2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Unified State Estimation for a Ballbot

Lionel Hertig[1,2], Dominik Schindler[1,2], Michael Bloesch[2], C. David Remy[3], and Roland Siegwart[2]
[1]The authors assert equal contribution and joint first authorship.
[2]Autonomous Systems Lab, ETH Zurich, Switzerland, E-Mail: bloeschm@ethz.ch
[3]Department of Mechanical Engineering, University of Michigan, USA

***Abstract*—This paper presents a method for state estimation on a ballbot; i.e., a robot balancing on a single sphere. Within the framework of an extended Kalman filter and by utilizing a complete kinematic model of the robot, sensory information from different sources is combined and fused to obtain accurate estimates of the robot's attitude, velocity, and position. This information is to be used for state feedback control of the dynamically unstable system. Three incremental encoders (attached to the omniwheels that drive the ball of the robot) as well as three rate gyroscopes and accelerometers (attached to the robot's main body) are used as sensors. For the presented method, observability is proven analytically for all essential states in the system, and the algorithm is experimentally evaluated on the Ballbot Rezero.**

I. Introduction

Ballbots are a newly emerging class of robots [1], [2], [3], [4] which move by dynamically balancing on a sphere. They show a very small footprint and are able to perform highly dynamic motions. Ballbots are dynamically stabilized by closed-loop control, which preferably applies state feedback. That is, estimates of position, velocity, attitude, and angular rates of the body are used to form the control law. Current research platforms often exhibit similar sensor setups, but differ in how the sensor readings are processed.

For example, the Ballbot built by U. Nagarajan et al. at Carnegie Mellon University [2] utilizes an inertial measurement unit (IMU) and rotary encoders in the ball drive mechanism. Relying on a precise estimate of the attitude of the main body, the rotary encoder readings are directly transformed into velocity and position estimates of the robot whereas attitude is obtained from the IMU which internally fuses measurements from three rate gyroscopes and three accelerometers. Very similarly, BallIP, a Ballbot designed by M. Kumagai and T. Ochiai [3], uses a ball drive mechanism with stepping motors which inherently deliver position and velocity information when actuated. The state estimation process is similar to the CMU robot, but instead of obtaining attitude from an off-the-shelf IMU, it is computed with a complementary filter fusing raw gyroscope and accelerometer data. Others apply similar techniques [1], [4].

In contrast to the aforementioned approaches, which separate the attitude and position estimation processes, this paper presents a unified filter, which estimates the complete state of the robot. In this approach, information can flow in both directions; that is, the attitude estimate will not only rely on the IMU data, but will also profit from the encoder measurements. This is achieved by implementing an extended Kalman filter (EKF), which fuses data from encoders, accelerometers, and gyroscopes. A kinematic model is employed for the prediction of the states by integrating the accelerometer and gyroscope measurements while the update step is using the encoder measurements by exploiting the kinematic contact constraints of the robot and ball. Attitude is represented as a non-parameterized transformation matrix while uncertainties of the estimated attitude are represented and propagated by an error rotation vector [6]. An observability analysis shows that only absolute positions and rotations about the direction of gravity are not observable within this setup.

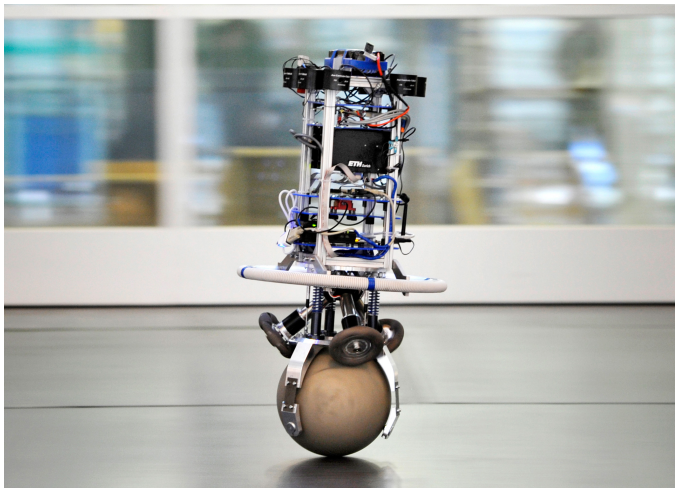

Fig. 1. The Ballbot Rezero measures 1 m in height, weights 14.5 kg, and reaches speeds of up to 2 m/s while pitching up to 17° [1], [5]. Photo: Karl-Heinz Hug, Ringier AG

The presented filter is evaluated on a dataset collected on the Ballbot Rezero (Fig. 1). Rezero is controlled at 160 Hz by a linear-quadratic regulator based on a linearized dynamic model. Similarly to the above mentioned platforms, It is equipped with rotary encoders for each ball-driving omniwheel as well as an IMU. However, only the raw accelerometer and gyroscope data are used for state estimation.

The remainder of this paper is organized as follows: Section II introduces the filter states and measurement signals, discusses the system model, and derives the filter equations for the prediction and update steps. Section III analyses the observability of the system, and in Section IV, the filter performance is evaluated on the basis of 190 seconds of exemplary driving in which attitude and velocity estimates are compared to data obtained from a motion capture system.

978-1-4673-5643-5/13/$31.00 ©2013 IEEE

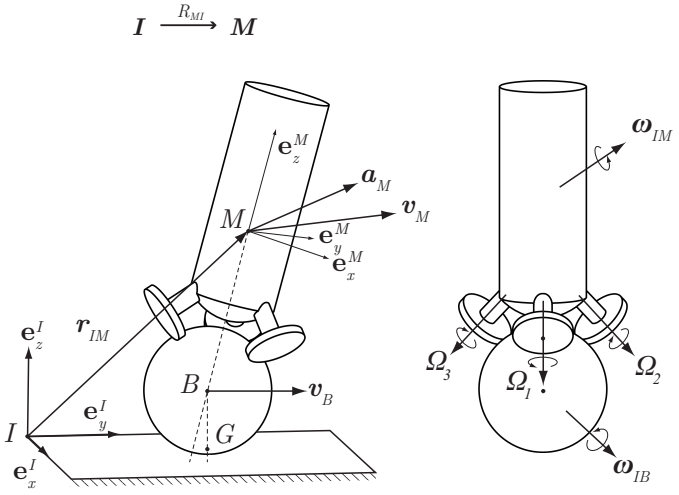

Fig. 2. Coordinate frames and kinematic Ballbot model

II. Filter Setup

A. Notation

For the derivation of the EKF, three coordinate frames and one point are introduced. The inertial frame I and the body-fixed frame M (Fig. 2) are used to represent the attitude of the upper body. The third coordinate frame B is fixed to the ball. It is only used to define the angular rate of the ball $\boldsymbol{\omega}_{IB}$ and is hence not shown in the figure. G stands for the point of contact with the ground.

The left index of a vector denotes its coordinate frame of reference. The right hand indices indicate the start and end point of relative positions $\boldsymbol{r}$, the point of reference for velocities $\boldsymbol{v}$, or the rotational rate $\boldsymbol{\omega}$ between two coordinate frames. For example, the angular rate of the ball relative to the inertial frame denoted in the body fixed frame is ${}_M\boldsymbol{\omega}_{IB}$. Rotations are handled by the transformation matrix $\boldsymbol{R}_{MI}$, which transforms the coordinates of a vector in the coordinate frame I to the coordinate frame M.

In order to write cross products in matrix form, skew symmetric matrices are introduced. Given a vector $\boldsymbol{v} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T$, the corresponding skew symmetric matrices $\lfloor \boldsymbol{v} \times \rfloor$ is defined as

$$\lfloor \boldsymbol{v} \times \rfloor = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \tag{1}$$

Given arbitrary vectors $\boldsymbol{v}$, $\boldsymbol{w}$ and a transformation matrix $\boldsymbol{R}$ we can state the following useful properties:

$$\boldsymbol{v} \times \boldsymbol{w} = \lfloor \boldsymbol{v} \times \rfloor \boldsymbol{w} = -\lfloor \boldsymbol{v} \times \rfloor^T \boldsymbol{w} \tag{2}$$

$$\lfloor \boldsymbol{R}\, \boldsymbol{v} \times \rfloor = \boldsymbol{R} \lfloor \boldsymbol{v} \times \rfloor \boldsymbol{R}^T. \tag{3}$$

B. Filter States and Measurements

The states of the EKF are chosen to be the position of the IMU ${}_M\boldsymbol{r}_{IM}$, its velocity ${}_M\boldsymbol{v}_M$, and its attitude parametrized by the transformation matrix $\boldsymbol{R}_{MI}$. For the sake of readability, the indices are dropped for these specific quantities. Nevertheless, it is important to keep in mind that all states are expressed in the robocentric coordinate frame M.

$$\boldsymbol{x} = ({}_M\boldsymbol{r}_{IM},\, {}_M\boldsymbol{v}_M,\, \boldsymbol{R}_{MI}) = (\boldsymbol{r},\, \boldsymbol{v},\, \boldsymbol{R}) \tag{4}$$

The rate gyroscope readings $\tilde{\boldsymbol{\omega}}$ correspond to the Ballbot's body angular rate ${}_M\boldsymbol{\omega}_{IM}$. They are corrupted by additive white Gaussian noise $\boldsymbol{n}_g$ and a constant bias $\boldsymbol{b}_g$:

$$\tilde{\boldsymbol{\omega}} = {}_M\boldsymbol{\omega}_{IM} + \boldsymbol{n}_g + \boldsymbol{b}_g \tag{5}$$

The accelerometers measurements $\tilde{\boldsymbol{f}}$ are based on the acceleration ${}_M\boldsymbol{a}_M$ of point M as well as gravity ${}_M\boldsymbol{g}$ (expressed in the IMU coordinate frame). Again, the signal is corrupted by additive white Gaussian noise $\boldsymbol{n}_a$ and a constant bias $\boldsymbol{b}_a$:

$$\tilde{\boldsymbol{f}} = {}_M\boldsymbol{a}_M + \boldsymbol{n}_a + \boldsymbol{b}_a - {}_M\boldsymbol{g} \tag{6}$$

The accelerometer and gyroscope noise covariance have been calculated from measurements of the idle Ballbot. Apart from an anti-aliasing filter, no further filtering is applied and the assumption of white sensor noise is justified. Online estimation of the bias was evaluated as well. However, since it was shown that the bias values are not fully observable, this would have led to divergence of the filter. Consequently, their values are estimated in a preliminary calibration step and assumed to remain constant during a trial. The bias corrected measurement quantities are:

$$\hat{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}} - \boldsymbol{b}_g = {}_M\boldsymbol{\omega}_{IM} + \boldsymbol{n}_g \tag{7}$$

$$\hat{\boldsymbol{f}} = \tilde{\boldsymbol{f}} - \boldsymbol{b}_a = {}_M\boldsymbol{a}_M + \boldsymbol{n}_a - R_{MI}\, {}_I\boldsymbol{g} \tag{8}$$

The omniwheels' angular rates Ω_i are obtained from incremental encoders. Measurements are corrupted by an additive noise term $\boldsymbol{n}_e$ that is primarily a result of slip between the ball and the wheels:

$$\hat{\boldsymbol{\Omega}} = \tilde{\boldsymbol{\Omega}} = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 \end{bmatrix}^T + \boldsymbol{n}_e \tag{9}$$

$\boldsymbol{n}_e$ is modeled as white Gaussian noise. This assumption is in conflict with the observation that wheel-slip can show a considerable correlation in time. Consequently, more conservative values have been chosen for the noise covariance.

C. Prediction Equations

To be able to predict the state variables, a discrete time system model needs to be derived. For each continuous time variable $\boldsymbol{x}$ its derivative is time-integrated to obtain the corresponding discrete time quantity $\boldsymbol{x}_k$ at time t_k. The prediction of the filter state $\hat{\boldsymbol{x}}_k^-$ is based on the inputs $\hat{\boldsymbol{f}}_k$ and $\hat{\boldsymbol{\omega}}_k$ as well as on the estimate of the previous state $\hat{\boldsymbol{x}}_{k-1}$.

In continuous time, the derivative of the position in robocentric coordinates is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t}\, {}_M\boldsymbol{r}_{IM} = {}_M\boldsymbol{v}_M - {}_M\boldsymbol{\omega}_{IM} \times {}_M\boldsymbol{r}_{IM} \tag{10}$$

Using euler forward integration, we can derive the discrete time prediction equation for the position $\boldsymbol{r}$ as a function of the input signal $\hat{\boldsymbol{\omega}}_k$ and the previous state:

$$\hat{\boldsymbol{r}}_k^- = \hat{\boldsymbol{r}}_{k-1} + \Delta t \cdot (\hat{\boldsymbol{v}}_{k-1} - \hat{\boldsymbol{\omega}}_k \times \hat{\boldsymbol{r}}_{k-1}) \tag{11}$$

Similarly, the derivative of the velocity is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\,{}_M\boldsymbol{v}_M = {}_M\boldsymbol{a}_M - {}_M\boldsymbol{\omega}_{IM} \times {}_M\boldsymbol{v}_M\,, \tag{12}$$

which can be used with the bias corrected acceleration measurements $\hat{\boldsymbol{f}}_k$ to obtain:

$$\hat{\boldsymbol{v}}_k^- = \hat{\boldsymbol{v}}_{k-1} + \Delta t \cdot (\hat{\boldsymbol{f}}_k + \hat{\boldsymbol{R}}_{k-1}\,{}_I\boldsymbol{g} - \hat{\boldsymbol{\omega}}_k \times \hat{\boldsymbol{v}}_{k-1}). \tag{13}$$

Finally, we derive the prediction equation for the transformation matrix. The derivative of this matrix is defined as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{R}_{MI} = \lfloor {}_M\boldsymbol{\omega}_{MI}\times\rfloor\,\boldsymbol{R}_{MI} = -\lfloor {}_M\boldsymbol{\omega}_{IM}\times\rfloor\,\boldsymbol{R}_{MI}\,, \tag{14}$$

and solving the matrix differential equations yields:

$$\hat{\boldsymbol{R}}_k^- = \exp(-\Delta t \cdot \lfloor \hat{\boldsymbol{\omega}}_k\times\rfloor)\,\hat{\boldsymbol{R}}_{k-1} \tag{15}$$

In an extended Kalman filter setup, stochastic arguments define the filter output. It is therefore required to track the covariances of the state and the measurements. The state covariance is given as

$$\boldsymbol{P}_k = \mathrm{Cov}\,(\delta\boldsymbol{x}_k) = \mathrm{Cov}\left(\begin{bmatrix}\delta\boldsymbol{r}_k & \delta\boldsymbol{v}_k & \delta\boldsymbol{\theta}_k\end{bmatrix}^T\right) \tag{16}$$

with the error terms $\delta\boldsymbol{x}_k$ being defined by the following three equations.

$$\hat{\boldsymbol{r}}_k = \boldsymbol{r}_k + \delta\boldsymbol{r}_k \tag{17}$$

$$\hat{\boldsymbol{v}}_k = \boldsymbol{v}_k + \delta\boldsymbol{v}_k \tag{18}$$

$$\hat{\boldsymbol{R}}_k = \boldsymbol{R}_k\,(I - \lfloor\delta\boldsymbol{\theta}_k\times\rfloor) \tag{19}$$

Equation (19) does take into account that rotation matrices live on $\mathrm{SO}(3)$: To represent uncertainties of a rotational quantity, the three dimensional error rotation vector $\delta\boldsymbol{\theta}_k$ is used [6]. The small error transformation matrix $(\boldsymbol{I} - \lfloor\delta\boldsymbol{\theta}\times\rfloor)$ is post-multiplied to the true transformation matrix. In contrast to a pre-multiplication, this isolates the non-observable angle about the gravity axis (see Section III).

In order to write the covariance of the state prediction, the Jacobians of (11), (13) and (15) must be calculated with respect to the previous state and the inputs. Using the error term notation (17)-(19) and the identities (2) and (3), the Jacobians can be defined and evaluated [6]:

$$\begin{bmatrix}\delta\boldsymbol{r}_k\\ \delta\boldsymbol{v}_k\\ \delta\boldsymbol{\theta}_k\end{bmatrix} = A_k \begin{bmatrix}\delta\boldsymbol{r}_{k-1}\\ \delta\boldsymbol{v}_{k-1}\\ \delta\boldsymbol{\theta}_{k-1}\end{bmatrix} + B_k \begin{bmatrix}\delta\boldsymbol{f}_k\\ \delta\boldsymbol{\omega}_k\end{bmatrix} \tag{20}$$

with

$$\boldsymbol{A}_k = \begin{bmatrix}\boldsymbol{I} - \Delta t\,\lfloor\hat{\boldsymbol{\omega}}_k\times\rfloor & \Delta t\,\boldsymbol{I} & 0\\ 0 & \boldsymbol{I} - \Delta t\,\lfloor\hat{\boldsymbol{\omega}}_k\times\rfloor & \Delta t\,\hat{\boldsymbol{R}}_{k-1}\,\lfloor {}_I\boldsymbol{g}\times\rfloor\\ 0 & 0 & \boldsymbol{I}\end{bmatrix} \tag{21}$$

and

$$\boldsymbol{B}_k = \begin{bmatrix}0 & \Delta t\,\lfloor\hat{\boldsymbol{r}}_{k-1}\times\rfloor\\ \Delta t\,\boldsymbol{I} & \Delta t\,\lfloor\hat{\boldsymbol{v}}_{k-1}\times\rfloor\\ 0 & \Delta t\,\hat{\boldsymbol{R}}_{k-1}^T\end{bmatrix}. \tag{22}$$

With this, we can now calculate the covariance of the predicted states according to:

$$\boldsymbol{P}_k^- = \boldsymbol{A}_k\,\boldsymbol{P}_{k-1}\,\boldsymbol{A}_k^T + \boldsymbol{B}_k\,\boldsymbol{Q}\,\boldsymbol{B}_k^T + \boldsymbol{Q}_n. \tag{23}$$

Hereby, $\boldsymbol{Q}$ denotes the input noise matrix, which is derived from the noise properties of the gyroscopes and accelerometers and is given as

$$\boldsymbol{Q} = \begin{bmatrix}\mathrm{Cov}\,(\boldsymbol{n}_a) & 0\\ 0 & \mathrm{Cov}\,(\boldsymbol{n}_g)\end{bmatrix}. \tag{24}$$

Since we are employing an Euler forward method, it is implicitly assumed that the integrated values are constant throughout each integration step. Since this does not hold in reality and the impact of this approximation – particularly for velocity prediction – is assumed to be significant, an additional noise term $\boldsymbol{Q}_n$ is added to account for integration errors.

D. Update Equations

In the update step of a standard EKF, the difference between expected and actual measurements (also called *innovation*) is used in order to refine the estimate of the state. For the given system, the problem arises that with a simple point contact model of the Ballbot, the expected encoder measurements cannot be explicitly computed (the vector product in (28) is not invertible). To circumvent this, we use a better suited, *virtual* measurement of the ball velocity ${}_M\boldsymbol{v}_B$ which is obtained as a function of the robot's states and measurements. The corresponding innovation is hence defined as the difference between two equations for the ball velocity; one relying solely on the filter states, and the other taking into account the sensor readings from the omniwheels.

Using simple rigid body kinematics, we can write for the first equation

$${}_M\boldsymbol{v}_B = {}_M\boldsymbol{v}_M + {}_M\boldsymbol{\omega}_{IM} \times {}_M\boldsymbol{r}_{MB}\,. \tag{25}$$

The second equation is based on the encoder values [5]. For its derivation, we start with the regular matrix $\boldsymbol{\Gamma}$, which, in the absence of slipping between omniwheels and ball, maps the ball angular rate ${}_M\boldsymbol{\omega}_{MB}$ relative to the Ballbot's body to the omniwheel angular rates $\hat{\boldsymbol{\Omega}}$.

$$\hat{\boldsymbol{\Omega}} = \boldsymbol{\Gamma}\,{}_M\boldsymbol{\omega}_{MB} \tag{26}$$

The matrix $\boldsymbol{\Gamma}$ is only a function of the geometric configuration of the ball-drive mechanism and therefore constant for a given Ballbot.

The angular rate of the ball relative to the inertial frame is:

$${}_M\boldsymbol{\omega}_{IB} = {}_M\boldsymbol{\omega}_{IM} + {}_M\boldsymbol{\omega}_{MB} = {}_M\boldsymbol{\omega}_{IM} + \boldsymbol{\Gamma}^{-1}\,\hat{\boldsymbol{\Omega}}\,. \tag{27}$$

Again, assuming no slipping, the velocity of the center of the ball can be computed from the cross product with the vector from the ground to the center of the ball:

$${}_M\boldsymbol{v}_B = {}_M\boldsymbol{\omega}_{IB}\times{}_M\boldsymbol{r}_{GB} = ({}_M\boldsymbol{\omega}_{IM} + \boldsymbol{\Gamma}^{-1}\,\hat{\boldsymbol{\Omega}})\times{}_M\boldsymbol{r}_{GB}\,. \tag{28}$$

If we further assume that the Ballbot is rolling on an even surface, the vector ${}_I\boldsymbol{r}_{GB}$ is constant and ${}_M\boldsymbol{r}_{GB} = \boldsymbol{R}_{MI}\,{}_I\boldsymbol{r}_{GB}$ is computed straightforwardly.

With (25) to (28), the virtual innovation can be stated as

$$\begin{aligned}\boldsymbol{y}_k = &(\hat{\boldsymbol{\omega}}_k + \boldsymbol{\Gamma}^{-1}\hat{\boldsymbol{\Omega}}_k) \times (\hat{\boldsymbol{R}}_k^- \, {}_I\boldsymbol{r}_{GB}) \\ &-(\hat{\boldsymbol{v}}_k^- + \hat{\boldsymbol{\omega}}_k \times {}_M\boldsymbol{r}_{MB}) \,. \end{aligned} \tag{29}$$

NOTE: the resulting virtual innovation is not only a function of the predicted state estimate $\hat{\boldsymbol{x}}_k^-$ and measurements $\hat{\boldsymbol{\Omega}}_k$, but also depends on the inputs used for prediction (i.e., $\hat{\boldsymbol{\omega}}_k$). This is in conflict with the assumption of a Kalman filter that the stochastic noise on the input is statistically independent from the measurements noise. The effect of this issue on the filter performance is expected to be low and therefore not further investigated in the scope of this paper.

Similarly to the computation of the Jacobians of the prediction equations, we again use infinitesimal errors on the states and measurements for the evaluation of the Jacobians of the update equation. This yields

$$\delta\boldsymbol{y}_k = \boldsymbol{C}_k \begin{bmatrix} \delta\boldsymbol{r}_{k-1} \\ \delta\boldsymbol{v}_{k-1} \\ \delta\boldsymbol{\theta}_{k-1} \end{bmatrix} + \boldsymbol{D}_k \begin{bmatrix} \delta\boldsymbol{\omega}_k \\ \delta\boldsymbol{\Omega}_k \end{bmatrix} \tag{30}$$

with

$$\boldsymbol{C}_k = \begin{bmatrix} 0 & -\boldsymbol{I} & \boldsymbol{c}_{3,k} \end{bmatrix} , \tag{31}$$

$$\boldsymbol{c}_{3,k} = \left(\lfloor \boldsymbol{\Gamma}^{-1} \, \hat{\boldsymbol{\Omega}}_k \times \rfloor + \lfloor \hat{\boldsymbol{\omega}}_k \times \rfloor \right) \hat{\boldsymbol{R}}_k^- \, \lfloor {}_I\boldsymbol{r}_{GB} \times \rfloor \tag{32}$$

and

$$\boldsymbol{D}_k = \begin{bmatrix} \boldsymbol{d}_{1,k} & -\lfloor \hat{\boldsymbol{R}}_k^- \, {}_I\boldsymbol{r}_{GB} \times \rfloor \, \boldsymbol{\Gamma}^{-1} \end{bmatrix} , \tag{33}$$

$$\boldsymbol{d}_{1,k} = \lfloor {}_M\boldsymbol{r}_{MB} \times \rfloor - \lfloor \hat{\boldsymbol{R}}_k^- \, {}_I\boldsymbol{r}_{GB} \times \rfloor \,. \tag{34}$$

With this, the innovation residual covariance is given by

$$\boldsymbol{S}_k = \boldsymbol{C}_k \, \boldsymbol{P}_k^- \, \boldsymbol{C}_k^T + \boldsymbol{D}_k \, \boldsymbol{R} \, \boldsymbol{D}_k^T \,, \tag{35}$$

where $\boldsymbol{R}$ denotes the measurement noise matrix, which is obtained from the encoder and gyroscope noise properties

$$\boldsymbol{R} = \begin{bmatrix} \mathrm{Cov}\,(\boldsymbol{n}_g) & 0 \\ 0 & \mathrm{Cov}\,(\boldsymbol{n}_e) \end{bmatrix} . \tag{36}$$

The Kalman gain $\boldsymbol{K}_k$ is calculated and partitioned according to

$$\boldsymbol{K}_k = \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{r},k}^T & \boldsymbol{K}_{\boldsymbol{v},k}^T & \boldsymbol{K}_{\boldsymbol{R},k}^T \end{bmatrix}^T = \boldsymbol{P}_k^- \, \boldsymbol{C}_k^T \, \boldsymbol{S}_k^{-1} \,, \tag{37}$$

which leads to the following update equations for the state vector $\hat{\boldsymbol{x}}_k$ and state covariance $\boldsymbol{P}_k$:

$$\hat{\boldsymbol{r}}_k = \hat{\boldsymbol{r}}_k^- - \boldsymbol{K}_{\boldsymbol{r},k} \, \boldsymbol{y}_k \tag{38}$$

$$\hat{\boldsymbol{v}}_k = \hat{\boldsymbol{v}}_k^- - \boldsymbol{K}_{\boldsymbol{v},k} \, \boldsymbol{y}_k \tag{39}$$

$$\hat{\boldsymbol{R}}_k = \hat{\boldsymbol{R}}_k^- \, \exp(\lfloor \boldsymbol{K}_{\boldsymbol{R},k} \, \boldsymbol{y}_k \times \rfloor) \tag{40}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \, \boldsymbol{C}_k) \, \boldsymbol{P}_k^- \tag{41}$$

III. Observability Analysis

Since non-observable states can strongly affect the performance of a filter, it is necessary to prove that this is not the case with the presented algorithm. Intuitively, it is to be expected that the position and the yaw angle (rotation around the gravity axis) of the Ballbot are not observable, but that the drift in these states will not impede the general function of the filter. This is proven formally in the remainder of this section.

We start by analyzing the observability of the time-varying discrete linear system at hand. For this, we apply the concept of local observability matrices [7], in which the subspace spanned by the rows of a local observability matrix is analyzed:

$$\boldsymbol{O}_k = \begin{bmatrix} \boldsymbol{C}_k \\ \boldsymbol{C}_{k+1}\boldsymbol{A}_k \\ \boldsymbol{C}_{k+2}\boldsymbol{A}_{k+1}\boldsymbol{A}_k \\ \boldsymbol{C}_{k+3}\boldsymbol{A}_{k+2}\boldsymbol{A}_{k+1}\boldsymbol{A}_k \\ \vdots \end{bmatrix} . \tag{42}$$

For the proposed filter this yields:

$$\boldsymbol{O}_k = \begin{bmatrix} 0 & \boldsymbol{I} & \boldsymbol{O}_{1,k}\lfloor {}_I\boldsymbol{g} \times \rfloor \\ 0 & 0 & \boldsymbol{O}_{2,k}\lfloor {}_I\boldsymbol{g} \times \rfloor \\ \vdots & \vdots & \vdots \end{bmatrix} . \tag{43}$$

In general, the matrices $\boldsymbol{O}_{i,k}$ are non-singular matrices and shall not be further discussed in the scope of this paper. For infinitesimal disturbance, $\delta\boldsymbol{x}$, we can now analyze the effect $\boldsymbol{O}_k\delta\boldsymbol{x}$ on the measured outputs. If the effect is equal to zero, the perturbation belongs to the unobservable subspace of the filter states. For further considerations we thus evaluate the nullspace of the observability matrix $\boldsymbol{O}_k$, which is equivalent to the column space of the matrix

$$\boldsymbol{U}_k = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & 0 \\ 0 & {}_I\boldsymbol{g} \end{bmatrix} . \tag{44}$$

It satisfies $\boldsymbol{0} = \boldsymbol{O}_k\boldsymbol{U}_k$. The first column corresponds to the unobservable absolute position while the second represents rotations around the gravity axis ($\lfloor {}_I\boldsymbol{g} \times \rfloor \, {}_I\boldsymbol{g} = \boldsymbol{0}$). The remaining degrees of freedom with respect to attitude as well as the 3D velocity of the robot are fully observable.

Since we have chosen robocentric coordinates, the drift in the aforementioned states does not influence the other states during prediction. This can be checked by regarding (13) and (15), in which the position states do not appear. Similarly, a drift in yaw $\boldsymbol{R}_z$ affect neither the prediction of velocities nor that of rotations other than around the yaw-axis:

$$\begin{aligned} \hat{\boldsymbol{v}}_{k,dist} &= \hat{\boldsymbol{v}}_{k-1} + \Delta t \cdot (\hat{\boldsymbol{f}}_k + \hat{\boldsymbol{R}}_{k-1}\boldsymbol{R}_z \, \boldsymbol{g} - \hat{\boldsymbol{\omega}}_k \times \hat{\boldsymbol{v}}_{k-1}) \\ &= \hat{\boldsymbol{v}}_{k-1} + \Delta t \cdot (\hat{\boldsymbol{f}}_k + \hat{\boldsymbol{R}}_{k-1} \, \boldsymbol{g} - \hat{\boldsymbol{\omega}}_k \times \hat{\boldsymbol{v}}_{k-1}) \\ &= \hat{\boldsymbol{v}}_k \end{aligned} \tag{45}$$

$$\begin{aligned} \hat{\boldsymbol{R}}_{k,dist}^- &= \exp(-\Delta t \cdot \lfloor \hat{\boldsymbol{\omega}}_k \times \rfloor) \, \hat{\boldsymbol{R}}_{k-1}\boldsymbol{R}_z \\ &= \hat{\boldsymbol{R}}_k^- \, \boldsymbol{R}_z \end{aligned} \tag{46}$$

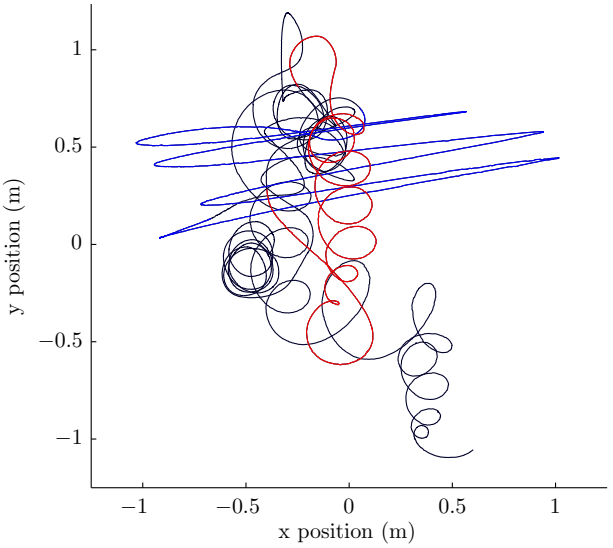

Fig. 3. Planar trajectory of the Ballbot during 190 s of driving. The approximated linear motion from 55 to 80 s (blue) and the circular motion from 110 to 135 s (red) will be used to evaluate performance.

IV. Experimental Setup and Results

To evaluate the performance of the presented filter, estimates from an exemplary trajectory (Fig. 3) were recorded over 190 seconds in which the Ballbot performed different kinds of motions and stationary balancing. The estimates are compared to data from a motion capture system, from which velocity was obtained by numerical differentiation and smoothing. The non-observable filter states, i.e., position and yaw angle, were initialized with ground truth measurements. Their variances were initialized to be zero. The observable states were initially set to zero as they would converge towards the true state. The velocity variances were initialized to $0.25\,\mathrm{m}^2/\mathrm{s}^2$ and the roll and pitch angle variances were set to $0.01\,\mathrm{rad}^2$. For better visualization and quantization of the EKF performance, attitudes are parameterized by a roll-pitch-yaw Euler sequence.

For two 25 s segments of this trajectory[1], the estimates of attitude and velocity are compared to ground truth. From 55 to 80 s the Ballbot was driving back and forth along a straight line. The corresponding attitude estimation is shown in (Fig. 4) and the velocity estimation in (Fig. 5). As can be seen, the error between estimated and motion capture state is extremely small (compare average errors given in the figure captions).

From 110 to 135 s the Ballbot was driving in circles (Fig. 6) and (Fig. 7). The velocity estimates exhibit a roughly constant offset from the true velocity. Since circle driving requires high accelerations towards the center of the circle, high forces are acting on the ball and it is suspected that this deviation is caused by slippage of the omniwheels and the ball. To check this assumption, the filter was run off-line using simulated encoder values, which were based on motion capture data and thus guaranteed to be slip-free. The corresponding velocity estimates did indeed not exhibit any offset (Fig. 8), which confirms our interpretation of the cause of the offset.

[1]See the accompanying video: http://youtu.be/Xyt0Oqlcrcw

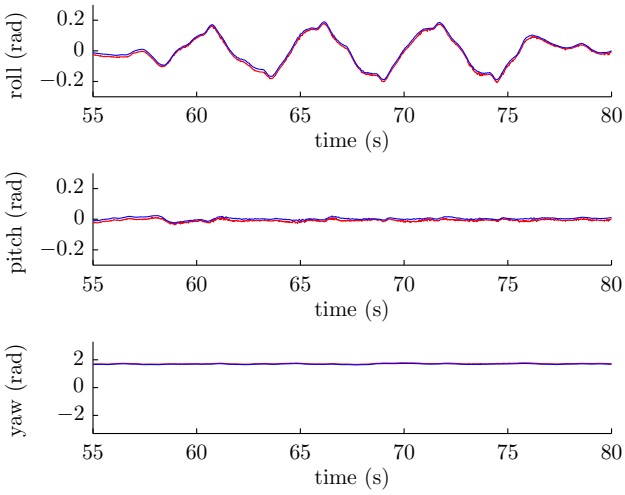

Fig. 4. EKF attitude estimation (blue) and ground truth (red) during linear driving. From top to bottom: Euler angle sequence roll, pitch and yaw. Average errors for roll: 0.0115 rad, pitch: 0.0112 rad and yaw: 0.0354 rad.

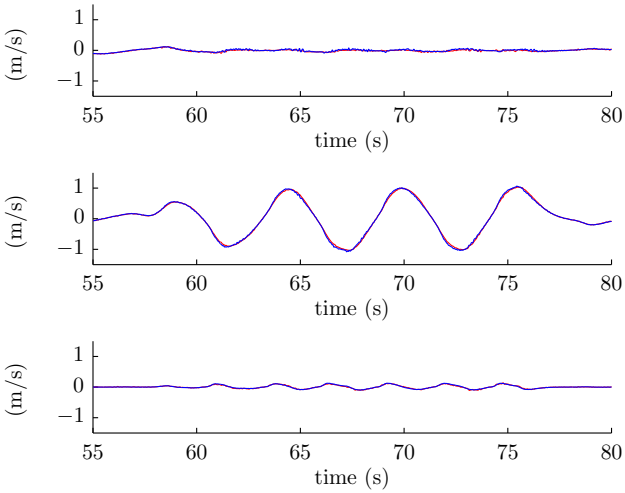

Fig. 5. EKF velocity estimation (blue) and ground truth (red) during linear driving. From top to bottom: velocity in body-fixed x, y and z direction. Average errors for x: 0.0163 m/s, y: 0.0203 m/s and z: 0.0082 m/s.

In Table I, the average errors of the estimated states over the full 190 s are shown. Due to the drift of unobservable states, estimates of position and yaw angle exhibit rather high average errors.

TABLE I
Average errors, total 190 s driving sequence

axis	position	velocity	angle	Euler angle
x	0.6617 m	0.0233 m/s	roll	0.0098 rad
y	0.4957 m	0.0138 m/s	pitch	0.0103 rad
z	0.0427 m	0.0077 m/s	yaw	0.1321 rad

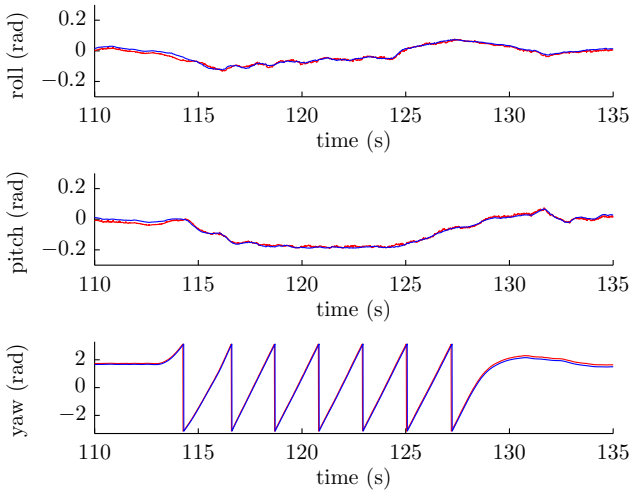

Fig. 6. EKF attitude estimation (blue) and ground truth (red) during circular driving. From top to bottom: Euler angle sequence roll, pitch and yaw. Average errors for roll: 0.0072 rad, pitch: 0.0074 rad and yaw: 0.1608 rad.

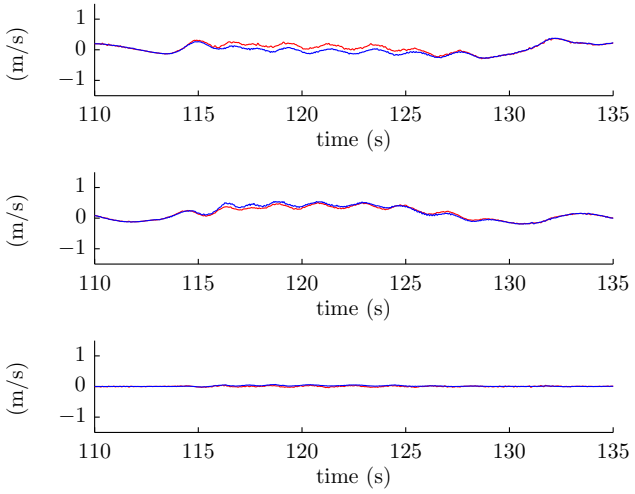

Fig. 7. EKF velocity estimation (blue) and ground truth (red) during circular driving. From top to bottom: velocity in body-fixed x, y and z direction. Average errors for x: 0.0659 m/s, y: 0.0341 m/s and z: 0.0159 m/s.

V. Summary and Conclusion

This paper presents a state estimation approach for a Ballbot. An unified extended Kalman filter exploits the information obtained from on-board accelerometers, gyroscopes, and encoders in order to produce estimates of the robot's position, velocity and attitude. The consistency of the approach was tested by means of an observability analysis and the performance was evaluated on a real Ballbot. Apart from exhibiting a slight offset for high slippage situations, the obtained estimation errors are very small and estimated states are sufficiently accurate for feedback control. To improve the performance, slippage between ball and omniwheels could be eliminated by mounting a separate sensor measuring the ball

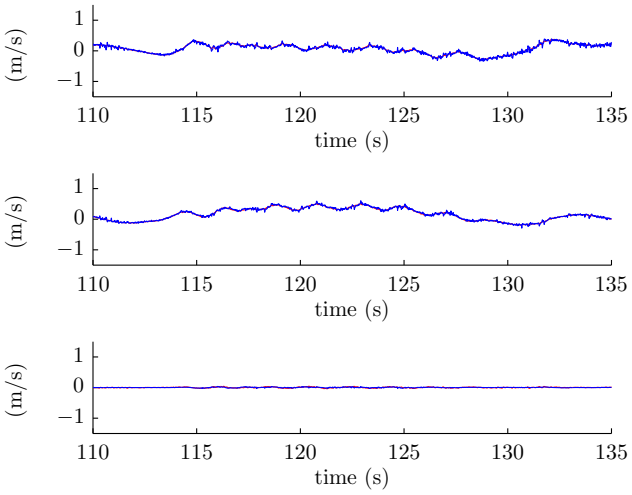

Fig. 8. For this figure, the EKF has been run off-line with simulated encoder values based on ground truth data to eliminate possible slippage effects between ball and ground and between omniwheels and ball as a cause for filter errors. EKF velocity estimates (blue) and ground truth (red) during circle driving. From top to bottom: velocity in body-fixed x, y and z direction. Average errors for x: 0.0247 m/s, y: 0.0191 m/s and z: 0.0097 m/s.

velocity [8] or by including additional sensory data, e.g., vision or laser ranger measurements. The modular structure of the filter would facilitate such an inclusion which would also allow for a wide range of application-oriented Ballbot research.

Acknowledgment

We would like to express our thanks to Michael Neunert and Péter Fankhauser for sharing their expertise on the Ballbot system.

References

[1] M. Neunert, P. Fankhauser, S. Leutenegger, C. Pradalier, F. Colas, and R. Siegwart, "Ballbot Rezero: Mechanical design, system modeling and control," in *IEEE Robotics and Automation Magazine*, 2013, in preparation.

[2] U. Nagarajan, A. Mampetta, G. A. Kantor, and R. L. Hollis, "State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot," in *Robotics and Automation, IEEE International Conference on*, May 2009, pp. 998 – 1003.

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

[4] C.-W. Liao, C.-C. Tsai, Y. Y. Li, and C.-K. Chan, "Dynamic modeling and sliding-mode control of a ball robot with inverse mouse-ball drive," in *SICE Annual Conference*, Aug. 2008, pp. 2951 – 2955.

[5] P. Fankhauser and C. Gwerder, "Modeling and Control of a Ballbot," Bachelor thesis, ETH Zurich, June 2010.

[6] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," University of Minnesota, Tech. Rep., Mar. 2005.

[7] Z. Chen, K. Jiang, and J. Hung, "Local observability matrix and its application to observability analyses," in *Industrial Electronics Society, 16th Annual Conference of IEEE*, vol. 1, Nov. 1990, pp. 100 – 103.

[8] M. Kumagai and R. Hollis, "Development of a three-dimensional ball rotation sensing system using optical mouse sensors," in *Robotics and Automation, IEEE International Conference on*, May 2011, pp. 5038 – 5043.