_i}r_i + d_i \\
f_i(q_i) = C_{\alpha_i}(S_{\alpha_i}d_i - C_{\alpha_i}S_{\alpha_i}r_i) - S_{\alpha_i}(C_{\alpha_i}r_i + r_i) \\
f_i(q_i) = S_{\alpha_i}(S_{\alpha_i}d_i - C_{\alpha_i}S_{\alpha_i}r_i) + C_{\alpha_i}(C_{\alpha_i}r_i + r_i) \\
\]

Similarly, noting the equations gi quadrant a point expressed in coordinates robot gets:

\[
g_i(q_i) = F_i(\theta_i, q_i) + d_i \\
g_i(q_i) = C_{\alpha_i}F_i(q_i, q_i) - S_{\alpha_i}F_i(r_i, q_i) \\
g_i(q_i) = S_{\alpha_i}F_i(q_i, q_i) + C_{\alpha_i}F_i(r_i, q_i) \\
\]

resulting equations for the angular coordinates of the robot in a certain position depending on the point of support expressed in the coordinates of the foot robot environment:
Starting from equation (5) - (8) has been developed a new method of calculating the coordinates of the robot system [10, 11, 18] for the PERO-WALK robot control.

3. Linear invariants method

A very effective method of motion trajectory of robots walking leg is determining invariants. In the construction of a mechanism for moving robots are necessary three degrees of freedom for positioning and three for orientation. Almost all robots have concurrent axis orientation mechanism which is a classical solution and known as an “uncoupled” structure. By adopting this solution (by reason of the existence of the solution to the problem of inverse kinematics), the problem of positioning is separated from the problem of orientation.

The practice requires determination coordinates joint for both problems of positioning, and orientation mechanisms. Instead, the orientation problem is much more complex, which is why different ways of defining it were sought and identified, such as: directors cosines, Hartenberg-Denavit parameters (HD), Euler angles, invariants.

Invariants were imposed because of their remarkable properties of not being affected by changing axes. There are several types of which the most important invariants are linear and Olinde-Rodrigues’s invariants. The linear invariants are defined by the vectors and trace of a rotation matrix Q. Orientation of the endeffector towards a fixed reference trihedral is done using the matrix Q:

\[ Q = \prod_i Q_i \]  

(9)

where \( Q \) is defined as the rotation of trihedral i+1 towards trihedral i, modeled by the D-H method. To define the rotation it is enough to define its rotation axis and rotation angle \( \Phi \). The rotation axis can be defined by the unit vector which is determined by the three projections on axes.

\[ \text{vect}(Q) = u \sin \Phi \]  

(10)

\[ \text{tr}(Q) = 1 + 2 \cos \Phi \]  

(11)

\( \text{vect}(Q) \) is a vectorial invariant. It however does not correspond to a position of a rotation axis, but does to two identical rotation axis with additional rotational angles. Therefore relationship (11) has been introduced, where \( \text{tr}(Q) \) is the scalar invariant. We have four relationships that define the three parameters that provide the body orientation in space. Apparent indetermination is removed by the relation between the vector and the scalar invariant:

\[ \text{[vect}(Q)]^2 + \frac{1}{4} \text{[tr}(Q)-1]^2 = 1 \]  

(12)

Matrix components \( v = \text{vect}(Q) \) are calculated using the formula:

\[ v_i = \frac{1}{2} e_{ijk} q_k j \]  

(13)

Where \( q \) and \( k \) are the components of matrix Q and \( e_{ijk} \) is the Ricci symbol and is described by the phrase:

\[
\begin{cases}
1, & \text{if } i + j + k = 123, 231, 312 \\
-1, & \text{if } i + j + k = 123, 231, 312 \\
0, & \text{if } i + j + k = \text{not in } 231\text{ or } 123 \text{ or } 312
\end{cases}
\]

(14)

In these conditions for \( v_i \):

\[ v_1 = \frac{1}{2} (q_{23} - q_{32}) \]

\[ v_2 = \frac{1}{2} (q_{31} - q_{13}) \]

\[ v_3 = \frac{1}{2} (q_{12} - q_{21}) \]

(15)

We get:

\[ \text{vect}(Q) = \begin{bmatrix} -\frac{1}{2}(\cos \theta \sin \alpha + \sin \alpha) \\ \sin \theta \sin \alpha \\ -\frac{1}{2}(\cos \theta \sin \alpha + \cos \theta) \end{bmatrix} \]

(16)

Finally, the results obtained in the determination of invariants for walking robots PERO, calculated using Mathematica 2.2, are (17):

\[
q_1 = \cos(q_1) - \cos(q_1) \cos(q_2) - \cos(q_1) \sin(q_2) + \cos(q_2) \sin(q_1) \sin(q_2) - \cos(q_2) \sin(q_2) \sin(q_1) - \cos(q_2) \sin(q_2) \sin(q_1) + \cos(q_2) \sin(q_2) \sin(q_1)
\]

(17)

Starting from equations (9)- (17) a new method has been developed for the calculation of the coordinates in a robot system for a support point of robot’s leg in its robot environment (MGI) with implementation in a control system multiprocessor with real-time multi-tasking execution, presented below, which allows control of motion trajectory of the support point of PERO-WALK robot.

4. Multi-tasking architecture for PERO-Walk control

Based on modeling presented in the above chapters, it a real-time control of the motion trajectory has been developed for a robot mechanism PERO. Starting with the robot’s position control with position transformations from robot coordinates into
Carthesian coordinates using the Denevit-Hartenberg method and coordinate calculation on a robot axis using linear invariants methods for closing the feedback loop is shown in figure 5. The acquisition and processing multiprocessor system ensures the real time implementation for the control algorithm in Carthesian coordinates for the position of industrial shaping robots using numerical processing of the Jacobean matrix obtain through direct cinematic of the robot and of the robot axis respectively, using linear invariants matrix vect(Q) and invariant scalar tr(Q).

Each axis is provided by the CPU with the correspondent positioning set for the programmed applications and control of the dependences between axes. The UCP program, with data machine (MDS) transmission and initialization, using execution vectors INIT, MDS_1, MDS_2 and MDS_3. If one of the parameters of the machine data set (MDS_n) is beyond the acceptable limits, the error can be read using the ERRFI component. In this case, the error ends with parameter “Default” (error). The functional structure of UCP program and the communication structure with the axis processor (PE) are shown in fig 4. The parameters and commands that are pre-established by the positioning vectors POS_x and the communications vectors KOM_x are processed by the interpolator in axis CPU. The interpolator forms the programmed position and velocity following pre-submitted and transmitted data to the positioning control circuit.

The structure of the control system and the action of the MDS parameters on the regulating positioning circuit are shown in figure 5 with its main functions and components: $K_p$, the amplification coefficient is generating the position size in the regulating circuit, the static defect tracking which indicate maximum allowable deviation from the prescribed position and
the current position respectively, the dynamic tracking defect is proportional to the speed.

5. Experimental Results and Conclusions

In view of technological developments in making experiments there was developed and designed a control system presented in figure 6. The system allows operation of a control experiment in real time walking robots [10, 16, 19], by virtual projection on graphical interface that runs on a PC. The motion trajectory generated by GRMR module is handled by a PLC which permits closed loop control according to the method chosen (SCC module) to drive each axis of movement by positioning frequency converters ACSM. To simulate the operation of the robot in charge, the actuators module MA, which contains the actuators on each axis, is coupled with a rigid set of actuators MS with variable load.

Fig. 6. Technological Development of the experiments

Force control was archived by using the torque traducers TC, and position control by using incremental traducers IGR. Over-current protection signals, overvoltage, movement speed, robot race switches and start position are taken from ACSM converter, specialized in precision positioning and control at high velocity.

Application of walking robots simulation through virtual projection. The application’s purpose is to generate, visualize and send referrals of walking robot consisting of 3 modules of 2 feet each, positioned at the triangle corners.

Application of walking robots simulation through virtual projection. The application’s purpose is to generate, visualize and send referrals of walking robot consisting of 3 modules of 2 feet each, positioned at the triangle corners.

Two of the windows of walking robots motion control HFPC are presented in figures 7 and 8. The trajectories of both feet are made of manually input segments or by coordinates of points in mm, received by the serial transmission of a file, relative to an axis system attached to the platform.

Segments can be added one by one, forming a visible path through the representation of 2D and 3D. You can choose to generate referrals through serial transmission, ie to generate 16-bits related angles of each joint for the robot to follow the path defined variables set.

Fig. 7. Simulated robot control window

The results show that the time necessary to perform the program for the walking robot position control in Cartesian coordinates is 30% shorter by applying the new modelling method presented in chapter 3 in which the vect \( Q \) and \( tr(Q) \) are used, as opposed to the classical method of inverted kinematics.

Fig. 8. View from the right side camera movement

Moreover, the short time execution will ensure a faster feedback, allowing other programs to be
performed in real time as well, like theprehension force control, objects recognition, making it possible that the control system have a human flexible and friendly interface.

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. The work took place as part of research project ID 005/2007, no. 263/2007-2010 in the framework of Grants of CNCSIS Program IDEAS, PN II (National Program for Scientific Research and Innovation Technologies).

REFERENCES