2020 17th International Conference on Ubiquitous Robots (UR)
June 22-26, 2020. Kyoto, Japan

# Contact Force based Balancing and Tracking Control of a Ballbot using Projected Task Space Dynamics with Inequality Constraints

Joonhee Jo[1,2] and Yonghwan Oh[1]

*Abstract*— **Since the ball-balancing robot has been researched, many kinds of platforms and control methods have been developed for the ballbot. Even though the behavior of the robot has been achieved by previous studies, there are few studies considering the contact forces between the robot and ball. In this paper, we propose balancing and tracking control of the ballbot, with unilateral constraints. Using an adequate task transformation matrix, the task space dynamics can be divided into the dynamics of the robot and the ball respectively. This decomposition has advantages to obtain the input torque through the ball task dynamics with constraint forces. Through the proposed formalism, the contact force can be computed from the ball task space dynamics with the quadratic programming(QP) with inequality constraints such as unilateral constraints and friction constraints. The obtained contact force is used in the robot task space dynamics to get control input. In addition, the balancing force is computed using CoM reflex as a reference. Hence, using the synthesized controller, the contact force based balancing and tracking control simulation is performed.**

## I. INTRODUCTION

The mobility of the robot has been studied in a variety of robots which has multiple legs/wheels including one leg or wheel. A legged robot kinematically linked by limbs is widely known to be more versatile and can traverse various terrains with walking and jumping although it is more complex and consumes more power. On the other hand, wheeled robot systems are widely considered implying difficulties in navigating over obstacles, sharp declines or low friction areas however it is much simpler in design and control.

A balancing ability, another issue, has been studied with the wheeled mobile robot [1], [2]. Simply, the system is shown to be more stable with more wheels because the stability region represented by the supporting polygon is bigger compared to fewer wheels.

The balancing and navigating a ball-bot, which is equipped with omnidirectional wheels on a ball, is extremely challenging because the system has a small stability margin. The researches have started from CMU [3] which has inverse mouse-ball type robot since 2005. Then, the ballbot with omnidirectional wheels has developed in many kinds of research [4] and it was shown better performance.

New mechanisms led to the development of the ballbot system performance, it still has challenging issues in the tracking of motion and balancing itself. In the beginning, the researches designed the robot as multiple 2D models or inverted pendulum [5], and made the controllers in each plane using PD/PID controller [6], and widely LQR [7] for achieving the behavior of the ballbot. Since 2011, the system as a 3D model has been studied with different point linearization [8], inverse dynamics control [9] and LQR [10], [11]. In addition to this, fuzzy sliding mode control [12] and energy-shaping control [13] has been performed in the simulation.

Although the control schemes got better performance, there are difficulties with the contact between the ball and the ground in the system. In the case of ground contact, there is no way to measure the contact information and other difficulty arises due to the unilateral contact forces [14]. It should be carefully considered in modelling. In addition, the system is rigid so the frictional point contact with rolling constraint [15] is used to assume slip-free in the formulation. In the case of a grasp-able manipulator, it has bilateral constraints at the contact with an environment, however, the ballbot has unilateral constraints at the ball-wheel contact similar to humanoid foot contact [16]. To maintain the contact, the relative velocity of the ball-wheel contact should remain in the friction cone and the velocity of the ball-ground contact should be zero to avoid slipping.

In this paper, 3-dimensional modelling of an underactuated ballbot system and tracking control based on the projected task space dynamics with contact constraints using quadratic programming(QP) are introduced. The robot has 9 degrees of freedom(DoF), 4 DoF tasks and 9 contact constraints. Since the ball-ground contact is not a measurable force, the null space projection of the ground contact constraint is used to obtain a projected dynamics. In addition, properly chosen tasks are used to form the task space dynamics and the task space dynamics are decomposed into the dynamics of the robot and the ball. Through the ball task space dynamics, the optimal contact force is computed using QP to solve inequality constraints such as friction, omni-wheel constraint, and unilateral constraints because the solution is not unique.

The paper derives equation of motion of the ballbot system with contact constraints in section II and task space dynamics is formulated in section II-C for tracking and balancing control. In addition, the optimal contact force is obtained from quadratic programming(QP) with inequality constraints and control input is formed with the task force. Using the proposed controller, the simulation is performed to establish the improved performance and experimental results are illustrated in section IV. Finally, we characterize the

[1]J.-H. Jo and Y. Oh is with Center for Intelligent & Interactive Robotics, Korea Institute of Science and Technology (KIST), Seoul, 136-791, Korea (email: jhjo@kist.re.kr and oyh@kist.re.kr )

[2]Department of HCI & Robotics, University of Science and Technology(UST), Daejeon, 305-350, Korea

978-1-7281-5714-6/20/$31.00 ©2020 IEEE

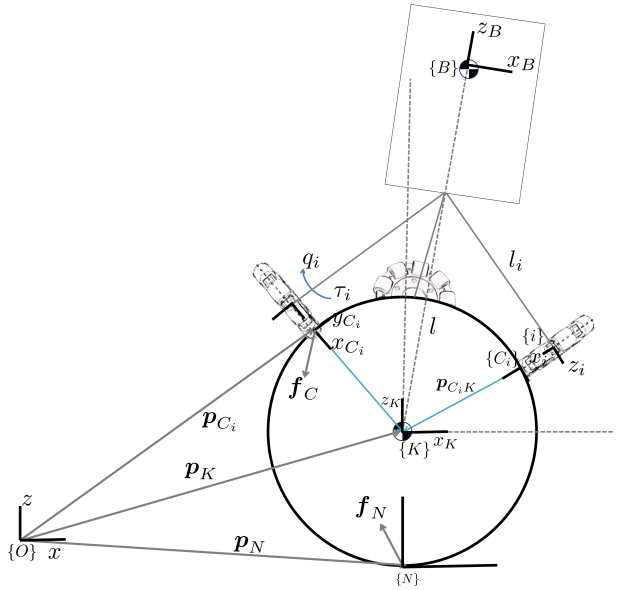



Fig. 1. A ball-balancing robot. Three omni-directional wheels are equipped and each contacts are considered as frictional point contact. {B} is the body frame, {i} is the wheel frame, {C} is the contact frame, {K} is the ball frame, {N} is the ground frame and {O} is the inertial frame.

results in the conclusion in section V.

## II. Ballbot Modelling and Problem Formulation

For a ballbot which has wheels with 3 active degrees of freedom(DoF) and a floating body with 6 DoF, a velocity of generalized coordinates of a body and a ball is defined as

$$\dot{\boldsymbol{\xi}}_B = \begin{bmatrix} \dot{\boldsymbol{p}}_B \\ \boldsymbol{\omega}_B \\ \dot{\boldsymbol{q}} \end{bmatrix} \in \Re^9, \quad \dot{\boldsymbol{\xi}}_K = \begin{bmatrix} \dot{\boldsymbol{p}}_K \\ \boldsymbol{\omega}_K \end{bmatrix} \in \Re^6 \tag{1}$$

where $\dot{\boldsymbol{p}}_B, \boldsymbol{\omega}_B \in \Re^3$ are linear and angular velocities of the floating body and $\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^T \in \Re^3$ are wheel joint velocities. $\dot{\boldsymbol{p}}_K, \boldsymbol{\omega}_K \in \Re^3$ are linear and angular velocities of the ball.

### A. A ballbot model

The equations of motion(EoM) of the ballbot system is expressed as

$$\begin{cases} \boldsymbol{M}_B^{'}\ddot{\boldsymbol{\xi}}_B + \boldsymbol{C}_B^{'}\dot{\boldsymbol{\xi}}_B + \boldsymbol{g}_B^{'} = \boldsymbol{B}_B^{'}\boldsymbol{\tau} + \boldsymbol{J}_{CB}^T\boldsymbol{f}_C \\ \boldsymbol{M}_K^{'}\ddot{\boldsymbol{\xi}}_K + \boldsymbol{C}_K^{'}\dot{\boldsymbol{\xi}}_K + \boldsymbol{g}_K^{'} = -\boldsymbol{G}_{CK}^T\boldsymbol{f}_C + \boldsymbol{G}_{NK}^T\boldsymbol{f}_N \\ \dot{\boldsymbol{p}}_C = \boldsymbol{J}_{CB}\dot{\boldsymbol{\xi}}_B = \boldsymbol{G}_{CK}\dot{\boldsymbol{\xi}}_K \\ \dot{\boldsymbol{p}}_N = \boldsymbol{G}_{NK}\dot{\boldsymbol{\xi}}_K = \boldsymbol{0} \end{cases} \tag{2}$$

where $\boldsymbol{M}_B^{'}, \boldsymbol{C}_B^{'} \in \Re^{9\times9}$ are an inertia and a Coriolis and Centrifugal matrices of the body and $\boldsymbol{g}_B^{'} \in \Re^9$ is a gravitational force vector of the body. $\boldsymbol{B}_B^{'} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{E}_3 \end{bmatrix}^T \in \Re^{9\times3}$ is a selection matrix and $\boldsymbol{E}_3$ is a 3-dimensional identity matrix. $\boldsymbol{f}_C = \begin{bmatrix} \boldsymbol{f}_{C_1}^T & \boldsymbol{f}_{C_2}^T & \boldsymbol{f}_{C_3}^T \end{bmatrix}^T \in \Re^9$ is the contact force vector between ball and wheel. In addition, $\boldsymbol{M}_K^{'} = \begin{bmatrix} m_K\boldsymbol{E}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_K \end{bmatrix}, \boldsymbol{C}_K^{'} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\omega}_K \times \boldsymbol{I}_K \end{bmatrix} \in \Re^{6\times6}$ are an inertia and a Coriolis and Centrifugal matrices of the ball and $\boldsymbol{g}_K^{'} \in \Re^9$ is a gravitational force vector of the ball. $\boldsymbol{G}_{CK}^T = \begin{bmatrix} \boldsymbol{G}_{C_1K}^T & \boldsymbol{G}_{C_2K}^T & \boldsymbol{G}_{C_3K}^T \end{bmatrix} \in \Re^{6\times9}, \boldsymbol{G}_{NK}^T \in \Re^{6\times3}$ is the contact map of the velocity of the ball-wheel contact and ball-ground contact. $\boldsymbol{f}_N \in \Re^3$ is the ball-ground contact force. In addition, $\dot{\boldsymbol{p}}_C \in \Re^9$ is a velocity at the contact point and $\boldsymbol{J}_{CB}$ is the ball-wheel contact Jacobian with joint angles and body posture. Additionally, $\dot{\boldsymbol{p}}_N \in \Re^3$ is a ground contact velocity.

In this model, the ground velocity $\dot{\boldsymbol{p}}_N \in \Re^3$ is assumed to be zero with the assumption that there is no slip between the ground and the ball because $\boldsymbol{f}_N$ is unmeasurable and it makes controller design difficult. If the ball-ground contact force is ejected from the dynamics, much essential form of the robot system dynamics can be obtained. Hence, the equation of motion is reformed using a null space projector of a ball-ground contact constraint.

### B. Projected dynmaics of the robot system

The system dynamics in Eq. (2) can be described as

$$\begin{cases} \underbrace{\begin{bmatrix} \boldsymbol{M}_B^{'} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_K^{'} \end{bmatrix}}_{\boldsymbol{M}^{'}}\ddot{\boldsymbol{\xi}} + \underbrace{\begin{bmatrix} \boldsymbol{C}_B^{'} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_K^{'} \end{bmatrix}}_{\boldsymbol{C}^{'}}\dot{\boldsymbol{\xi}} + \underbrace{\begin{bmatrix} \boldsymbol{g}_B^{'} \\ \boldsymbol{g}_K^{'} \end{bmatrix}}_{\boldsymbol{g}^{'}} \\ = \underbrace{\begin{bmatrix} \boldsymbol{B}_B^{'} \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{B}^{'}}\boldsymbol{\tau} + \underbrace{\begin{bmatrix} \boldsymbol{J}_{CB}^T \\ -\boldsymbol{G}_{CK}^{'T} \end{bmatrix}}_{\boldsymbol{G}_C^{'T}}\boldsymbol{f}_C + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{G}_{NK}^{'T} \end{bmatrix}}_{\boldsymbol{G}_N^{'T}}\boldsymbol{f}_N \end{cases} \tag{3}$$

where $\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{\boldsymbol{\xi}}_B & \dot{\boldsymbol{\xi}}_K \end{bmatrix}^T \in \Re^{15}$ is the generalized coordinates of the whole system. The ball-ground contact map is $\boldsymbol{G}_N^{T'}$ and is used to compute projection matrix

$$\boldsymbol{P}_N = \boldsymbol{I} - \boldsymbol{G}_N^{\dagger'}\boldsymbol{G}_N^{'} = \begin{bmatrix} \boldsymbol{E}_9 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_N^{'} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E}_9 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}_6 - \boldsymbol{G}_{NK}^{'\dagger}\boldsymbol{G}_{NK}^{'} \end{bmatrix}. \tag{4}$$

The projected EoM with ball-wheel constraints is obtained from premultiplying $\boldsymbol{P}_N$ by Eq. (3),

$$\begin{cases} \underbrace{\begin{bmatrix} \boldsymbol{M}_B^{'} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_N^{'}\boldsymbol{M}_K^{'} \end{bmatrix}}_{\bar{\boldsymbol{M}}}\ddot{\boldsymbol{\xi}} + \underbrace{\begin{bmatrix} \boldsymbol{C}_B^{'} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_N^{'}\boldsymbol{C}_K^{'} \end{bmatrix}}_{\bar{\boldsymbol{C}}}\dot{\boldsymbol{\xi}} + \underbrace{\begin{bmatrix} \boldsymbol{g}_B^{'} \\ \boldsymbol{0} \end{bmatrix}}_{\bar{\boldsymbol{g}}} \\ = \underbrace{\begin{bmatrix} \boldsymbol{B}_B^{'} \\ \boldsymbol{0} \end{bmatrix}}_{\bar{\boldsymbol{B}}}\boldsymbol{\tau} + \underbrace{\begin{bmatrix} \boldsymbol{J}_{CB}^T \\ -\boldsymbol{P}_N^{'}\boldsymbol{G}_{CK}^{'T} \end{bmatrix}}_{\bar{\boldsymbol{G}}_C^T}\boldsymbol{f}_C \end{cases} \tag{5}$$

and it becomes

$$\begin{cases} \bar{\boldsymbol{M}}\ddot{\boldsymbol{\xi}} + \bar{\boldsymbol{C}}\dot{\boldsymbol{\xi}} + \bar{\boldsymbol{g}} = \bar{\boldsymbol{B}}\boldsymbol{\tau} + \bar{\boldsymbol{G}}_C^T\boldsymbol{f}_C \\ \dot{\boldsymbol{p}}_C = \boldsymbol{J}_{CB}\dot{\boldsymbol{\xi}}_B = \boldsymbol{G}_{CK}\dot{\boldsymbol{\xi}}_K. \end{cases} \tag{6}$$

where $\bar{\boldsymbol{M}}, \bar{\boldsymbol{C}} \in \Re^{15\times15}$ are an inertia and a Coriolis and Centrifugal matrices of total system with 15 DoF and $\bar{\boldsymbol{g}} \in \Re^{15}$ is a gravitational force vector. $\bar{\boldsymbol{G}}_C^T \in \Re^{15\times9}$ is the ball-wheel contact force map. This projected EoM does not contain ball-ground contact part due to the null space projection.

### C. Task space representation

The task of the ballbot is complicated because the motion of the robot and the ball is dependant on each other. The change of the orientation of x-axis and y-axis leads to the change of x-y position of the robot and the ball respectively. Hence, x-y motion and z-axis rotation are only independent relations in the system. Therefore, it is important to define the task properly. In this paper, a balancing task is focused so that x-y orientation is intuitively proper as a task. In that sense, the common way of defining the task variable is choosing the body orientation and the ball position [8] as minimal representation. However, defining tasks with body frame variable makes the task space dynamics not to relate with the joint variable. Because the body position/orientation are obtained from the sensor such as IMU and that information does not contain the joint-body relation. If the joint-body related task is defined, the dynamics formalism can be derived easily with decomposed task dynamics discussed below. Hence, the x-y position of robot CoM $\boldsymbol{p}_G$ is proper compared to the orientation for a motion task and z-axis orientation is neglected because it is not critical to the balancing of the robot.

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{x}_K \\ \dot{y}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_{G,T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_{K,T} \end{bmatrix} \dot{\boldsymbol{\xi}} = \boldsymbol{T}\dot{\boldsymbol{\xi}} \tag{7}$$

where $\boldsymbol{T} \in \Re^{4\times 15}$ is the transformation matrix from general coordinates to task variables. $\boldsymbol{B}_{G,T} \in \Re^{2\times 9}$ are the xy components of Jacobian from {B} to {G} that includes joint related term and $\boldsymbol{B}_{K,T} \in \Re^{2\times 6}$ is the selection matrix of xy components for linear velocity of the ball. With the task transformation matrix, task space dynamics can be derived using pre-multiplication by $\boldsymbol{T}^{\dagger T}$ and substitution $\dot{\boldsymbol{\xi}}$ with $\boldsymbol{T}^{\dagger}\dot{\boldsymbol{x}}$,

$$\begin{aligned} &\underbrace{\boldsymbol{T}^{\dagger T}\bar{\boldsymbol{M}}\boldsymbol{T}^{\dagger}}_{\boldsymbol{\Lambda}}\ddot{\boldsymbol{x}} + \underbrace{\boldsymbol{T}^{\dagger T}\left(\bar{\boldsymbol{C}} - \bar{\boldsymbol{M}}\boldsymbol{T}^{\dagger}\dot{\boldsymbol{T}}\right)\boldsymbol{T}^{\dagger}}_{\boldsymbol{C}}\dot{\boldsymbol{x}} + \underbrace{\boldsymbol{T}^{\dagger T}\bar{\boldsymbol{g}}}_{\boldsymbol{g}} \\ &\quad = \underbrace{\boldsymbol{T}^{\dagger T}\bar{\boldsymbol{B}}}_{\boldsymbol{B}}\boldsymbol{\tau} + \underbrace{\boldsymbol{T}^{\dagger T}\bar{\boldsymbol{G}}_C^T}_{\boldsymbol{G}_C^T}\boldsymbol{f}_C \\ &\begin{bmatrix} \boldsymbol{\Lambda}_G & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Lambda}_K \end{bmatrix}\ddot{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{C}_G & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\dot{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{g}_G \\ \boldsymbol{0} \end{bmatrix} \\ &\quad = \begin{bmatrix} \boldsymbol{B}_G \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{\tau} + \begin{bmatrix} \boldsymbol{G}_{C,G}^T \\ -\boldsymbol{G}_{C,K}^T \end{bmatrix}\boldsymbol{f}_C. \end{aligned} \tag{8}$$

where $\boldsymbol{\Lambda}_G \in \Re^{2\times 2}$ represents the task robot CoM inertia of the task inertia matrix $\boldsymbol{\Lambda} \in \Re^{4\times 4}$ and $\boldsymbol{\Lambda}_K \in \Re^{2\times 2}$ represents the task ball CoM inertia of the task inertia matrix $\boldsymbol{\Lambda}$. $\boldsymbol{C}_G \in \Re^{2\times 2}$ and $\boldsymbol{T}^{\dagger T}\bar{\boldsymbol{C}}_{C,K}$ is zero because $\boldsymbol{T}^{\dagger T}$ contains linear part of the ball motion and $\bar{\boldsymbol{C}}_{C,K}$ contains orientation part of the ball motion. So, the multiplication of two are equal to zero. The selection matrix $\boldsymbol{B} \in \Re^{4\times 3}$ and $\boldsymbol{B}_G \in \Re^{2\times 3}$. Each dynamics can affect other by contact force $\boldsymbol{f}_C$.

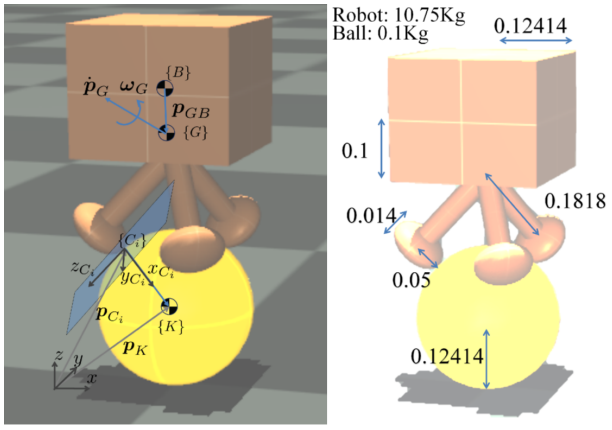



Fig. 2. A ball-balancing robot is modelled in Mujoco. The wheels in the simulation is ellipsoid and have omni-directional friction as omni wheel. $x_{C_i}$ is a normal contact direction, $z_{C_i}$ is a non-frictional direction and $y_{C_i}$ is a friction-effective direction that is tangential to the plane. In addition, a size and specification of ball-balancing robot is described in the right figure.

## III. Controller Design

Since the dynamics of the robot system is projected into the null space of ground contact constraint, only the ball-wheel contact force can be considered in the ball task space dynamics equation, below part in Eq. (8). The contact forces are obtained from quadratic programming(QP) with friction and unilateral constraints.

Hence, the below dynamics is formed as

$$\boldsymbol{f}_K = \boldsymbol{\Lambda}_K\ddot{\boldsymbol{x}}^{ref} = \boldsymbol{G}_{C,K}^T\boldsymbol{f}_C^{QP} \tag{9}$$

where $\ddot{\boldsymbol{x}}^{ref}$ is the reference acceleration.

To compute contact force, QP solver [17] is used.

$$\begin{aligned} &\min \ \frac{1}{2}\left(\triangle\boldsymbol{f}_C^T\boldsymbol{W}_C\triangle\boldsymbol{f}_C + \boldsymbol{f}_C^T\boldsymbol{f}_C\right) \\ &\text{where} \ \ \triangle\boldsymbol{f}_C = \boldsymbol{f}_K - \boldsymbol{G}_{C,K}^T\boldsymbol{f}_C^{QP} \\ &\text{subject to} \ \ f_{C_i,x}^{QP} \geq f_{C_i,min} \\ &\qquad\qquad |\boldsymbol{f}_{C_i,y}| \leq \mu_i\boldsymbol{f}_{C_i,x} \\ &\qquad\qquad f_{C_i,z}^{QP} = 0 \end{aligned} \tag{10}$$

In contact frame, x-axis is normal contact direction and z-axis is omni wheel direction. Hence, y-axis is the tangential direction to move and friction occurs in the direction as described in Fig. 2. The condition of ball-ground is ejected because it is projected out by null space projector.

In addition, the control torque can be derived by the upper equation in Eq. (8).

$$\boldsymbol{\tau} = \boldsymbol{B}_G^W\left(\boldsymbol{\Lambda}_G\ddot{\boldsymbol{x}}^{ref} + \boldsymbol{C}_G\dot{\boldsymbol{x}} + \boldsymbol{g}_G - \boldsymbol{G}_{C,G}^T\boldsymbol{f}_C^{QP}\right) \tag{11}$$

where $\boldsymbol{B}^W$ is the generalized inverse of $\boldsymbol{B}$ which satisfies $\boldsymbol{B}\boldsymbol{B}^W\boldsymbol{B} = \boldsymbol{B}$.

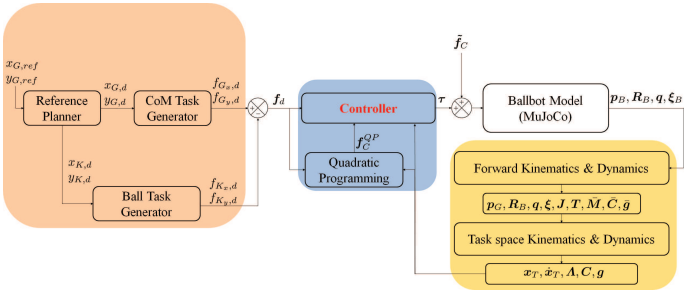



Fig. 3. Block diagram of ball balancing robot system.

TABLE I
TASK TIME SCHEDULE

| | Time(s) | reference $(x, y)$ position $(m)$ |
|---|---|---|
| Task 1 | 0.001 | $(0, -1)$ |
| Task 2 | 3.301 | $(-1, 0)$ |
| Task 3 | 6.601 | $(0, 1)$ |
| Task 4 | 9.901 | $(1, 0)$ |
| Task 5 | 14.301 | cannon ball |

In addition, the null space generalized coordinate force is defined as potential

$$\boldsymbol{f}^* = \boldsymbol{K}_0 \boldsymbol{T}^{+T} \nabla \boldsymbol{h}\left(\boldsymbol{\xi}\right) \tag{12}$$

where $\nabla \boldsymbol{h}\left(\boldsymbol{\xi}\right) = \sum_i \frac{1}{2}\boldsymbol{\eta}_i \left(\boldsymbol{\xi}_d - \boldsymbol{\xi}\right)$. $\boldsymbol{K}_0$ is the potential gain.

Hence, the null space force in the task space is

$$\boldsymbol{f}_0 = \boldsymbol{P}_T \boldsymbol{f}^* \tag{13}$$

where $\boldsymbol{P}_T \in \Re^{n\times n}$ is the null space projector of task transformation matrix and general solution of the system is

$$\boldsymbol{\tau}_c = \boldsymbol{B}_G^W \left(\boldsymbol{\Lambda}_G \ddot{\boldsymbol{x}}^{ref} + \boldsymbol{C}_G \dot{\boldsymbol{x}} + \boldsymbol{g}_G - \boldsymbol{G}_{C,G}^T \boldsymbol{f}_C^{QP} + \boldsymbol{f}_{0,G}\right) \tag{14}$$

where $\boldsymbol{\tau}_c \in \Re^n$ is generalized control torque and the solution contains a particular solution in the first part for a position tracking control and a homogeneous solution $\boldsymbol{f}_0$ and $\boldsymbol{f}_{0,G}$ is robot part force of null space task force. This control force is used to make the robot upright. Overall control block diagram is described in Fig. 3.

## IV. Simulation

The simulation model is developed with the ballbot model as shown in Fig. 2. The Robot has 3 wheels and weight of 10.75 $Kg$ including the body, leg and wheel. In addition, the ball has the radius of 0.12414 $m$ and the weight of 0.1 $Kg$. The system weight is decided to be extremely unstable to show the control performance. The heavier the robot weight is, the higher the CoM position is and it makes the robot unstable easily. ***quadprog++*** [18] is used to solve the QP problem and control rate is 1KHz.

Simulation is performed under the WINDOWS 10 64bit system with the VISUAL STUDIO 2017 C++ compilation. The body position/orientation and velocity are given values as estimated from MuJoCo sensor data. Although both the body and the ball task are chosen to be the control task, the task cannot be assigned independently to each task. For example, the body angle should be determined if the desired ball position is given to follow the desired trajectory of the ball.

### A. Reference For Task Control

The task is tracking the desired position generated from the reference value. The reference of the ball is given by

$$\ddot{\boldsymbol{x}}_K^{ref} = \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_{K,p}\left(\boldsymbol{x}_d - \boldsymbol{x}\right) + \boldsymbol{K}_{K,d}\left(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}\right). \tag{15}$$

In addition, the body should consider the balancing on the ball while the robot tracks the desired trajectory. To following the trajectory with balancing on the ball, CoM reflex [19] from the falling moment is used,

$$\boldsymbol{E}_{CoM} = \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} = \frac{1}{2}\boldsymbol{r}_{CoM}^T \boldsymbol{K}_{CoM} \boldsymbol{r}_{CoM}. \tag{16}$$

where $\boldsymbol{K}_{com,p} = \left[M\boldsymbol{g}\right]^T \left[M\boldsymbol{g}\right]$ and $\boldsymbol{\mu} = \boldsymbol{r}_{CoM} \times \boldsymbol{m}_B\boldsymbol{g}$ is the falling moment. $\boldsymbol{m}_B$, $\boldsymbol{g}$ are the mass of the robot and the gravity vector and $\boldsymbol{r}_{CoM}$ is the radius from the ball position to CoM of the robot in the x-y plane. Using the gradient decent method, the CoM reflex can generate the reflexive motion through the falling moment. Hence, the reference of the body can be

$$\ddot{\boldsymbol{x}}_B^{ref} = \alpha \boldsymbol{K}_{CoM,p} \boldsymbol{r}_{CoM} \tag{17}$$

where $\alpha$ is CoM complex gain. Hence, the reference can be obtained using (15) and (17),

$$\ddot{\boldsymbol{x}}^{ref} = \left[\ddot{\boldsymbol{x}}_B^{T,ref} \ \ddot{\boldsymbol{x}}_K^{T,ref}\right]^T \tag{18}$$

By substituting $\boldsymbol{\tau}$, $\boldsymbol{f}_K$ with $\ddot{\boldsymbol{x}}^{ref}$ in task space dynamics, the closed loop behavior is obtained,

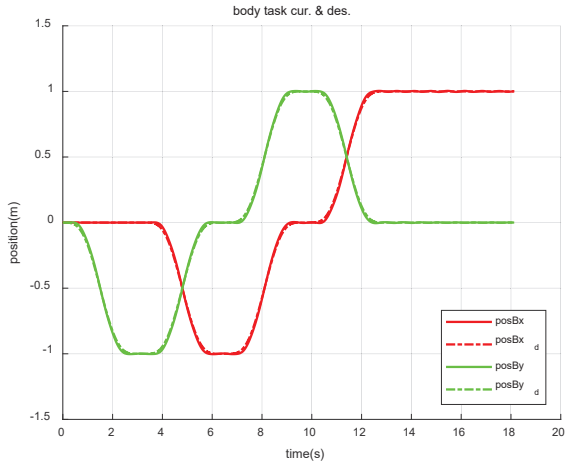
$$\begin{cases} \boldsymbol{\Lambda}_G \ddot{\boldsymbol{e}} = \boldsymbol{G}_{C,G}^T \triangle \boldsymbol{f}_C \\ \boldsymbol{G}_{C,K}^T \triangle \boldsymbol{f}_C = \boldsymbol{\delta}_C \end{cases} \tag{19}$$

If $\boldsymbol{\delta}_C \rightarrow 0$ steady state error will be 0. Moreover, the potential reference $\boldsymbol{\xi}_d$ designed as Eq. 12 has $\boldsymbol{\xi}_{i,d} = 0$ and $\boldsymbol{\eta}_i = 0$ for all $i$ except $\boldsymbol{\eta}_4 = \boldsymbol{\eta}_5 = 1$.

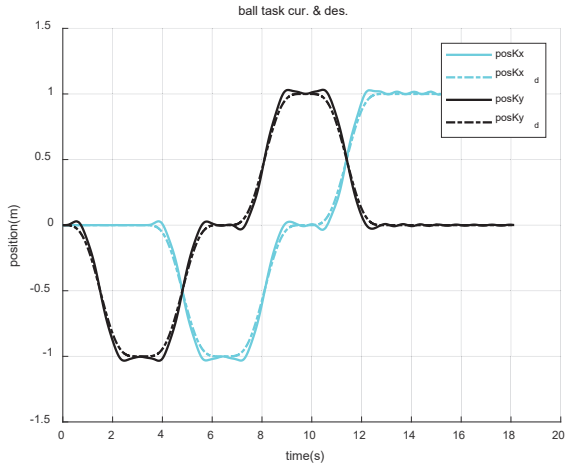
### B. Simulation Results

The task of the robot and the ball is given the same position value because the desired position of the robot and the ball would not be different. Hence, the same tasks are given to the robot and the ball by xy axis motion and last tasks will perform to take a balance. The task is given in the order in Table I. 4 tasks are performed for 3 seconds with 0.3 seconds rest respectively, and the cannon ball is shot at the robot to show the compliant motion against the disturbance. In addition, the desired task profile is generated by using the task reference positions.

As shown in Fig. 4. (a), a solid line is an actual position of the body and a dashed line is the desired position. In




(a) A task and a desired task of the ballbot.




(b) A task and a desired task of and the ball.

Fig. 4. A task and a desired task is compared. A red line is x-axis position, a green line is y-axis position. A solid line is actual task and dashed line is desired task.

addition, a red line is the x-axis and a green line is the y-axis. The actual CoM position of the robot tracks the desired CoM position and there are about 5 mm of steady-state error. In the case of ball task in Fig. 4 (b), a cyan line is the x-axis and a black line is the y-axis of the ball. The solid line is the actual position and the dashed line is desired position also. The steady state error of the ball task is about 2 cm. The simulation results are improved compared to the previous researches in position tracking.

Althogh two tasks have same reference trajectory, the CoM reflex works to interfer between robot task and ball task so the errors in transition are occured as shown in Fig. 4. (b). Instead, the CoM reflex makes the robot motion more faster and dynamic as shown in the ball velocity figure which is addressed in Fig. 5.

The simulation discusses mainly CoM position control so the desired velocity is set to 0. The modelled ballbot system has the big mass body compared to the ball mass and it makes that the ballbot has the small stable region on the ball. It means that the robot can be tilted at a small angle.

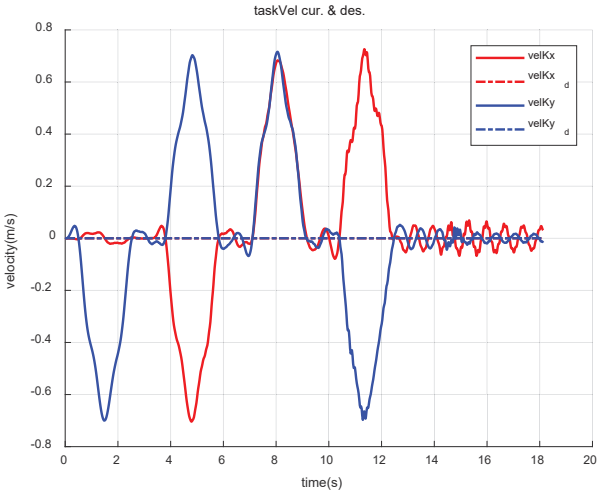



Fig. 5. A ball velocity comparison: A red line is x-axis velocity, a blue line is y-axis velocity.

With that difficulty, the proposed controller can move the ballbot at about 0.7 $m/s$.

In addition to this, multiple cannon balls with 0.01 Kg(10% weight of the ball weight) respectively is shot to the body. The ballbot can endure the disturbance from the cannon ball, although it has a smaller stability region. Described results are shown as in the snapshot Fig. 6.

## V. Conclusion

In this paper, the projected task space dynamics using QP to solve the optimal ball-wheel contact forces and the balancing reference force generation using falling moment is proposed. From the null space of the ball-ground contact constraint, task space dynamics can be obtained which is not related to the ball-ground condition. With adequate tasks, the task dynamics can be obtained and decomposed into the task dynamics of the robot and the ball. In this paper, x-y motion of the CoM is chosen to be the task reference and z-axis rotation is neglected because the z-axis motion is not primary task. Althogh there is 3 system inputs, 4 tasks can be taken because 2 inputs are used to control the x-y motion of the CoM and 1 is used to take a balance.

This formalism has advantages to obtain the contact force for balancing and tracking of the robot using the ball task space dynamics. To obtain the contact force with friction and unilateral constraints, QP is used to compute the optimal solution. Through this formalism, contact force can be synthesized to the controller and the reference force considers the position tracking and the body balancing.

To validate the algorithm, the simulation is performed using Mujoco, the dynamics engine, with omni-wheel like behavior at the ball-wheel contact. The ballbot model has 9 DoF with 3 joints and a body. With the diamond-shpaed motion trajectory, the position tracking performance is discussed and body balancing are tested with multiple cannon ball shooting in the simulation to see the robustness of the

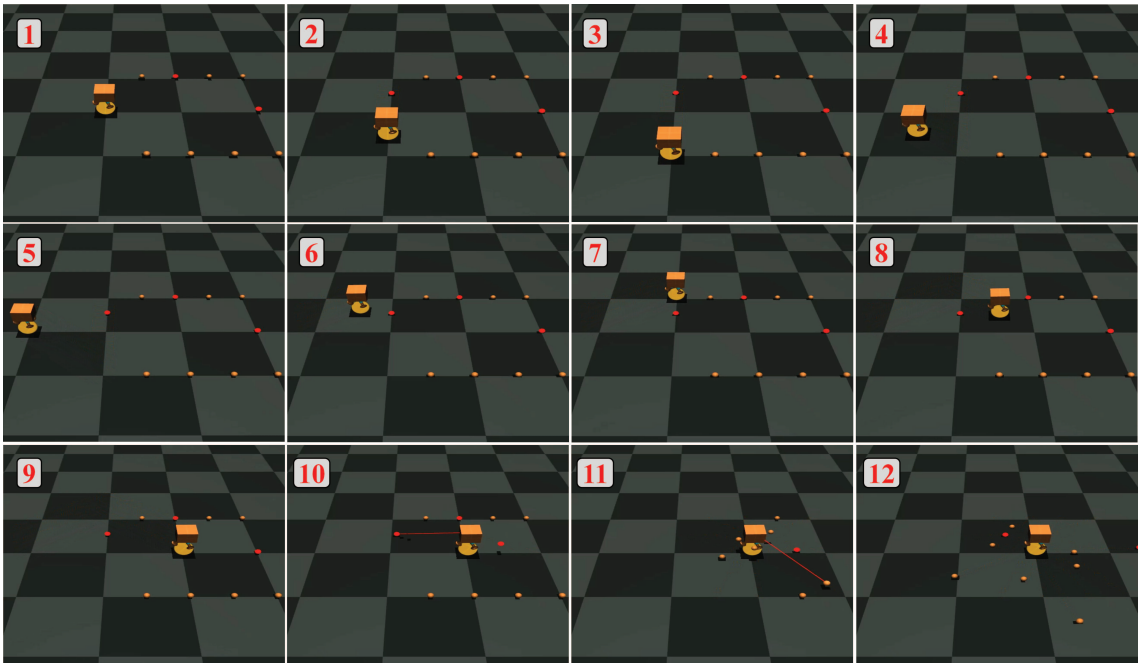



Fig. 6. Snapshots of the experiment. The Snapshot is obtained as in the order of Table I.

control and the adequate results are obtained.

This system has limitations due to the lightweight ball and contact condition. Considering the z-axis orientation of the body is a quite difficult issue because the ball rotates before the body rotates. As further research, the orientation of the body will be discussed.

## REFERENCES


[1] Y.-S. Ha *et al.*, "Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot," *Robotics and autonomous systems*, vol. 17, no. 1-2, pp. 65–80, 1996.

[2] B. Browning, P. E. Rybski, J. Searock, and M. M. Veloso, "Development of a soccer-playing dynamically-balancing mobile robot," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 2. IEEE, 2004, pp. 1752–1757.

[3] G. K. Lauwers, Tom and R. L. Hollis., "One is enough!" *ISRR*, pp. 327–336, 2005.

[4] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball," in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*. IEEE, 2008, pp. 433–438.

[5] M. Kumaga and T. Ochiai, "Development of a robot balanced on a ball—application of passive motion to transport—," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 4106–4111.

[6] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2884–2889.

[7] C.-C. Tsai, M.-H. Juang, C.-K. Chan, C.-W. Liao, and S.-J. Chan, "Self-balancing and position control using multi-loop approach for ball robots," in *2010 International Conference on System Science and Engineering*. IEEE, 2010, pp. 251–256.

[8] E. Pellegrini, K. J. Diepold, R. Dessort, and H. Panzer, "3d-modeling of a robot balancing on a ball," Lehrstuhl für Regelungstechnik, Tech. Rep., 2011.

[9] O. M. A. N. Inal and U. Saranli, "A 3d dynamic model of a spherical wheeled self-balancing robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5381–5386.

[10] P. Fankhauser and C. Gwerder, "Modeling and control of a ballbot," B.S. thesis, Eidgenössische Technische Hochschule Zürich, 2010.

[11] K. van der Blonk, J. Sandee, and A. Stoorvogel, "Modeling and control of a ball-balancing robot," *Faculty of Electrical Engineering, Mathematics and Computer Science*, 2014.

[12] H. Navabi, S. Sadeghnejad, S. Ramezani, and J. Baltes, "Position control of the single spherical wheel mobile robot by using the fuzzy sliding mode controller," *Advances in Fuzzy Systems*, vol. 2017, 2017.

[13] A. C. Satici, A. Donaire, and B. Siciliano, "Intrinsic dynamics and total energy-shaping control of the ballbot system," *International Journal of Control*, vol. 90, no. 12, pp. 2734–2747, 2017.

[14] P. Pagilla and M. Tomizuka, "Contact transition control of nonlinear mechanical systems subject to a unilateral constraint," *Journal of dynamic systems, measurement, and control*, vol. 119, no. 4, pp. 749–759, 1997.

[15] E. Paljug, X. Yun, and V. Kumar, "Control of rolling contacts in multi-arm manipulation," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 441–452, 1994.

[16] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[17] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical programming*, vol. 27, no. 1, pp. 1–33, 1983.

[18] L. D. Gaspero and E. Moyer, "Quadprog++," *http://quadprog.sourceforge.net/*, 2015.

[19] D.-h. Lee, D. T. Tran, and Y. Oh, "An approach toward human-like motion control of a dual arm robot for picking heavy objects," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 1069–1074.