

Article

Modelling of the Biped Robot with 10 DoF

Ahmet Shala ¹, Rame Likaj ^{1,*} and Xhevahir Bajrami ²

¹ Faculty of Mechanical Engineering, University of Prishtina, Prishtina, Kosovo; ahmet.shala@uni-pr.edu

² Intelligent Handling and Robotics, Favoritenstraße 9/E325 A6, Vienna, Austria; xhbajrami@hotmail.com

* Correspondence: rame.likaj@uni-pr.edu; Tel.: +37744214947

Abstract: First, a brief overview is provided on humanoid robots, and also models for the dynamic behavior are discussed. As base for these models these two methods Denavit-Hartenberg and Newton-Euler are used. Main aim of this work is to investigate the stability of a biped robot developed from IHRT. There is currently the low base of robot - consisting of feet, legs, hips and upper part of robots body. This structure currently has ten degrees of freedom.

Keywords: kinematic; dynamic; biped; Simulink; Archie

1. Introduction

The goal is to approximate the trajectory of the humanoid foot of a robot which is produced by the torque generated by the actuators input variables. Such a model has to fulfil two conflicting objectives. It must include enough details to represent the real behavior of the robot with sufficient accuracy, and it should permit an efficient, stable evaluation not only of the model equations but also of their derivatives that are needed in optimization. Such a model for the dynamic behavior is necessary for the mechanical design of the structure [6], choice of actuators, development of control strategies, and simulation humanoid motion.

2. Modeling – Kinematics and Dynamics

The forward kinematics problem deals with the relation between the individual joints of the humanoid robot and position and orientation of the tool or end-effector.

Forward Kinematics can be described as: (angular position)

- The length of each link, the angle of each joint,
- The position of each point of the robot (i.e. it's (x, y, z) coordinates.

In inverse kinematics the orientation of articulated parts is calculated from the desired position of certain points on the model.

It is also distinguished from other animation systems by the fact that the motion of the model is defined directly by the animator, no account is taken of any physical laws that might be in effect on the model, such as gravity or collision with other models:

- The length of each link, the position of some point on the robot
- The angles of each joint needed to obtain that position

Bipedal locomotion requires an accurate forward and an inverse kinematic model in order to specify desired joint angles related to the base and effector trajectories.

In addition, controller design and development justifies employing precise inverse kinematics and dynamic models to satisfy stability and agility requirements in biped robots such as Archie, Figure 1.

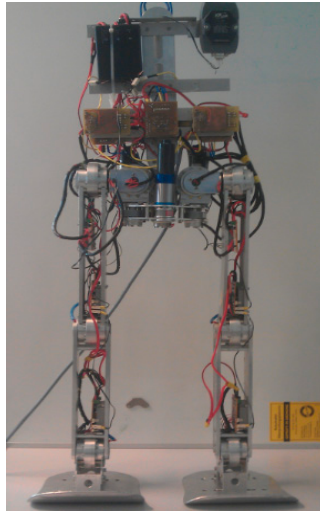


Fig. 1. Biped robot - Archie [2, 3]

Archie has (30) degrees of freedom (DOF), including 7 DOF in each leg, 3 DOF for each hip joint, 6 DOF on each arm including, 2 DOF for neck and head joints, 2 DOF for Torso which are introduced as ankle roll, ankle pitch, knee pitch, hip pitch, hip roll, and hip yaw-pitch. Yaw-pitch joints of the hips are physically bound and driven with one servo motor [4].

Table 1 D-H parameters for 10 DOF lower body

#Link	a_i [cm]	α_i [degree]	d_i [cm]	θ_i [degree]
1	a_1	0	0	$\theta_1^* + \pi/2$
2	a_2	0	0	θ_2^*
3	a_3	0	0	θ_3^*
4	0	$-\pi/2$	$-d_4$	$\theta_4^* + \pi/2$
5	0	$\pi/2$	0	$\theta_5^* + \pi/2$
6	a_6	0	0	θ_6^*
7	0	$\pi/2$	0	$\theta_7^* + \pi/2$
8	0	$-\pi/2$	0	$\theta_8^* + \pi/2$
9	$-a_9$	0	d_9	θ_9^*
10	$-a_{10}$	0	0	θ_{10}^*

* Joint variable

Denavit-Hartenberg transformation matrix for adjacent coordinate frames, i and $i-1$

$$A_i = \begin{bmatrix} c(\theta_i) & -s(\theta_i) \cdot c(\alpha_i) & s(\theta_i) \cdot s(\alpha_i) & a_i \cdot c(\theta_i) \\ s(\theta_i) & c(\theta_i) \cdot c(\alpha_i) & -c(\theta_i) \cdot s(\alpha_i) & a_i \cdot s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

In Figure 2 and 3 are represented angles for right and left hip-ankle-knee for two steps, first from 0 to 30 cm and second step from 30 cm to 60 cm.

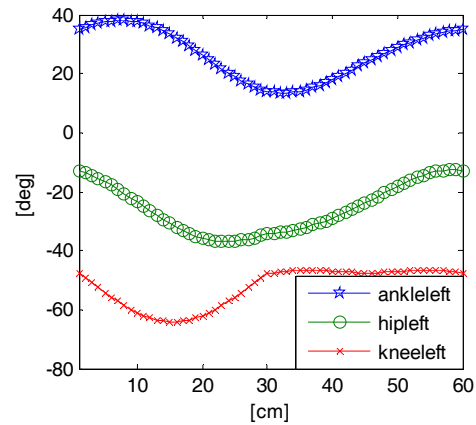
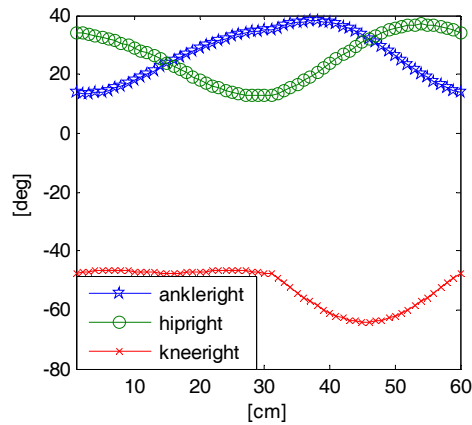


Fig. 2. Joint angle; hip-ankle-knee right for two steps Fig. 3. Joint angle; hip-ankle-knee left for two steps

A humanoid robot is basically a positioning device. To control the position we must know the dynamic properties of the Humanoid in order to know how much force to exert on it to cause it to move: too little force and the Humanoid is slow to react; too much force and the foot may crash into objects or oscillate about its desired position. A humanoid robot is a nonlinear, nonholonomic multivariable Systems because of the large number of DOF's. In the following the equations for the kinematics and dynamics behavior for the lower part of a humanoid robot with 10 DOF's will be derived. This part is concerned with the development of the dynamic model for 10-Dof robot and their kinematic and dynamic equations [2].

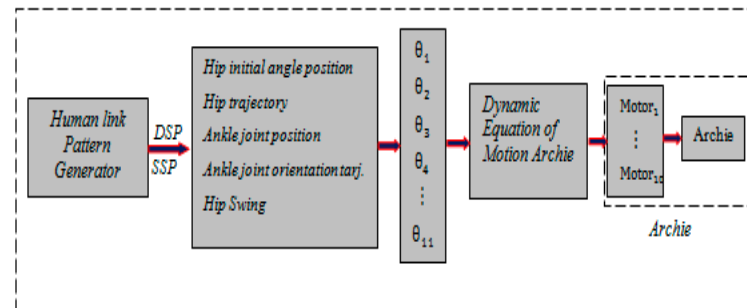


Fig. 4. Architecture for the dynamic model [2]

In the literature two methods are available; the Euler-Lagrange formulation and the Newton-Euler formulation. The Newton-Euler formulation is quite different because each link of the humanoid is treated in turn. First there is a forward recursion describing its linear and angular motion, then a backward recursion to calculate the forces and torques. Both of these formulations are derived from first principles in (Mark W.S 2010), and including examples of how the methods can be applied. The resulting dynamic model is the same for both methods and can be written in matrix form as Angular velocity of the i -th coordinate frame (Siciliano 2000). Orthogonal rotation matrix R_i which transforms a vector in the i -th coordinate frame to a coordinate frame which is parallel to the $(i-1)$ -th coordinate frame is first 3×3 sub-matrices of R_i :

$$R_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \cdot \sin(\theta_i) & \sin(\alpha_i) \cdot \sin(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cdot \cos(\theta_i) & -\sin(\alpha_i) \cdot \cos(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \quad (2)$$

2.1. Forward Recursion

The forward recursion describes the linear and angular motion of the links, from link 1 to link 10. As a part of the forward recursion it is necessary to compute b_i , the axis of rotation for each joint i expressed in frame i . Prior to the recursions, these computations will be carried out right away for all joints to emphasize a great advantage of the Newton-Euler formulation. The rotation axis in frame 0 is given directly as:

$$z_0 = [0 \ 0 \ 1]^T \quad (3)$$

$$\text{Link 1 to Link 10: } \left. \begin{array}{l} \omega_0 \wedge v_0 = 0 \\ d\omega_0 \wedge dv_0 = 0 \end{array} \right\} \quad (4)$$

Angular velocity and acceleration are calculated from Equation (5) and (6) respectively, and becomes.

$$\left. \begin{array}{l} \omega_1 = R_1^T [\omega_0 + z_0 \cdot \dot{q}_1], \\ \vdots \\ \omega_{10} = R_{10}^T [\omega_0 + z_0 \cdot \dot{q}_{10}] \end{array} \right\} \quad (5)$$

$$\left. \begin{array}{l} d\omega_1 = R_1^T [d\omega_0 + z_0 \cdot \ddot{q}_1 + \omega_0 \times (z_0 \cdot \dot{q}_1)], \\ \vdots \\ d\omega_{10} = R_{10}^T [d\omega_0 + z_0 \cdot \ddot{q}_{10} + \omega_0 \times (z_0 \cdot \dot{q}_{10})] \end{array} \right\} \quad (6)$$

Acceleration of the end of the link and the center of the link are calculated from Equation (5) and (6) respectively, and becomes.

$$\left. \begin{array}{l} dv_1 = R_1^T [dv_0 + d\omega_1 \times p_1 + z_0 \cdot \ddot{q}_1 + \omega_1 \times (\omega_1 \times p_1)], \\ \dots \\ dv_{10} = R_{10}^T [dv_0 + d\omega_{10} \times p_{10} + z_0 \cdot \ddot{q}_{10} + \omega_{10} \times (\omega_{10} \times p_{10})], \end{array} \right\} \quad (7)$$

$$\left. \begin{array}{l} da_1 = d\omega_1 \times s_{1x} + \omega_1 \times (\omega_1 \times s_{1x}) + dv_1, \\ \vdots \\ da_{10} = d\omega_{10} \times s_{10x} + \omega_{10} \times (\omega_{10} \times s_{10x}) + dv_{10} \end{array} \right\} \quad (8)$$

2.2. Backward Recursion

The backward recursion calculates the forces and joint torques acting on the links, starting with link 10 and ending with link 1. Determining the joint torques is the ultimate goal of the Newton-Euler formulation, because the torques are the externally applied input to the model. Note that the force equation includes the gravity vector. This gravity vector differs for each link, but can easily be calculated with the use of rotation matrices as shown in the recursions below.

$$\text{Forces at Joints are: } \left. \begin{array}{l} F_1 = m_1 \cdot da_1, \\ \vdots \\ F_{10} = m_{10} \cdot da_{10} \end{array} \right\} \quad (9)$$

The net moments (N_i) exerted on link $i=1,2,3,\dots,10$ are:

$$\left. \begin{array}{l} N_1 = I_{1yy} \cdot d\omega_1 + \omega_1 \times (I_{1yy} \cdot d\omega_1), \\ \vdots \\ N_{10} = I_{10yy} \cdot d\omega_{10} + \omega_{10} \times (I_{10yy} \cdot d\omega_{10}) \end{array} \right\} \quad (10)$$

Moment of inertia is the rotational analog of the mass. According to the mechanical structure of a humanoid robot, all the movements are based on revolute joints.

Thus, finding the moment of inertia is necessary for modelling the joints. These coordinate's axes are called the principal axes. Information about the values of inertia for every link which are given in Table 2:

Table 2: Values of principal axes corresponding to every link of the robot

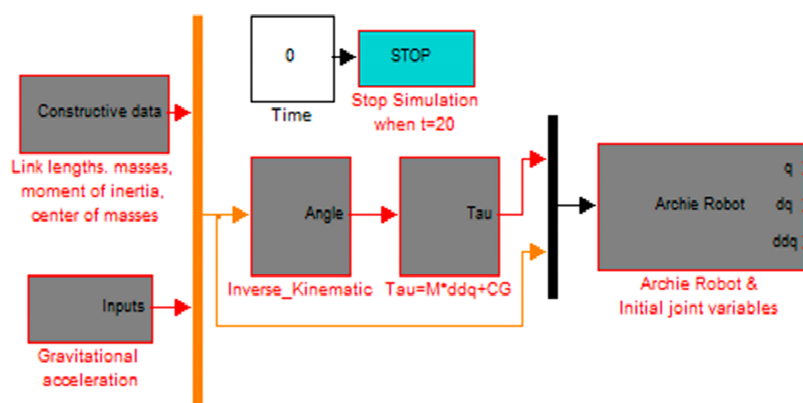
#Link	Inertia Matrix $I_{xx} [kgm^2]$	Inertia Matrix $I_{yy} [kgm^2]$	Inertia Matrix $I_{zz} [kgm^2]$
1	7.345×10^{-4}	6.4×10^{-4}	1.365×10^{-3}
2	1.765×10^{-4}	4.935×10^{-4}	4.12×10^{-3}
3	2.143×10^{-4}	6.375×10^{-3}	6.589×10^{-3}
4	9.333×10^{-4}	8.4×10^{-4}	9.333×10^{-4}
5	0.017	0.012	0.029
6	3.28×10^{-3}	1.313×10^{-4}	1.641×10^{-4}
7	9.333×10^{-3}	8.4×10^{-4}	9.333×10^{-4}
8	2.143×10^{-4}	6.375×10^{-3}	6.589×10^{-3}
9	1.765×10^{-4}	4.935×10^{-4}	4.12×10^{-3}
10	7.345×10^{-4}	6.4×10^{-4}	1.365×10^{-3}

Table 3 shows the weight and the position of the center of mass for each link based on their coordinate system.

#Link	Mass	Center of mass X	Center of mass Y	Center of mass Z
1	0.125 [kg]	0 [mm]	-53 [mm]	-4 [mm]
2	0.111 [kg]	0 [mm]	-4 [mm]	-26 [mm]
3	0.08 [kg]	0 [mm]	0 [mm]	-5 [mm]
4	0.075 [kg]	28 [mm]	0 [mm]	-130 [mm]
5	0.131 [kg]	98 [mm]	0 [mm]	-155 [mm]
6	0.048 [kg]	58 [mm]	23 [mm]	0 [mm]
7	0.049 [kg]	0 [mm]	0 [mm]	25 [mm]
8	0.346 [kg]	75 [mm]	70 [mm]	-15 [mm]
9	0.049 [kg]	0 [mm]	0 [mm]	22 [mm]
10	2.992 [kg]	0 [mm]	33 [mm]	249 [mm]

3. Simulation of the human a lower body

The simulation scheme used for the robot is developed in the Matlab/Simulink. To develop the motions and to predict the real results, the simulation scheme calculates ankle, velocity, acceleration and torque/force for all ten joints-Lower body [2, 5].

**Fig. 5.** Simulation scheme for lower body

The walking controller proposed above has been tested by simulation environment which makes use of two separated software packages. The mathematical model dynamic of the robot has been implemented in Maple.

This dynamical simulation environment takes care of all the dynamics, feet-ground interactions included. The control algorithm is then implemented in Simulink of the Matlab package.

In Fig. 5 the Simulink model is shown, and the block "Archie Robot" is the communication interface with m-file. The biped robot parameters used for simulations are those of the robot "Archie". In Fig. 6 and 7 joint torques are plotted in below.

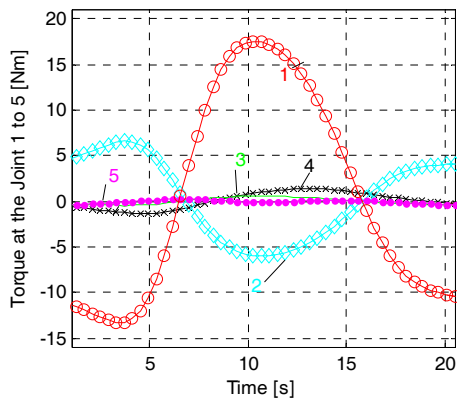


Fig. 6. Torques at Joint / left foot

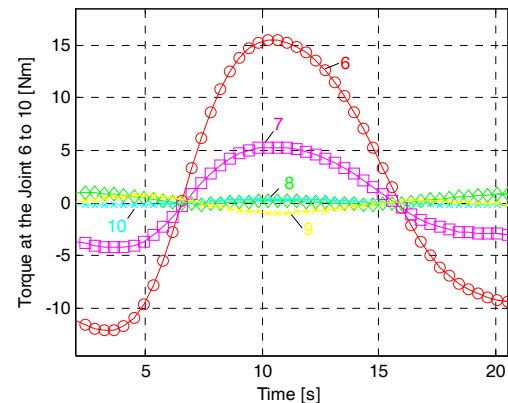


Fig. 7. Torques at Joint for left foot

5. Conclusions

Using Newton-Euler equations enable the computation of dynamic equations numerically without going through analytical derivation procedure which is unpractical for a complex system and to calculate reaction forces and moments between bodies which might be beneficial for the preliminary stage of mechanical design. Represented model of Archie (only lower body) in this paper has 10 DOF. Dynamical model of Archie with 10 DOF derived in Maple software, was not so complicated for formulation but for Maple when we request dynamic equations to be converted for Matlab use was too complex. Reach of maximum 10, characters for one expression was so far and not predicated. For better performances of Archie robot, the usage of Fuzzy Neural Controllers will be considered in our future work.

References

1. Sh. Kajita et al.: "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point", Proceedings IEEE Conference on Robotics & Automation Taipei, Taiwan, pp. 1620-1626, 14-19/09, 2003.
2. Bajrami, Xh. (2013): Dynamic modeling and simulation of a biped robot. PhD thesis. Vienna University of Technology, Austria.
3. Bajrami, Xh., Kopacek, P., Shala, A., Likaj, R., (2013): Modeling and control of a humanoid robot. Received November 20, 2012, accepted February 18, 2013, published online March 9, 2013 © Springer Verlag Wien 2013.
4. Kopacek, P. (2011): Cost oriented humanoid robot. In Proceedings of the 18th IFAC world congress, Milano, Italy, 28 August–2 September 2011.
5. Shala, A. (2004): Planning walking patterns for a BIPED robot using FNN-GA. Brussels, Belgium.
6. Guzmán, V., Blanco, O.A., Quintero, M.E., Oliver, S.A. (2012): "Development of a Biped Robot Based on Dynamic Walking." Morelos, México: Eighth International Conference on Intelligent Environments, 978-0-7695-4741-1/12 \$26.00 © 2012 IEEE