2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Modeling and state estimation of a Micro Ball-balancing Robot using a high yaw-rate dynamic model and an Extended Kalman Filter

Eric Sihite, Daniel Yang, Thomas Bewley, *UCSD Coordinated Robotics Lab*[1]

***Abstract*— The state estimation and control of a ball-balancing robot under high yaw rate is a challenging problem due to its highly nonlinear 3D dynamic. The small size and low-cost components in our Micro Ball-Balancing Robot makes the system inherently very noisy which further increases the complexity of the problem. In order to drive the robot more aggressively such as translating and spinning at the same time, a good state estimator which works well under high yaw rates is required. This paper presents the derivation of a high yaw-rate Ball-Balancing Robot dynamic model and the implementation of said model in an Extended Kalman Filter (EKF) using raw on-board sensor measurements. The EKF using the new model is then compared to a Kalman Filter which uses a linearized dynamic model. The accuracy of the attitude estimates and the controller performance under high yaw rates were verified using a motion capture system.**

I. INTRODUCTION

A Ball-Balancing Robot (BBR) is a robot that balances itself on top of a ball by applying torque through omni-directional wheels such as in [1][2][3] or through an inverse mouse ball mechanism such as in [4][5]. This class of robots features complex and nonlinear 3D dynamics. However, its dynamic equation can be linearized by assuming trivial yaw dynamics, small roll and pitch angles, and small angular speeds. The controls of most existing BBRs were designed based on this linear dynamic model. However, driving the robot at high yaw rates violates the linear model's trivial yaw dynamics assumption and the controller tends to become more unstable during the spin. Our Micro Ball-Balancing Robot (MBBR) [1], shown in Fig. 1, is one of the smallest, if not the smallest, BBR in the world. It weights 650 g and is 25 cm tall, which is much smaller and lighter than the other existing BBRs. As a type of an inverted pendulum robot, the natural frequency of the system is proportional to the square root of the length of the center of mass from the center of rotation. Therefore, the MBBR's smaller body accelerates faster which demands the controller to react quicker to the change in the system states. The low-cost nature of the components contributes to the high friction and noise in the system which further increases the difficulty of the estimation and controls. In particular, the omniwheels can have microslips when transitioning between the two rows of the wheel's rollers, adding vibrations and noise into the measurements. All of the challenges above motivate us to develop a nonlinear dynamic model which works well under high yaw rate and use it in controllers and state estimators.

*This work was supported by WowWee Robotics.
[1]emails: {esihite,djyang,bewley}@ucsd.edu

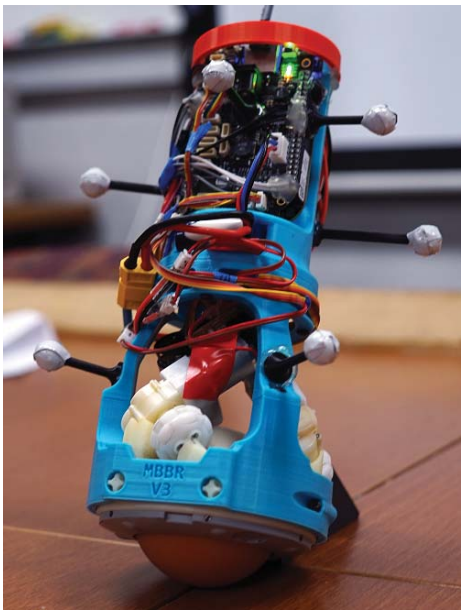


Fig. 1: The most recent iteration of the Micro Ball-Balancing Robot, with motion capture markers attached.

This new model can then be used to drive this robot more aggressively, e.g. translating while spinning at the same time.

Past works in this topic include the indirect Kalman Filtering used by Rezero [2]. In their work, the states were estimated using Kalman Filter (KF) from the kinematic relationship between the IMU and the encoder measurements. The formulation of a BBR 3D dynamic model has been done by [6] and [7]. However, [6] did not use the model in an estimator or controller. The KF used in [7] was not explained in detail and they assumed that robot's attitude angles can be directly measured. The linearized BBR dynamic model was used by most of the existing BBRs; the model was simplified down into two decoupled Mobile Inverted Pendulum (MIP) problems in the x-y plane (roll and pitch). The MIP dynamic model is well known and can easily be linearized [8], and used in linear controllers and estimators. We also have shown that the attitude estimation accuracy can be improved by using a high yaw-rate dynamic model in an Extended Kalman Filter (EKF) on a MIP robot [9].

In this paper, we derive the nonlinear high yaw-rate BBR model, implement said model in an EKF and use the estimated states in a state feedback controller. The accuracy of the state estimation were evaluated by comparing the the EKF attitude estimates with motion capture measurements.

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

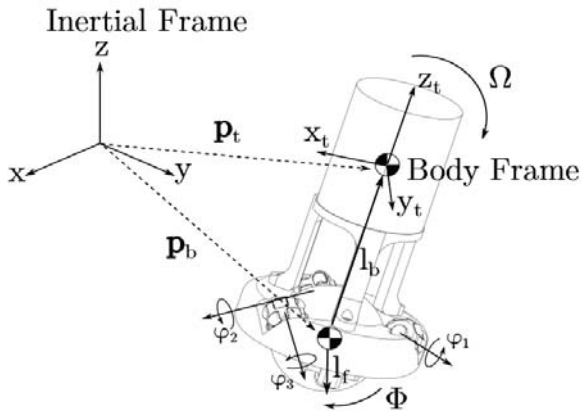


Fig. 2: Diagram of the frame of references, some position vectors, and angular speed.

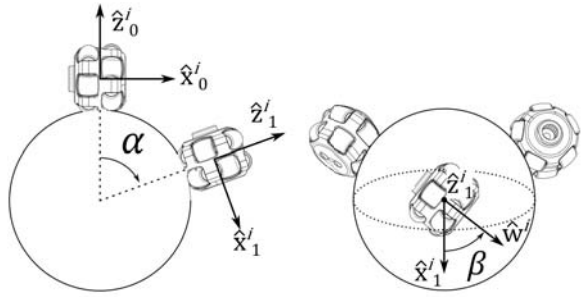


Fig. 3: Perpendicular omniwheel alignment diagram, represented by the α and β angles.

A KF using linearized BBR model is used as a comparison to show that the high yaw-rate model EKF has a better estimation accuracy and stability compared to the linear model KF. The implementation of a high yaw-rate BBR dynamic model in an EKF using on board raw measurement data is novel and the methodology is extensible to other lightweight low-cost robots. The EKF is chosen instead of the other nonlinear KFs, such as Unscented KF, because it is simple to implement. The Jacobian of the high yaw-rate model is also easy to calculate which makes the implementation of the EKF easier than other nonlinear KFs.

II. DYNAMIC MODELING

This section contains the derivations of the BBR dynamic equation of motion using Lagrangian Dynamics, which follows similar derivations done by Hoshino [3]. The equation of motion is derived symbolically in 'Wolfram Mathematica 10.0', which then is simplified into the high yaw-rate model used in the EKF. The list of parameters and variables used in the derivations can be seen in Table I.

A. Frames of Reference and Wheel Transformation

The frames of reference used in the derivations are the inertial and the body frames, as shown in Fig. 2. The variables defined in the body frame are represented by using the superscript B. The transformation from body to inertial frame is shown in (1) below:

$$\begin{gathered} \boldsymbol{x} = R_B(\boldsymbol{\theta})\, \boldsymbol{x}^B, \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_x & \theta_y & \theta_z \end{bmatrix}^T \\ R_B(\boldsymbol{\theta}) = R_z(\theta_z)\, R_y(\theta_y)\, R_x(\theta_x), \end{gathered} \tag{1}$$

where $R_x(\theta_x)$, $R_y(\theta_y)$ and $R_z(\theta_z)$ are the rotation matrices of the Euler rotations about the inertial frame's x, y and z axis respectively. Then $R_B(\boldsymbol{\theta})$ is the body's intrinsic rotation in the body's z-y'-x'' axis (intrinsic yaw-pitch-roll), which is one of the standard Tait-Bryan angles.

The omniwheels in the MBBR are aligned perpendicularly as shown in Fig. 3. α is the omniwheel contact angle from the ball's north pole about the body frame and β is the omniwheel's tilt angle about the wheel's axis perpendicular to the surface of the ball. The normalized vectors $\hat{\boldsymbol{w}}^i$, $i = \{1, 2, 3\}$, represent the omniwheel's spinning axis and the direction of the torque τ_i applied by the motor i. The optical encoder attached on the motor i measures the angle φ_i which can be used to determine the ball rotation angle $\boldsymbol{\phi}$. Assuming that there is no slip between the omniwheels and the ball, the applied torque and encoder measurement at omniwheel i in body coordinates are $(r/r_w)\tau_i\hat{\boldsymbol{w}}_i$ and $(r_w/r)\varphi_i\hat{\boldsymbol{w}}_i$ respectively. Let $\boldsymbol{\varphi}^B$ be the ball rotation angle as measured by the encoders and $\boldsymbol{\tau}^B$ be the total torque applied to the ball. Then $\boldsymbol{\varphi}^B$ and $\boldsymbol{\tau}^B$ can be calculated from τ_i and φ_i as shown below:

$$\boldsymbol{\varphi}_w = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}^T, \quad \boldsymbol{\tau}_w = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T \tag{2}$$

$$\boldsymbol{\varphi}^B = (r_w/r) \textstyle\sum_{i=1}^{3} \varphi_i \hat{\boldsymbol{w}}^i = (r_w/r)\, T_{ob}\, \boldsymbol{\varphi}_w \tag{3}$$

$$\boldsymbol{\tau}^B = (r/r_w) \textstyle\sum_{i=1}^{3} \tau_i \hat{\boldsymbol{w}}^i = (r/r_w)\, T_{ob}\, \boldsymbol{\tau}_w, \tag{4}$$

where the matrix $T_{ob} = \begin{bmatrix} \hat{\boldsymbol{w}}^1 | \hat{\boldsymbol{w}}^2 | \hat{\boldsymbol{w}}^3 \end{bmatrix}$ is the transformation matrix from the omniwheel axis to the body frame. Our omniwheels are perpendicular, therefore $(T_{ob})^{-1} = (T_{ob})^T$.

B. Kinematics Formulation

This section derives the body and the ball kinematic equations which will be used for deriving the kinetic and

TABLE I: List of parameters and time varying variables. CoM = Center of Mass, CoR = Center of Rotation.

Parameter List	
m_t = top body mass.	$\hat{I}_t$ = top body inertia about CoM.
m_b = ball mass.	$\hat{I}_b$ = ball inertia about CoM.
r = ball radius.	I_w = omniwheel inertia about its CoR.
r_w = omniwheel radius.	l = length of body's CoM from CoR.
k_1 = motor torque gain.	k_2 = motor back EMF gain.
g = gravity constant.	
Time Varying Variable List, $i = \{1, 2, 3\}$	
θ_x = roll angle.	ϕ_x, ϕ_y, ϕ_z = ball rotation angles.
θ_y = pitch angle.	φ_i = encoder i rotation.
θ_z = yaw angle.	u_i = motor i PWM command $\in [-1, 1]$.
τ_i = motor i torque.	
$\boldsymbol{p}_t$ = top body CoM position vector.	
$\boldsymbol{p}_b$ = ball CoM position vector.	
$\boldsymbol{l}_b$ = length vector from ball's CoM to the body CoM.	
$\boldsymbol{l}_f$ = length vector from ball's CoM to the ground.	
$\boldsymbol{l}_a$ = length vector from ball's CoM to the IMU.	

potential energy of the system. The inertia of the body and the ball, represented by the matrix $\hat{I}_t$ and $\hat{I}_b$ respectively, are shown below:

$$\hat{I}_t^B = \text{diag}(I_{t1}, I_{t2}, I_{t3}) \qquad \hat{I}_b = I_b\, I_{3\times 3}. \tag{5}$$

The length vectors used in the derivation are defined below:

$$\boldsymbol{l}_b^B = \begin{bmatrix} 0 & 0 & l \end{bmatrix}^T, \qquad \boldsymbol{l}_f = \begin{bmatrix} 0 & 0 & -r \end{bmatrix}^T. \tag{6}$$

The body rotational speed $\boldsymbol{\Omega}$ about the inertial frame:

$$\boldsymbol{\Omega} = R_z(\theta_z)\, R_y(\theta_y) \begin{bmatrix} \dot{\theta}_x \\ 0 \\ 0 \end{bmatrix} + R_z(\theta_z) \begin{bmatrix} 0 \\ \dot{\theta}_y \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_z \end{bmatrix}. \tag{7}$$

The ball rotational speed $\boldsymbol{\omega}$ about the inertial frame:

$$\boldsymbol{\omega} = \begin{bmatrix} \dot{\phi}_x & \dot{\phi}_y & \dot{\phi}_z \end{bmatrix}^T = R_B(\boldsymbol{\theta})\, \dot{\boldsymbol{\varphi}}^B + \boldsymbol{\Omega}. \tag{8}$$

The ball linear velocity $\dot{\boldsymbol{p}}_b$ can be derived by using the no slip conditions between the ball and the ground, as shown below:

$$\dot{\boldsymbol{p}}_b = \boldsymbol{l}_f \times \boldsymbol{\omega}, \qquad \dot{\phi}_z = 0. \tag{9}$$

This constraint is applied here in order to avoid the use of Lagrange Multiplier in the dynamic equation. This no slip condition also constrains the $\dot{\phi}_z$ to zero, which affects the encoder measurement $\dot{\boldsymbol{\varphi}}$ in (8). Then $\dot{\boldsymbol{\varphi}}$ can be expressed in terms of $\dot{\phi}_x$ and $\dot{\phi}_y$ as shown below:

$$\dot{\boldsymbol{\varphi}}^B = R_B(\boldsymbol{\theta})^T \left(\begin{bmatrix} \dot{\phi}_x & \dot{\phi}_y & 0 \end{bmatrix}^T - \boldsymbol{\Omega} \right). \tag{10}$$

Finally, we have the body linear velocity:

$$\dot{\boldsymbol{p}}_t = \tfrac{d}{dt} \left(R_B\left(\boldsymbol{\theta}\right) \boldsymbol{l}_b^B \right) + \dot{\boldsymbol{p}}_b. \tag{11}$$

C. Motor Dynamics

The back electromotive force (EMF) from the DC brushed motors used in the robot also contributes to the system dynamic. The torque applied by each motor is:

$$\tau_i = k_1 u_i - k_2 \dot{\varphi}_i, \qquad i = 1, 2, 3, \tag{12}$$

where $u_i = [-1, 1]$ is the PWM command into the motor. Using the transformations in (3) and (4) into (12) yields:

$$\boldsymbol{\tau}^B = k_1 \boldsymbol{u} - k_2 \left(r/r_w\right)^2 \dot{\boldsymbol{\varphi}}^B \tag{13}$$

$$\boldsymbol{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T = (r/r_w)\, T_{ob} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T. \tag{14}$$

D. Lagrangian Dynamics

The BBR dynamic equation is derived using Lagrangian Dynamics which then simplified and transformed into the state space form. The energy equations for the Lagrangian are derived for the top body, the ball and the wheels. The potential and the linear kinetic energy of the wheels are assumed to be negligible compared to the body and the ball due to their small mass. However, the no slip constraint between the ball and the wheels creates a coupling. This coupling and the fast wheel speed may cause a nontrivial increase in the rotational energy. In order to keep the equations simple and prevent cross terms between the $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$, we assume that $\boldsymbol{\Omega}$ is much smaller than the wheel speed $\dot{\varphi}_i$. Then the wheel i's angular velocity vector $\boldsymbol{\omega}_{wi}$ is shown below:

$$\boldsymbol{\omega}_{wi} = \boldsymbol{\Omega} + \dot{\varphi}_i\, \hat{\boldsymbol{w}}^i \approx \dot{\varphi}_i\, \hat{\boldsymbol{w}}^i. \tag{15}$$

Let I_w be the wheel inertia about its axis of rotation. Then the total angular kinetic energy of the wheels K_w is:

$$\begin{aligned} K_w &= \sum_{i=1}^{3} \frac{I_w}{2} \boldsymbol{\omega}_{wi}^T\, \boldsymbol{\omega}_{wi} = \frac{I_w}{2} \dot{\boldsymbol{\varphi}}_w^T\, \dot{\boldsymbol{\varphi}}_w \\ &= (I_w/2)(r/r_w)^2 (\dot{\boldsymbol{\varphi}}^B)^T \dot{\boldsymbol{\varphi}}^B. \end{aligned} \tag{16}$$

The kinetic and potential energy of the body and the ball are:

$$\begin{aligned} K_t &= \tfrac{1}{2} (R_B^T\, \boldsymbol{\Omega})^T\, \hat{I}_t^B\, (R_B^T\, \boldsymbol{\Omega}) + \tfrac{1}{2}\, m_t\, \dot{\boldsymbol{p}}_t^T\, \dot{\boldsymbol{p}}_t \\ K_b &= \tfrac{1}{2}\, \boldsymbol{\omega}^T\, \hat{I}_b\, \boldsymbol{\omega} + \tfrac{1}{2}\, m_b\, \dot{\boldsymbol{p}}_b^T\, \dot{\boldsymbol{p}}_b \\ U_t &= -m_t \begin{bmatrix} 0 & 0 & -g \end{bmatrix} \boldsymbol{p}_t, \qquad U_b = 0. \end{aligned} \tag{17}$$

The following states $\boldsymbol{x}(t)$ are used in the dynamic equation:

$$\boldsymbol{x}(t) = \begin{bmatrix} \boldsymbol{q} & \dot{\boldsymbol{q}} \end{bmatrix}^T, \quad \boldsymbol{q} = \begin{bmatrix} \theta_x & \theta_y & \theta_z & \phi_x^* & \phi_y^* \end{bmatrix}^T, \tag{18}$$

where $\dot{\phi}_x^*$ and $\dot{\phi}_y^*$ are the inertial frame ball rotation speeds rotated about the inertial z-axis by θ_z as shown below:

$$\dot{\phi}_x^* = \cos\left(\theta_z\right) \dot{\phi}_x + \sin\left(\theta_z\right) \dot{\phi}_y \tag{19}$$

$$\dot{\phi}_y^* = -\sin\left(\theta_z\right) \dot{\phi}_x + \cos\left(\theta_z\right) \dot{\phi}_y. \tag{20}$$

By choosing ϕ_x^* and ϕ_y^* as the states, we can eliminate all $\sin\left(\theta_z\right)$ and $\cos\left(\theta_z\right)$ terms after the simplification in Section II-F. Then the Lagrangian of the system is:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = K_t + K_b + K_w - U_t. \tag{21}$$

The system dynamic equation can be solved by using the Lagrange's Equation:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\partial \boldsymbol{q}} - \boldsymbol{\tau}_L = \boldsymbol{0}, \tag{22}$$

where $\boldsymbol{\tau}_L$ is the total generalized force applied by all the motors through the omniwheels. $\boldsymbol{\tau}_L$ is defined as follows:

$$\tau_{Lj} = (\boldsymbol{\tau}^B)^T \left(\frac{\partial \dot{\boldsymbol{\varphi}}^B}{\partial \dot{q}_j} \right), \qquad j = \{1, 2, 3, 4, 5\}, \tag{23}$$

where τ_{Lj} and $\dot{q}_j$ are the j-th component of $\boldsymbol{\tau}_L$ and $\dot{\boldsymbol{q}}$ respectively. The no slip constraints are already applied during the kinematic formulations, so there is no Lagrange Multiplier due to system constraints. Then using the Lagrange's Equation in (22), we can form the system dynamic equation $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ which needs to be simplified before being used in the EKF because of its sheer length and nonlinearities.

E. Sensor Dynamics

The MBBR uses the following sensors to estimate the states: the optical encoders and IMU gyrometer and accelerometer measurements. In particular, the accelerometer is greatly affected by the dynamic of the robot. Therefore, we need to derive the sensor dynamics before they can be used in the EKF. The encoders measure the ball rotation angle with respect to the body frame ($\boldsymbol{y}_{en}^B = \boldsymbol{\varphi}^B$), where $\dot{\boldsymbol{\varphi}}^B$ was derived in (10). The gyrometer measures the body angular velocity ($\boldsymbol{y}_{gy}^B = \boldsymbol{\Omega}^B$) which was derived in (7). The accelerometer measures the linear acceleration at the IMU's position about the body frame. Let $\boldsymbol{p}_a$ be the position of the IMU in the inertial frame and $\boldsymbol{l}_a^B = [l_{ax}, l_{ay}, l_{az}]^T$ is the length vector from ball's center of mass to the IMU. Then using a similar kinematics derivation to (11), we can derive the accelerometer measurement dynamics below:

$$\dot{\boldsymbol{p}}_a = \tfrac{d}{dt}\left(R_B(\boldsymbol{\theta})\,\boldsymbol{l}_a^B\right) + \dot{\boldsymbol{p}}_b \tag{24}$$

$$\boldsymbol{y}_{ac}^B = R_B(\boldsymbol{\theta})^T\left(\begin{bmatrix}0 & 0 & -g\end{bmatrix}^T - \ddot{\boldsymbol{p}}_a\right). \tag{25}$$

The acceleration components of $\ddot{\boldsymbol{p}}_a$ can be derived from the system dynamic equation $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ solved in the Lagrangian Dynamics Section above. Nonzero l_{ax} and l_{ay} can cause a bias in the accelerometer measurement due to the centripetal force. However, we assume that the l_{ax} and l_{ay} values are zero in order to keep the equations simple.

F. Simplification

The dynamic equation and sensor dynamics must be simplified before they can be used in the EKF due to the sheer size and nonlinearities in the equations. We can use the linear BBR model's assumptions without the trivial yaw rate assumption. Therefore, we assume that θ_x, θ_y, $\dot{\theta}_x$, and $\dot{\theta}_y$ are small. This assumption works under the knowledge that a stable BBR system has small perturbations on these variables. Then we can use the small angle approximation for the sine and cosine functions such that $\sin(\theta) \approx \theta$, $\cos(\theta) \approx 1$ for both θ_x and θ_y. Also, all of the multiplications between θ_x, θ_y, $\dot{\theta}_x$, and $\dot{\theta}_y$, or with themselves are approximately equal to zero (e.g. $\theta_x\,\theta_x \approx 0, \dot{\theta}_x\,\theta_y \approx 0$). If ϕ_x^* and ϕ_y^* are used as the states, then there is no $\sin(\theta_z)$ and $\cos(\theta_z)$ left in the dynamic equation. The high yaw-rate model can't be simplified further, but the linear model can be derived from here by using $\dot{\theta}_z = 0$ and $u_z = 0$.

III. KALMAN FILTERS AND CONTROLLER SETUP

This section describes the linear model KF, high yaw-rate model EKF and the controller used by the MBBR during the motion capture experiment. Table II lists the numerical values of the MBBR parameters used in the dynamic model. The high yaw-rate model is the linear model with some additional nonlinear terms, so we derive the linear model first in the following section.

TABLE II: MBBR Parameter Values.

Parameter	Value	Parameter	Value
m_t	500 g	I_{t1}	4.39 10^{-3} kg.m^2
m_b	150 g	I_{t2}	4.39 10^{-3} kg.m^2
r	32 mm	I_{t3}	1.44 10^{-3} kg.m^2
r_w	12.5 mm	I_b	9.13 10^{-5} kg.m^2
l	100 mm	I_w	1.83 10^{-6} kg.m^2
l_{ax}	0 mm	k_1	0.176 N.m
l_{ay}	0 mm	$k_2(r/r_w)^2$	0.011 N.m.s/rad
l_{az}	130 mm	g	9.8 m/s^2
dt	0.005 s		

A. Linear Model Kalman Filter

The linear BBR model is simply a MIP problem about roll and pitch. The linear model uses the following states, input and output vectors respectively:

$$\begin{aligned}\boldsymbol{x}^L &= [\theta_x, \theta_y, \phi_x^*, \phi_y^*, \dot{\theta}_x, \dot{\theta}_y, \dot{\phi}_x^*, \dot{\phi}_y^*]^T\\ \boldsymbol{u}^L &= [u_x, u_y]^T\\ \boldsymbol{y}^L &= [\Omega_x^B, \Omega_y^B, \varphi_x^B, \varphi_y^B, y_{ac1}^B, y_{ac2}^B]^T.\end{aligned} \tag{26}$$

Then by using the parameter values listed in Table II, the continuous time linear dynamic model for the MBBR is:

$$\begin{aligned}\dot{\boldsymbol{x}}^L &= \boldsymbol{f}^L(\boldsymbol{x}^L, \boldsymbol{u}^L)\\ f_1^L &= \dot{\theta}_x, \quad f_2^L = \dot{\theta}_y, \quad f_3^L = \dot{\phi}_x^*, \quad f_4^L = \dot{\phi}_y^*\\ f_5^L &= 80.1\,\theta_x - 5.35\,\dot{\theta}_x + 5.35\,\dot{\phi}_x^* - 87.9\,u_x\\ f_6^L &= 80.1\,\theta_y - 5.35\,\dot{\theta}_y + 5.35\,\dot{\phi}_y^* - 87.9\,u_y\\ f_7^L &= -165\,\theta_x + 25.0\,\dot{\theta}_x - 25.0\,\dot{\phi}_x^* + 410\,u_x\\ f_8^L &= -165\,\theta_y + 25.0\,\dot{\theta}_y - 25.0\,\dot{\phi}_y^* + 410\,u_y,\end{aligned} \tag{27}$$

where the f_i^L is the i-th component of the vector $\boldsymbol{f}^L$. The sensor dynamic equation for this model is:

$$\begin{aligned}\boldsymbol{y}^L &= \boldsymbol{h}^L(\boldsymbol{x}^L, \boldsymbol{u}^L)\\ h_1^L &= \dot{\theta}_x, \qquad h_2^L = \dot{\theta}_y\\ h_3^L &= \phi_x^* - \theta_x, \quad h_4^L = \phi_y^* - \theta_y\\ h_5^L &= 0.69\,\theta_y + 0.17\,\dot{\theta}_y - 0.17\,\dot{\phi}_y^* + 2.70\,u_y\\ h_6^L &= -0.69\,\theta_x - 0.17\,\dot{\theta}_x + 0.17\,\dot{\phi}_x^* - 2.70\,u_x,\end{aligned} \tag{28}$$

where the h_i^L is the i-th component of the vector $\boldsymbol{h}^L$. Then the discrete time equation for the KF can be calculated using a simple Explicit Euler scheme below:

$$\begin{aligned}\boldsymbol{x}_{k+1}^L &= \boldsymbol{x}_k^L + dt\,\boldsymbol{f}^L(\boldsymbol{x}_k^L, \boldsymbol{u}_k^L) + \boldsymbol{v}_k^L\\ \boldsymbol{y}_k^L &= \boldsymbol{h}^L(\boldsymbol{x}_k^L, \boldsymbol{u}_k^L) + \boldsymbol{w}_k^L,\end{aligned} \tag{29}$$

where dt is the measurement period, $\boldsymbol{v}_k^L$ and $\boldsymbol{w}_k^L$ are the process and the measurement noise vectors respectively.

B. High Yaw-Rate Model Extended Kalman Filter

The high yaw-rate model uses the following states, input and output vectors respectively:

$$\begin{aligned}\boldsymbol{x}^N &= [\theta_x, \theta_y, \theta_z, \phi_x^*, \phi_y^*, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z, \dot{\phi}_x^*, \dot{\phi}_y^*, \varphi_x^B, \varphi_y^B, \varphi_z^B]^T\\ \boldsymbol{u}^N &= [u_x, u_y, u_z]^T\\ \boldsymbol{y}^N &= [\Omega_x^B, \Omega_y^B, \Omega_z^B, \varphi_x^B, \varphi_y^B, \varphi_z^B, y_{ac1}^B, y_{ac2}^B]^T.\end{aligned} \tag{30}$$

The continuous time high yaw-rate dynamic model is:

$$\begin{aligned}
\dot{\boldsymbol{x}}^N &= \boldsymbol{f}^N(\boldsymbol{x}^N, \boldsymbol{u}^N) \\
f_1^N &= \dot{\theta}_x, \quad f_2^N = \dot{\theta}_y, \quad f_3^N = \dot{\theta}_z, \quad f_4^N = \dot{\phi}_x^*, \quad f_5^N = \dot{\phi}_y^* \\
f_6^N &= f_5^L + \dot{\theta}_z(-5.58\,\theta_y + 1.76\,\dot{\theta}_y + 0.00196\,\dot{\phi}_y^*) \\
&\quad + 0.76\,\theta_x\,\dot{\theta}_z^2 - 179\,\theta_y\,u_z \\
f_7^N &= f_6^L + \dot{\theta}_z(5.58\,\theta_x - 1.76\,\dot{\theta}_x - 0.00196\,\dot{\phi}_x^*) \\
&\quad + 0.76\,\theta_y\,\dot{\theta}_z^2 + 179\,\theta_x\,u_z \\
f_8^N &= \dot{\phi}_x^*\,(7.15\,\theta_y + 0.0082\,\dot{\theta}_y) - \dot{\phi}_y^*\,(1.80\,\theta_x + 0.0082\,\dot{\theta}_x) \\
&\quad + \dot{\theta}_z\,(0.0063\,\theta_x\dot{\phi}_x^* + 0.0082\,\theta_y\dot{\phi}_y^* - 7.36) \\
&\quad - 121\,u_z + 3.38\,\theta_y\,u_x - 91.2\,\theta_x\,u_y \\
f_9^N &= f_7^L + \dot{\theta}_z(-3.78\,\theta_y + 0.51\,\dot{\theta}_y - 0.0040\,\dot{\phi}_y^*) \\
&\quad + 0.51\,\theta_x\,\dot{\theta}_z^2 + 348\,\theta_y\,u_z \\
f_{10}^N &= f_8^L + \dot{\theta}_z(3.78\,\theta_x - 0.51\,\dot{\theta}_x + 0.0040\,\dot{\phi}_x^*) \\
&\quad + 0.51\,\theta_y\,\dot{\theta}_z^2 - 348\,\theta_x\,u_z \\
f_{11}^N &= \dot{\phi}_x^* - \dot{\theta}_x + \theta_y\,\dot{\theta}_z, \qquad f_{12}^N = \dot{\phi}_y^* - \dot{\theta}_y - \theta_x\,\dot{\theta}_z \\
f_{13}^N &= \theta_y\,\dot{\phi}_x^* - \theta_x\,\dot{\phi}_y^* - \dot{\theta}_z.
\end{aligned} \tag{31}$$

The additional states $\boldsymbol{\varphi}^B$ in (30) are used for the encoder measurements. The $\dot{\boldsymbol{\varphi}}^B$ defined in (8) needs to be integrated to determine the $\boldsymbol{\varphi}^B$ which is represented by these additional states. The sensor dynamic model is:

$$\begin{aligned}
\boldsymbol{y}^N &= \boldsymbol{h}^N(\boldsymbol{x}^N, \boldsymbol{u}^N) \\
h_1^N &= \dot{\theta}_x - \theta_y\,\dot{\theta}_z, \qquad h_2^N = \dot{\theta}_y + \theta_y\,\dot{\theta}_z \\
h_3^N &= \dot{\theta}_z, \quad h_4^N = \varphi_x^B, \quad h_5^N = \varphi_y^B, \quad h_6^N = \varphi_z^B \\
h_7^N &= h_5^L + \dot{\theta}_z\,(0.20\,\theta_x - 0.027\,\dot{\theta}_x + 0.00022\,\dot{\phi}_x^*) \\
&\quad + 0.027\theta_y\,\dot{\theta}_z^2 + 0.56\,\theta_x\,u_z \\
h_8^N &= h_6^L + \dot{\theta}_z\,(0.20\,\theta_y - 0.027\,\dot{\theta}_y + 0.00022\,\dot{\phi}_y^*) \\
&\quad - 0.027\theta_x\,\dot{\theta}_z^2 + 0.56\,\theta_y\,u_z.
\end{aligned} \tag{32}$$

Similarly to the linear case, the discrete time model for the EKF can be derived using Explicit Euler method as in (29) with $\boldsymbol{v}_k^N$ and $\boldsymbol{w}_k^N$ as the nonlinear model noise vectors. Let $Q = E(\boldsymbol{v}_k^T\,\boldsymbol{v}_k)$ and $R = E(\boldsymbol{w}_k^T\,\boldsymbol{w}_k)$ be the process and measurement noise covariance matrix respectively. We use the following Q and R matrices for the linear and high yaw-rate model:

$$\begin{aligned}
Q^L &= \mathrm{diag}(a_1, a_1, a_2, \ldots, a_2), \quad a_1 = 10^{-6} \\
Q^N &= \mathrm{diag}(a_1, a_1, a_2, \ldots, a_2), \quad a_2 = 0.1 \\
R^L &= \mathrm{diag}(b_1, b_1, b_2, b_2, b_3, b_3) \\
R^N &= \mathrm{diag}(b_1, b_1, b_1, b_2, b_2, b_2, b_3, b_3) \\
b_1 &= 2.08 \cdot 10^{-6}, \quad b_2 = 7.62 \cdot 10^{-5}, \quad b_3 = 0.70.
\end{aligned} \tag{33}$$

C. Controller Setup

The MBBR was controlled by using a linear state feedback controller and the estimated states from either the KF or the EKF. The controller gains were determined from the linear model's LQR controller gains which then were tuned by hand afterwards. The same controller were used to test both the KF and the EKF in the Section IV. The KF doesn't estimate θ_z and $\dot{\theta}_z$, so we used the IMU's DMP yaw angle estimates and raw gyro measurements for the controller in the KF experiment. Let $\boldsymbol{x}_c$ be the states for the linear state feedback controller and $\boldsymbol{x}_r$ be the reference states. Then we have the following linear state feedback controller:

$$\begin{aligned}
\boldsymbol{x}_c &= [\theta_x, \theta_y, \theta_z, \phi_x^*, \phi_y^*, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z, \dot{\phi}_x^*, \dot{\phi}_y^*] \\
\begin{bmatrix} u_x, u_y, u_z \end{bmatrix}^T &= K_c\,(\boldsymbol{x}_c - \boldsymbol{x}_r) \\
K_c &= \begin{bmatrix}
c_1 & 0 & 0 & c_2 & 0 & c_3 & 0 & 0 & c_4 & 0 \\
0 & c_1 & 0 & 0 & c_2 & 0 & c_3 & 0 & 0 & c_4 \\
0 & 0 & c_5 & 0 & 0 & 0 & 0 & c_6 & 0 & 0
\end{bmatrix} \\
c_1 &= 9, \qquad c_2 = 0.06, \quad c_3 = 0.6 \\
c_4 &= 0.075, \quad c_5 = 0.4, \qquad c_6 = 1.2.
\end{aligned} \tag{34}$$

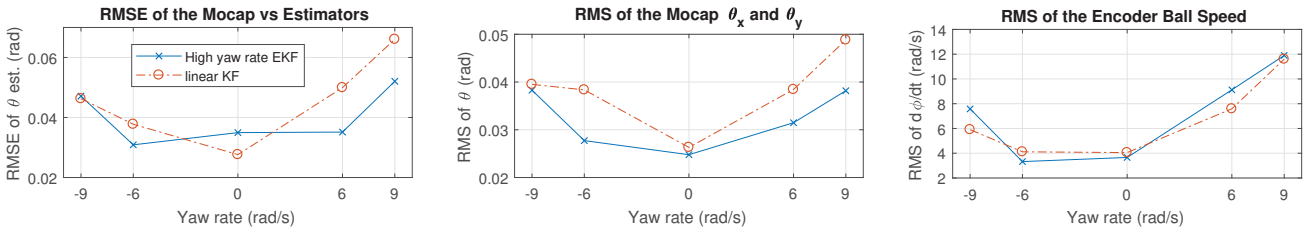
We were stabilizing the states ϕ_x^* and ϕ_y^* which are not the inertial frame ball angles ϕ_x and ϕ_y. Controlling for ϕ_x and ϕ_y is possible for the EKF which can achieve a stable spinning and translating at the same time with respect to the inertial frame. However, attempting this with the KF is highly unstable, so we used ϕ_x^* and ϕ_y^* as the controller states for the motion capture experiments. This means that both controllers used the same states and we might get a fairer comparison in the motion capture experiment.

IV. MOTION CAPTURE EXPERIMENT

In this experiment, the accuracy of the EKF and the KF were evaluated by comparing the estimated attitude angles with the motion captured measurements. In addition, the stability of the controller using the estimated states was also evaluated from the fluctuations in body angles and ball speed.

A. Experimental Setup

A motion capture (mocap) system using four Optitrack Prime 13 cameras was used to verify the accuracy of the estimated θ_x and θ_y angles by the high yaw-rate model EKF and linear model KF. The MBBR was attached with 8 mocap markers, as shown in Fig. 1, and was balanced with the controller in (34) using the state estimates from either the EKF or the KF. The estimated states and the mocap measurements were recorded under several yaw rates: 0, ± 6 and ± 9 rad/s. These yaw rates were chosen due to the motor limitations and we believe that these speeds are high enough to be nontrivial for the system dynamics. The accuracy of both estimators were compared by calculating the RMSE value of the estimated θ_x and θ_y with respect to the motion captured measurements. We only evaluated the accuracy of these variables because the actual ball rotation is extremely difficult to measure. The ball is mostly hidden by the robot's casing and the on board sensors (encoders) are not accurate enough due to the following factors: encoder skipping counts, friction, and the slip between the omniwheels and the ball. Most of these problems are caused by the small size and low-cost nature of the components. Therefore, we are unable to determine the accuracy of the ball speed estimate with our current setup. Additionally, θ_x and θ_y are the most important states for balancing, so we prioritize on their accuracy more than the ball position and speed estimates. The controller



(a) RMSE of the $[\theta_x, \theta_y]$ estimates from both estimators vs motion capture.

(b) RMS value of the $[\theta_x, \theta_y]$ from the motion capture.

(c) RMS value of the ball speed estimate $[\dot{\phi}_x, \dot{\phi}_y]$ from the encoder measurements.

Fig. 4: Plot of RMSE between both estimators vs motion capture measurement, RMS of the robot's attitude from motion capture and RMS of the ball speed from the encoder estimates.

stability can be determined from the RMS values of the θ_x, the θ_y, and the ball speed. The θ angles can be measured by the mocap while the ball speed can be estimated from the encoder measurements. While inaccurate as we mentioned above, the RMS of the ball speed estimated by the encoders can be a good indicator of how well the robot maintains its position (position hold).

B. Experimental Result

The RMSE values of the θ_x and θ_y estimates, RMS values of the motion captured θ and the ball speed can be seen in Fig. 4. Fig. 5 shows the motion captured measurements vs the high yaw-rate EKF estimates during idling and spinning at 9 rad/s. The data showed that the high yaw-rate model EKF performed better than the linear model KF under high yaw rates for both estimator accuracy and the upper body stability of the robot. The KF has better estimation accuracy than EKF when idling, which is the condition where we expect the linear model to perform well. The KF has better performance during -9 rad/s spin than the 9 rad/s spin. We believe that this is caused by the design limitation of the MBBR's rotated wheel axis shown in Fig. 3. This design choice affects the spinning differently: spinning left and right pushes the ball away and into the wheels respectively. The spinning right case (positive yaw rate) significantly increases the system friction and we observed less stability while spinning in this direction. However, the controller with the EKF performed equally well on both directions which is a great result. The RMS values of the estimated ball speed by the encoders in Fig. 4c showed that both estimators have approximately the same difficulty of maintaining position hold under high yaw rates. This could be an issue caused from using ϕ_x^* and ϕ_y^* as the states instead of the inertial ball angles. Drifting during position hold has always been an issue for our MBBR which is also caused by the inaccuracy of the encoder measurements. However, the increased stability on θ_x and θ_y for the EKF is an improvement over the KF and the overall performance of the robot while spinning fast is better with the EKF estimates.

V. CONCLUSION AND FUTURE WORK

In this paper, a BBR model that allows for high yaw rates is derived and implemented in an EKF. The motion capture data showed that the estimation accuracy and controller

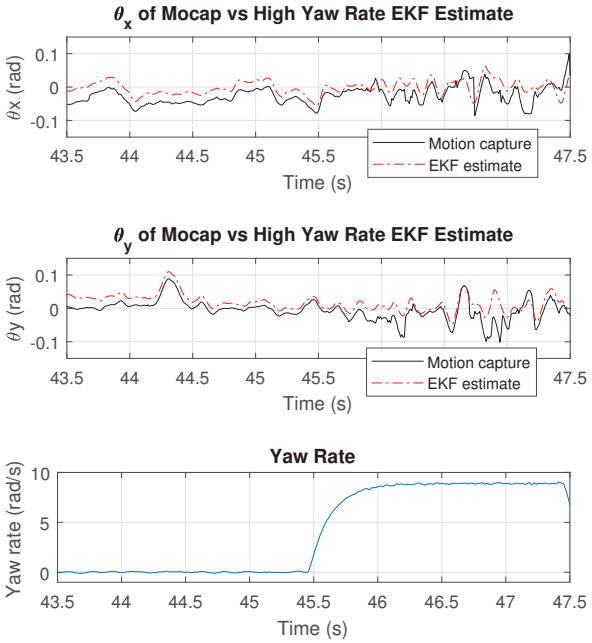


Fig. 5: Plot of the motion captured measurement, high yaw-rate model EKF attitude estimates, and the yaw rate.

stability under high yaw rates using the high yaw-rate model EKF is better than with the linear model KF. For future work, we can improve the accuracy of our model parameters by using system identification techniques or implement a proper friction model into the EKF. We believe that the friction present in our MBBR is very significant and a proper friction compensation or model may improve the position tracking performance that the current build lacks. Improving the performance of the ball position tracking under high yaw rates can also improve the robustness of the controller. The performance of the EKF has been shown, so we can change the controller states in (34) to use inertial ball positions ϕ_x and ϕ_y, instead of ϕ_x^* and ϕ_x^*, for future MBBR controllers.

ACKNOWLEDGMENT

Special thanks to Wowwee for supplying us with the motors, encoders, motor mounts and the ball. Also special thanks to Clark Briggs from ATA Engineering for offering our team to use their motion capture system.

REFERENCES

[1] Yang, Daniel, et al., "Design and control of a micro ball-balancing robot (MBBR) with orthogonal midlatitude omniwheel placement," in *IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, pp. 4098-4104, 2015.

[2] Hertig, Lionel, et al. "Unified state estimation for a ballbot." *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[3] T. Hoshino, S. Yokota, T.Chino, "OmniRide: A personal vehicle with 3 DOF mobility," in *Int'l Conf. on Control, Automation, Robotics and Embedded Systems* , Dec 2013.

[4] T.B. Lauwers, G.A. Kantor, R.L Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 2884-2889, May 2006.

[5] U. Nagarajan, G. Kantor, R. Hollis, "The ballbot: An omnidirectional balancing mobile robot," in *The Int'l Journal of Robotics Research*, 2013.

[6] Bonci, Andrea. "New dynamic model for a Ballbot system." Mechatronic and Embedded Systems and Applications (MESA), 2016 12th IEEE/ASME International Conference on. IEEE, 2016.

[7] Fankhauser, Peter, and Corsin Gwerder. Modeling and Control of a Ballbot. BS thesis. Eidgenssische Technische Hochschule Zrich, 2010.

[8] Bewley, Thomas R. "Numerical renaissance: simulation, optimization, and control." pp.475, Renaissance Press, San Diego, 2012.

[9] Sihite, Eric, and Thomas Bewley. "Attitude estimation of a high-yaw-rate Mobile Inverted Pendulum; comparison of Extended Kalman Filtering, Complementary Filtering, and motion capture." American Control Conference (ACC), pp. 5831-5836. IEEE, 2018.