Analysis of Position and Angular Velocity of Four-Legged Robot (Mini-Bot) from Dynamic Model Using Euler-Lagrange Method

Soe Yan Naing
Department of Mechanics, Mechatronics and Robotics
South-West State University
Kursk, Russia
boyan.243@gmail.com

Thu Rain
Department of Software and Information Systems Administration
Kursk State University
Kursk,Russia
thurein.48@gmail.com

***Abstract*—This paper presents the structure of the four-legged robot (Mini-Bot), dynamical equation of the multibody system, numerical results for the generation of the positions of each leg in the Cartesian coordinate system and angular velocities of each at different time and experimental results of locations for each links calculated in the Cartesian coordinate system. In this paper we used the Euler-Lagrange method to generate the dynamical equations of the Mini-Bot. The calculation of dynamic equations is performed using Mupad toolbox from MATLAB software. The experimental results and dynamical equations will be applied to the prototype of Mini-bot.**

Keywords—Forward dynamics, Inverse dynamics, Euler-Lagrange method, Mini-bot, four-legged robot, MATLAB, Mupad toolbox

I. INTRODUCTION

Legged Robots [1, 2, and 3] are used for many purposes, including research and professional usages. Legged robots are the types of robots that can make advance movements and go around on the uneven terrain. The legged robots are easier to move, more adaptable and useful than wheeled robots, and can make fast locomotion on many different kinds of terrains. Because of these reasons, legged robots are more popular in these days and researchers in robotics field are paying attentions for this kind of robots. In this work, we reviewed the techniques of the movements of the legs of the four-legged robots. These movements intimate the movements of four legged animals. It is more difficult in handling four legs to move around. Different variables in the legs overcome the size, weight and efficiency of the problem [4, 5] and required numerical methods, kinematics modeling, dynamics modeling and sensitivity information [6].

In this paper, the dynamic model of the four-legged [7, 8, and 9] robot is generated using the Euler-Lagrange method. Euler-Lagrange method is famous for the calculations of dynamic models of rigid-body systems. The calculation of dynamic equations is performed using Mupad toolbox from MATLAB software. The dynamic analysis for multibody system is performed by using Euler-Lagrange equation [10, 11] and Newton-Euler equation [12, 13]. This paper focuses on four-legged robot design, calculation and analysis of positions and velocities from the dynamic equation using the Euler-Lagrange method.

II. SIMULATION MODEL OF MINI-BOT

The Mini-Bot consists of a rigid body and four legs. Each leg has 3 degrees of freedom (DOF) with 3 revolute joints per leg (each leg has the same structure). The 3-D model of Mini-Bot is shown Fig.1. Fig.2 shows coordinate systems of one leg of the Mini-Bot Fig.3 shows the link-length definitions and centers of masses for one leg of the Mini-Bot. The kinematic and dynamic parameters of Mini-Bot are given in Table I.

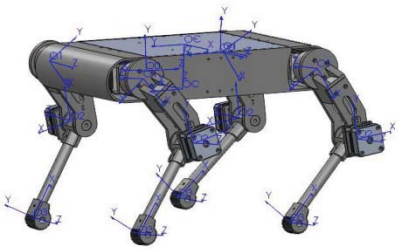

Fig.1. The 3-D model of the Mini-Bot.

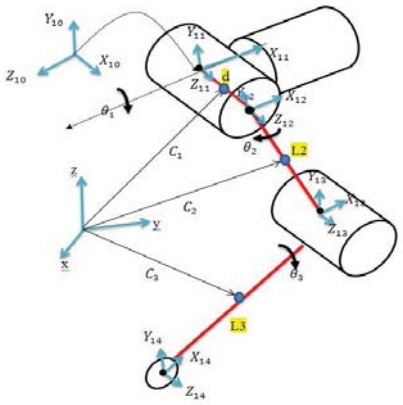

Fig.2. Coordinate systems of one leg of the Mini-Bot.

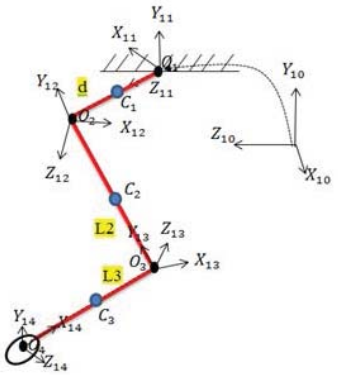

Fig.3. The link-length definitions and centers of masses for one leg of the Mini-Bot

978-1-5386-8119-0/19/$31.00 ©2019 IEEE

TABLE I. THE KINEMATIC AND DYNAMIC PARAMETERS OF MINI-BOT

Physical Model of Dimensions	The length of the body of Mini-Bot	l= 36[cm]
	The width of the body of Mini Bot	w=16.6[cm]
	The length and mass of side swing hip joint	d=7.1 [cm], m1=748 [g]
	The length and mass of hip joint	L2=12.3[cm], m2=360 [g]
	The length and mass of knee joint	L3=16.5[cm], m3=72 [g]
Variables	The angle of side swing hip joint	$\boldsymbol{\theta_1}$
	The angle of hip joint	$\boldsymbol{\theta_2}$
	The angle of knee joint	$\boldsymbol{\theta_3}$
Coordinate System	The base coordinate system for each leg	$[\mathbf{X_0}, \mathbf{Y_0}, \mathbf{Z_0}]$
	The coordinate system of side swing hip joint	$[\mathbf{X_1}, \mathbf{Y_1}, \mathbf{Z_1}]$
	The coordinate system of hip joint	$[\mathbf{X_2}, \mathbf{Y_2}, \mathbf{Z_2}]$
	The coordinate system of knee joint	$[\mathbf{X_3}, \mathbf{Y_3}, \mathbf{Z_3}]$
	The coordinate system of Endpoint of leg	$[\mathbf{X_4}, \mathbf{Y_4}, \mathbf{Z_4}]$

III. DYNAMIC MODELLING

In the dynamic modelling of robotic systems, Newton-Euler and Euler-Lagrange methods are commonly used. Newton-Euler method formulates the dynamic model based on force balance of each rigid link. Whereas Euler-Lagrange method is an energy-based method which doesn't need internal forces to formulate dynamic model [14]. In this paper, Euler-Lagrange method is used to formulate the dynamic equation of the Mini-Bot. In Euler-Lagrange method the kinetic and potential energy of a mechanical system is firstly developed and then the equations of motion are derived using a scalar function called the Lagrangian. The Lagrangian is the difference between the kinetic and potential of a mechanical system and can be expressed as

$$L(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - V(\theta), \tag{1}$$

where L is the Lagrangian, θ is the generalized coordinate of the joint, $\dot{\theta}$ is the joint velocity, T is the kinetic energy and V is the potential energy.

The translational and rotational kinetic energy of a system can be defined as:

$$T_i = \frac{1}{2} m_i \left(\dot{x}_{c_i}^2 + \dot{y}_{c_i}^2\right) + \frac{1}{2} I_{zz-cm_i}\, \dot{\theta}_i, \tag{2}$$

where $\boldsymbol{m_i, i = 1 \cdots n}$ is the mass of each link, $\boldsymbol{\dot{x}_{c_i}^2}$ is the velocity of the center of mass of the *i-th* link in horizontal coordinates system and $\boldsymbol{\dot{y}_{c_i}^2}$ is the velocity of the center of mass of the *i-th* link in vertical coordinate system, $\boldsymbol{I_{zz-cm_i}}$ is the moment of inertia about the center of mass of the *i-th* link for rotational kinetic energy and $\dot{\boldsymbol{\theta}}_\mathbf{i}$ is the angular velocity of the *i-th* joint.

The total kinetic energy of a system can be defined as:

$$T = T_1 + T_2 + \cdots + T_i. \tag{3}$$

The generalize potential energy and total potential energy is defined as:

$$V_i = m_i g y_{c_i} \tag{4}$$

$$V = V_1 + V_2 + \cdots + V_i \tag{5}$$

where $\boldsymbol{V_i}$ is the potential energy of the center of mass of the *i-th* link, g is acceleration due to gravity and $\boldsymbol{y_{c_i}}$ is the vertical position of the center of mass of the *i-th* link and V is the total potential energy.

The equations of motion of the Mini-Bot are then given by

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{\theta}}\right) - \frac{\delta L}{\delta \theta} = \tau, \tag{6}$$

where $\boldsymbol{\tau}$ is the $\boldsymbol{n \times 1}$vector of actuator torques.

Now the dynamic model of the Mini-Bot is calculated using Mupad toolbox in MATLAB. Assuming each link as a stick, the moment of inertia about the center of mass of the *i-th* link is

$$I_{zz-cm_i} = \frac{m_i {L_i}^2}{12}. \tag{7}$$

The required data of Mini-Bot for the calculation of dynamic equations are described in Table II.

TABLE II. REQUIRED DATA OF MINI-BOT

The Center of position	The velocity of the center of mass	The moments of inertia for each link
xc_1 := t-->d/2*cos($\boldsymbol{\theta_1}$(t)); yc_1:=t--> d/2*sin($\boldsymbol{\theta_1}$ (t));	vc_x_1:=t->xc_1'(t); vc_y_1:=t->yc_1'(t);	$\frac{\mathbf{m_1 d^2}}{\mathbf{12}}$
xc_2:=t-->x_1(t)+cos($\boldsymbol{\theta_2}$ (t)+ $\boldsymbol{\theta_1}$ (t))*L2/2; yc_2:=t-->y_1(t)+sin($\boldsymbol{\theta_2}$ (t)+ $\boldsymbol{\theta_1}$ (t))*L2/2;	vc_x_2:=t->xc_2'(t); vc_y_2:=t->yc_2'(t);	$\frac{\mathbf{m_2 {L_2}^2}}{\mathbf{12}}$
xc_3:=t-->x_2(t)+L3/2*cos($\boldsymbol{\theta_1}$ (t)+ $\boldsymbol{\theta_2}$(t)+ $\boldsymbol{\theta_3}$ (t)); yc_3:=t-->y_2(t)+L3/2*sin($\boldsymbol{\theta_1}$ (t)+ $\boldsymbol{\theta_2}$ (t)+ $\boldsymbol{\theta_3}$ (t));	vc_x_3:=t->xc_3'(t); vc_y_3:=t->yc_3'(t);	$\frac{\mathbf{m_3 {L_3}^2}}{\mathbf{12}}$

The equations of motion given to formulate (1), (2), (3), (4), (5) and (6), calculated partial derivative of the

Lagrangian with respect to $\{\boldsymbol{\theta_i}, \dot{\boldsymbol{\theta}}_\mathbf{i}\}$ and the torque of each joint can be defined as:

$$\tau_1 = \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\delta L}{\delta \dot{\theta}_1}\right) - \frac{\delta L}{\delta \theta_1}, \quad (8)$$

$$\tau_2 = \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\delta L}{\delta \dot{\theta}_2}\right) - \frac{\delta L}{\delta \theta_2}, \quad (9)$$

$$\tau_3 = \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\delta L}{\delta \dot{\theta}_3}\right) - \frac{\delta L}{\delta \theta_3}. \quad (10)$$

In this case, for the numerical solution assumed that there are no external forces or torques acting on the joints. Using 'solve' command angular acceleration were computed in Mupad toolbox

$$\alpha_i = \text{solve}(T_i, \theta_i(t), (\text{IgnoreSpecialCase}) \quad (11)$$

IV. Experimental Results of Simulation

Here, we present numerical results for calculations of angular positions and velocities of each joint at different time (*t*). In the process of dynamic modelling, the second order differential equations are solved by using the ODE solver's 'odesolve2' command in Mupad toolbox. This command returns the numerical solution of the function $\boldsymbol{\theta_i(t)}$ of the first order differential equation (dynamical system). The command 'numeric::ode2vectorfield' was used to generate the input parameters for the numerical ODE solver [15]. The commands used for calculation of numerical solutions are described in Table.3.

TABLE III. Commands for Calculation of Numerical Solutions

fields:=[th1(t),th1'(t),th2(t),th2'(t),th3(t),th3'(t)];
q0:=[0,0,-PI,0,0,0];
sys:={ $\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2}$, $\boldsymbol{\alpha_3}$,th1(0)=q0[1],th1'(0)=q0[2],th2(0)=q0[3], th2'(0)=q0[4], th3(0)=q0[5],th3'(0)=q0[6]};
solveth:=numeric::ode2vectorfield(sys,fields);
Rotateangle:=numeric::odesolve2(solveth);

Where 'q0' is initial condition,$\mathbf{th_i}$ represents angular position $\boldsymbol{\theta_i(t)}$ and $\mathbf{th_i'}$ represents angular velocity $\dot{\boldsymbol{\theta}}_\mathbf{i}(\mathbf{t})$. The values of the states { $\boldsymbol{\theta_i(t)}$, $\dot{\boldsymbol{\theta}}_\mathbf{i}(\mathbf{t})$ } were evaluated at the different times and the experimental results are presented in the following Table IV. $\boldsymbol{\theta_1(t)}$ and $\dot{\boldsymbol{\theta}}_\mathbf{1}(\mathbf{t})$ represent the angular position and velocity of the side swing hip joint, $\boldsymbol{\theta_2(t)}$ and $\dot{\boldsymbol{\theta}}_\mathbf{2}(\mathbf{t})$ represent the angular position and velocity of the hip joint, $\boldsymbol{\theta_3(t)}$ and $\dot{\boldsymbol{\theta}}_\mathbf{3}(\mathbf{t})$ represent the angular position and velocity of the knee joint respectively. Fig.4 shows the graphs of angular positions and velocities of each joint within the time duration from 0 to 5 seconds.

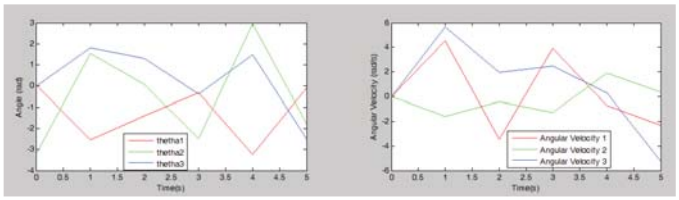

Fig. 4 .Angular positions and velocities of each joint

TABLE IV. Numerical Results of Angular Positions and Velocities of Each Joint

Time (*s*)	$\boldsymbol{\theta_1}$ (rad)	$\dot{\boldsymbol{\theta}}_\mathbf{1}$ (rad/s)	$\boldsymbol{\theta_2}$ (rad)	$\dot{\boldsymbol{\theta}}_\mathbf{2}$ (rad/s)	$\boldsymbol{\theta_3}$ (rad)	$\dot{\boldsymbol{\theta}}_\mathbf{3}$ (rad/s)
0	0.0	0.0	–3.14 15	0.0	0.0	0.0
1	–2.55 786	4.5512 3	1.5445	–1.61 413	1.8166 2	5.6822 8
2	–1.41 2	–3.43 45 1	0.0622 7	–0.41 39	1.2894 7	1.9779 82
3	–0.30 6	3.9318 2	–2.47 410	–1.29 889	-0.369 1	2.4960 2
4	–3.24 6	–0.72 209	2.9501 8	1.8800 8	1.4697 9	0.3280 7
5	–0.06 7	–2.30 303	–1.79 77	0.3631 8	-2.523 95	–5.25 564

The link-ends of the links d, L2 and L3 are described by $\mathbf{O_2, O_3}$ **and** $\mathbf{O_4}$ (see Fig.3). The positions of $\mathbf{O_2, O_3}$ **and** $\mathbf{O_4}$ are described in the Cartesian coordinate system. The position of $\mathbf{O_2}$ represents position of the side swing hip link, the position of $\mathbf{O_3}$ represents position of the hip link and the position of $\mathbf{O_4}$ represents position of the knee link respectively. Fig.5, Fig.6 and Fig.7 show the results of the positions of $\mathbf{O_2, O_3}$ **and** $\mathbf{O_4}$ within the time duration from 0 to 5 seconds.

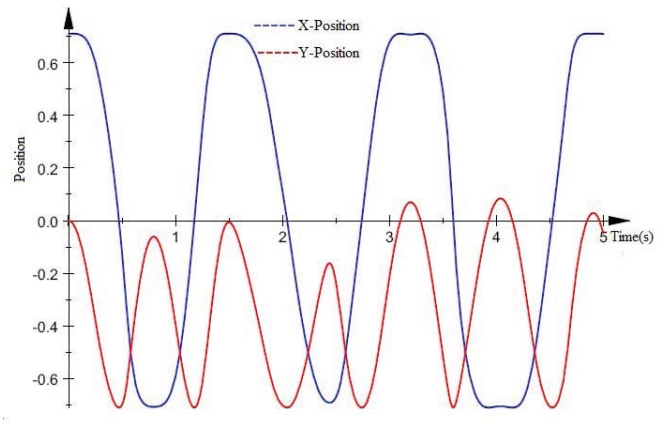

Fig.5. Positions of $\mathbf{O_2}$

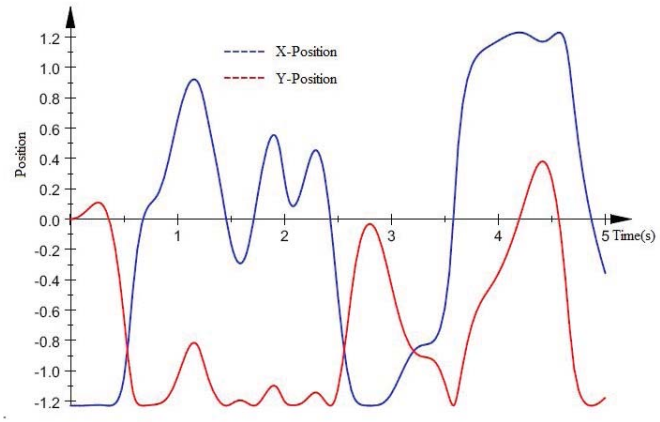

Fig.6. Positions of O_3

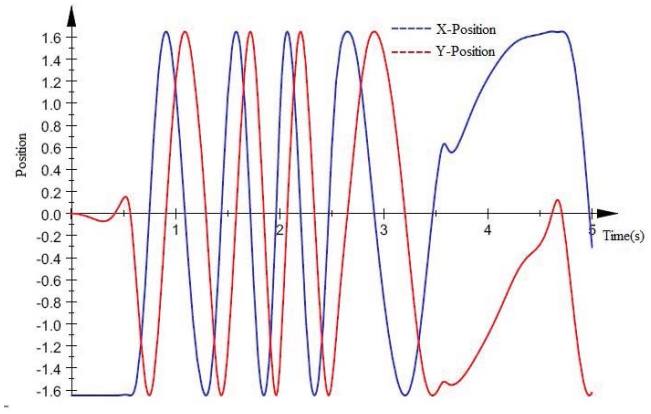

Fig.7. Positions of O_4

V. Conclusions and Future Work

In this study, the dynamic model of the Mini-Bot robot was formulated by using Euler-Lagrange method and calculated by using Mupad toolbox in MATLAB. This paper presented a dynamic model of multibody computations for one leg and evaluated the angular positions and velocities at different time. The dynamic model can be generated the required driving torque of each leg and can be implemented in four legged robot control algorithm design. In future work, we will develop the hardware prototype design of Mini-Bot (Fig.8) using simulation results and dynamic equations. Controlling motors will be performed by using raspberry PI 3B+ and L298N motor driver.

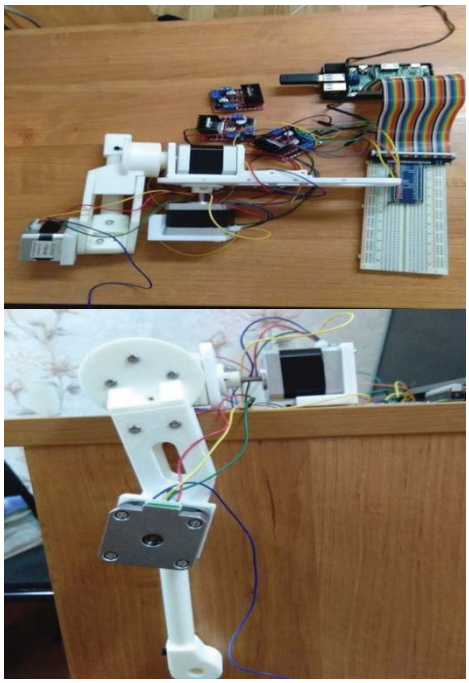

Fig.8.Hardware Prototype of a leg

REFERENCES

[1] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-Free Humanoid State Estimation fusing Kinematic, Inertial and LIDAR sensing," 14th IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids), 2014.

[2] S. Han, S. Um, and S. Kim," Mechanical Design of Robot Lower Body based on Four-Bar Linkage Structure for Energy Efficient Bipedal Walking," IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR)EPFL, 2016.

[3] Y. Huang and Q. Wang, "Torque-Stiffness-Controlled Dynamic Walking Analysis of the Behaviors of Bipeds with Both Adaptable Joint Torque and Joint Stiffness," IEEE Robotics and automation magazine, 2016.

[4] M. M. Gor, P. M. Pathak, and J. M. Yang, "Dynamic Modeling and Simulation of Complaint Legged Quadruped Robot," Proc. of the 1st Int. and 16th National Conf. on Machines and Mechanisms, 2013.

[5] B. H. Kim, "Analysis of Balance of Quadrupedal Robotic Walk using Measure of Balance Margin," Int. Journal of Fuzzy Logic and Intelligent Systems, vol. 13, no. 2, pp. 100–105, 2013.

[6] R. Höpler, M. Stelzer, and O. von Stryk, "Object-oriented Dynamics Modeling for legged Robot Trajectory Optimization and Control," Proc. IEEE Int. Conf. On Mechatronics and Robotics, pp. 972–977, 2004.

[7] J. Yang, W. Jia, Y. Sun, H. Pu, and S. Ma, "Mechanical Design of a Compact and Dexterous Quadruped Robot," Proc. of 2017 IEEE Int. Conf. on Mechatronics and Automation, 2017.

[8] C. Semini, V. Barasuol, J. Goldsmith, and M. Frigerio, "Design of the Hydraulically Actuated, Torque-Controlled Quadruped Robot HyQ2Max," IEEE/ASME Transactions on Mechatronics, 2016. DOI 10.1109/TMECH.2016.2616284

[9] F. I. Sheikh and S. S. Haq, "Dynamic Maneuverability Through Voluntary Morphosis in a Four-Legged Robot," 6th IEEE Conf. on Robotics, Automation and Mechatronics, 2013.

[10] S. F. Yatsun, S. I. Savin, O. V. Emelyanova, and A. S. Yatsun, "Analysis of structures, principles of creation, basics of modeling, monograph," pp. 44–57, 2015.

[11] R. N. Jazar, Theory of Applied Robotics Kinematics, Dynamics and Control, 2010. DOI 10.1007/978-1-4419-1750-8

[12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Dynamics and Control, 2004.

[13] M. D. Ardema, Newton-Euler Dynamics. Springer Science-Business Media, 2005.

[14] J. Zhang, J. Zhang, and C. Wang, "Dynamic Analysis and Simulation on Bionics Quadruped Robot," The Open Automation and Control Systems Journal, 7, pp. 1088–1092, 2015.

[15] Site Mathworks. [Online]. Available: https://www.mathworks.com /help/ symbolic/mupad_ref/numeric-odesolve2.html.