Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Sliding mode control of a ball balancing robot ⋆

Ioana Lal * **Alexandru Codrean** ** **Lucian Buşoniu** **

* *Engineering Center Cluj, Robert Bosch and Technical University of Cluj-Napoca, Romania (e-mail: ioana.lal@ro.bosch.com).*
** *Technical University of Cluj-Napoca, Romania (e-mail: alexandru.codrean@aut.ucluj.ro, lucian@busoniu.net)*

Abstract: This paper presents a sliding mode control design for a ball-balancing robot (ballbot), with associated real-time results. The sliding mode control is designed based on the linearized plant model, and is robust to matched uncertainties. The design is considerably simpler than other nonlinear control strategies presented in the literature, and the experimental results for stabilization and tracking show much better performances than those obtained with linear control (in particular, a linear quadratic regulator).

Keywords: Control applications, mobile robots, sliding mode control, robust control

1. INTRODUCTION

Using mobile robots in closed spaces – like offices or homes – has received considerable attention in the last few years. One of the main tasks for such robots is to move in an agile and stable manner in narrow environments, between people or other possible obstacles. With this goal in mind, lately the so called ballbots (ball balancing robots) have been developed, which can move easily in any direction in the plane (Lauwers et al., 2006; Fankhauser and Gwerder, 2010; Pham et al., 2018; Nagarajan and Hollis, 2013; Hoshino et al., 2013). The most often encountered design is that from Figure 1, where the robot is actuated by three motors via omni-directional wheels that are in contact with the ball. The control objective is to keep the robot around the unstable equilibrium point (vertical position), while ensuring that it moves according to the desired trajectory.

Ballbots fall into the class of underactuated electromechanical systems (3 actuators, 5 degrees of freedom), with complex nonlinear dynamics, unstable and non-minimum phase, with kinematic and controllability constraints. Moreover, there is significant uncertainty in the models, due to different modeling simplifications, friction forces that are not considered, noise and varying load torques. Linear controllers designed based on the linearized model around the unstable equilibrium point work only for low velocities and are not robust to uncertainties or external disturbances. Nonlinear control strategies are thus required in order to surpass these limitations.

In practice, however, most experiments are reported with PID or LQR controllers (or both in a cascaded loop) (Lauwers et al., 2006; Fankhauser and Gwerder, 2010;

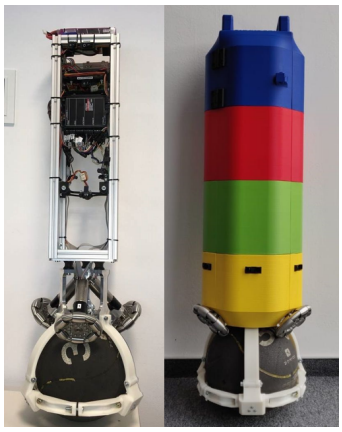

Fig. 1. The ballbot, without and with cover

Pham et al., 2018; Nagarajan and Hollis, 2013; Blonk, 2014). The wast majority of nonlinear controllers reported for ballbots are tested only through simulations. The only exceptions that we are aware of are Fankhauser and Gwerder (2010) and Pham and Lee (2018). Fankhauser and Gwerder (2010) linearize the model in different operating points, and then design a gain-scheduling LQR. Although the design is pretty straightforward, it is not obvious how to choose the operating points, and the interpolation between different controllers requires a lot of a memory. Also, no stability guarantees are provided. Pham and Lee (2018) design a hierarchical sliding mode controller based on the nonlinear dynamical model of the robot, and prove asymptotic stability using Lyapunov functions. Due to the complexity of the model, the calculations are fairly elaborate, and involve significant online computation in generating the control signals at each sampling period.

In contrast, the current paper aims to provide a nonlinear control strategy directed towards control practitioners, that is easy to tune and adapt to different types of ballbots and control scenarios (tracking, stabilization, disturbance rejection), with relatively low online computational effort. To that end, we propose a sliding mode control based on

⋆ This work was supported by a grant of the Romanian Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2016-0670, within PNCDI III, and by the Post-Doctoral Programme "Entrepreneurial competences and excellence research in doctoral and postdoctoral programs - ANTREDOC", project co-funded from European Social Fund, contract no. 56437/24.07.2019.

the *linearized* model of the process. We also introduce an adaptation law for the gain corresponding to the discontinuous part of the control law, based on the reference velocity. Although the approach has only local stability guarantees, experimental results show that it surpasses linear control. Another argument for using the linear model is because in deriving a nonlinear dynamic model for the robot, different simplifications and approximations are always made, like ignoring the coupling between planes or the friction components. We believe that the linearized model is more reliable and much easier to validate experimentally than the nonlinear one. Finally, the control exhibits robustness to matched model uncertainties and external disturbances.

Our previous paper (Lal et al., 2019) focuses mainly on the hardware construction and electronics of the robot, and also presents the design of a linear quadratic regulator and balancing experimental results. The present paper fully focuses on control, and provides a novel sliding mode control design, for both stabilization and tracking.

The paper has the following structure: first, in Section 2 the methodology is presented, namely the control methods used throughout this paper. Then, the mathematical model of the ballbot is introduced in Section 3, and later the design of the controllers and the experimental results using them are shown in Section 4. In the end, in Section 5 some conclusions are given.

2. METHODOLOGY

In this section we present the control methods used throughout the paper. First, the LQR design is presented, which will be used as a baseline for comparison. Then, a nonlinear control technique, namely the sliding mode control is introduced. This is the main focus of our paper.

2.1 LQR for stabilization

Consider a linear state space model of the form:

$$\dot{x} = Ax + Bu \tag{1}$$

where x is the state vector, u is the control vector, $A \in \mathbf{R}^{n\times n}$ and $B \in \mathbf{R}^{n\times m}$. We consider the cost function

$$J = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)dt \tag{2}$$

where $Q \in \mathbf{R}^{n\times n}$ is positive-definite or positive-semidefinite, and $R \in \mathbf{R}^{m\times m}$ must be positive-definite hermitian or real symmetric. We want to minimize the cost function. For that, it is sufficient to apply the control law $u = -Kx$, where $K = R^{-1}B^T P$ and P is the stabilizing solution to the Ricatti equation (Ogata and Yang, 2002):

$$Q + A^T P + PA - PBR^{-1}B^T P = 0 \tag{3}$$

2.2 Sliding mode control for reference tracking

In this subsection, we introduce sliding mode control (SMC), a nonlinear robust control method (Edwards and Spurgeon, 1998). The principle of SMC is to drive the states of the system to a surface in the state space, named the sliding surface and then keep them on this surface.

Let us consider a nominal linear system of the form:

$$\begin{aligned} \dot{x} &= Ax + Bu + Bd_m \\ y &= Cx \end{aligned} \tag{4}$$

where $C \in \mathbf{R}^{p\times n}$. d_m represents an unknown matched uncertainty, with known upper bound $\|d_m\| \leq k_u\|u\| + \alpha(t,x)$, where $0 < k_u < \sqrt{\lambda_{min}(B^T B)}$. Here, $\lambda_{min}(\cdot)$ represents the minimum eigenvalue. System (4) is assumed to be square and the pair (A, B) is assumed to be in regular form. This means that the matrix B can be written as $B = [0_{(n-m)\times m}, B_2]^T$. All the signals are functions of time, but we omit the time argument, for simplicity. We shall consider an additional state $x_r \in \mathbf{R}^p$, which satisfies:

$$\dot{x}_r = r - y \tag{5}$$

Here, tracking reference signal r must be differentiable and must satisfy $\dot{r} = \Gamma(r - R)$. Γ is a stable design matrix and R is a constant. Let us now define the augmented system state, and its partition as follows:

$$\tilde{x} = \begin{bmatrix} x_r \\ x \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{6}$$

with $x_1 \in \mathbf{R}^n$ and $x_2 \in \mathbf{R}^p$. Note that $x_r \neq x_1$ and $x \neq x_2$. We also consider the following partitions for A and C:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},\ C = [C_1\ C_2] \tag{7}$$

with $A_{11} \in \mathbf{R}^{(n-p)\times(n-p)}$, $A_{12} \in \mathbf{R}^{(n-p)\times p}$, $A_{21} \in \mathbf{R}^{p\times(n-p)}$, $A_{22} \in \mathbf{R}^{p\times p}$, $C_1 \in \mathbf{R}^{p\times(n-p)}$ and $C_2 \in \mathbf{R}^{p\times p}$. We can now write the augmented system dynamics as:

$$\begin{aligned} \dot{x}_1 &= \tilde{A}_{11}x_1 + \tilde{A}_{12}x_2 + B_r r \\ \dot{x}_2 &= \tilde{A}_{21}x_1 + \tilde{A}_{22}x_2 + B_2 u + d_m \end{aligned} \tag{8}$$

where

$$\begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} = \left[\begin{array}{cc|c} 0 & -C_1 & -C_2 \\ 0 & A_{11} & A_{12} \\ \hline 0 & A_{21} & A_{22} \end{array}\right], \qquad B_r = \begin{bmatrix} I_p \\ 0 \end{bmatrix}. \tag{9}$$

The sliding surface is defined as:

$$\mathcal{S} = \{\tilde{x} \in \mathbf{R}^{n+p} : Sx - S_r r = 0\} \tag{10}$$

with S and S_r being design parameters which give the reduced-order motion.

We use another partition, corresponding to that in (6), for the hyperplane system matrix ($S_1 \in \mathbf{R}^n$ and $S_2 \in \mathbf{R}^p$):

$$S = [S_1\ S_2] \tag{11}$$

We further define the switching function

$$s(x, r) = Sx - S_r r = S_2 M x_1 + S_2 x_2 - S_r r. \tag{12}$$

where $M = S_2^{-1}S_1$ can be regarded as a state feedback gain that stabilizes the pair (A_{11},A_{12}) during ideal sliding mode (i.e. when $s = 0$). By assuming that $S_2 = \Lambda B_2^{-1}$, with Λ being a non singular diagonal design matrix, it results that, in order to have S, we only need to design the matrix M. This can easily be done through pole placement or quadratic minimization, as described by Edwards and Spurgeon (1998) – Ch. 4.

Consider the transformation

$$\begin{bmatrix} x_1 \\ s \end{bmatrix} = \begin{bmatrix} I & 0 \\ S_1 & S_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{13}$$

for which (8) becomes

$$\begin{aligned} \dot{x}_1 &= \bar{A}_{11}x_1 + \bar{A}_{12}s + B_r r \\ \dot{s} &= S_2\bar{A}_{21}x_1 + S_2\bar{A}_{22}S_2^{-1}s + \Lambda u + S_1 B_r r + S_2 d_m \end{aligned} \tag{14}$$

where $\bar{A}_{11} = \tilde{A}_{11} - \tilde{A}_{12}M$, $\bar{A}_{21} = M\bar{A}_{11} + \tilde{A}_{21} - A_{22}M$, $\bar{A}_{22} = M\tilde{A}_{12} + A_{22}$, and $\bar{A}_{12} = \tilde{A}_{12}S_2^{-1}$.

The control law has a linear and a nonlinear component:

$$u = u_L + u_N. \tag{15}$$

The linear part has both feedback and feedforward terms:

$$u_L = L\tilde{x} + L_r r + L_{\dot{r}}\dot{r}, \tag{16}$$

where the gains are computed as:

$$\begin{aligned} L &= -\Lambda^{-1}(S\tilde{A} - \Phi S), \\ L_r &= -\Lambda^{-1}(\Phi S_r + S_1 B_r), \\ L_{\dot{r}} &= \Lambda^{-1} S_r. \end{aligned} \tag{17}$$

The nonlinear part is a discontinuous term, given by

$$u_N = \begin{cases} -\rho_c \, \Lambda^{-1} \dfrac{\bar{P}_2(Sx - S_r r)}{\|\bar{P}_2(Sx - S_r r)\|} & \text{if } Sx \neq S_r r, \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

with $\bar{P}_2$ being symmetric positive definite and satisfies:

$$\bar{P}_2\Phi + \Phi^T\bar{P}_2 = -I \tag{19}$$

The gain ρ_c is designed based on the following proposition:

Proposition 1: The control law (15) induces a sliding motion despite the matched uncertainty if

$$\rho_c \geq \frac{\|S_2\|(k_u\|u_L\| + \alpha(t,x)) + \gamma}{1 - k_u\|\Lambda^{-1}\|\|\Lambda\|\|B_2^{-1}\|}, \tag{20}$$

where γ is a design parameter.

The proof follows directly from the argument of Edwards and Spurgeon (1998) – Sec. 3.6.1.

Proof: By using the Lyapunov function $V(s) = s^T\bar{P}_2 s$ for system (14), along with control law (15), it can be shown that if (20) holds, then

$$\rho_c \geq \|S_2\|(k_u\|u\| + \alpha(t,x)) + \gamma, \tag{21}$$

and the derivative of the Lyapunov function is negative:

$$\dot{V}(s) \leq -\|s\|^2 - 2\gamma\|\bar{P}_2 s\|. \tag{22}$$

Next, the mathematical model of the ballbot is presented.

3. SYSTEM DESCRIPTION AND MODEL

The ballbot is a vertical platform, mounted on a ball. Here, the driving mechanism is comprised of 3 brushless DC motors and 3 omni-wheels, which are distributed equidistantly at 120°, and fall perpendicular to the ball.

3.1 Nonlinear mathematical model

The mathematical model consists of a combination of three 2D models, which separate the planes YZ, XZ and XY. Two of these planes can be seen in Figure 2. It should be noted that the X axis is aligned with motor 1. Mathematical modeling is done with the Euler-Lagrange technique, based on the derivation in Pham et al. (2018). The minimal coordinates for the planes are:

$$q_x = \begin{bmatrix} y_k \\ \theta_x \end{bmatrix}, q_y = \begin{bmatrix} x_k \\ \theta_y \end{bmatrix}, q_z = [\theta_z] \tag{23}$$

where x_k and y_k give the ball's position on the floor, and θ_x, θ_y and θ_z are the Tait-Bryan angles of the body (Fankhauser and Gwerder, 2010). We use subscript x to

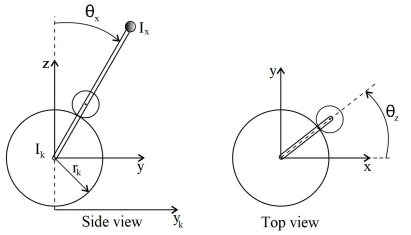

Fig. 2. The coordinates for the YZ and XY planes

denote the YZ plane, y for XZ and z for XY. The robot dynamics in the YZ plane are:

$$D(q_x)\ddot{q}_x + C(q_x,\dot{q}_x)\dot{q}_x + G(q_x) = B_x\tau_x \tag{24}$$

where:

$$\begin{aligned} D(q_x) &= \\ &\begin{bmatrix} m_k + \dfrac{I_k}{r_k^2} + m_a + \dfrac{3I_w\cos^2\alpha}{2r_w^2} & \dfrac{3I_w\cos^2\alpha}{2r_w^2}r_k - m_a l\cos\theta_x \\ \dfrac{3I_w\cos^2\alpha}{2r_w^2}r_k - m_a l\cos\theta_x & m_a l^2 + \dfrac{3I_w r_k^2\cos^2\alpha}{2r_w^2} + I_x \end{bmatrix} \\ C(q_x,\dot{q}_x) &= \begin{bmatrix} 0 & m_a l\dot{\theta}_x\sin\theta_x \\ 0 & 0 \end{bmatrix} \\ G(q_x) &= [0 \;\; -m_a g l\sin\theta_x]^T \\ B_x &= \begin{bmatrix} \dfrac{1}{r_w} & \dfrac{r_k}{r_w} \end{bmatrix}^T \end{aligned} \tag{25}$$

For the XZ plane, the dynamics are described by analogous equations. The dynamics for the horizontal plane are:

$$\ddot{\theta}_z = \frac{I_k}{I_k I_z + 3(I_k + I_z)I_w\frac{r_k^2}{r_w^2}\sin^2\alpha}\frac{r_k}{r_w}\tau_z \tag{26}$$

The inputs for the models τ_x, τ_y, τ_z are the equivalent torques for each plane, but since we need the model in terms of motor torques, a conversion must be applied. This conversion is represented by the following transformation:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \dfrac{2}{3\cos\alpha} & 0 & \dfrac{1}{3\sin\alpha} \\ -\dfrac{1}{3\cos\alpha} & \dfrac{\sqrt{3}}{3\cos\alpha} & \dfrac{1}{3\sin\alpha} \\ -\dfrac{1}{3\cos\alpha} & -\dfrac{\sqrt{3}}{3\cos\alpha} & \dfrac{1}{3\sin\alpha} \end{bmatrix} \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{27}$$

where τ_1, τ_2, τ_3 are the torques of each motor, respectively. See Pham et al. (2018) for further details regarding the model, and Lal et al. (2019) for the parameter values.

3.2 Linear model

We can observe that the developed model is nonlinear for the vertical planes. Nevertheless, we will apply both linear as well as nonlinear control techniques on the linearized model of the ballbot. Therefore, we must first linearize the previously derived model. We can choose any point in the horizontal plane, when the robot is upright. For simplicity, we shall consider the linearization around the point represented by the origin of the coordinate system, which to us is the initial position on the floor.

We linearize the two models for vertical planes separately, as in Lal et al. (2019). Therefore we have different state vectors and inputs corresponding to each of the planes. We shall present here just the linearization for the YZ plane, as it is analogous for XZ. Each state vector is represented

by the minimal coordinates and their velocities, and the input is the equivalent torque for that plane:

$$x_x = [q_x^T \ \dot{q}_x^T]^T, \ \ u_x = \tau_x \tag{28}$$

Considering that $\dot{x}_x = [\dot{q}_x^T \ddot{q}_x^T]^T$ and $\ddot{q}_x$ can be computed from (24), by multiplying to the left with $M^{-1}(q_x)$, we can now write the model of the system as:

$$\dot{x}_x = f_x(x_x, u_x) \tag{29}$$

with f_x being a nonlinear vector function. Its expression is complex, so it will not be provided here. The equilibrium point is $x_{x_0} = [0\ 0\ 0\ 0]^T, u_{x_0} = 0$. The linear model of the system becomes:

$$\dot{x}_x = A_x x_x + B_x u_x \tag{30}$$

where

$$A_x = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 3.4586 & 0 & 0 \\ 0 & 20.6773 & 0 & 0 \end{bmatrix}, \ B_x = \begin{bmatrix} 0 \\ 0 \\ 2.1950 \\ 2.7587 \end{bmatrix} \tag{31}$$

The equation for the horizontal plane was already linear, so by substituting parameters, considering the system state as $x_z = [q_z \ \dot{q}_z]^T$, and the input as τ_z, we get:

$$A_z = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ B_z = \begin{bmatrix} 0 \\ 50.0343 \end{bmatrix} \tag{32}$$

4. CONTROL DESIGN AND EXPERIMENTAL RESULTS

This section presents the design of the two controllers, LQR and SMC, and the experimental results obtained with them for balancing (stabilization), and then the ones with SMC for tracking. A sampling period of 0.01 seconds was adopted for both controllers.

4.1 LQR control for balancing

This subsection discusses the balancing results when the controller used is a linear quadratic regulator. First, based on the method described in Section 2.1, we design the three gains, for the three planes. In order to do this, we must choose suitable Q and R matrices corresponding to the planes. Matrix Q determines the weight of the states and R that of the input. Therefore, we must find a balance between how fast the states must reach their equilibrium state and the magnitude of the control signal. Since there is no precise rule when it comes to these weight matrices, we try several values. While in Lal et al. (2019) we also provided similar LQR stabilization results, the weight matrices here are different. The best results are with the following matrices:

$$\begin{gathered} Q_x = Q_y = \mathrm{diag}(35, 35, 1, 1) \quad R_x = R_y = 5 \\ Q_z = \mathrm{diag}(30, 5) \quad R_z = 10 \end{gathered} \tag{33}$$

With these weight matrices, the resulting gain vectors are:

$$\begin{gathered} K_x = [-2.6458 \ \ 32.8594 \ \ -3.1148 \ \ 7.8317] \\ K_y = [\,2.6458 \ \ 32.8594 \ \ 3.1148 \ \ 7.8317] \\ K_z = [1.7321 \ \ 0.7589] \end{gathered} \tag{34}$$

However, in practice, we observe that some gains are too high, making the robot too violent. Therefore, we make some adjustments. Specifically, for the vertical planes, we divide the gains that correspond to the inclination of the ballbot body (angles and angular velocities) by 2. Therefore, the new gains are:

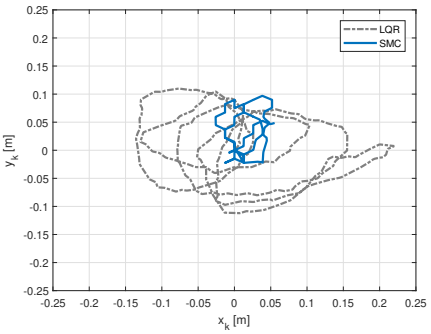

Fig. 3. LQR vs. SMC - stabilization: position in the plane

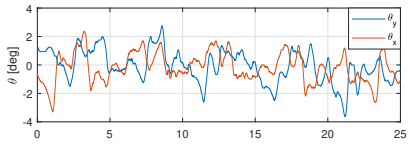

Fig. 4. LQR - stabilization: angles

$$\begin{gathered} K_x = [-2.6458 \ \ 16.4297 \ \ -3.1148 \ \ 3.9158] \\ K_y = [2.6458 \ \ 16.4297 \ \ 3.1148 \ \ 3.9158] \end{gathered} \tag{35}$$

The balancing results with these gains can be seen in Figure 3, in a dotted line, where we can observe the ballbot movement on the floor. The robot has circular movements around the initial point, which is due to the fact that robot always inclines to one side or another and must thus regain its balance. In order to do so, it has to move in the plane. In Figure 4 we can see the evolution of the angles in time. We notice that the robot is almost upright and only slightly inclines, up to roughly 3 degrees.

4.2 SMC control for balancing

This subsection shows the stabilizing result for the ballbot when a sliding mode controller is used. Recall that all the shown results are experimental. Since we only want the robot to balance around the initial point in the plane, we shall consider the reference signal to be $r(t) = \dot{r}(t) = 0$. We will only present the design of the controller for the YZ plane, as it is analogous for the XZ plane. For XY, we will keep the linear controller designed in Section 4.1, as the model for that plane is linear. First, we bring the system to a regular form, by applying the following transformation, $x_{reg} = T_r x$, determined with QR factorization, where:

$$T_r = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ -0.6585 & 0 & 0.56637 & -0.4956 \\ -0.7525 & 0 & -0.4956 & 0.4337 \\ 0 & 0 & -0.6585 & -0.7525 \end{bmatrix}. \tag{36}$$

The new system matrices are:

$$\begin{gathered} A_{reg} = \begin{bmatrix} 0 & -0.4956 & 0.4337 & -0.7525 \\ -7.7159 & -0.3729 & 0.3264 & 0.4337 \\ 6.7521 & -0.4262 & 0.3729 & 0.4956 \\ -17.5233 & 0 & 0 & 0 \end{bmatrix} \\ B_{reg} = \begin{bmatrix} 0 \\ 0.0000 \\ 0.0000 \\ -3.1564 \end{bmatrix} \end{gathered} \tag{37}$$

Now $B_2 = -3.1564$ and we can adopt any nonzero value for Λ. We choose it to be 1. Therefore, we will have $S_2 = B_2^{-1} = -0.3168$. We now derive the vector M using quadratic minimization (Shtessel et al., 2014), in order to find S_1. This vector is

$$M = [-1.0000 \quad -18.6849 \quad 1.0851 \quad -7.9555], \tag{38}$$

which leads to the following S vector:

$$S = [0.3168 \quad 5.9198 \quad -0.3438 \quad 2.5205 \quad -0.3168]. \tag{39}$$

We choose $\bar{P}_2$ to be 1, meaning that Φ is -0.5. In the end, this leads to:

$$L = [-0.1584 \quad -28.1826 \quad 3.8429 \quad -4.8937 \quad 3.5133]. \tag{40}$$

In order to reduce the chattering effect, due to the discontinuous term the control law, we replace (18) with:

$$u_N = \begin{cases} -\rho_c \Lambda^{-1} \dfrac{\bar{P}_2(Sx - S_r r)}{\|\bar{P}_2(Sx - S_r r)\| + \epsilon} & \text{if } Sx \neq S_r r, \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

where we adopt $\epsilon = 0.002$. This value has to be small enough to approximate the sign function well; we chose it based on a few experimental trials. The value of ρ_c should be high enough to stabilize the robot, but not too high, to not make it violent. Again, after running a few experiments we picked $\rho_c = 0.3$.

We first want to analyze the results with this controller for balancing (stabilization). These can be seen in Figure 3, in a continuous line. As it can be seen, the result with SMC is better than when using LQR, the robot remaining closer to the initial point on the floor.

We also want to test the robustness to a matched disturbance when using SMC. Therefore, we consider a scenario in which we give the robot an additive disturbance d_m on u_y at 5s. The disturbance is an impulse signal that lasts for 0.5s and has an amplitude of 0.5Nm. The results are shown in Figure 5, where we can observe the position of the ballbot in the plane. As we can see, the robot does not have greater oscillations than the initial case. Also, in Figure 6 we can see the evolution in time of the inclination of the ballbot. The maximum angles of inclination are smaller than when using LQR. Figure 7 shows an example of a command current for the first motor.

Figure 8 shows the switching functions s for the two planes. As we can see on s_y, there are certain oscillations around 0 in the beginning. Up until 5s, the oscillatory behaviour is due to the stabilization process, since we have non-zero initial conditions (θ_y starts from approximately 2°). Without the introduced disturbance, the oscillations would decrease and we would reach a small chattering effect around 0. However, due to the additive matched disturbance, the damping of these oscillations lasts longer. It is nonetheless difficult to highlight just the effect of this disturbance, since there are many other unmatched uncertainties (such as the coupling between the planes) and external perturbations (such as friction) that act simultaneously. Conclusively, the matched disturbance is rejected by the sliding mode controller.

4.3 SMC control for tracking

In this subsection, we present the results for trajectory tracking, when the implemented controller is a sliding

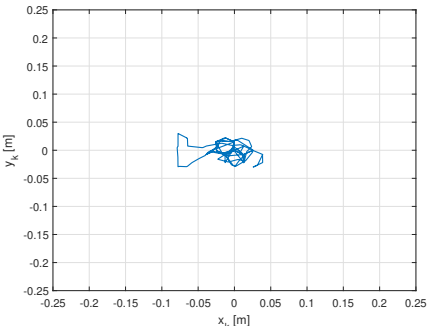

Fig. 5. SMC - stabilization with disturbance: position in the plane

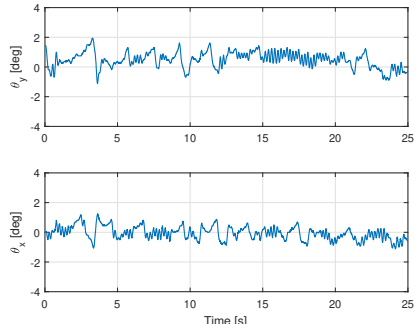

Fig. 6. SMC - stabilization with disturbance: angles

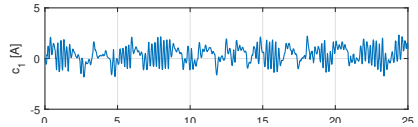

Fig. 7. SMC - stabilization with disturbance: command current for first motor

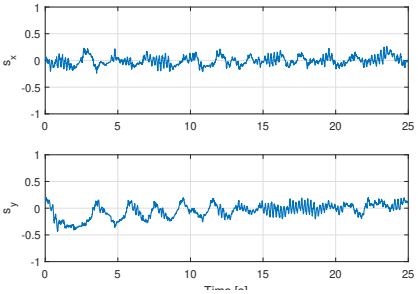

Fig. 8. SMC - stabilization with disturbance: switching function s

mode one. We are interested in seeing the robot follow a straight line in the XY plane, but also in the switching between the balancing phase and the tracking one. Therefore, we consider a ramp reference trajectory for the position in the plane, after an initial balancing phase. We choose a ramp trajectory for x_k and 0 for y_k. After reaching the desired point in the plane, the robot must then stabilize in that position. We will in the end have 2 switches: first between balancing and tracking, and later between tracking and balancing.

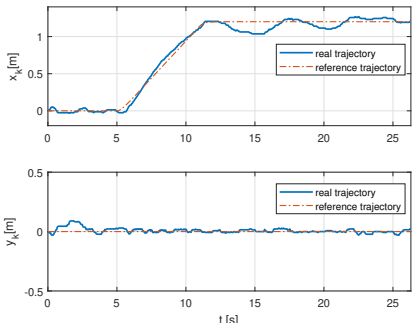

Fig. 9. SMC - tracking: Positions in time

We will now design the controller using the method presented in Section 2.2. Again, we only consider the YZ plane, as the model is analogous for the XZ plane. As introduced before, we must first bring the system to a regular form, which means applying the transformation in (36) and reaching the new system in (37). We consider:

$$y = y_k = Cx \tag{42}$$

with $C = [0,\ -0.6585,\ -0.7525,\ 0]$.

The matrix S is designed the same way as in the previous subsection, with the result already given in (39). S_r is chosen such that in a steady state regime, the state x_r is 0 (Edwards and Spurgeon, 1998, Sec. 7.3.4). In our case, this value is -1.6704. However, by running several tests, we notice that this value is too small, and the robot is moving too slowly. Therefore, we increase its value by multiplying with 3. The final value is $S_r = -1.6704 \cdot 3 = -5.0112$.

We choose the same $\bar{P}_2 = 1$, meaning $\Phi = -0.5$ and $\Lambda = 1$. Therefore, the value for L remains the same as the one computed before, in (40). In a tracking scenario, L_r and $L_{\dot{r}}$ are important as well, since they are the feedforward gains for the reference and its derivative. The computed values are $L_r = -2.8224, L_{\dot{r}} = -5.0112$.

An important parameter in our design is ρ_c. We notice that an increased value is desired for the stabilizing phase; however it makes the robot too violent when it must follow a line trajectory. However, a smaller value, suitable for the tracking phase, gives poor results when the robot balances in the same spot. Therefore, we propose a slow variation between values based on the reference velocity in the plane ($\dot{r}$), using the affine function $\rho_c = a|\dot{r}| + b$. Considering the maximum speed of the robot to be $0.2m/s$, and that in experiments we obtained the best results with $\rho_c = 0.3$ for balancing, and $\rho_c = 0.15$ for tracking at maximum speed, we get $a = -0.75$ and $b = 0.3$.

The results with this control law are shown in Figure 9. As we can observe, the robot succeeds in following the line trajectory quite well. When going from tracking to balancing, damped oscillations can be noticed around the final reference position in the plane.

Finally, the results are difficult to compare with other experiments reported in the literature because often the robot structure differs: the actuators (motors) are different, the surface of the ball is different, the robot is bigger (taller, heavier). All of these greatly influence control performances. Nevertheless, if we limit the comparison to Pham and Lee (2018), which implemented a sliding mode strategy based on the nonlinear model, the performances are comparable: stabilization in a disc with radius smaller than 10 cm, tracking with errors less than 15 cm.

5. CONCLUSIONS

This paper presented an SMC strategy for stabilization and reference tracking with ballbot. The control design is based on a linearized model of the process, and is robust to matched uncertainties due to unmodeled dynamics or external disturbances. Experimental results for stabilization and tracking illustrated the efficiency of the control, and the fact that it outperforms linear control (LQR).

As future work, we plan to use algorithms inspired from artificial intelligence to determine the switching values of ρ_c depending on the tracking scenario (Buşoniu et al., 2017). We also intend to tackle the issue of the unmatched disturbance, by estimating and then rejecting it.

REFERENCES

Blonk, J. (2014). *Modeling and control of a ball-balancing robot*. Master's thesis, University of Twente.

Buşoniu, L., Daafouz, J., Bragagnolo, M.C., and Morărescu, I.C. (2017). Planning for optimal control and performance certification in nonlinear systems with controlled or uncontrolled switches. *Automatica*, 78, 297–308.

Edwards, C. and Spurgeon, K., S. (1998). *Sliding Mode Control*. CRC Press.

Fankhauser, P. and Gwerder, C. (2010). *Modeling and Control of a Ballbot*. B.S.C. thesis, Eidgenossische Technische Hochschule Zurich.

Hoshino, T., Yokota, S., and Chino, T. (2013). Omniride: A personal vehicle with 3 dof mobility. In *2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE)*, 1–6. IEEE.

Lal, I., Nicoara, M., Codrean, A., and Busoniu, L. (2019). Hardware and control design of a ball balancing robot. In *2019 IEEE 22nd International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, 1–6. IEEE.

Lauwers, T.B., Kantor, G.A., and Hollis, R.L. (2006). A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive. In *Proceedings 2006 IEEE International Conference on Robotics and Automation*, 2884–2889. IEEE.

Nagarajan, U., K.G. and Hollis, R. (2013). The ball robot: An omnidirectional balancing mobile robot. *The International Journal of Robotics Research*, 33(6), 917–930.

Ogata, K. and Yang, Y. (2002). *Modern control engineering*. Prentice Hall.

Pham, D.B., Kim, H., Kim, J., and Lee, S.G. (2018). Balancing and transferring control of a ball segway using a double-loop approach [applications of control]. *IEEE Control Systems*, 38(2), 15–37.

Pham, D.B. and Lee, S.G. (2018). Hierarchical sliding mode control for a two-dimensional ball segway that is a class of a second-order underactuated system. *Journal of Vibration and Control*, 25(1), 72–83.

Shtessel, Y., Edwards, C., Fridman, L., and Levant, A. (2014). *Sliding mode control and observation*. Springer.