The 2012 IEEE/ASME International Conference on
Advanced Intelligent Mechatronics
July 11-14, 2012, Kaohsiung, Taiwan

LQR Motion Control of a Ball-Riding Robot

Ching-Chih Tsai, *Senior Member, IEEE*, Cheng-Kai Chan, Lung-Chun Kuo

***ABSTRACT*—The paper presents techniques and design methodologies for modeling and LQR motion control of a ball-riding robot driven by three omnidirectional wheels. A completely dynamic model of the robot moving on a flat terrain is derived using Lagrangian mechanics. Two LQR controllers are synthesized to achieve station keeping and point stabilization. Through computer simulations and experimental results, both proposed controllers together with the built ball-riding robot are successfully shown to give a satisfactory control performance.**

Keywords*- *Ball-riding robot, Lagrangian mechanics, modeling, point stabilization.

I. INTRODUCTION

Recently, many kinds of ball robots have been constructed and shown to achieve personal transportation or mobile robot platforms in human environments [1-7]. Such ball robots in [1-7] are classified into two types: ballbot and ball-riding, which are particularly used for personal transportation or material handling movement. The studies in [1-4] proposed the ballbot with a two-motor or four-motor inverse mouse-ball driving mechanism, while the ball riding robot devised by the authors in [5-7] were constructed using three omnidirectional wheels driven by three independent direct-current servomotors. Ballbots and ball-riding robots have been designed as mobile platforms for indoor autonomous service robots. Moreover, those robot developed in [1-7] have high center of gravity, narrow bases of support, agile mobility toward any poses, and dynamical stability, in order to achieve efficient navigation around the human living space, and establish friend and convenient man-machine interactions.

In comparison with the ballbots, ball-riding robots have attracted much attention in recent years [5-7]. Kumagai et al. [5-6] presented a double PD-like control method for the type of ball robot driven by three omnidirectional wheels. However, the stability of the proposed double PD-like control in [5-6] was not analyzed as well because a complete dynamic model of the ball robot has not been established as well. On the other hand, the modeling and control of the ball robot developed in [6] have not been seen in the literature although the authors released the experimental videos to show how the robot worked successfully. Although researchers have paid considerable effort on design of the ball-riding robots, the modeling and stable control problems have not been completely studied yet [5-6]. Moreover, the trajectory tracking problem of the robot-riding remains unsolved.

Linear quadratic regulator (LQR) method has been invented in 1960 by Rudolf Kalman, a Hungarian-American mathematical system theorist. This control method is one branch of linear optimal control for model-based systems. LQR has been adopted by [2] to achieve motion control of a kind of ball robot driven by two or four-motor inverse-mouse drive. However, LQR has been not used to accomplish station keeping and point stabilization and trajectory tracking of the ball-riding robot yet.

The paper is devoted to the modeling and LQR control of the ball-riding robot with a driving mechanism using three omnidirectional wheels. The paper is written in two principal contributions. One is the use of Lagrangian mechanics to derive a completely dynamic model of the robot by incorporating with kinematics of an inverse Atlas spherical motion platform. The other is the design, simulation and experimentation of the LQR controller, thus achieving station keeping and point stabilization for the robot with parameter variations.

The rest of the paper is organized as follows. Section II derives the completely dynamic model of the robot, while Section III is devoted to developing two LQR controllers for the robot. Simulation results are presented in Section IV, in order to explore the feasibility of the proposed LQR controllers. The applicability of the proposed controller together with the built ball-riding robot is exemplified in Section V. Section VI concludes the paper.

II. SYSTEM MODELING

The section aims to describe the system structure and physical configuration of the designed ball robot, and then derive its dynamic model using Lagrangian mechanics. Note that the modeling process is based on a simplification assumption that the ball robot is constructed by two major components: the body and the ball.

2.1 Kinematic and Dynamic Models of the Inverse Atlas Spherical Motion Platform

To steer the designed ball robot, it is necessary to develop its mathematical model such that a stable controller can be designed to achieve any desired control goals. As mentioned in [5-6], the ball robot is realized using two mechanisms: one is a two-dimensional inverted pendulum and the other is an omnidirectional mechanism using an inverse Atlas spherical motion platform for driving the ball as mentioned above. For the control purposes, two decoupled, independent dynamic equations for the two-dimensional mobile inverted pendulum are required in order to achieve self-balancing. Additionally, two dynamic equations for the translation of the ball robot are also needed

Ching-Chih Tsai ,Cheng-Kai Chan and Lung-Chun Kuo are with Department of Electrical Engineering, National Chung Hsing University,250, Kuo-Kuang Road, Taichung 40227, Taiwan, R.O.C. Ching-Chih Tsai is the corresponding authors is, whose email is cctsai@nchu.edu.tw. The authors deeply acknowledge the financial support from National Science Council, Taiwan under the grant number NSC 97-2628-E-005-004- MY3

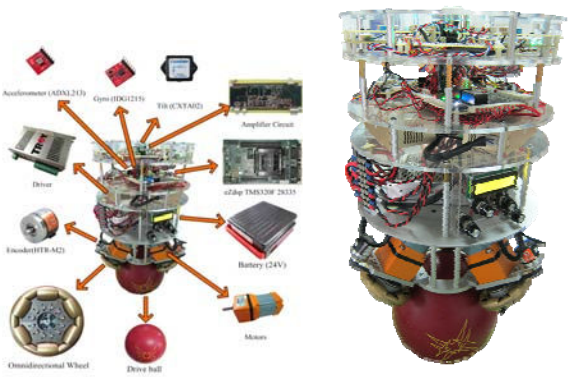
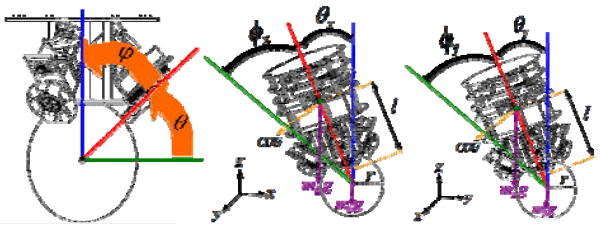


Fig.1 The experimental ball-riding robot; (a)physical diagram; (b)laboratory-built prototype.



(a) (b) (c)

Fig. 2 Illustration of (a)the elevation angle θ, the zenith angle φ; (b) the tilt angle θ_x and the motors' angular positions ϕ_x in the median sagittal plane; (c) the tilt angle θ_y and the motors' angular positions ϕ_y in the median coronal plane.

Table.1 List of the robot parameters and their actual values

Symbol	Parameter description	Actual value
m_B	mass of the body	25kg
m_b	mass of the ball	4.95kg
l	distance between the center of the ball and the center of mass of the body	0.225m
r_B	radius of the body	0.15m
r_b	radius of the ball	0.105m
$I_B=m_B(r_B^2/4+l^2/3)$	moment of inertia of the Body with respect to the center of the body	0.7875 $[kg\cdot m^2]$
$I_b=(2/5)m_b r^2$	moment of inertia of the ball with respect to the center of mass	0.0218 $[kg\cdot m^2]$

in order to accomplish out point stabilization and trajectory tracking. The basic concept there is to obtain a relationship between the angular velocity vectors of the center of mass of the sphere $\vec{\Omega}$ and the angular speeds of the three omnidirectional wheels $\vec{\omega}_i$, which would account for zero kinematic slip between the sphere and the omnidirectional wheels. The condition is met by requiring that the projection of the velocities of the sphere at all contact points in the actuation direction of the omnidirectional wheel be the same or expressed mathematically.

$$(\vec{\Omega}\times\vec{R})\cdot\vec{v}_i = V_i, i=1,2,3, \vec{v}_1=\vec{i}, \vec{v}_2=\vec{j}, \vec{v}_3=\vec{k} \qquad (1)$$

where $\vec{R}$ is the position vector of contact point, $\vec{v}_i$ is the vector of direction cosines of the omnidirectional wheel contact point velocities in the actuation directions, and V_i is the speed of omnidirectional wheel i at the contact point. Furthermore, the relationship between $\vec{\Omega}$ and $\vec{\omega}$ was discussed and given in [8]; the relationship is expressed by

$$\vec{\Omega} = \mathbf{J}\vec{\omega} \qquad (2)$$

where $\vec{\omega}=[\omega_1 \quad \omega_2 \quad \omega_3]^T$ and $\mathbf{J}$ is the Jacobian matrix of the architecture, which project the angular velocity of each wheel to the angular velocity of the sphere. Moreover, below are both Jacobian matrix and the inverse Jacobian matrix of the Atlas spherical motion platform with the elevation angle θ [8], as Fig.2 shows.

$$J^{-1}=\frac{r_b}{r}\begin{bmatrix}-\sin\theta & 0 & \cos\theta\\ \frac{1}{2}\sin\theta & -\frac{\sqrt{3}}{2}\sin\theta & \cos\theta\\ \frac{1}{2}\sin\theta & \frac{\sqrt{3}}{2}\sin\theta & \cos\theta\end{bmatrix}, J=\frac{2r}{3r_b}\begin{bmatrix}-\frac{1}{\sin\theta} & \frac{1}{2\sin\theta} & \frac{1}{2\sin\theta}\\ 0 & -\frac{\sqrt{3}}{2\sin\theta} & \frac{\sqrt{3}}{2\sin\theta}\\ \frac{1}{2\cos\theta} & \frac{1}{2\cos\theta} & \frac{1}{2\cos\theta}\end{bmatrix} \qquad (3)$$

where r_b and r respective denote the radius of the driving ball and the wheel. Since the sum of the elevation angle θ and the zenith angle φ is 90°, i.e., $\varphi+\theta=90^\circ$, as shown in Fig. 2(a), one obtains $\sin\theta=\cos\varphi$, $\cos\theta=\sin\varphi$ .Hence the Jacobian matrix is now rewritten by

$$J^{-1}=\frac{r_b}{r}\begin{bmatrix}-\cos\varphi & 0 & \sin\varphi\\ \frac{1}{2}\cos\varphi & -\frac{\sqrt{3}}{2}\cos\varphi & \sin\varphi\\ \frac{1}{2}\cos\varphi & \frac{\sqrt{3}}{2}\cos\varphi & \sin\varphi\end{bmatrix}, J=\frac{2r}{3r_b}\begin{bmatrix}-\frac{1}{\cos\varphi} & \frac{1}{2\cos\varphi} & \frac{1}{2\cos\varphi}\\ 0 & -\frac{\sqrt{3}}{2\cos\varphi} & \frac{\sqrt{3}}{2\cos\varphi}\\ \frac{1}{2\sin\varphi} & \frac{1}{2\sin\varphi} & \frac{1}{2\sin\varphi}\end{bmatrix} \qquad (4)$$

Moreover, the angular velocity vector of the sphere $\vec{\Omega}$ is computed by the product of the Jacobian matrix (4) and the angular speeds of the three omnidirectional wheels $\vec{\omega}_i$ using equation (5)

$$\vec{\Omega}=\begin{bmatrix}\vec{\Omega}_1\\ \vec{\Omega}_2\\ \vec{\Omega}_3\end{bmatrix}=J\vec{\omega}=\frac{2r}{3r_b}\begin{bmatrix}-\frac{1}{\cos\varphi} & \frac{1}{2\cos\varphi} & \frac{1}{2\cos\varphi}\\ 0 & -\frac{\sqrt{3}}{2\cos\varphi} & \frac{\sqrt{3}}{2\cos\varphi}\\ \frac{1}{2\sin\varphi} & \frac{1}{2\sin\varphi} & \frac{1}{2\sin\varphi}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}=\frac{2r}{3r_b}\begin{bmatrix}\frac{-2\omega_1+\omega_2+\omega_3}{2\cos\phi}\\ \frac{\sqrt{3}}{2\cos\phi}(-\omega_2+\omega_3)\\ \frac{1}{2\sin\phi}(\omega_1+\omega_2+\omega_3)\end{bmatrix} \qquad (5)$$

Let the contact point $\vec{R}_s$ between the sphere and the ground have the position vector defined by $\vec{R}_s=[0 \quad 0 \quad -r_b]$. Therefore, $\vec{\mathbf{V}}$ the velocity of the contact point between the driving ball and the ground, is computed by .

$$\vec{\mathbf{V}}=\vec{\Omega}\times\vec{R}_s=\begin{bmatrix}\vec{i} & \vec{j} & \vec{k}\\ \Omega_1 & \Omega_2 & \Omega_3\\ 0 & 0 & -r_b\end{bmatrix}=-r_b\Omega_2\vec{i}+r_b\Omega_1\vec{j}+0\vec{k}=V_x\vec{i}+V_y\vec{j}+V_z\vec{k} \qquad (6)$$

Since the vector $\vec{\mathbf{V}}$ is expressed by $\vec{\mathbf{V}}=V_x\vec{i}+V_y\vec{j}+V_z\vec{k}$, where the three components, V_x, V_y and V_z, in the Cartesian coordinates are found from (6) and given as below.

$$\begin{aligned}V_x&=(\vec{\Omega}\times\vec{R})\cdot\vec{i}=-r_b\Omega_2=-r_b\frac{2r}{3r_b}\frac{\sqrt{3}}{2\cos\varphi}(-\omega_2+\omega_3)=\frac{r}{\sqrt{3}\cos\varphi}(\omega_2-\omega_3)\\ V_y&=(\vec{\Omega}\times\vec{R})\cdot\vec{j}=r_b\Omega_1=r_b\frac{2r}{3r_b}\frac{(-2\omega_1+\omega_2+\omega_3)}{2\cos\varphi}=\frac{r(-2\omega_1+\omega_2+\omega_3)}{3\cos\varphi}\\ V_z&=0\end{aligned} \qquad (7)$$

On the other hand, the optimal linear velocities: v_{s1} , v_{s2} ,and v_{s3} of the three wheels have been proposed by [3] using the virtual linear wheel speeds, v_x, v_y, integrated from the acceleration control commands in [3]. The following are the relations between the three optimal linear velocities, v_{s1} , v_{s2} and v_{s3} , and the virtual linear wheel speeds, v_x and v_y

$$\begin{bmatrix} v_{s1} \\ v_{s2} \\ v_{s3} \end{bmatrix} = \begin{bmatrix} -v_y \cos\varphi + K_z\omega_z \\ (\frac{\sqrt{3}}{2}v_x + \frac{1}{2}v_y)\cos\varphi + K_z\omega_z \\ (-\frac{\sqrt{3}}{2}v_x + \frac{1}{2}v_y)\cos\varphi + K_z\omega_z \end{bmatrix} \tag{8}$$

Note that $K_z = r_b \sin\varphi$, $\dot{v}_x = a_x = u_x$ and $\dot{v}_y = a_y = u_y$, where a_x and a_y are respectively acceleration commands in both x and y axes; u_x and u_y are the acceleration control commands in both x and y axes. Moreover, because $\omega_i = v_{si}/r$ where r is the radius of the wheel, the relationship between the angular velocities of three real $\bar{\omega}_i$ and the velocities of the three real wheels $\bar{V}_s$ are obtained from (8) and expressed by

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r}\begin{bmatrix} -v_y \cos\varphi + K_z\omega_z \\ (\frac{\sqrt{3}}{2}v_x + \frac{1}{2}v_y)\cos\varphi + K_z\omega_z \\ (-\frac{\sqrt{3}}{2}v_x + \frac{1}{2}v_y)\cos\varphi + K_z\omega_z \end{bmatrix} \tag{9}$$

Thus the angular velocity vector $\vec{\Omega}$ of the spherical ball is obtained from (5) and given by

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \frac{2r}{3r_b}\begin{bmatrix} \frac{-2\omega_1 + \omega_2 + \omega_3}{2\cos\varphi} \\ \frac{\sqrt{3}}{2\cos\varphi}(-\omega_2 + \omega_3) \\ \frac{1}{2\sin\varphi}(\omega_1 + \omega_2 + \omega_3) \end{bmatrix} = \begin{bmatrix} v_y / r_b \\ -v_x / r_b \\ \omega_z \end{bmatrix} \tag{10}$$

From (7), it easily obtains the subsequent three relations of the three velocity components

$$\begin{aligned} V_x &= -r_b\Omega_2 = v_x \Rightarrow \dot{x} = V_x = v_x \\ V_y &= r_b\Omega_1 = v_y \Rightarrow \dot{y} = V_y = v_y \\ V_z &= 0 \end{aligned} \tag{11}$$

Taking the time derivative of equation (11), one easily obtains the dynamic equations of the inverse Atlas spherical motion platform in the following.

$$\ddot{x} = \dot{v}_x = a_x = u_x, \ddot{y} = \dot{v}_y = a_y = u_y \tag{12}$$

2.2 Dynamic Modeling of the Two-Dimensional Mobile Inverted Pendulum

The section aims to derive a simplified planner dynamic model of the two-dimensional mobile inverted pendulum using Lagrangian mechanics. Similar to the modeling process investigated by Nagarajan et al. [3-4], our modeling approach makes the following three assumptions: (*i*) there is no slip between the spherical wheel and the floor, (*ii*) the motion in the median Sagittal plane and median coronal plane is decoupled, and (*iii*) the equations of motion in these two planes are identical. With these assumptions, one obtains two decoupled, independent planar dynamic models for the robot, thereby easily designing two independent motion controllers. In the sequel, as per the assumptions, the equations of motion in the median sagittal plane (x-z) are derived and then the vehicle dynamics in the median coronal plane (y-z) is assumed to have identical dynamics with different symbols.

2.2.1 Vehicle Dynamics in the Median Sagittal plane

Notice that the planar modeling process is based on a simplification assumption that the ballbot in each plane is constructed by two major components: the body and the ball. To describe the dynamics of the robot, one needs to obtain two expressions of the kinetic and potential energy of the body and the ball. For the ball one obtains

$$K_{ball} = \frac{1}{2}m_b(\dot{x}^2) + \frac{1}{2}I_b w_b^2, \quad V_{ball} = 0 \tag{13}$$

where I_b , m_b and r_b are, respectively the moment of inertia, mass, and radius of the ball. Assume no slip between the ball and the floor. For sake of simplicity, the motion control can be design into two separate and identical planar control systems [4-5]. Figs. 2(b) and 2(c) depict the simplified diagram of the simplified ballbot. For start to designing the one of the decoupling systems, the kinetic energy and potential energy of the ball in the single axis can be rewritten by

$$K_{ball} = \frac{1}{2}m_b(r_b(\dot{\phi}_x + \dot{\theta}_x))^2 + \frac{1}{2}I_b(\dot{\phi}_x + \dot{\theta}_x)^2, V_{ball} = 0 \tag{14}$$

For the body it follows that

$$\begin{aligned} K_{body} &= \frac{1}{2}m_B(r_b^2(\dot{\phi}_x + \dot{\theta}_x)^2 + 2r_b l\dot{\theta}_x(\dot{\phi}_x + \dot{\theta}_x))\cos(\theta_x) + l^2\dot{\theta}_x^2) + \frac{1}{2}I_B(\dot{\theta}_x)^2 \\ V_{body} &= m_c g l\cos(\theta_x) \end{aligned} \tag{15}$$

where I_B and m_B are the moments of inertia and mass. Therefore, the total kinetic energy of the robot is

$$K = K_{ball} + K_{body} \tag{16}$$

and the total potential energy of the robot is

$$V = \underbrace{0}_{V_{ball}} + \underbrace{m_B g l\cos(\theta_x)}_{V_{body}} \tag{17}$$

Moreover, define the generalized coordinates and the friction vector as

$$\mathbf{q} = \begin{bmatrix} \theta_x & \phi_x \end{bmatrix}^T \tag{18}$$

$$D_x(\dot{\mathbf{q}}) = \begin{bmatrix} 0 & \mu_{\phi x}\dot{\phi}_x + \mu_{c\phi x}\,\mathrm{sgn}(\dot{\phi}_x) \end{bmatrix}^T \tag{19}$$

where μ_ϕ represents the viscous coefficients and $\mu_{c\phi}$ denotes the static friction coefficient. Since the Lagrangian function L is given by

$$L(q, \dot{q}) = K - V \tag{20}$$

The Euler-Lagrange equations of motion for the ballbot model are governed by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} = \begin{bmatrix} 0 & \tau_x \end{bmatrix}^T - \mathrm{D}(\dot{\mathbf{q}}) \tag{21}$$

After manipulations, the Euler-Lagrange equations are rewritten by the following matrix form

$$\mathrm{M}_x(\mathbf{q}_\mathbf{x})\ddot{\mathbf{q}}_\partial + \mathrm{C}_x(\mathbf{q}_\mathbf{x}, \dot{\mathbf{q}}_x)\dot{\mathrm{q}}\dot{\mathrm{q}} + \mathrm{G}_x(\mathbf{q}_\mathbf{x}) + \mathrm{D}_x(\dot{\mathbf{q}}_x) = \begin{bmatrix} 0 & \tau_x \end{bmatrix}^T \tag{22}$$

where

$$\mathrm{M_x}(\mathbf{q_x}) = \begin{bmatrix} \alpha+\gamma+2\beta\cos(\theta_x) & \alpha+\beta\cos(\theta_x) \\ \alpha+\beta\cos(\theta_x) & \alpha \end{bmatrix} \quad (23)$$

$$\mathrm{C_x}(\mathrm{q_x},\dot{\mathrm{q}}_x)\dot{\mathrm{q}}_x = \begin{bmatrix} -\beta\dot{\theta}_x^{\,2}\sin(\theta_x) & -\beta\dot{\theta}_x^{\,2}\sin(\theta_x) \end{bmatrix}^T \quad (24)$$

$$\mathrm{G_x}(\mathrm{q}_x) = \begin{bmatrix} -\dfrac{\beta g}{r_b}\sin(\theta_x) & 0 \end{bmatrix}^T \quad (25)$$

where $\alpha = I_b + (m_B + m_b)r_b^{\,2}$; $\beta = m_B r_b l$; $\gamma = m_B l^2 + I_B$.

Equation (26) represents the single-input-multiple-output dynamic equation of the robot with the applied torque τ_x. However, since $\dot{\phi}_x$ and $\ddot{\phi}_x$ can be calculated from the kinematic equations of motion of the ball, i.e., $\dot{\phi}_x = \dfrac{r}{\sqrt{3}r_b\cos\varphi}(\omega_2 - \omega_3) = \dfrac{v_x}{r_b}$ and $\ddot{\phi}_x = \dfrac{r}{r_b\sqrt{3}\cos\varphi}(\dot{\omega}_2 - \dot{\omega}_3)$ where $a_x = dv_x/dt$, then we have the following second-order equation of motion of θ_x from (22)

$$M_{x11}\ddot{\theta}_x - \beta\dot{\theta}_x^{\,2}\sin(\theta_x) - \frac{\beta g}{r_b}\sin\theta_x = -M_{x12}\ddot{\phi}_x = -\frac{M_{x12}}{r_b}u_x \quad (26)$$

where $M_{x11} = \alpha+\gamma+2\beta\cos(\theta_x)$, $M_{x12} = \alpha+\beta\cos(\theta_x)$ and $u_x = a_x$. Equation (26) represents the dynamic equation of the inverted pendulum in the Sagittal plane or in the x axis.

2.2.2 Vehicle Dynamics in the Median Coronal plane

Similar to Section 2.2.1, the Euler-Lagrange equations of the robot in the median coronal plane or in the y axis can be derived and expressed by the following matrix form

$$\mathrm{M_y}(\mathbf{q}_y)\ddot{\mathbf{q}}_y + \mathrm{C_y}(\mathbf{q}_y,\dot{\mathbf{q}}_y)\dot{\mathrm{q}} + \mathrm{G_y}(\mathbf{q}_y) + \mathrm{D_y}(\dot{\mathbf{q}}_x) = \begin{bmatrix} 0 & \tau_y \end{bmatrix}^T \quad (27)$$

where

$$\mathrm{M_y}(\mathbf{q}_y) = \begin{bmatrix} \alpha+\gamma+2\beta\cos(\theta_y) & \alpha+\beta\cos(\theta_y) \\ \alpha+\beta\cos(\theta_y) & \alpha \end{bmatrix} \quad (28)$$

$$\mathrm{C_y}(\mathbf{q}_y,\dot{\mathbf{q}}_y)\dot{\mathbf{q}}_y = \begin{bmatrix} -\beta\dot{\theta}_y^{\,2}\sin(\theta_y) & -\beta\dot{\theta}_y^{\,2}\sin(\theta_y) \end{bmatrix}^T \quad (29)$$

$$\mathrm{G_y}(\mathbf{q}_y) = \begin{bmatrix} -\dfrac{\beta g}{r_b}\sin(\theta_y) & 0 \end{bmatrix}^T, D_y(\dot{\mathbf{q}}_y) = \begin{pmatrix} 0 \\ \mu_{\phi y}\dot{\phi}_y + \mu_{c\phi y}\,\mathrm{sgn}(\dot{\phi}_y) \end{pmatrix} \quad (30)$$

Equation (27) represents the single-input-multiple-output dynamic equation of the robot with the applied torque τ_y. However, since $\dot{\phi}_y$ and $\ddot{\phi}_y$ can be calculated from the kinematic equations of motion of the ball, i.e.,

$$\dot{\phi}_y = \frac{r(-2\omega_1+\omega_2+\omega_3)}{3r_b\cos\varphi} = \frac{v_y}{r_b} \text{ and } \ddot{\phi}_y = \frac{r(-2\dot{\omega}_1+\dot{\omega}_2+\dot{\omega}_3)}{3r_b\cos\varphi} = \frac{a_y}{r_b}$$

where $a_y = dv_y/dt$, then we have the following second-order equation of motion of θ_x from (27)

$$M_{y11}\ddot{\theta}_y - \beta\dot{\theta}_y^{\,2}\sin(\theta_y) - \beta g/r_b\sin\theta_y = -M_{y12}\ddot{\phi}_y = -M_{y12}/r_b\,u_y \quad (31)$$

where $M_{y11} = \alpha+\gamma+2\beta\cos(\theta_y)$, $M_{y12} = \alpha+\beta\cos(\theta_y)$, and $u_y = a_y$. Equation (31) represents the dynamic equation of the inverted pendulum in the median coronal plane.

Before closing this section, it is worthwhile to note that the dynamic equations of motion of the robot in the two planes are identical except the notation. Hence, only one kind of controller is required during the controller design process. Once the controller has been synthesized in one plane, the control method can be straightforward applied to control the robot in the other plane. On the other hand, the two tilt angles θ_x and θ_y have been known, the actual tilt angle θ can be obtained from the equality $l\cos\theta = l\cos\theta_y\cos\theta_x$. Thus, $\theta = \cos^{-1}(\cos\theta_y\cos\theta_x)$. If both tilt angles θ_x and θ_y are controlled to approach zero, then the nutation angle θ must tend to zero simultaneously.

2.2.3 Linearized Models

2.2.3.1 Linearized Model in the Median Sagittal Plane

From (26), one linearized the dynamic equation of the inverted pendulum in the X axis about $\theta_x \cong 0$ and obtains the following second-order linearized model

$$M_{x11}\ddot{\theta}_x - \frac{\beta g}{r_b}\theta_x = -M_{x12}\ddot{\phi}_x = -\frac{M_{x12}}{r_b}u_x \Rightarrow \ddot{\theta}_x = \frac{\beta g}{M_{x11}r_b}\theta_x - \frac{M_{x12}}{M_{x11}r_b}u_x \quad (32)$$

where $M_{x11} = \alpha+\gamma+2\beta$, $M_{x12} = \alpha+\beta$, $\ddot{x} = a_x = u_x$. Combing the dynamics of the translation of the robot in the x axis yields

$$\ddot{\theta}_x = \frac{\beta g}{M_{x11}r_b}\theta_x - \frac{M_{x12}}{M_{x11}r_b}u_x, \ddot{x} = u_x \quad (33)$$

2.2.3.2 Linearized Model in the Median Coronal Plane

Similarly, from (31), one also linearized the dynamic equation of the inverted pendulum in the Y axis about $\theta_y \cong 0$, thereby giving the subsequent linearized second-order model

$$M_{y11}\ddot{\theta}_y - \frac{\beta g}{r_b}\theta_y = -M_{y12}\ddot{\phi}_y = -\frac{M_{y12}}{r_b}u_y \Rightarrow \ddot{\theta}_y = \frac{\beta g}{M_{y11}r_b}\theta_y - \frac{M_{y12}}{M_{y11}r_b}u_y \quad (34)$$

where $M_{y11} = M_{x11} = \alpha+\gamma+2\beta$, $M_{y12} = M_{x12} = \alpha+\beta$, $\ddot{y} = a_y = u_y$. Similarly, by integrating the dynamics of the translation of the robot in the y axis, one obtains

$$\ddot{\theta}_y = \frac{\beta g}{M_{y11}r_b}\theta_y - \frac{M_{y12}}{M_{y11}r_b}u_y, \ddot{y} = u_y \quad (35)$$

III. STATION KEEPING AND POINT STABILIZATION USING LQR

This section is devoted to design two state feedback controllers for both self-balancing and position control of the robot in the median sagittal and coronal planes, respectively. The two aforementioned control goals (station keeping and position control) can be reduced to the same regulation problem; namely that two tilt angle θ_x and θ_y are maintained at 0 (θ_x=0, θ_y=0), and the position errors $e_x = x - x_o$ and $e_y = y - y_o$ must be maintained at the origin, respectively, where the pair (x_o, y_o) is the desired position. Note that both control goals regarding two variables, tilt angle θ and the position error e must be achieved. The two state feedback laws for u_x and u_y are given by

$$\begin{aligned} u_x = a_x &= K_A\theta_x + K_{AV}\dot{\theta}_x + K_T\,e_x + K_V\dot{e}_x \\ u_y = a_y &= K_A\theta_y + K_{AV}\dot{\theta}_y + K_T\,e_y + K_V\dot{e}_y \end{aligned} \quad (36)$$

The four control parameters, K_A, K_{AV}, K_T and K_V, in (36) are found by the LQR approach as below.

To find the four controller's parameters by using the linear quadratic regulator approach, one has to rewrite the linearized system model (33) with the four state variables,

$x_1 = \theta_x, x_2 = \dot{\theta}_x, x_3 = e_x, x_4 = \dot{e}_x$,and the state vector $\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T$, in the following form.

$$\dot{\mathbf{X}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \mathbf{X} + \underbrace{\begin{bmatrix} 0 \\ -b \\ 0 \\ 1 \end{bmatrix}}_{B} u_x = \mathbf{A}\,\mathbf{X} + \mathbf{B}\,u_x \tag{37}$$

where $a = \dfrac{\beta g}{M_{x11} r_b}$ and $b = \dfrac{M_{x12}}{M_{x11} r_b}$. By checking the rank of the controllability matrix $\mathbf{C}_x = [\mathbf{B}\ \mathbf{AB}\ \mathbf{A^2B}\ \mathbf{A^3B}]$, one obtains that the rank of $\mathbf{C}_x$ is four, thereby implying that the linearized system (37) is completely controllable. This is because the determinant of the matrix $\mathbf{C}_x$ is a^2b^2 and is nonsingular. Since the system model (37) is completely controllable, the all the controller's parameters, K_A, K_{AV}, K_T and K_V, in (36) can be easily found via some well-known pole placement method, i.e.,

$$u_x(t) = [K_A \quad K_{AV} \quad K_T \quad K_V]\mathbf{X} = -\mathbf{K}_x\mathbf{X} \tag{38}$$

where the matrix K_x represents the state feedback gain matrix. In what follows, particular attention is paid to find a linear state feedback controller that stabilizes the overall system variables about the origin, and minimizes the cost function

$$J = \int_0^\infty \left(X(t)^T QX(t) + Ru_x(t)^2 \right) dt \tag{39}$$

where $\mathbf{Q}$ and R are respectively semi-positive definite and positive-definite matrices; for example, they can be of the following forms

$$\mathbf{Q} = \begin{bmatrix} 5 & 5 & 0 & 0 \\ 5 & 5 & 0 & 0 \\ 0 & 0 & 10 & 10 \\ 0 & 0 & 10 & 10 \end{bmatrix},\ R=1 \tag{40}$$

The optimal state feedback law is given by

$$u_x(t) = -\mathbf{K}_\mathbf{x}^\mathbf{o}\mathbf{X} \tag{41}$$

where $\mathbf{K}_\mathbf{x}^\mathbf{o} = \mathbf{PB}R_1^{-1}$ and the matrix $\mathbf{P}$ satisfies the Riccati equation, $\mathbf{A}^\mathbf{T}\mathbf{P} + \mathbf{PA} - \mathbf{PB}R^{-1}\mathbf{BP} + \mathbf{Q} = 0$.

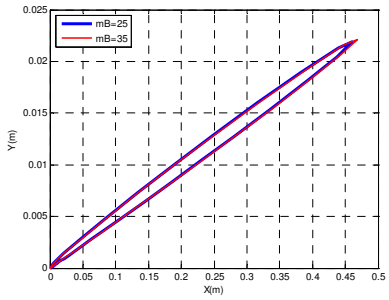
In practice, these parameters are re-tuned based on simulation results. For given choices of $\mathbf{Q}$ and R, an LQR command, lqr, in Matlab can be used to compute the associated state feedback gain matrix K_x^0

$$\mathbf{K}_\mathbf{x}^\mathbf{o} = [K_{x1}^o \quad K_{x2}^o \quad K_{x3}^o \quad K_{x4}^o] = \text{lqr}(\mathbf{A},\mathbf{B},\mathbf{Q},\text{R}) \tag{42}$$

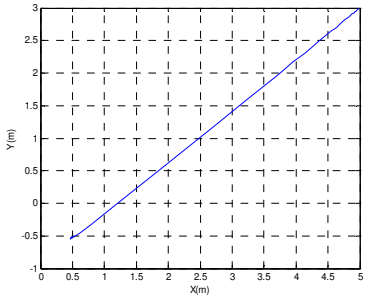
which defines the optimal stabilizing feedback control law of (38) so as to minimize the LQR cost function. In comparison with (39), all the controller's parameters, K_A, K_{AV}, K_T and K_V, are obtained from the following equality

$$[K_A \quad K_{AV} \quad K_T \quad K_V] = [-K_{x1}^o \quad -K_{x2}^o \quad -K_{x3}^o \quad -K_{x4}^o] \tag{43}$$

It is worthwhile to note that the four controller's parameters in (36) are the same for both control laws of u_x and u_y. Therefore, there is no further information for the four controller's parameters. The main result in this section is summarized as below.



(a)



(b)

Fig.4 Simulation of the proposed LQR controllers : (a) station keeping; (b) point stabilization.

Theorem 1: *Let the proposed LQR laws for u_x and u_y be given in (36).Then, the four parameters, K_A, K_{AV}, K_T and K_V, can be found using the LQR method, and satisfy the equalities (43), thus showing the asymptotical stability of the overall closed-loop control system, namely that self-balancing, position control and trajectory tracking can be achieved by using (36).*

IV. SIMULATIONS AND DISCUSSION

To examine the performance and merit of the proposed two LQR controllers, this section will conduct two simulations:(1)station keeping; (2) point-to-point stabilization. The parameters of the robot are given in Table 1. Both simulations are executed using Matlab/Simulink and the proposed controller with the following calculated LQR gains: K_A=30.9507 , K_{AV}=9.1924 , K_T=3.1623 and K_V=5.9628 .

4.1 Station Keeping

The first simulation is aimed to investigate station keeping in the origin of the ball robot. For the simulation, the initial conditions are given by $x = 0\text{m}, \dot{x} = 0.05\text{m/s}$, $\dot{\theta}_y = 0.005\text{rad/s}, y = 0\text{m}, \dot{y} = 0.05\text{m/s}, \theta_x = 0.3\text{rad}, \theta_y = 0.01$ rad , and the mass of body is 25kg. In order to test the robust property of the proposed controllers, the body mass of the robot is altered from 25kg to 35kg. Fig. 4(a) depicts the simulation results for the robot with the weights of 25kg and 35kg. The results in Fig. 4(a) indicate that the proposed LQR controllers are shown capable of keeping the robot at the origin, and they are robust against weight variations of the robot.

4.2 Point-To-Point Stabilization

In the second simulation, position control from the initial position (0.5m, -0.5m) to the end (5m, 3m) is simulated when the initial conditions are given by $\dot{x} = -0.01$ m/s $\theta_x = 0.15$ rad, $\dot{\theta}_x = 0.1$ rad/s, $\dot{y}$ = -0.02 m/s, $\theta_y = 0.1$ rad, $\dot{\theta}_y = -0.015$ rad/s. Fig.4(b) illustrates the simulation result for this case, showing that the proposed controllers successfully steer the robot to reach the desired position (0.5m, -0.5m). Worthy of mention is that the robot is controlled to reach the end position along a straight line. Furthermore, the results in Figs. 4(a) and (b) indicate that the proposed LQR controllers perform well in achieving self-balancing and point stabilization.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The objective of this section is to conduct one self-balancing experiment to explore the actual performance of both designed LQR control laws for the ball-riding robot. Fig. 5 depicts the experimental data of the tilt and gyro of the ball-riding robot for station keeping. Note that the experimental data were taken by the serial communication port, RS232, from the main controller, DSP 28335. As can be seen in Fig. 5 the tilt angle and the inclination rate of the controller ball-riding robot were almost maintained at zero. Fig.6 shows that the robot was commanded to stand still on a ball; although the robot fluctuated around the origin, it did achieve station keeping.

VI. CONCLUSIONS

This paper has presented techniques for dynamic modeling and LQR control of a ball-riding robot with three omnidirectional wheels, which can be thought of as a combination of a two-dimensional inverted pendulum and the inverse Atlas spherical motion platform. The completely nonlinear dynamic model of the mobile platform has also been established using Lagrangian mechanics; this new model is composed of both dynamic parts: one is from the two-dimensional inverted pendulum and the other is from the inverse Atlas spherical motion platform. The model has been shown valid in comparison with the simplified one in [3]. Two stable LQR control laws have been presented for station keeping and point stabilization of the ball-riding robot. For the ball robot with the tilt sensor, gyroscope, accelerometer and three encoders, both self-balancing and point stabilization problems can be regarded as two regulation problems, namely that the controllers are designed to bring nonzero initial errors towards zero. Moreover, both proposed controllers are shown to be capable of achieving station keeping and point stabilization. Through computer simulations and experimental results, the proposed double PD control methods together with the constructed ball robot has been successfully shown powerful and useful in achieving station keeping and point stabilization. Interesting topics for future research would be to conduct the experimental results for position control, and to address the trajectory tracking problem of this ball-riding robot.

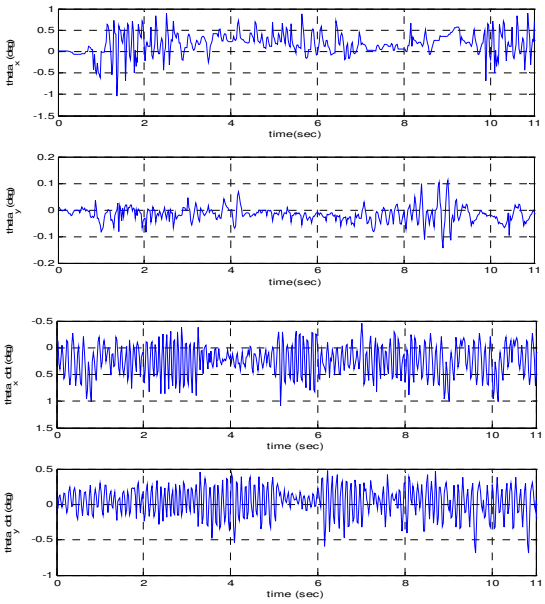


Fig.5. Experimental results of both tilt angles and rates for self balancing and station keeping.

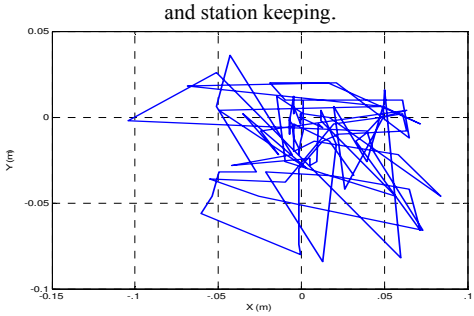


Fig.6. Experimental robot positions in both x and y axes.

ACKNOWLEDGMENT

The authors deeply acknowledges the financial support from the National Science Council, Taiwan, R.O.C., under the grant number NSC 100-2221-E-005-033.

REFERENCES

[1] R. Hollis, "Ballbots," *Scientific American Magazine*, pp. 72-77, Oct. 2006.

[2] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A Dynamical Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive," *Proceedings of the IEEE International Conference on Robotics and Automation,* Orlando, USA, pp. 2884-2889, 2006.

[3] U. Nagarajan and A. Mampetta, G. A. Kantor and R. L. Hollis, "State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot" in *Proc.IEEE Int. Conf. Robot. and Autom.*, pp. 998-1003.,2009.

[4] U. Nagarajan, G. A. Kantor and R. L. Hollis, "Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot" in *Proc. IEEE Int. Conf. Robot. And Autom.*, pp. 3743-3748, 2009

[5] M. Kumagai, T. Ochiai, "Development of a Robot Balancing on a Ball" in *Proc. IEEE Int. Conf. Contr. Autom. And Systems*, pp. 433-438, 2008

[6] M. Kumagai and T. Ochiai, "Development of a Robot Balancing on a Ball- Application of passive motion transportation," in *Proc. IEEE Int. Conf. Robot. And Autom.*, pp. 4106-4111, 2009.

[7] http://rezero.ethz.ch/project_en.html [online] 2011-07.

[8] A. Weiss, R. G. Langlois, and M. J. D. Hayes, "The Effects of Dual Row Omnidirectional Wheels on the Kinematics of the Atlas Spherical Motion Platform," *Mechanism and Machine Theory*, vol. 44, pp. 349-358, 2009.