ISBN 978-1-84626-xxx-x
Proceedings of 2011 International Conference on Optimization of the Robots and Manipulators (OPTIROB 2011)
Sinaia, Romania, 26-28 Mai, 2011, pp. xxx-xxx

Aspects of biped robot stabilization

Mate Csaba Zoltan[1] and Cristea Luciana [1]

[1] Precision Mechanics and Mechatronics Department, Transilvania University of Braşov, Braşov, Romania

Abstract. This paper aims to present special issues concerning the analysis of mobile robots with kinematic motion effects on the stability study. In the analysis, the authors used inverse kinematics, which enables rapid modeling and identifying solutions as regards the stability of bipedal robots.

Keywords: Mechatronics, Biped robot, Kinematics, Simulation, Stability

1. Introduction

For a biped robot the sole position and orientation is known, defined within the domain of exterior coordinates, if a $\vec{q}$ vector is given with join coordinates. In the case of a robot with n freedom degree, the vector of joint variables is [5] the following:

$$\vec{q} = [q_1, q_2, \ldots\ldots, q_{n-1}, q_n]^T \tag{1}$$

And the vector of unknown exterior coordinates is the following:

$$x_q = \left[x_{q1}, x_{q2}, \ldots\ldots, x_{qn-1}, x_{qn} \right]^T \tag{2}$$

The equation below is the only solution for the so called direct kinematics problem.

$$x_q = f(\vec{q}) \tag{3}$$

If we know sum of the joint's setup and from this we define the coordinate system's position, according to the sole's centre point, as well as its orientation, thus we solved the direct kinematics problem.

Inverse kinematics problem means that if the sole's expected position and orientation (within the exterior coordinates) is known, and then with which joint setups can we obtain this. In other words we can say that we are looking for only solution.

$$\vec{q} = f^{-1}(x_q) \tag{4}$$

This task is more complex then the direct kinematics problem, since it is not linear, we have to solve equations containing trigonometric functions. The symbolic solution for kinematics equations of biped robots is of great importance for the efficient controllability of these robots. In the world of low-cost computers, the real-time motion control is an increasingly important requirement. In order to achieve this feature, the lowest

+ Corresponding author. Tel.: +40268416352; fax: 0040268416352.
E-mail address: mcsabazoltan@yahoo.com.

computational demand requesting method has to be used. The use of symbolic [5] solution, opposing to numerical methods, is important because accelerates the manipulator trajectory control signals needed to be determined according to the track.

The symbolic form of the kinematics equations describes explicitly in trigonometric form the biped robots' sole's position and orientation according to the joint coordinates. In this case, the equation in the range of real numbers can be solved with the minimal possible operations.

2. Kinematic modelling

Direct kinematics problem is to define all relationships that end-effector position (foot of biped robot) based on joint coordinates practically [3] [4], it ensures internal coordinates conversion (joint) Coordinate external (operational).

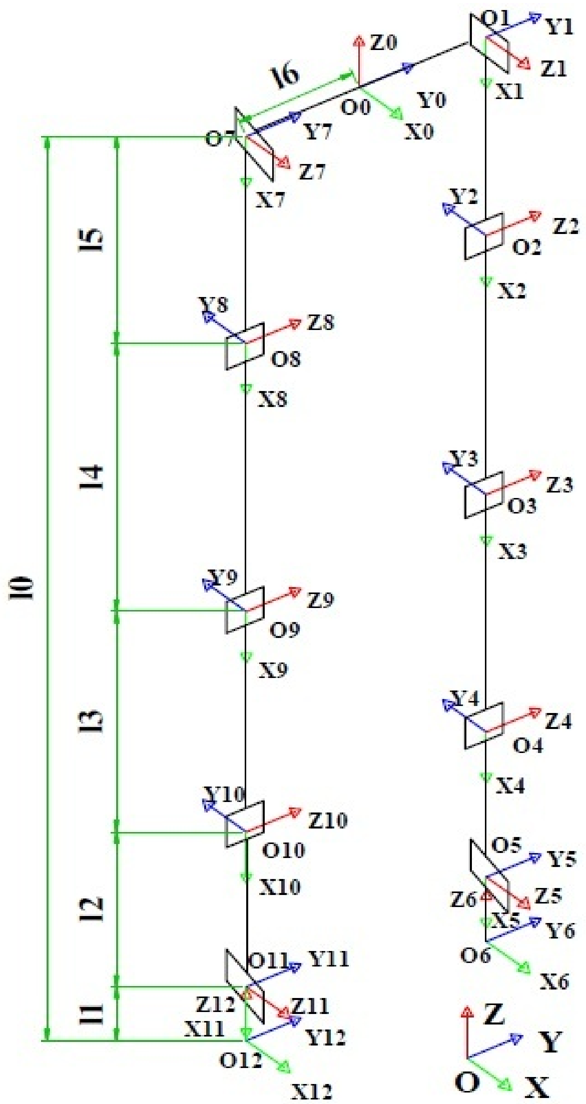

Figura.1. Kinematic model

Biped robot kinematics equations are (Fig. 1.):

$$T_{06} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (5)$$

Where:

$a_{11} = -\cos(q_2 + q_3 - q_4)$

$a_{12} = \sin(q_5) * \sin(q_2 + q_3 - q_4)$

$a_{13} = \cos(q_5) * \sin(q_2 + q_3 - q_4)$

$a_{14} = l_2 * \sin(q_2 + q_3 - q_4) + l_1 * \cos(q_5) * \sin(q_2 + q_3 - q_4) - l_3 * \sin(q_2 + q_3)$

$a_{21} = -\sin(q_1) * \sin(q_2 + q_3 + q_4)$

$a_{22} = \cos(q_1) * \cos(q_5) + \sin(q_1) * \sin(q_5) * \cos(q_2 + q_3 + q_4)$

$a_{23} = \cos(q_1) * \cos(q_5) + \sin(q_1) * \cos(q_5) * \cos(q_2 + q_3 + q_4)$

$a_{24} = -l_1 * \cos(q_1) * \cos(q_5) - l_2 \sin(q_1) * \cos(q_2 + q_3 + q_4) - l_3 * \sin(q_1) * \cos(q_2 + q_3) - l_5 \sin(q_1) + l_6$

$a_{31} = \cos(q_1) * \sin(q_2 + q_3 + q_4)$

$a_{32} = -\sin(q_1) * \cos(q_5) + \cos(q_1) * \sin(q_5) * \cos(q_2 + q_3 + q_4)$

$a_{33} = -\sin(q_1) * \sin(q_5) + \cos(q_1) * \cos(q_5) * \cos(q_2 + q_3 + q_4)$

$a_{34} = -l_1 * \sin(q_1) * \cos(q_5) - l_2 \cos(q_1) * \cos(q_2 + q_3 + q_4) - l_3 * \cos(q_1) * \cos(q_2 + q_3) - l_5 \cos(q_1) + l_0$

$a_{41} = 0$

$a_{42} = 0$

$a_{43} = 0$

$a_{44} = 1$

Convert coordinate joint operational details is done by solving the direct kinematics problem and coordinate joint operational coordinate conversion is done by solving the inverse kinematics problem.

Inverse kinematics problem allows the calculation [5], [2] coordinates of the joints, which provide end-effector in the desired position and orientation, given the absolute coordinates (operational). When the problem is the inverse kinematics solution, it is the inverse geometrical model. If we cannot find an analytical solution for inverse kinematics problem (which happens quite frequently) we resort to numerical methods, but whose weakness is the sheer volume of calculations. The most common method is Newton-Raphson method. Among these features is remarkable for the way it offers and Khalil Pieper and Paul's method. Pieper and Khalil's method allows solving inverse kinematics problem regardless of the values of the robot geometrical

3. Centre of gravity

During walking, the feet are subjected to the action of forces [1], [2], [6] and moments of inertia and gravity forces. Balancing the forces of gravity is to reduce the mechanical work consumed for drive motor. The position of equilibrium of a system subject to stationary and links under the action of forces is given, is called stable equilibrium, if for a sufficiently small arbitrary variation of the coordinates of its points and arbitrary speeds sufficiently small print of these points, the system will move all the time remaining in the vicinity of equilibrium position. To determine the center of gravity G of a facility plan is sufficient to determine the position vector r_G thereof with the relationship:

$$\overrightarrow{r_g} = \frac{\sum_{i=0}^{10} m_i * r_i}{\sum_{i=0}^{10} m_i} \quad (6)$$

$$\overrightarrow{r_g} = x_g * \vec{i} + y_g * \vec{j} + z_g * \vec{k} \quad (7)$$

Where: m_i is the mass of the i element; r_i is the position vector of center of gravity of element i.

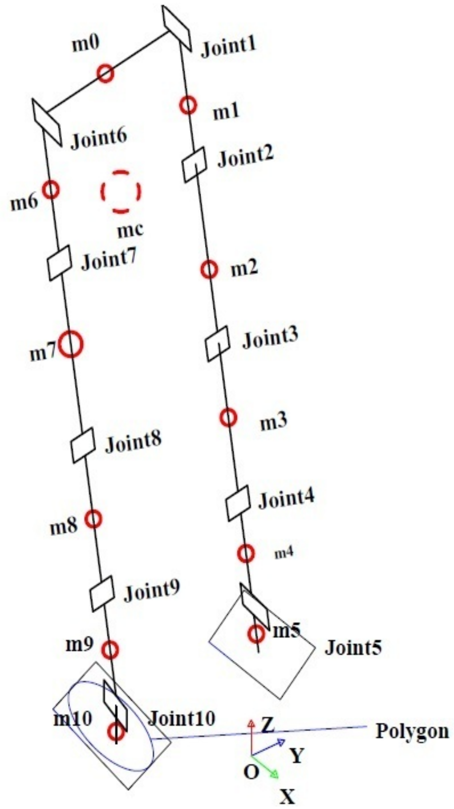

Figura.3. Robot model and polygon by one leg

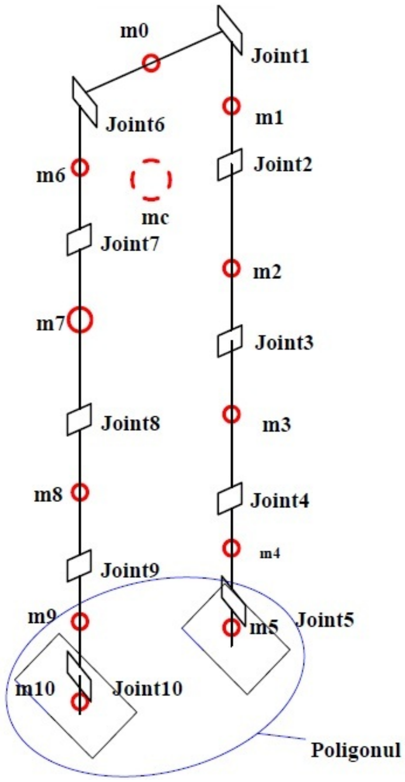

Figura.2. Robot model and polygon by two leg mc-weight center

During walking robot must be stable. This means that its centre of gravity must fall within the polygon (figura.2.) which consists of the two legs of the robot. If the left leg is raised then the centre of gravity must fall within the polygon of the right leg (Figure 3). If the projection centre of gravity doesn't fall within the polygon, the robot is unstable. Then it is necessary to tilt (figura.3.) robot to the right. We know direct and inverse kinematics of the robot and in this way we can calculate the coordinates of the joints. After calculating the joint coordinates it must be checked if the centre of gravity falls within the polygon or, if other compensation is needed for the robot to tilt to the right until the centre of gravity projection falls within the polygon. Easier to solve the problem can be done with an algorithm based on direct and inverse kinematics, with which to verify and ensure the necessary compensation to realise that the position of gravity centre projection of the robot is falling inside the stability polygon.

4. Conclusion

If a biped robot's inverse kinematics problem is solved well, then this helps a lot on the stability, because the well positioned ligament's overall centre of weight has to fall in the given sole's polygon, so that the robot wouldn't tumble over.

Inverse kinematics problem allows the calculation coordinates of the joints, which provide end-effector in the desired position and orientation, given the absolute coordinates (operational).

Easier to solve the problem can be done with an algorithm based on direct and inverse kinematics, with which to verify and ensure the necessary compensation to realise that the position of gravity centre projection of the robot is falling inside the stability polygon.

5. Acknowledgements

This paper is supported by the sectional Operational Programmer Human Resources Development (SOP HRD), ID59321 financed from the European Social Fund and by the Romanian Government References.

6. References

[1] Christine Azevedo a, Philippe Poignet b, Bernard Espiau c, Artificial locomotion control: from human to robots, [a]*CNRS DPA P3M, 31 chemin J. Aiguier, 13402 Marseille Cedex 20, France, b LIRMM, 161 rue Ada, 34392 Montpellier Cedex 05, France, c INRIA Rhˆone-Alpes, 655 avenue de l'Europe, 38334 St Ismier Cedex, France,* Robotics and Autonomous Systems 47 (2004) 203–223

[2] Jimmy Or, A hybrid CPG_ZMP control system for stable walking of a simulated flexiblespine humanoid robot, Neural Networks 23 (2010) 452_460

[3] Peiman Naseradin Mousavi, Ahmad Bagheri, Mathematical simulation of a seven link biped robot on various surfaces and ZMP considerations, *Department of Mechanical Engineering, Guilan University, Rasht, Iran*, Applied Mathematical Modelling 31 (2007) 18–37

[4] Peiman Naseradin Mousavi[a] , C. Nataraj[b] , Ahmad Bagheri[a] ,Mahdi Alizadeh Entezari[c], Mathematical simulation of combined trajectory paths of a seven link biped robot,*a Department of Mechanical Engineering, Guilan University, Rasht, Iran, bDepartment of Mechanical Engineering, Villanova University, Villanova, USA, c Department of Mechanical Engineering, Amir Kabir University, Tehran, Iran ,Applied Mathematical Modelling 32 (2008) 1445–1462*

[5] Vámossy Zoltán, "Automatizált eszközök", Budapesti Műszaki Főiskola, Budapest, 2002. Június

[6] Yutaka Nakamura a,b, Takeshi Moria, Masa-aki Satoc, Shin Ishiia, Reinforcement learning for a biped robot based on a CPG-actor-critic method, *a Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan, b Osaka University, 2-1 Yamadaoka, Suita, Osaka 565-0871, Japa, c ATR Computational Neuroscience Laboratories, 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan,* Neural Networks 20 (2007) 723–735