Open-Loop Trajcetory Planning and Nonlinear Control For Underactated Spherical Wheel Mobile Robot (Ballbot)

H. Zabihi
Department of Electrical Engineering
The Center of Excellence in Control and Robotics
Amirkabir University of Technology
Tehran, Iran
Email: zabihi@aut.ac.ir

H.A. Talebi*
Department of Electrical Engineering
The Center of Excellence in Control and Robotics
Amirkabir University of Technology
Tehran, Iran
Email: alit@aut.ac.ir

A.A Suratgar
Department of Electrical Engineering
The Center of Excellence in Control and Robotics
Amirkabir University of Technology
Tehran, Iran
Email a-suratgar@aut.ac.ir

***Abstract*— This paper investigate three dimensional dynamic model of underactuated spherical wheel mobile robot (Ballbot) in the presence of input coupling and friction. The way of how such a robot moves is considered and an approach is proposed for accurate trajectory generation for linear and circular paths. Due to the high reliability of the approach, it simulated open-loop and no position feedback is required. A collocated partial feedback linearization is introduced for Ballbot that ensures body angles trajectory tracking. The open-loop trajectory generation with collocated partial feedback linearization is simulated which shows a very low position error.**

Keywords—Underactuation; spherical wheel; trajectory generation; inverse dynamics control

I. INTRODUCTION

Underactuated systems are mechanical control systems with more degrees of freedom than control inputs [1]. There are different underactuated systems in aerospace, marine and robotics which open broad area of researches. Walking robot, flexible-link robot and balancing robots like Segway and Ballbot [2] are included in Underactuated robotics. Lack of control inputs for these robot makes the trajectory generation and control a challenging task.

Ballbot is a robot dynamically balancing on a ball and moves to wherever direction by actuating the ball. The dynamically stable robots unlike statically stable robots, can: (i) be tall and skinny with high centers of gravity, (ii) have smaller footprints, and (iii) accelerate or decelerate quickly [3]. The Ballbot first is introduced by Roph Holis and consisted of two orthogonal actuated rollers attached to the body and provided torque to the ball [3]. The two orthogonal motors couldn't provide yaw motion. This defect lids to representing a new actuating mechanism utilized three motors along with omnidirectional wheels which could have yaw motion [4]. Adding ball roller grippers to the three motors mechanism and proposing a Three Dimensional (3D) model of the robot, improved the mobility and control of the Ballbot [5]. Other attempts implemented timing belt gearbox to reduce actuating backlash [6] and calibrated measurement devices were used to improve set point stability [7].

A Balancing Underactuated mobile robot like the Ballbot has constraint on its system dynamic which causes difficulties in planning and control. These constraints are second-order, non-holonomic constraints that inherently restrict trajectories which the Ballbot can follow. In addition, there is strong coupling between its configuration variables; such that you can't move the Ballbot ball, without changing its body angles. Previous work has shown that, However, constraints and coupling effects make control strategies complicated, they are used to generate ball trajectories which could be followed by appropriate body angles [8]. More recent work has focused on computationally efficient methods for generating feasible trajectories based on constraints linearization and the concept of flat outputs [9] [10].

Dynamic constraints reveal the relation between shape space (the body) and the position space (the ball). As much as the relation accuracy, the trajectory is generated by means of it, would be more precise. Such approaches [8] [9] [10], have failed to address accurate system constraints because of: they (i) used simple Two Dimensional (2D) Ballbot model which its inaccuracy has proved [5], (ii) has linearized constraints which hide effective aspects of the constraints, (iii) has neglected friction in trajectory generation which causes accumulative position error. Hence, due to the trajectory inaccuracy the robot wouldn't get to the desired position by following the trajectory. To overcome this error, position compensation loop should be added aggressively [3].

Control of the Ballbot toward the generated trajectory is another concern due to its natural instability and underactuation. In [2] [3] a PID and LQR controllers are used based on the 2D decoupled model which is not covering coupling effects [5]. Moreover, linear control couldn’t handle such nonlinear systems farther away from the upright set point. In [11] a path-following controller is proposed, eliminating the need for following time-parameterized state trajectory. But it's not well-developed and the result of following circle path shows

*Corresponding author

significant steady-state errors. Hence there is a need for an accurate nonlinear controller.

In this paper the latest actuating mechanism for Ballbot is used which consists of three motors instead of two orthogonal motors. According to this mechanism, the authors improved the 3D model proposed in [5] by considering friction, hence a more accurate dynamic constraint has been attained. The constraint is investigated deeply and a novel online open-loop trajectory planning algorithm is introduced. An inverse dynamic controller in the presence of inputs coupling is introduced to follow the novel trajectory. The open loop trajectory following is simulated in Matlab, to prove the trajectory planning and the controller proposed.

The paper is organized as follows: Section II discusses the improved 3D Ballbot model and analyzing dynamic constraints and its effects on Ballbot movement. Section III describes the trajectory planning procedure. Section V presents the inverse dynamic controller, while section VI shows simulation results. Finally, section VII presents the conclusion and future works.

II. Ballbot 3D Model with friction

A. 2D Ballbot model

The first models of the Ballbot relied on 2D decoupled dynamic system. It modeled the Ballbot as a rigid cylinder on top of a sphere. Fig. 1. Three assumptions were made in that model: (i) no slip between the ball and the floor, and also between the wheels and the ball. (ii) The motions in three planes are decoupled (XY, YZ and XZ plane). (iii) The equations of motion in the sagittal and frontal planes are identical [12]. But the representation of such a nonlinear coupled system as a planar model implies multiple drawbacks [5]. It neglected the coupling effects between each plane which the maneuvers shows it up. In addition, many conversions are required to map the model to the real Ballbot, hence there are an urgent need to a 3D coupled model without any artificial conversions.

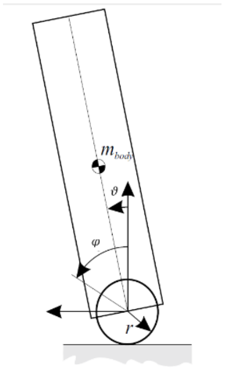


Fig. 1. plannar Ballbot model

B. 3D Ballbot model with friction

The configuration space of each dynamic system can be divided into position space and shape space. The position space in the Ballbot are ball angles $q_x = [\varphi_x \ \varphi_y]^T$ and the shape space are body angles $q_s = [\vartheta_x \ \vartheta_y \ \vartheta_z]^T$. The ball angles φ are the angles between the ball and the L coordinate, located at the center of the ball, while the body angle ϑ are the angles between the body and the word coordinate Fig. 2. The proposed 3D model has 5 rigid bodies as shown in the Fig. 2. No slip is the only assumption in this model.

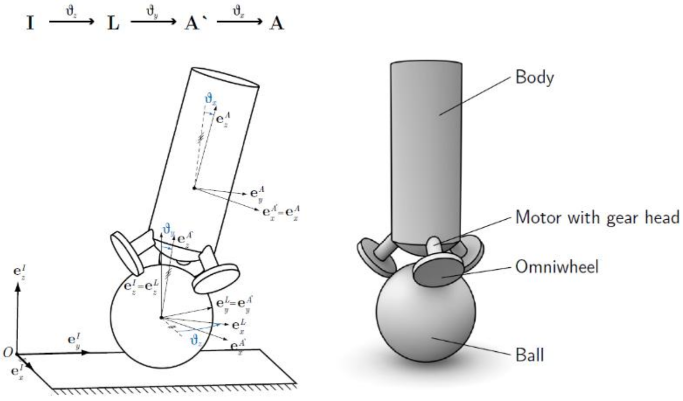


Fig. 2. 3D Ballbot Model

Euler-Lagrange equations are used to derive the dynamic equations of motion. Derivation of these equations for the 3D Ballbot model can be found in [5]. In this paper the model is improved by adding friction terms. Finally the equations of motion can be written in matrix form as follows:

$$M(q_s)\ddot{q} + C(q_s,\dot{q}_s)\dot{q} + G(q_s) + D(\dot{q}_x) = \Gamma(q_s)\tau \qquad (1)$$

Where $q = [\vartheta_x \ \vartheta_y \ \vartheta_z \ \ \varphi_x \ \ \varphi_y]^T$ is the generalized coordinate vector, $M(q_s) \in \mathbb{R}^{5\times5}$ is the mass/inertia matrix, $C(q_s,\dot{q}_s) \in \mathbb{R}^{5\times5}$ is the vector of Coriolis and centrifugal forces, $G(q_s) \in \mathbb{R}^{5\times1}$ is the vector of gravitational forces, $D(\dot{q}_x) \in \mathbb{R}^{5\times1}$ is the friction matrix, $\Gamma \in \mathbb{R}^{5\times3}$ is the coupling matrix and $\tau \in \mathbb{R}^{3\times1}$ is the input torque matrix. As can be seen the system is underactuated with input coupling.

The friction matrix added viscous damping friction term to the model. The friction is considered only between the ball and the ground which is more significant than friction between the ball and the wheels.

$$D(\dot{q}_x) = \begin{bmatrix} 0_{3\times1} \\ [D_s\,(\dot{\varphi})]_{2\times1} \end{bmatrix} \qquad (2)$$

C. Coupled dynamic constrainst

Because of the coupling matrix Γ, dynamic constraints aren't shown seperately and none of the right sides of the dynamic equations equals to zero explicitly. Hence there is no way to follow the approach given in [3]. The approach in [3] used the dynamic constraints to show the relation between ball acceleration and body angles. It proved that if the body preserve a nonzero angle, the ball should have continuous acceleration. But neither dynamic system could have constant acceleration forever and acceleration would converge to zero in a time. So the question is when does it become zero and consequently what is the Ballbot velocity in that time? To answer this question the dynamic equation is studied in the following. According to Eq.1 the accelerations would be:

$$\ddot{q} = M(q_s)^{-1}\,[\,C(q_s,\dot{q}_s)\dot{q} + G(q_s) + D(\dot{q}_x) - \Gamma(q_s)\tau] \qquad (3)$$

In the steady state (SS) time, $\ddot{q} \rightarrow 0$ due to the friction term and bounded input forces, hence it can be written:

$$C(q_s,\dot{q}_s)\dot{q} + G(q_s) = -D(\dot{q}_x) + \Gamma(q_s)\tau \qquad (4)$$

The equation is the general constraint of the Ballbot and it is more accurate than the constraints attained by simple 2D models. This equation is investigated to catch significant information which lids us to trajectory generation.

Assuming the Ballbot is controlled and has catch specific angles in SS time. Then the $\dot{q}_s$ should be zero in SS time, furthermore, any changes in body angles would accelerate the ball [3] and on the other side, it mentioned that the ball acceleration is zero in SS time, so the $\dot{q}_s$ is zero and there would be no change in the q_s. Thereby, the Eq. 4 makes a system of equations consist of 8 unknowns:

$$q_s \in \mathbb{R}^{3\times1}, \qquad \dot{q}_x \in \mathbb{R}^{2\times1}, \qquad \tau \in \mathbb{R}^{3\times1}$$

and 5 equations. So this system of equation is unsolvable. But he ϑ_z may be relaxed and assumed to zero, because the Ballbot is symmetric around z axis and being the Ballbot in any ϑ_z angle, wouldn't affect its performance and activity. Hence, an unknown is relaxed and we can get three important information by solving the equ.4 in three ways:

1) Determining desired $\dot{q}_x$ and assuming $\vartheta_z = 0$, causes the Eq. 4 to be solved for needed torque and configuration angles to get to the desired $\dot{q}_x$.

2) Determining desired q_s, asumming $\vartheta_z = 0$ and solving Eq. 4 would show maximum reachable ball velocity $\dot{q}_s$ by that angle with the torque needed.

3) Determining three torques and solving Eq.4 would show maximum reachable ball velocity $\dot{q}_s$ by the torques and it would occured in the solved body angles q_s.

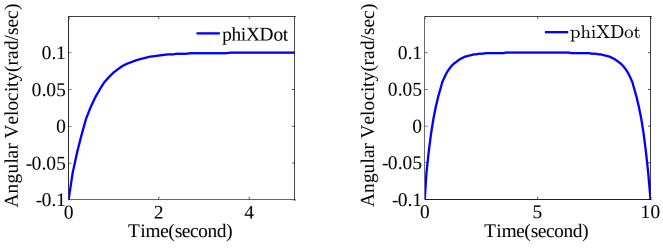
These information could help Ballbot designer to change its parameter in order to reach to a desired performance. Furthermore, they would guide us to plan an accurate trajectory which is the next section subject.

III. GENERATE ACCURATE TRAJECTORY

Trajectory generation for motion from an initial configuration to a desired final configuration is an important task. If the trajectory is accurate and the controller could follow it, the result state errors degrades and there are less needs to compensate the errors.

A. General movement approximation

It is shown that, the Ballbot body should catch specific angles in order to get to a desired constant ball velocity, In addition, the physical understanding of the ballbot behavior indicates that leaning the body to any direction is done by ball actuating. To lean forward, first an abrupt negative acceleration is imposed on the ball by the wheels to get to the desired body angles. Consequently, the wheels rotation direction changes to make the Ballbot keep the desired body angles which causes positive acceleration to the ball. This acceleration leads to the desired ball velocity $\dot{\varphi}_d$. It constitutes the ball velocity trajectory like Fig. 3(a). This ball velocity is well approximated by an exponential term plus an offset term:



(a) Constant ball velocity *(b) P. to P. ball velocity*

Fig. 3. Ball velocity

$$\dot{\varphi}_{apr}(t) = c_1 + c_2 e^{\alpha t} \qquad (5)$$

It is analyzed that the equation should converge to $\dot{\varphi}_d$ as $t \rightarrow \infty$ hence:

$$\dot{\varphi}_{apr}(\infty) = c_1 = \dot{\varphi}_d$$

It would be a well approximation if we assume:

$$\dot{\varphi}_{apr}(0) = c_2 \cong -2\,\dot{\varphi}_d$$

Hence the equation of approximated ball velocity become:

$$\dot{\varphi}_{apr}(\mathrm{t}) = \dot{\varphi}_d - 2\,\dot{\varphi}_d e^{\alpha t} \qquad (6)$$

the α is the exponential speed rate of reaching the ball to the desired velocity. It depends on ground friction parameters and ball composition materials.

B. Linear path generation

Any linear path in a two dimensional plane could be decomposed to its position primitives. In a XY ground plane, following both primitives $x(t)$ and $y(t)$, would lead us to the desired point in a certain time. Hence we have investigated two path primitives in X and Y axes according to reaching any linear path. By this definition we have proposed a symmetric trajectory generation approach for both axis.

In order to reach to a desired position from the initial rest configuration with a constant velocity, the ball should be accelerated. Furthermore, it should be deaccelerated at a time to cause the zero ball velocity. The acceleration and deacceleration of the ball approximately constitute a point to point (P. to P.) velocity curve like Fig. 3(b). To calculate the final position, we can integral half of the curve by Eq. 6 and multiply it by two:

$$\begin{aligned}\varphi(t) &= 2\int_0^{\frac{t}{2}} \dot{\varphi}_d - 2\,\dot{\varphi}_d\,e^{\alpha t} \\ &= \dot{\varphi}_d\,(t + \tfrac{2}{\alpha} - \tfrac{2}{\alpha}\,e^{\frac{\alpha t}{2}}) \qquad (7)\end{aligned}$$

The above equation is solved for the velocity is needed to pass a certain distance in a specific time t. This ball velocity is catch by corresponding body angles which derived from numerical solution of Eq. 4. Otherwise, the Eq. 6 is solved to find the specific time needed for passing a distance by a certain ball velocity.

C. Generalized path generation

Any linear paths became achievable by its velocity in accordance with each axis. It could be well extended to any nonlinear path. To do so, there is a need for having a decoupled curvature motion in XY axis which represents ball velocity in each axis. A circle path is an example of such a curvature. It is obvious that the ball velocities for following this curvature in XY planes are:

$$\dot{\varphi}_x(t) = \mathrm{p}\, cos(\omega\, t) \qquad 0 < t < 2\pi$$
$$\dot{\varphi}_y(t) = \mathrm{p}\, \sin(\omega\, t) \qquad 0 < t < 2\pi \tag{8}$$

Where p is a constant coefficient which determines the desired radius of the path circle. The Eq. 8 is integraled to calculate the circle position primitives:

$$\varphi_x(t) = \int_0^t p\cos(\omega\, t)\, dt = \frac{p}{\omega}\sin(\omega\, t)$$
$$\varphi_y(t) = \int_0^t p\sin(\omega\, t)\, dt = \frac{p}{\omega}(1 - cos(\omega\, t)) \tag{9}$$

As the integral shows, it constitute a circle. But to following a circle, the body angles command should be given as:

$$\vartheta_x(t) = \beta\, cos(\omega\, t) \qquad 0 < t < 2\pi$$
$$\vartheta_y(t) = \beta\, sin(\omega\, t) \qquad 0 < t < 2\pi \tag{10}$$

This is the same as Eq. 8 with the difference β coefficient that should be solved by using Eq. 4. Although the non-constant ball velocity violates the $\ddot{q} = 0$ (the assumption to derive Eq. 4), the simulation results shows acceptable results.

IV. Trajectory Tracking Control

The entire trajectory planning procedure is presented in this paper is assumed there is a controller which can track the desired body angles. Any errors in body angles tracking would cause position tracking errors due to the Ballbot behavior. Hence it is vital to have a nonlinear controller as opposed to linear controllers proposed in [3][5] which can handle all the trajectories and robot nonlinearities.

As the 3D model with added friction covers all the dynamic aspects of the Ballbot such as coupling effects and its high nonlinearity, it could be well used to design an inverse dynamic controller. But using a full state inverse dynamic is impossible due to the underactuation, so a collocated partial feedback linearization (CPFL) is used [13].

Writing equation of motion in a matrix form, yields:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_s \\ \ddot{q}_x \end{bmatrix} + \begin{bmatrix} \varphi_s \\ \varphi_x \end{bmatrix} = \begin{bmatrix} \Gamma(q_s) \\ \Gamma(q_x) \end{bmatrix} \tau \tag{11}$$

$$\ddot{q}_x = -H_{22}^{-1}\, [\, H_{21}\, \ddot{q}_s\, \varphi_x - \Gamma(q_x)\, \tau] \tag{12}$$

In order to linearize the actuated terms by CPFL, Eq. 12 is substituted for the $\ddot{q}_x$ in the Eq. 11:

$$(H_{11} - \mu\, H_{21})\ddot{q}_s - \mu\, \varphi_x + \varphi_s = (-\,\mu\, \Gamma_x + \Gamma_s)\, \tau \tag{13}$$

Where $\mu = H_{12}H_{22}^{-1}$. The following control law is used to make a double integral equation:

$$\tau = (-\mu\, \Gamma_s + \Gamma_x)^{-1}(H_{11} - \mu\, H_{21})a_q$$
$$-(-\mu\, \Gamma_s + \Gamma_x)^{-1}(-\mu\, \Gamma_s + \Gamma_x) \tag{14}$$

By the above control law, the body configurations is controlled and setting a_q to a PD terms would result tracking. The a_q is:

$$a_q = \ddot{q}_{sd} - K_d\, (\dot{q}_s - \dot{q}_{sd}) - K_p\, (q_s - q_{sd}) \tag{15}$$

Where $\ddot{q}_{sd}, \dot{q}_{sd}, q_{sd}$ are desired body accelerations, velocites and angles, respectively. The closed loop systems would be:

$$\ddot{\tilde{q}}_s + K_d\, \dot{\tilde{q}}_s + K_d\, \tilde{q}_s = 0 \tag{16}$$

Where $\tilde{q}_s = (q_s - q_{sd})$. By setting the K_d, K_p to any positive definite matrix, the tracking of the desired body angles would occur. The Fig. 4. shows trajectory generation and tracking controller diagram. As can be seen there is no position feedback it is controlled open loop.

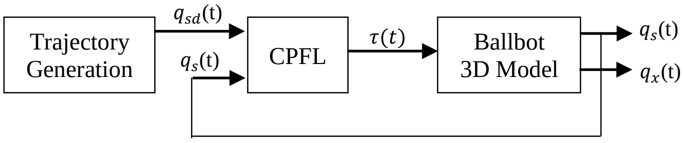


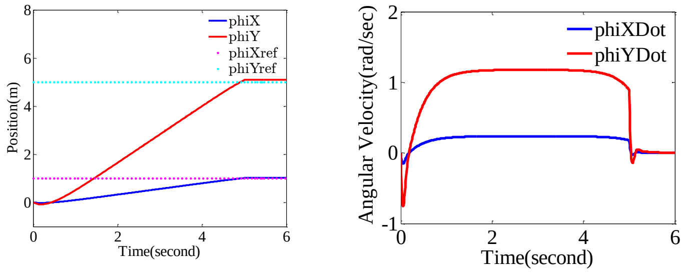
Fig. 4. Trajectory generation and tracking controller

V. Simulation Results

In this section, the proposed strategy is examined by linear point to point (P. to P.) motion and circle following scenarios. The simulation has used CPFL to control body angles and there is not any position feedbacks.

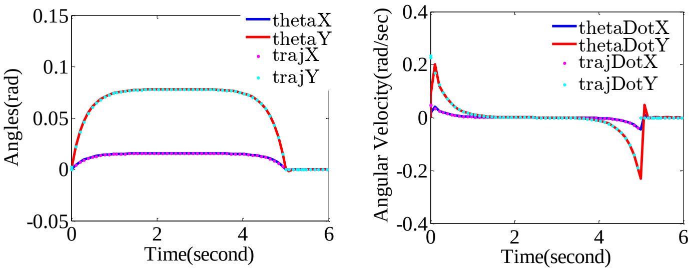
A. Linear path following

The first test is following a desired line in a specific time. The Ballbot starts from the initial and catch a ball position of $(\varphi_{xd}, \varphi_{yd}) = (1\,, 5)\ (m)$ in five seconds. By solving Eq. 7 to calculate the ball velocity primitives, the result would be $(\dot{\varphi}_{xd}, \dot{\varphi}_{yd}) = (0.235, 1.1773)\ (rad/\mathrm{sec})$.Where the α which



(a) Ball position *(b) Ball velocity*

Fig. 5



(a) Desired angles vs body angles *(b) desired vel. vs body vel.*

Fig. 6

is related to friction term is considered to be 3, equal to the viscous coefficient in [9]. Having the calculated ball velocity, the nonlinear Eq. 4 is solved by *fsolve* function in MATLAB in order to find the desired body angles which causes the ball to get to the desired velocity. The solved angles are $(\vartheta_{xd}, \vartheta_{yd}) = (0.0156\,,\ 0.0781)\ radians$. To avoid an abrupt set point change, we used an exponential body trajectory to get to the ϑ_d:

$$\begin{aligned} \vartheta_d &= \vartheta_d - \mathrm{e}^{-\mathrm{t}} && 0 < t < 5/2 \\ \vartheta_d &= \vartheta_d - \mathrm{e}^{-(5-\mathrm{t})} && 5/2 < t < 5 \end{aligned} \qquad (17)$$

The CPFL is used to follow ϑ_{xd} and ϑ_{yd}. The result of tracking the body trajectory for a point to point motion is shown in Fig. . The figure shows that the CPFL makes tracking. Desired body angles tracking would cause the ball to go to the desired planed position. Fig. indicates the result of ball position.

The final position are $(\varphi_x, \varphi_y) = (1.02\,,\ 5.1)\ (m)$ which shows 0.2 present tracking error. It should be mentioned that no position feedback is used and this position tracking accuracy is only gained from the body angles control.

B. *Generalized path following*

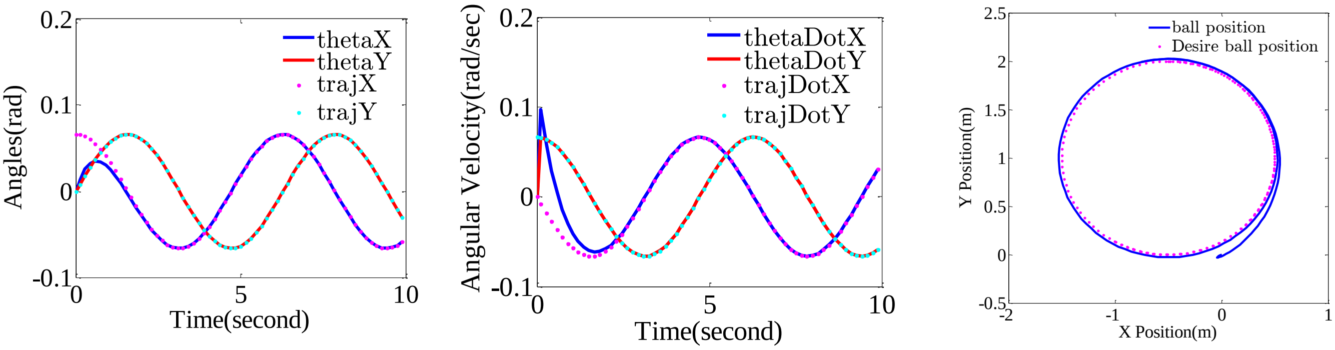
Here, the goal is to make Ballbot follow a circular path. Eq. 7 shows the ball velocity which generate a circle path. The p parameter should be calculated according to the circle parameters. The desired circle radius is one meter and the full circle passed in π seconds. So the $\omega = p = \frac{p}{\omega} = 1(rad/sec)$. Nonlinear Eq. 4 is solved by $\dot{\varphi}_d = p/\omega = 1(rad/sec)$ as the ball velocity, the result has derived $\beta = 0.0385(rad/sec)$ as the body angle.

It can be seen in Fig. (a , b), the CPFL controller makes the body angles and angular velocity to track the sin function and it's velocity. Fig. (c) shows the circle motion approximately tracked. The passed circle radius is 1.02 which shows 2% error.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, a novel accurate trajectory generation has been introduced for Ballbot robot. The approach stemmed from the dynamic constraint which is resulted from considering the 3D Ballbot model with added friction. The coupled dynamic constraints has been investigated deeply and three important information has been proposed. The constraint is shown the relation between the ball position and body angles and based on the accurate relation, a trajectory generation for linear and circular motion has been presented which decoupled the trajectory generation in each XY plane.

In order to follow the produced trajectory, a collocated partial feedback linearization in the presence of input coupling is calculated for Ballbot. The combination of controller and trajectory generator has been simulated on the model for linear and circular paths and the results has been shown 2% tracking error. This is an outstanding result, especially when there is no



(a) Desired angles vs body angles *(b) Desired body Vel. Vs body Vel.* *(c) Ball Pos. vs Desired ball Pos.*

Fig. 7

position feedback information. It shows the approach reliability against other approach which utilized feedback position to compensate their trajectory generation inaccuracy.

The proposed approach was based on α parameter which depends on floor friction. Attempt will be made to adaptively identify the parameter in different floors condition. Furthermore, the trajectory which is generated for circular path could be more generalized to any curvature paths.

References

[1] M. W. Spong, "The control of underactuated mechanical systems," *First international conference on mechatronics*. 1994.

[2] U. Nagarajan, G. Kantor, and R. Hollis, "The ballbot: An omnidirectional balancing mobile robot," *Int. J. Rob. Res.*, vol. 33, pp. 917–930, 2014.

[3] U. Nagarajan and R. Hollis, "Shape space planner for shape-accelerated balancing mobile robots," *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1323–1341, 2013.

[4] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball," *2008 Int. Conf. Control. Autom. Syst. ICCAS 2008*, pp. 433–438, 2008.

[5] P. Fankhauser and C. Gwerder, "Modeling and Control of a ballbot," Ph.D. dissertation, Bachelor Thesis ETH Zurich, 2010.

[6] M. Akbari, H. Z. Kheibari, A. Sadat, M. Nejad, S. Danesh, I. Of, and H. Education, "Timing Belt Gearbox In Ballbot Robot," *2013 First RSI/ISM Int. Conf. Robot. Mechatronics*, 2013.

[7] H. Z. Kheibari, M. Akbari, and a. S. M. Nejad, "Constructing IMU based On ADC and sensors calibration for Ballbot," *2013 First RSI/ISM Int. Conf. Robot. Mechatronics*, pp. 449–454, 2013.

[8] U. Nagarajan, "Dynamic Constraint-based Optimal Shape Trajectory Planner for Shape-Accelerated Underactuated Balancing Systems.," *Robot. Sci. Syst.*, 2010.

[9] M. Shomin and R. Hollis, "Differentially flat trajectory generation for a dynamically stable mobile robot," *Robot. Autom. (ICRA), 2013 IEEE Int. Conf. on. IEEE*, pp. 4467–4472, 2013.

[10] M. Shomin and R. Hollis, "Fast, dynamic trajectory planning for a dynamically stable mobile robot," *2014 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, no. Iros, pp. 3636–3641, 2014.

[11] and U. S. Inal, Ali Nail, Omer Morgul, "Path Following with An Underactuated Self-Balancing Spherical-Wheel Mobile Robot," *Adv. Robot. (ICAR), 2015 Int. Conf. on. IEEE*, 2015.

[12] U. Nagarajan, G. Kantor, and R. Hollis, "Trajectory Planning and Control of a Dynamically Stable Single Spherical Wheeled Mobile Robot," *IEEE Int. Conf. Robot. Autom. ICRA 2009*, pp. 3743 – 3748, 2009.

[13] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," *Intell. Robot. Syst. '94. "Advanced Robot. Syst. Real World", IROS '94. Proc. IEEE/RSJ/GI Int. Conf.*, vol. 1, pp. 314–321 vol.1, 1994.