Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 1/14

Dynamic Modeling and Simulation of Q-Baller—A Spherical Wheeled Robot

Jiamin Wang and Yuyi Lin

AQ1

Abstract In this paper we present the dynamic modeling of Q-Baller—a Spherical Wheeled Robot (Ball-Bot) design. A low cost prototype is constructed to verify the theoretical derivation. Low cost is achieved by using inexpensive structural design and electronic components, which, however, increases the difficulty for control. The Ball-Bot's dynamic model is important for the later studied and design suitable controller. Simulations are performed to test the design concepts, which includes detailed system modeling, noisy environment setup and the human machine interface. The results of the simulation reveal some potential problems of the current design and the improvements needed in both mechanical components and controller designs. The simulations also prepare us for improving the Ball-Bot prototype and future experiments.

Keywords Spherical wheeled robot · Dynamic modeling · Modern control · Real time simulation · Noisy simulation environment · Human machine interface

1 Introduction

To develop low cost mechatronic or robotic products that involves control issues can be quite challenging—inaccessibility of high-end or professional processors and sensors; lack of manufacturing accuracy in prototyping; shortage of human resources and experimental environment. These challenges may bring any ambitious but not well-funded project to a halt.

Since 2005 when Carnegie Mellon University (CMU) accomplished the first Spherical Wheeled Robot—a robot that can maintain balance standing on and moving around with a ball, several spherical wheeled robot designs as shown in Fig. 1 have been introduced. Inspired by these predecessors, we have been

J. Wang (✉) · Y. Lin
University of Missouri, Columbia, MO, USA
e-mail: jw2gd@mail.missouri.edu

Y. Lin
e-mail: liny@missouri.edu

X. Zhang et al. (eds.), *Mechanism and Machine Science*,
Lecture Notes in Electrical Engineering 408,
DOI 10.1007/978-981-10-2875-5_93

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 2/14

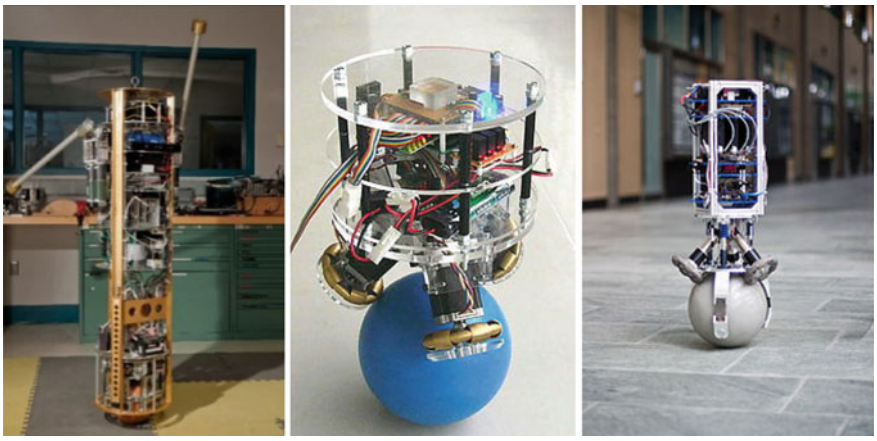

Fig. 1 *Left* the CMU Ball-Bot [1]; *Middle* the BallIP [2]; *Right* the Rezero [3]

developing a Spherical Wheeled Robot starting from 2015. Our design costs less than most of the current Spherical Wheeled Robots on the market. In addition to adopting reasonable mechanical design structure, the electronic components we use are, in most cases, less advanced than those in the other designs which, however, posed a different and more complex control task.

We performed detailed modeling and simulations to better understand the design and its potential problems. Our system model has combined mechanical electronics dynamics, and involves detailed parameters to derive random models for uncertainty simulation. Simulation processes include a noisy environment and a real-time human machine interface to facilitate a more realistic study of Ball-Bot dynamics. The process helps us better understand our control task and prepare for a suitable controller design. The study also helps us figure out the improvements to be made to our current design.

2 The Q-Baller Design

Our basic design goal is to design a Ball-Bot with great dynamic performance and possibly the lowest cost. Presented in Fig. 2 is our current Ball-Bot design named Q-Baller. The robot is about $300 \times 300 \times 400$ mm^3 in size, and 8 kg in weight. Q-Baller has 4 friction drive system to provide symmetric control performance. The spherical wheel is made from an iron spherical shell with a high traction coating. We also have a four-bearing system, which serves to secure the frictional surface contact and allow addition add-ons for future uses. All structural parts are sheet metal providing sufficient strength with low material and manufacture cost.

Instead of using high-end and expensive servo motors, we adopt four brushed DC motors without any speed feedback sensors (rotary encoders) in our design.

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 3/14

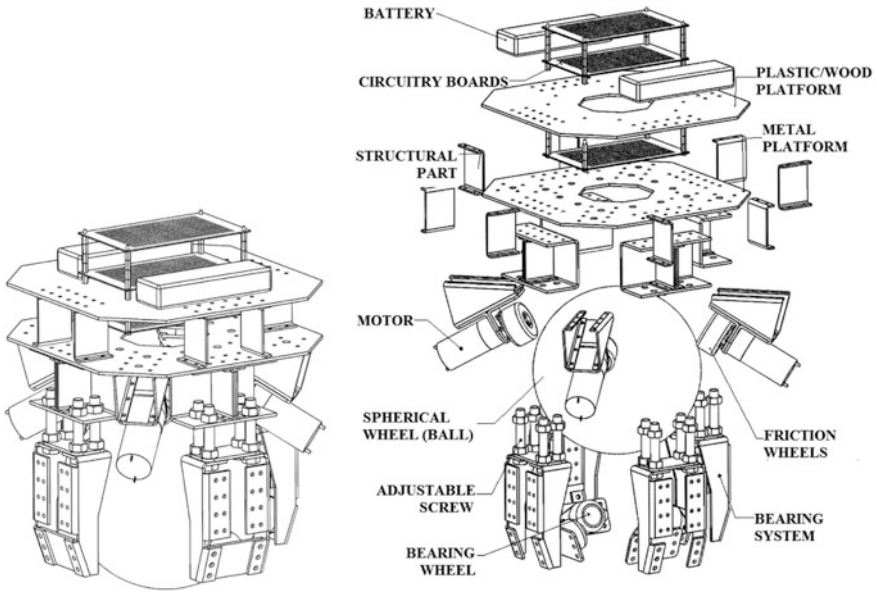

Fig. 2 Q-baller spherical wheeled robot design

These motors have less control accuracy, but are less expensive than the servo motors, and can generate greater power compared to the stepper motors of the same size. Rotary velocity feedback can be installed as add-ons if necessary (Fig. 3).

Since DC motors require only the simplest control hardware, the electronic circuitry of Q-Baller has also been minimized, leaving the gyroscope/accelerometer sensor currently the only sensor in our robotic system. A Bluetooth device provides communication between the Ball-Bot and the microcontroller. The control

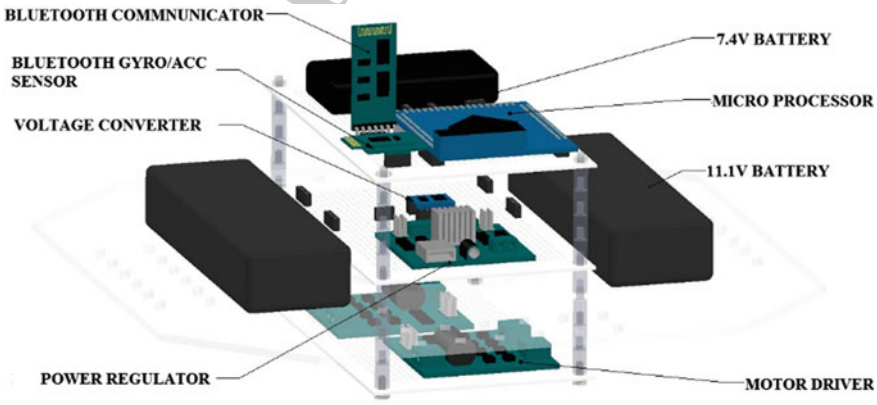

Fig. 3 Electronic system of Q-Baller

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 4/14

programs are processed by a 32-bit microprocessor, which sends a control signal to drive the motor through the motor driver module boards.

Some features of this ball-bot are:

1. Without highly reliable feedback and powerful processors, the control sensitivity, accuracy and frequency is limited. Many complicated control methods cannot be used due to the calculation limitation.
2. No guarantee exists that the friction surfaces have good contact. Since the DC motors are controlled without any state feedback, DC motor outputs may not be stable and identical, causing uncertainties in the system.
3. The prototype may not be the same as approximated ones from ideal modeling due to differences in manufacturing accuracies and actual component properties.

As a result, simulations must check the design feasibility before prototyping. Even though the simulation results may be limited, they may help improve the Q-Baller design, control algorithm and programing.

3 Dynamic Analysis

The Cartesian coordinate frames of the system used in the modeling process are depicted in Fig. 4. The ground coordinate frame (Frame G) will be set up by the time the robot starts running. Frame O's origin is at the center of the ball and it takes form after the orientation conversion around the Z axis (Yawing) from Frame G. Similarly, the Local Frame L takes form after the Pitching and Rolling of the coordinate system. The origins of Frame O and Frame L are both located at the

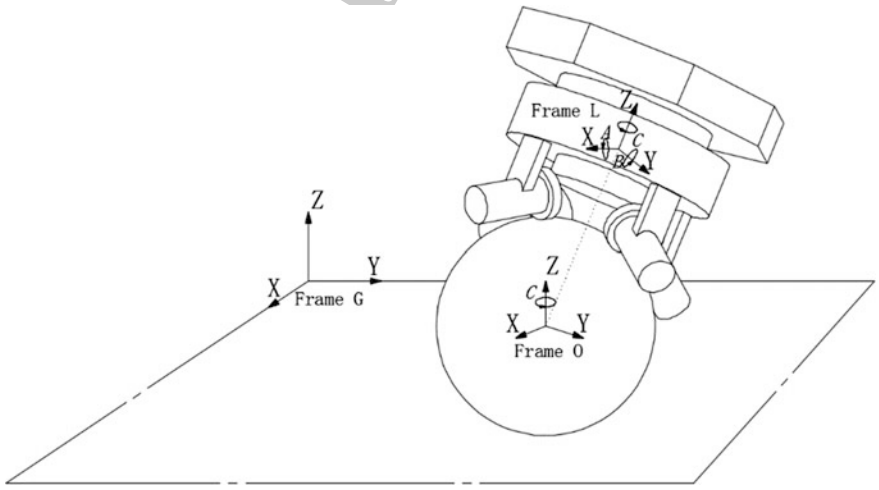

Fig. 4 Different coordinate frames of the system

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 5/14

center of the ball, while in Fig. 4 Frame L is moved away only for clearer presentation purpose.

Figure 5 describes vectors of forces and geometries in the system: $\vec{V}_M$ stands for the vector pointing from the origin to the center of mass of the robot body; $\vec{V}_R$ and $\vec{V}_W$ points from the origin to the contact point of the ground and one of the friction Omni-Wheels respectively; $\vec{V}_m$ and $\vec{V}_r$ stands for the positive direction of the rotary velocity and the positive translational velocity direction at the contact point on the Omni-Wheel from one of the motors respectively; $\vec{T}_m$ is the actuating torque effect on the robot body from one of the motors; $\vec{F}_G$ is the gravity force of the Ball-Bot's body (origins from the Center of Mass of the body); and $\vec{T}_d, \vec{F}_d, \vec{T}_D, \vec{F}_D$ are the 4 exterior perturbation torques and forces acting on the body and the ball (which are under Frame G) which will be used as process noises during simulation. (Note that $\bar{V}_w, \bar{V}_m, \bar{V}_r$ and $\bar{T}_m$ are sets of 3×4 matrixes because there are 4 motors in total, and direction vectors are unit vectors.)

To adopt a Lagrangian Method [4], we first choose the generalized coordinates according to the assumptions: (1). the Ball-Bot will always be on the ground; (2). there is no slipping between any of the contact surfaces; (3). the body never loses contact with the ball. The coordinate vectors chosen only include 5 states in total:

$$q = [A \quad B \quad C \quad X \quad Y]^T \tag{1}$$

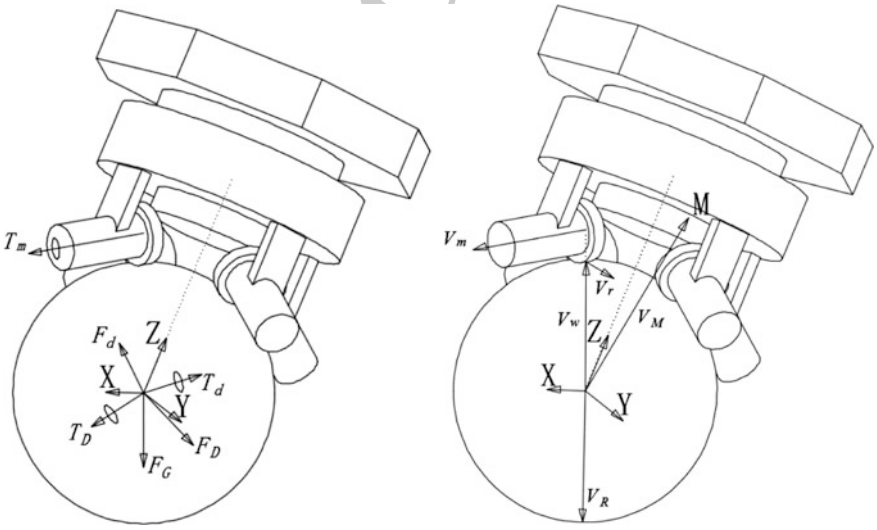

Fig. 5 Force and geometric vectors of the system

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 6/14

As shown in (1), A, B, C are the Euler Angles describing the rotation of the robot body around axis X (Rolling), Y (Pitching), Z (Yawing) in Frame L, and X, Y are the accumulated distance of the Ball-Bot traveled along X and Y directions in Frame O which are derived from the rotation of the ball. Some of the terms are derived from the coordinate vectors. Here are the terms for the Lagrange Equations:

$$W_{B-L} = M_J \begin{bmatrix} \dot{A} & \dot{B} & \dot{C} \end{bmatrix}^T \tag{2}$$

$$W_{b-O} = \begin{bmatrix} -\frac{\dot{Y}}{R} & \frac{\dot{X}}{R} & 0 \end{bmatrix}^T \tag{3}$$

$$V_G = C_L^G V_O = C_L^G * \begin{bmatrix} \dot{X} & \dot{Y} & 0 \end{bmatrix}^T \tag{4}$$

Here R is the radius of the ball, which is a scalar value. C_L^G is the coordinate conversion matrix from the local to the ground frame. The other coordinate conversion matrixes are similar (For example: C_X^Y is the coordinate transformation matrix from X frame to Y frame.) M_J is the Jacobian Matrix transforming the Euler Angle rates to the angular velocity of the robot body. Some of these conversion matrixes are shown below in (5), (6) [5, 6]:

$$M_J = \begin{bmatrix} 1 & 0 & -\sin(B) \\ 0 & \cos(A) & \sin(A)\cos(B) \\ 0 & -\sin(A) & \cos(A)\cos(B) \end{bmatrix} \tag{5}$$

$$C_L^G = \begin{bmatrix} \cos(C) & -\sin(C) & 0 \\ \sin(C) & \cos(C) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(B) & 0 & \sin(B) \\ 0 & 1 & 0 \\ -\sin(B) & 0 & \cos(B) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(A) & -\sin(A) \\ 0 & \sin(A) & \cos(A) \end{bmatrix} \tag{6}$$

According to the Lagrangian formulation, we have the following equations which govern the motion of the Ball-Bot's entire mechanical system [4, 6]:

$$T_L = \frac{1}{2} V_G^T (M+m) V_G \tag{7}$$

$$T_R = \frac{1}{2}\left(W_{B-L}^T J_B W_{B-L}\right) + \frac{1}{2}\left(W_{b-O}^T J_b W_{b-O}\right) \tag{8}$$

$$T_C = M\left(V_G^T C_L^G\right)\left(W_{B-L} \times V_M\right) \tag{9}$$

$$V = -F_G C_L^G V_M \tag{10}$$

$$L = T_L + T_R + T_C - V \tag{11}$$

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 7/14

$$Q_I = W_{b-O}^T C_L^O \left(T_W + C_G^L T_d\right) + W_{B-L}^T \left(-T_W + C_G^L T_D\right) + V_G^T (F_D + F_d) \quad (12)$$

$$Q_D = \frac{1}{2} V_G^T \left(C_L^G B_{BL} + B_{bL}\right) V_G + \frac{1}{2} \left(W_{B-L}^T B_{BR} W_{B-L}\right) + \frac{1}{2} \left(W_{b-O}^T B_{bR} W_{b-O}\right) \quad (13)$$

And finally the Lagrange Equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) = \left(\frac{\partial (Q_I dt)}{\partial (dq_i)}\right) - \left(\frac{\partial Q_D}{\partial \dot{q}_i}\right) (for\ i = 1, 2 \ldots 5) \quad (14)$$

Here $T_L + T_R + T_C$ make up the kinematic energy of the whole system including translation, rotary movement plus the coupling effect since the rotary coordinate of the robot body does not origin from its center of mass; V accounts for the potential energy from the non-conservative forces—the gravity forces, for which the Zero-Potential-Level is set at the horizontal plane that goes through the center of the ball; Q_I stands for the virtual work caused by internal and exterior inputs; and Q_D is the energy dissipation term according to Rayleigh Dissipation Function caused by viscosity dampers in the system. (M, J and B terms are translational inertias, rotary inertias and viscosity dampers measured in the components' local frames.)

The following part is the modeling of the mechatronic system and the constraints to complete the modeling of the whole mechatronic system. According to the geometry of the robot system, the working torque T_W as the sum of the motor outputs acting on the ball:

$$T_W = -\frac{R}{r} T_m * [\,1 \quad 1 \quad 1 \quad 1\,]^T \quad (15)$$

where each motor's torque T is determined by the dynamics of the DC motor [7, 8]:

$$\frac{T_{s_0} U}{U_0} - T = J_w \dot{\omega} + b_w \omega + \frac{60 T_{s_0}}{2\pi n_{0_0}} \omega \quad (16)$$

Equation (16) has omitted the fast dynamics from the inductances in the motor, and it is simplified by the free spin velocity n_{0_0}, stalling torque T_{s_0} and nominal voltage U_0 according to the typical steady performance plot of DC motors.

According to the nonslip condition, the contact point velocity at the friction wheel and the ball should be identical. Thus, for each motor the velocity equation should be:

$$\omega_j = V_{r_j}^T \left(\left(C_O^L W_{b-O}\right) \times V_{W_j} - W_{B-L} \times V_{W_j} \right) / r \quad (17)$$

where V_{XX_j} is the jth column of the V_{XX} matrix, and $j = 1, 2, 3, 4$. The equations couple the electronic systems of the motors with the mechanical system of the robot. Governing equations therefore can be rearranged into the standard form:

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 8/14

$$x = \begin{bmatrix} A & B & C & X & Y & \dot{A} & \dot{B} & \dot{C} & \dot{X} & \dot{Y} \end{bmatrix}^T \tag{18}$$

$$u = \begin{bmatrix} U_{x+y+} & U_{x+y-} & U_{x-y-} & U_{x-y+} \end{bmatrix}^T \tag{19}$$

$$v = \begin{bmatrix} T_D^T & F_D^T & T_d^T & F_d^T \end{bmatrix}^T \tag{20}$$

And the standard ODE functions are:

$$\dot{x} = f(x, u, v) \tag{21}$$

$$y = g(x, w) \tag{22}$$

where u is the input of the system, v is the process noise, y is the output (or measurement) of the system, and $w(x)$ is the measurement noise. This completes the system model with detailed dynamics including exterior perturbations and possible errors.

In order to study the effect of potential uncertainty that may occur because of modeling inaccuracies, we generated two different sets of governing equations based on 2 different property setups—the standard model whose properties are acquired from 3D modeling, and a randomly generated model whose properties are presumed unknown.

The system generally presents strong nonlinearity at most of the areas. However, when linearized at the point $x^T = [0 \cdots 0]_{1\times 10}$ (zero point) and presented in State-Space Form, the ideal system characteristic matrix A_{SS} and B_{SS} for $\dot{X}_{SS} = A_{SS}X_{SS} + B_{SS}U_{SS}$ [9] would be:

$$A_{SS} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 69.27 & 0 & 0 & 0 & 0 & -8.23 & 0 & 0 & 0 & -81.08 \\ 0 & 67.60 & 0 & 0 & 0 & 0 & -8.04 & 0 & 79.12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.98 & 0 & 0 \\ 0 & -5.17 & 0 & 0 & 0 & 0 & 1.11 & 0 & -11.12 & 0 \\ 5.30 & 0 & 0 & 0 & 0 & -1.13 & 0 & 0 & 0 & -11.27 \end{bmatrix} \tag{23}$$

$$B_{SS} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 9.68 & -9.44 & -5.68 & 1.32 & 1.34 \\ 0 & 0 & 0 & 0 & 0 & -9.68 & -9.44 & 5.68 & 1.32 & -1.34 \\ 0 & 0 & 0 & 0 & 0 & -9.68 & 9.44 & -5.68 & -1.32 & -1.34 \\ 0 & 0 & 0 & 0 & 0 & 9.68 & 9.44 & 5.68 & -1.32 & 1.34 \end{bmatrix}^T \tag{24}$$

Matrixes in (23) and (24) indicate the symmetric characteristic of the Q-Baller design. They also present the weak bonds between states and low nonlinearity around the zero point. The zero point is also an equilibrium point, which would be a good choice for designing a linear controller.

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 9/14

4 Controller Design

We presume that the random model properties cannot be acquired, along with the unidentified process and measurement noise that will take place in real practice. So the controller design is only based on the ideal standard model. The preliminary simulation only deploys a standard PID linear controller [9] to the system. Gain scheduling method [10] may take place when PID at a single operating point is insufficient to cover the required controllable areas. The system flowchart is presented in Fig. 6 Left.

We used a LQR designer [11] to find the initial parameters for the PID controller, These values optimize the control performance by minimizing a cost function $\left(J = \int_0^\infty (x^T Q x + u^T R u)\, dt\right)$ which balances performance and energy conservation. The PID controller will later be adjusted to enhance performance.

Linear controllers are only effective in vicinity areas, and may fail when the system goes beyond certain boundaries. The Gain Scheduling Method [10] is adopted to solve this problem especially when the Q-Baller is required to operate at high speed. We have effectively chosen several equilibrium points which provide enough coverage for the required velocity control areas. The idea is depicted in the right side of Fig. 6, where you can see that each small circle indicates the approximate effectiveness boundary of an operating point. Generally, a linear controller performs better in the vicinity of the zero point, but may have problems away from the zero point. A good algorithm should be circumspective and smooth to avoid drastic jumps in parameters during scheduled zone switches.

5 Simulation Layout

The simulation loop is shown in the flowchart in Fig. 7. In addition to discretize the originally continuous control model into code loops, we have added simulation modules for the sensor and filters along with noise generators. We have also separated the numerical integration process and included a time lag feature in the program to add realistic challenges to the control problem.

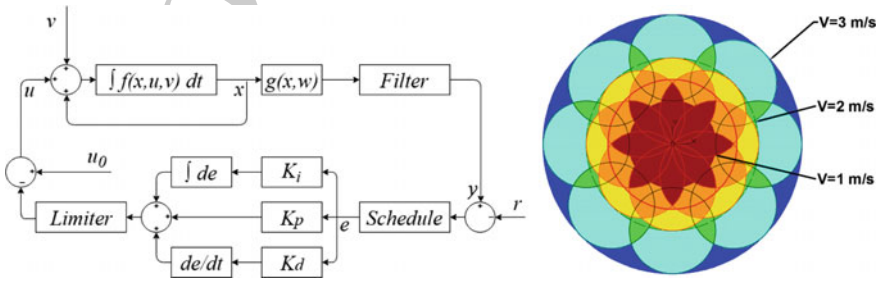

Fig. 6 *Left* control system flowchart; *Right* gain scheduling tiers

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 10/14

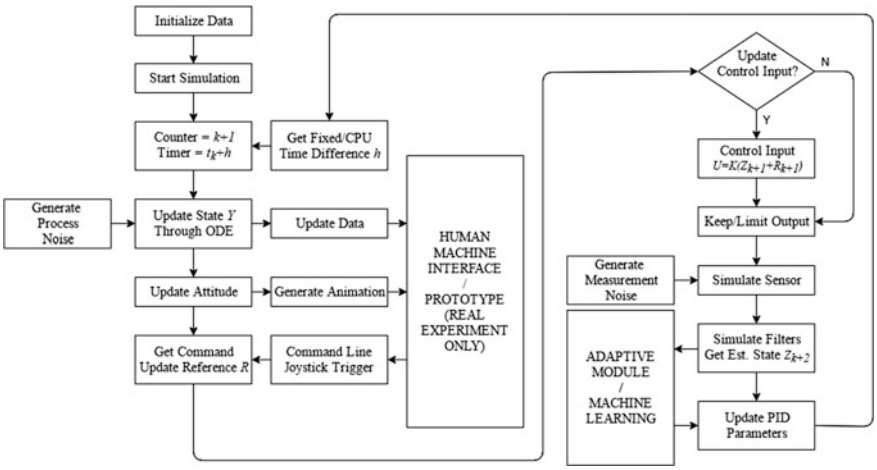

Fig. 7 Simulation flowchart

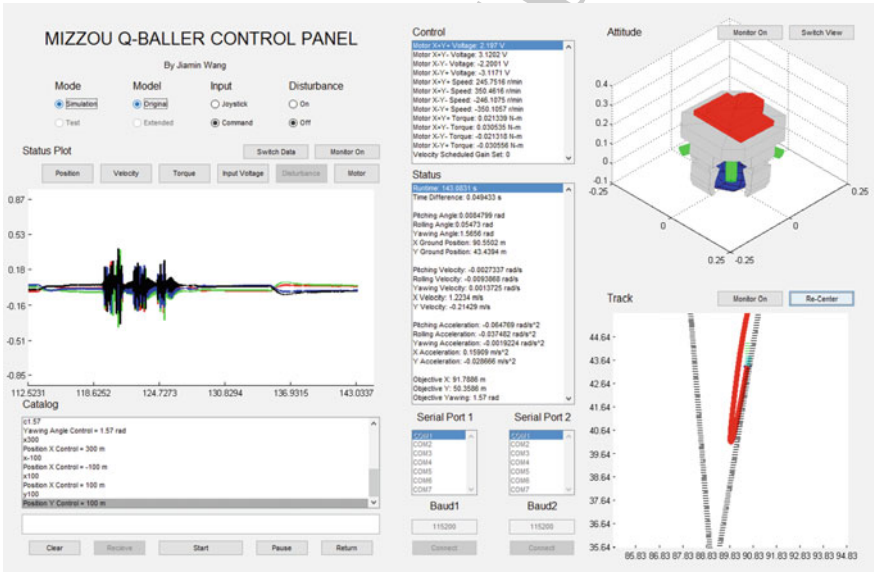

Fig. 8 MATLAB GUI human machine interface for Q-Baller

A Human Machine Interface based on MATLAB GUI has been constructed to support real time simulation as shown in Fig. 8. The interface is able to generate real time plotting, 3D animation and trajectory recording. In addition to type-in commands we use a gaming joystick as a more convenient alternative input. The interface will also be modified into an experimental interface for the Q-Baller after

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 11/14

prototyping is completed, facilitating observation, comparing models, data collecting and possibly recognizing dynamic pattern.

Adaptive and machine learning modules are something we have just started working on. Future planned improvements include adaptive control t and neural network techniques to improve the controller design for maneuvering a prototype whose model is vastly different from the ideal one we established before.

6 Some Results and Conclusion

Currently we are able to control the ideal model well under noiseless condition. We also realized that PD controller would be enough for position control, but Integral Control has to be included during velocity control to eliminate static error. Under noiseless conditions, a velocity controller can cover up to 3 m/s in any direction on the horizontal plane with Gain Scheduling.

However, the situation becomes very different with noise. A random model simulation shows this with reference following at low speed noise. The sensor feedback frequency is 50 Hz and the control input update frequency is 25 Hz. Both process and measurement noise are generated randomly with amplitudes realistically set. The feedbacks are digitally filtered and later used in state estimation through a Discrete Kalman Filter presented from (25) to (30) [13]:

$$x_{n+1,n} = Ax_n + B\left(u_n + v\right) \tag{25}$$

$$E = y_n - Cx_n - Du_n \tag{26}$$

$$P_{n+1,n} = AP_nA^T + Q_n \tag{27}$$

$$K_n = P_{n+1,n}C^T\left(CP_{n+1,n}C^T + R_n\right)^{-1} \tag{28}$$

$$P_{n+1} = \left(I - K_nC\right)P_{n+1,n} \tag{29}$$

$$x_{n+1} = x_{n+1,n} + K_nE \tag{30}$$

where x is the state estimation through integration of gyroscope sensor data; y is the measurement of Pitch and Roll angle through accelerometer data; Initial parameters P, Q, R are empirically determined which can be either fixed or variable [13, 14]. AQ4

Kalman filters are very useful tools when dealing with the digital signal feedback from a gyroscope/accelerometer under noisy conditions for pitch and roll states. As displayed in Fig. 9 Left, the Kalman Filter successfully kept the rolling angle from drifting away during numerical integration. However, Fig. 9 Right shows the integrals of accelerometer (displacement and velocity) are not reliable due to the truncation error and amplification of noise.

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 12/14

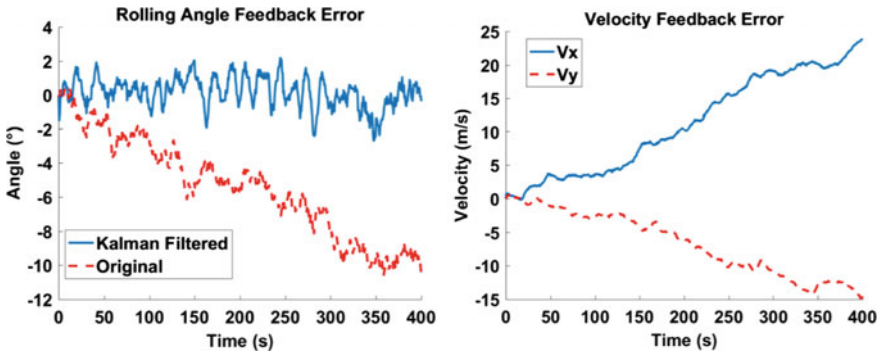

Fig. 9 *Left* performance of kalman filter; *Right* accelerometer data integration

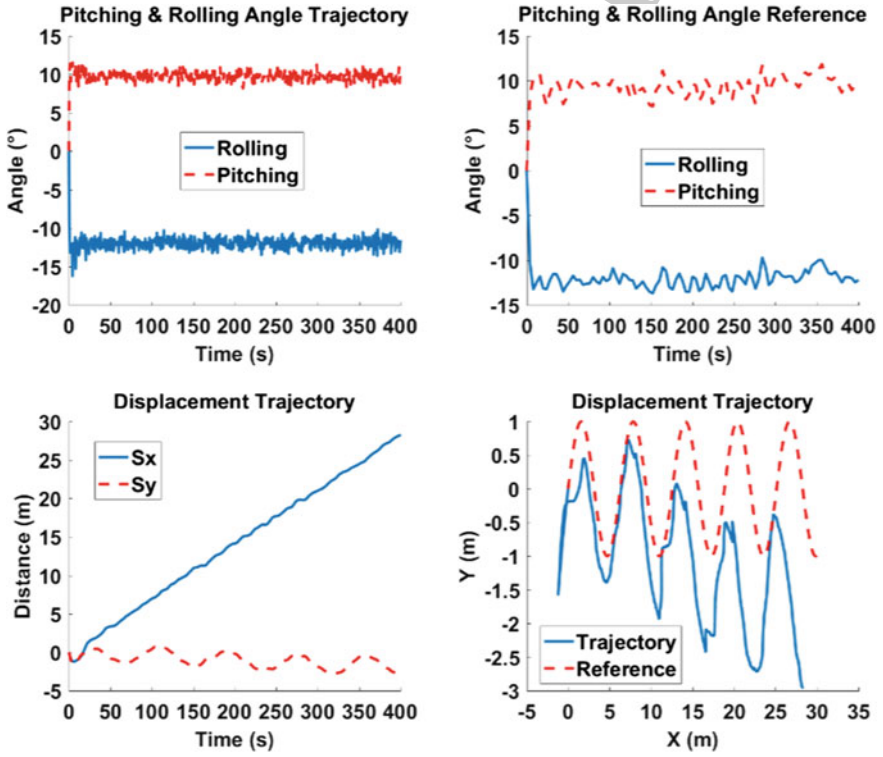

Fig. 10 Trajectory following simulation under noisy condition

Here we adopted some simple adaptive control algorithm. Assuming that the displacement and velocity feedback of Q-Baller are reliable (directly updated by ODE), and we use sensor simulator for Pitch, Roll and Yaw feedbacks, then the robot can be stably controlled as shown in Fig. 10. The adaptive control mainly

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 13/14

works on adjusting the Pitch and Roll references so as to eliminate static error in position control caused by inaccurate modeling. The angle references tend to converge to 9 and −12 degrees respectively, indicating that those may be the 'Zero Point' States of this random model.

It is also easy to see that the Q-Baller cannot follow the trajectory well, and starts to deviate toward −Y direction. The phenomenon results from the lack of a Kalman Filter for Yawing to eliminate error accumulation.

Based on these results we have reached the following conclusions about the current Q-Baller design, control program and simulation:

1. Controllers and filters should be further improved for better product control performance.
2. Rotary encoders should be installed to monitor spherical wheel rotations. Rotary encoders provide more accurate translational velocity and position feedback.
3. A magnetometer should be installed to form a Kalman Filter along with the gyroscope sensor to eliminate Yawing errors
4. The current simulation has not yet been perfected since features like friction surfaces are very complicated to model. More accurate simulation should be made before experimenting on the prototype to possibly reveal more problems.

7 Conclusion and Future Plans

The simulation results we have acquired so far indicated that our design needs further improvement. The results have also shown us that our design may be feasible if better refined and controlled. Better structural design obviously will lead to better control conditions, while control theory would certainly help optimizing the product design. Simulation serves as the bond and facilitates the mutual improvement between product design and controller design.

For the moment Our Ball-Bot is already being prototyped and will be tested in the near future. As mentioned previously we will try some more complicated control algorithms and refine the current PID controller. Simulation will help us predict more problems before actually getting our hands on the robot. Our final goal would be successfully developing a Q-Baller software system with excellent control performance.

Acknowledgments The authors thank Dr. Ming Xin, Dr. Qingbin Tong, Mr. Zhengwei Nie from Department of Mechanical & Aerospace Engineering, University of Missouri—Columbia for their supports in fields of mechanical, control and signal process engineering. The authors also thank Mr. Yueqi Yu from Department of Electronic and Computer Engineering for his support in fields of electronic and computer engineering. Finally, the authors thank Dr. Peter Hodges for proof reading the manuscript.

Layout: T1 Standard | Book ID: 429718_1_En | Book ISBN: 978-981-10-2874-8
Chapter No.: 93 | Date: 7-10-2016 | Time: 10:05 am | Page: 14/14

References

1. Lauwers TB, Kantor GA, Hollis RL (2006) A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive. In: IEEE international conference on robotics and automation. pp 2884–2889
2. Kumagai M, Ochiai T (2008) Development of a robot balancing on a ball. In: International conference on control (Seoul, Korea: Automation and Systems). pp 433–438
3. Doessegger S, Fankhauser P, Gwerder C, Huessy J, Kaeser J, Kammermann T, Limacher L, Neunert M (2010) Rezero, focus project report. Autonomous Systems Lab, ETH Zurich
4. Greenwood DT (1987) Principle of dynamics, 2nd edn. Pearson Education, Upper Saddle River. ISBN-13: 978-0137099818
5. Kim P (2013) Rigid body dynamics for beginners: euler angles & quaternions. CreateSpace Independent Publishing Platform ISBN-13: 978-1493598205
6. Raffo GV, Ortega MG, Rubio FR (2010) An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter. Automatica 46(1):29–39
7. Emhemed AAA, Mamat RB (2012) Modelling and simulation for industrial DC motor using intelligent control. In: International symposium on robotics and intelligent sensors 2012 (IRIS 2012), Procedia Engineering vol 41, pp 420–425
8. Hao Y, Miao M, Luo X (2009) Electric drive control (Chinese edition). Huazhong University of Science and Technology Press, Wuhan. ISBN-13: 978-560958620
9. Ogata K (2009) Modern control engineering, 5th edn. Pearson Education, Upper Saddle River. ISBN-13: 978-0136156734
10. Khalil HK (2002) Nonlinear systems, 3rd edn. Pearson Education, Upper Saddle River. ISBN-13:978-9332542037
11. Naidu DS (2012) Optimal control systems, 1st edn. CRC Press, Boca Raton. ISBN-13: 978-0849308925
12. Kalman RE (1960) A new approach to linear filtering and prediction problems. Trans ASME J Basic Eng 82(Series D):35–45
13. Feng B, Fu M, Ma H, Xia Y, Wang B (2014) Kalman filter with recursive covariance estimation—sequentially estimating process noise covariance. IEEE Trans Ind Electron 54 (1):6253–6263