Advances in Engineering Research, volume 208
Proceedings of the International Conference on Innovation in Science and Technology (ICIST 2020)

Adaptive-Gain Second-Order Sliding Mode Control for Balancing an Underactuated Unicycle Robot

Yusie Rizal*, Feriyadi Marzuki
Department of Electrical Engineering
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
*yusie.rizal@poliban.ac.id

***Abstract*—This paper presents the balancing control of underactuated unicycle robot. The authors propose new approach to control such system using Adaptive Super-Twisting Sliding Mode Control. First, the simplified dynamic model of unicycle robot is adopted from previous results and then the normal forms are obtained to design controllers. Second, the adaptive super twisting sliding mode control is applied for all pitch, roll, and yaw dynamics such that the system can be balanced upright. Then the controller laws are implemented in virtual robot using Open Dynamics Engine (ODE) simulation. It is shown in simulations that the proposed controllers are effectively stabilized the unicycle robot upright.**

Keywords—adaptive super-twisting algorithm, balancing control, unicycle robot, underactuated system, second-order sliding mode control

I. INTRODUCTION

This paper presents the balancing control of unicycle robot which consists of a wheel, a reaction wheel, and a turntable. We propose a balancing controller for such robot using adaptive-gain second order sliding mode control. This work is inspired from the works of Chin [1] and Yang [2] or some other works in [3,4] and apply different approach to balance the system. Indeed, we use the mathematical model derived by Chin [1] and Yang [2] which is based on Euler-Lagrange method. Then, we develop the unicycle robot simulator using open-dynamic engine (ODE) simulation as a testbed to verify the effectiveness of proposed controller design.

The stabilizing control of unicycle robot is the fundamental problem in underactuated unicycle robot. In this control problem, the robot is kept balance upright by actuating the wheel, reaction wheel, and turntable for pitch, roll, and yaw axes, respectively. To control such robot, the dynamics model is decoupled into subsystems and control separately. These approaches have successfully implemented in previous works in [1–4]. The advanced control problem of unicycle robot is the motion control of such robot so that the robot can follow a path or to track a trajectory. In this paper, the author only discusses the fundamental problem in unicycle robot.

II. UNDERACTUATED UNICYCLE ROBOT

A. Related Works

This paper relates with the previous works as conducted by Chin [1] and Yang [2] in their thesis. In earlier design, the unicycle robots were developed using wheel and reaction wheel without turntable as discussed in Ho et al [3], while in Hohl et al [4], the robot has reaction wheel and turntable where the position of the turntable is located on the top of the body. Since the stabilities of the previous systems were harder to maintain, then in recent designs the position of the turntable is located near the centre of body as depicted in Figure 1 [1,2], furthermore, the size and mass of unicycle robot is smaller and lighter [3,4].

In his master thesis, Chin [1] and Yang [2] have derived the mathematical model of unicycle robot based on Euler Lagrange method. This method was different from Diaoxiong's approach where he derived his unicycle robot model using Lagrange-D'Alembert equation [5,6]. Based on simplified model derived [1,2], we propose a new approach to solve self-balancing controls problem using Adaptive Super Twisting Algorithm. Following the system model with similar dimension given in Shamsiri et al [2], we have successfully developed a unicycle robot simulator as a testbed to verify the control method presented in this paper. This simulation is developed using commercial package named AnyKode Marilou Robotic Studio [7]. This package is utilized Open-Dynamic Engine (ODE) libraries that serve as physics engine as many other robotics simulation [8–11] including Webots [12,13], Gazebo [14,15], and other realistic game software in the market today.

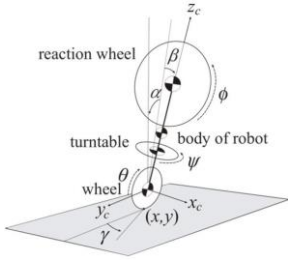

Fig. 1. Unicycle robot model.

B. Mathematical Model

The mathematical model presented in this paper were derived by Chin [1] and Yang [2] using Euler-Lagrange method. They both developed different robots, but they obtained similar mathematical models and this system model is depicted as Figure 1 [1,2]. Let α, β, and γ are the respective for the pitch, roll, and yaw angles, while $\{\theta,\varphi,\psi\}$ are the angular displacement of the wheel, reaction wheel, and turntable, respectively. By using dynamics model of unicycle robot in [1,2], the simplified dynamics model can be decoupled into pitch (Σ_1), roll (Σ_2), and yaw (Σ_3) as following:

$$\Sigma_1: \begin{cases} a_{11}\ddot{\alpha} + a_{12}\cos\alpha\,\ddot{\theta} - d_{11}\,g\sin\alpha = 0, \\ a_{21}\cos\alpha\,\ddot{\alpha} + a_{22}\ddot{\theta} - d_{12}\,\dot{\alpha}^2\sin\alpha = \tau_w \end{cases} \quad (1)$$

$$\Sigma_2: \begin{cases} b_{11}\ddot{\beta} + b_{12}\ddot{\phi} + d_{21}g\sin\beta = 0, \\ b_{21}\ddot{\beta} + b_{22}\ddot{\phi} = \tau_d. \end{cases} \quad (2)$$

$$\Sigma_3: \begin{cases} c_{11}\ddot{\psi} + c_{12}\ddot{\gamma} = 0, \\ c_{21}\ddot{\psi} + c_{22}\ddot{\gamma} = \tau_f. \end{cases} \quad (3)$$

with τ_w is generalized dc motor modelled for modelling τ_w, τ_d, and τ_f as:

$$\tau_m = \frac{K_t}{R_a}u \quad (4)$$

τ_m and u are the motor torque and control voltage applied to the motor, respectively. R_a is the armature resistance and K_t is motor constant. Furthermore, the parameters of system (1)–(3) are given by:

$$\begin{aligned}
a_{11} &= I_y + \frac{1}{4}l^2(m_b + 4m_d) + l_{wf}^2 m_f \\
a_{12} &= m_d l l_r + \frac{1}{2}m_b l l_r + m_f l_{wf} l_r \\
a_{21} &= m_d l_r l + \frac{1}{2}m_b l l_r + m_f l_r l_{wf} \\
a_{22} &= I_{wy} + l_r^2(m_b + m_d + m_f + m_w) \\
b_{11} &= I_x + (l + l_r)^2 m_d + (l_r + l_{wf})^2 m_f + l_r^2 m_w \\
&\quad + \frac{1}{4}(l + 2l_r)^2 m_b \\
b_{12} &= b_{21} = b_{22} = I_{dx} \\
c_{11} &= c_{21} = c_{22} = I_{fz} \\
c_{12} &= (I_{wz} + I_{fz} + I_{bz} + I_{dz}) \\
d_{11} &= m_d l l_r + \frac{1}{2}m_b l l_r + m_f l_{wf} l_r \\
d_{12} &= -(\frac{1}{2}m_b l l_r + m_d l l_r + m_f l l_r) \\
d_{21} &= -\left(\frac{1}{2}(l + 2l_r)m_b + (l + l_r)m_d + (l_r + l_{wf})m_f \right. \\
&\quad \left. + l_r m_w\right) \\
I_x &= I_{bx} + I_{dx} + I_{fx} + I_{wx} \\
I_y &= I_{by} + I_{dy} + I_{fy}
\end{aligned} \quad (5)$$

It is important to note that $\{I_{by},I_{wy},I_{dy},I_{fy}\}$ are the moment of inertia of body, wheel, reaction wheel, and turntable about y_c axis, respectively. $\{I_{bx},I_{wx},I_{dx},I_{fx}\}$ are the moment of inertia of body, wheel, reaction wheel, and turntable about x_c axis, respectively. Moreover, $\{I_{bz},I_{wz},I_{dz},I_{fz}\}$ are the moment of inertia of body, wheel, reaction wheel, and turntable about the z_c axis, respectively.

III. CONTROLLER DESIGN

A. Stabilizing Controller of Longitudinal Dynamics

Since the system is only operated near equilibrium point and so the system (1) can be linearized by assuming $\sin\alpha \approx \alpha$, $\cos\alpha \approx 1$ and $\cos^2\alpha \approx 1$. Furthermore, let $q = [q_1\ q_2]^T$, $p = [p_1\ p_2]^T$ in which $q_1 = \alpha$, $q_2 = \theta$, $p_1 = \dot{\alpha}$, and $p_2 = \dot{\theta}$. If the dc motor model in (4) is applied into τ_w of longitudinal dynamics model, and then, the linearized model can be rewritten as follow

$$\begin{aligned}
\dot{q}_1 &= p_1 \\
\dot{p}_1 &= f_1(q,p) + g_1(q_2)u \\
\dot{q}_2 &= p_2 \\
\dot{p}_2 &= f_2(q,p) + g_2(q_2)u
\end{aligned} \quad (6)$$

where:

$f_1(q,p) = \frac{-L_1 g q_1 + J_1 q_1 p_1^2}{M_1}$, $f_2(q,p) = \frac{L_2 g q_1 - J_2 q_1 p_1^2}{M_1}$, $g_1(q_2) = \frac{K_1 K_t}{M_1 R_a}$, and $g_2(q_2) = -\frac{K_2 K_t}{M_1 R_a}$.

Furthermore, K_1, K_2, L_1, L_2, J_1, J_2, M_1 are given:

$$\begin{aligned}
K_1 &= \frac{1}{2}l l_r m_b + l l_r m_d + l_r l_{wf} m_f, \\
L_1 &= \left(\frac{l m_b}{2} + l m_d + l_{wf} m_f\right)(I_{wy} + l_r^2 m_y), \\
J_1 &= l_r\left(\frac{l m_b}{2} + l m_d + l m_f\right) \\
&\quad \times \left(\frac{1}{2}l l_r m_b + l l_r m_d + l_r l_{wf} m_f\right), \\
M_1 &= -(I_{wy} + l_r^2 m_y) \\
&\quad \times \left(I_y + \frac{1}{4}l^2(m_b + 4m_d) + l_{wf}^2 m_f\right) \\
&\quad + \left(\frac{1}{2}l l_r m_b + l l_r m_d + l_r l_{wf} m_f\right)^2, \\
K_2 &= I_y + \frac{1}{4}l^2(m_b + 4m_d) + l_{wf}^2 m_f, \\
L_2 &= \left(\frac{l m_b}{2} + l m_d + l_{wf} m_f\right) \\
&\quad \times \left(\frac{1}{2}l l_r m_b + l l_r m_d + l_r l_{wf} m_f\right), \\
J_2 &= l_r\left(\frac{l m_b}{2} + l m_d + l m_f\right) \\
&\quad \times \left(I_y + \frac{1}{4}l^2(m_b + 4m_d) + l_{wf}^2 m_f\right) \\
I_y &= I_{by} + I_{dy} + I_{fy}, \\
m_y &= m_b + m_d + m_f + m_w.
\end{aligned} \quad (7)$$

In (6), the control input u appears on second and fourth rows and we need to decouple the system by using standard decoupling algorithm [16]. We use the change of coordinates

$z_1 = q_1 + \frac{K_1}{K_2}q_2$, $z_2 = p_1 + \frac{K_1}{K_2}p_2$, $\xi_1 = q_2$, and $\xi_2 = p_2$ to decouples (q_1,p_1) subsystem and (q_2,p_2) subsystem with respect to u. The balance point of unicycle robot for longitudinal dynamics model where $\alpha \to 0$, $\theta \to 0$, $\dot{\alpha} \to 0$, and $\dot{\theta} \to 0$ is equivalent to $z_1 \to 0$, $z_2 \to 0$ and $\xi_1 \to 0$, $\xi_2 \to 0$. It follows that

$$\begin{aligned}
\dot{z}_1 &= p_1 + \frac{K_1}{K_2}p_2 \\
\dot{z}_2 &= \left(-\frac{gL_1}{M_1} + \frac{gK_1L_2}{K_2M_1}\right)q_1 + \left(\frac{J_1}{M_1} - \frac{J_2K_1}{K_2M_1}\right)q_1p_1^2 \\
\dot{\xi}_1 &= p_2 \\
\dot{\xi}_2 &= \frac{L_2g}{M_1}q_1 - \frac{J_2}{M_1}q_1p_1^2 - \frac{K_2K_t}{M_1R_a}u
\end{aligned} \tag{8}$$

The relation of (q,p) and (z,ξ) are given:

$$\begin{aligned}
q_1 &= z_1 - \frac{K_1}{K_2}\xi_1 \\
p_1 &= z_2 - \frac{K_1}{K_2}\xi_2 \\
q_2 &= \xi_1 \\
p_2 &= \xi_2
\end{aligned} \tag{9}$$

and by substituting (9) into (8), we have:

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= A_1z_1 - B_1\xi_1 \\
\dot{\xi}_1 &= \xi_2 \\
\dot{\xi}_2 &= A_2z_1 + B_2\xi_1 + Ku + d
\end{aligned} \tag{10}$$

Where d is the higher order terms, and we assume as disturbances due to unmodelled dynamics. Furthermore, A_1, B_1, A_2, B_2, and K are given:

$$\begin{aligned}
A_1 &= \frac{2g(lm_b + 2lm_d + 2l_{wf}m_f)}{4I_y + l^2(m_b + 4m_d) + 4l_{wf}^2m_f}, \\
B_1 &= \frac{4gl_r(lm_b + 2lm_d + 2l_{wf}m_f)^2}{(4I_y + l^2(m_b + 4m_d) + 4l_{wf}^2m_f)^2}, \\
A_2 &= \left(g(\frac{lm_b}{2} + lm_d + l_{wf}m_f)(\frac{1}{2}ll_rm_b + ll_rm_d \right. \\
&\quad \left. + l_rl_{wf}m_f)\right) \Big/ \left[(\frac{1}{2}ll_rm_b + ll_rm_d + l_rl_{wf}m_f)^2 \right. \\
&\quad \left. -(I_y + \frac{1}{4}l^2(m_b + 4m_d) + l_{wf}^2m_f)(I_{wy} + l_r^2m_y)\right]
\end{aligned} \tag{11}$$

and:

$$\begin{aligned}
B_2 &= \left[-g\left(\frac{lm_b}{2} + lm_d + l_{wf}m_f\right)\left(\frac{1}{2}ll_rm_b + ll_rm_d \right.\right. \\
&\quad \left.\left. + l_rl_{wf}m_f\right)\left(\frac{1}{2}ll_rm_b + ll_rm_d + l_rl_{wf}m_f\right)\right] \\
&\quad \Big/\left[\left(l_{wf}^2m_f + I_y + \frac{1}{4}l^2(m_b + 4m_d)\right)\right. \\
&\quad \times \left((\frac{1}{2}ll_rm_b + ll_rm_d + l_rl_{wf}m_f)^2 - (I_{wy} + l_r^2m_y)\right. \\
&\quad \left.\left. \times (I_y + \frac{1}{4}l^2(m_b + 4m_d) + l_{wf}^2m_f)\right)\right], \\
K &= -K_t\left(I_y + \frac{1}{4}l^2(m_b + 4m_d) + l_{wf}^2m_f\right) \Big/ \\
&\quad \left[\left(\frac{1}{4}l_r^2(lm_b + 2lm_d + 2l_{wf}m_f)^2 - (I_{wy} + l_r^2m_y)\right.\right. \\
&\quad \left.\left. \times (\frac{1}{4}l^2(m_b + 4m_d) + I_y + l_{wf}^2m_f)\right)R_a\right].
\end{aligned} \tag{12}$$

The system in (10) can be rewritten as:

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= f_1(z_1,z_2,\xi_1) \\
\dot{\xi}_1 &= \xi_2 \\
\dot{\xi}_2 &= f_2(z_1,z_2,\xi_1,\xi_2) + b(z_1,z_2,\xi_1)u + d
\end{aligned} \tag{13}$$

With $|d| \le D$. To design sliding mode controller for underactuated system (13), there are three conditions need to be satisfied as given in Xu and Özgüner [17] and all three conditions are satisfied. Hence, we allow to design sliding manifold as:

$$s = \xi_2 - c_1z_1 - c_2z_2 - c_3\zeta_1. \tag{14}$$

It follows that:

$$\dot{s} = \dot{\xi}_2 - c_1\dot{z}_1 - c_2\dot{z}_2 - c_3\dot{\xi}_1. \tag{15}$$

When the system trajectory reaches and stay on sliding manifold at time t_s, we have $s = 0$. On this manifold as shown in (14) we may rewrite as:

$$\xi_2 = c_1z_1 + c_2z_2 + c_3\xi_1. \tag{16}$$

To choose c_i (i = 1,2,3) such that the system trajectory converges to zero, we let $\mu_1 = \xi_2$ and $\mu_2 = [z_1\ z_2\ \xi_1]^T$, then (16) becomes:

$$\mu_1 = C\mu_2 \tag{17}$$

Where $C = [c_1\ c_2\ c_3]$. When time $t \ge t_s$, we have $\dot{s} = 0$ and thus (15) can be represented in μ_1 and μ_2 by:

$$\begin{aligned}
\dot{s} &= \dot{\mu}_1 - C\dot{\mu}_2 \\
&= A_2z_1 + B_2\xi_1 + Ku + d - C\dot{\mu}_2
\end{aligned} \tag{18}$$

In (18), it is important to choose C such that $\dot{\mu}_2 \to 0$. This can be achieved by choosing C as Hurwitz as discussed in Liu [18]. To do that, let us take first derivative of μ_2, it yields:

$$\dot{\mu}_2 = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{\xi}_1 \end{bmatrix} = \begin{bmatrix} z_2 \\ A_1 z_1 - B_1 \xi_1 \\ \xi_2 \end{bmatrix} = C_1 \mu_1 + C_2 \mu_2 \quad (19)$$

where

$$C_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \; C_2 = \begin{bmatrix} 0 & 1 & 0 \\ A_1 & 0 & -B_1 \\ 0 & 0 & 0 \end{bmatrix} \quad (20)$$

From (17), the equation (19) can be rewritten as:

$$\dot{\mu}_2 = (C_1 C + C_2)\mu_2. \quad (21)$$

To guarantee $\mu_2 \rightarrow 0$, then $(C_1 C + C_2)$ must be Hurwitz. Calculating $C_1 C + C_2$ from (20), we have:

$$C_1 C + C_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [\, c_1 \;\; c_2 \;\; c_3 \,] + \begin{bmatrix} 0 & 1 & 0 \\ A_1 & 0 & -B_1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ A_1 & 0 & -B_1 \\ c_1 & c_2 & c_3 \end{bmatrix} \quad (22)$$

The pole for (22) can be designed by:

$$|sI - (C_1 C + C_2)| = s^3 - c_3 s^2 + (B_1 c_2 - A_1)s + (A_1 c_3 + B_1 c_1) = 0 \quad (23)$$

Let $(s + k)^3 = 0$ where $k > 0$ and equate with (23), then:

$$\begin{cases} -c_3 & = 3k \\ -A_1 + B_1 c_2 & = 3k^2 \\ A_1 c_3 + B_1 c_1 & = k^3 \end{cases} \quad (24)$$

By solving (24) we obtain:

$$c_1 = \frac{k^3}{B_1} + \frac{3kA_1}{B_1}$$
$$c_2 = -\frac{3k^2 + A_1}{B_1}$$
$$c_3 = -3k \quad (25)$$

If c_1, c_2, and c_3 are chosen properly, then (21) would make $\mu_2 \rightarrow 0$. Since $\mu_2 \rightarrow 0$ then $\mu_1 \rightarrow 0$ in (17), and thus, $s \rightarrow 0$ in (14) as time $t \rightarrow \infty$.

The system in (15) has relative degree one so that the super twisting algorithm can be applied. To design controller, we may choose

$$u = \frac{1}{K}\left(u_{sm} - u_{eq}\right) \quad (26)$$

for $K \neq 0$. Substituting (26)-(28) into (18), it yields:

$$\dot{s} = u_{sm} + d \quad (27)$$

where $u_{eq} = (A_2 - A_1 c_2) z_1 - c_1 z_2 + (B_2 + B_1 c_2)\xi_1 - c_3 \xi_2$ (28)

And d can be removed by control input u_{sm}. This controller is super-twisting algorithm [19] i.e.:

$$u_{sm} = u_1 + u_2$$
$$u_1 = -\alpha\, |s|^{\frac{1}{2}} \operatorname{sign}(s)$$
$$\dot{u}_2 = \begin{cases} -u_{sm} & \text{if } |u_{sm}| > u_m \\ -\beta \operatorname{sign}(s) & \text{if } |u_{sm}| \leq u_m \end{cases} \quad (29)$$

We consider α and β are adaptively changing by:

$$\dot{\alpha} = \begin{cases} \omega_1 \sqrt{\frac{\gamma_1}{2}} \operatorname{sign}(|s| - \mu), & \text{if } |\alpha| > \alpha_m \\ 0 & \text{if } |\alpha| \leq \alpha_m \end{cases}$$
$$\beta = 2\,\epsilon_1\,\alpha \quad (30)$$

where $\epsilon_1, \gamma_1, \omega_1 > 0$ with α_m is small positive value.

B. *Stabilization Control of Lateral Dynamics*

We ignore the state variable φ from equation (2) [3,20,21] and let $x_1 = \beta$, $x_2 = \dot{\beta}$, and $x_3 = \dot{\varphi}$. It yields:

$$\dot{x} = f(x) + g(x)u \quad (31)$$

$$f(x) = \begin{bmatrix} x_2 \\ -\frac{b_{22} d_{21}}{b_{11} b_{22} - b_{12} b_{21}} g \sin x_1 \\ \frac{b_{21} d_{21}}{b_{11} b_{22} - b_{12} b_{21}} g \sin x_1 \end{bmatrix} \quad (32)$$

and

$$g(x) = \begin{bmatrix} 0 \\ -\frac{b_{12} K_t}{(b_{11} b_{22} - b_{12} b_{21}) R_a} u \\ \frac{b_{11} K_t}{(b_{11} b_{22} - b_{12} b_{21}) R_a} u \end{bmatrix}. \quad (33)$$

The

Output equation is defined:

$$y = \frac{b_{11} K_t x_2 + b_{12} K_t x_3}{R_a}. \quad (34)$$

We take derivative of y using Lie Derivatives with respect to f and g [20,22] until we find $L_g L_f^{k-1} h \neq 0$ for $k = 1,2,\ldots,n$. By computing these derivatives, we obtain:

$$\dot{y} = L_f h + L_g h u = -\frac{d_{21} K_t g}{R_a} \sin x_1$$
$$\ddot{y} = L_f^2 h + L_g L_f h u = -\frac{d_{21} K_t g}{R_a} x_2 \cos x_1$$
$$\dddot{y} = L_f^3 h + L_g L_f^2 h u = -\frac{b_{22} d_{21}^2 K_t g^2}{(b_{12} b_{21} - b_{11} b_{22}) R_a} \times \cos x_1 \sin x_1 + \frac{d_{21} K_t g}{R_a} x_2^2 \sin x_1 - \frac{b_{12} d_{21} K_t^2 g}{(b_{12} b_{21} - b_{11} b_{22}) R_a^2} \cos x_1 u. \quad (35)$$

Since $L_g L_f^2 h \neq 0$. Following (34)-(35) and choosing new state variables as:

$$\xi_1 = \frac{b_{11}K_t}{R_a}x_2 + \frac{b_{12}K_t}{R_a}x_3$$
$$\xi_2 = -\frac{d_{21}K_t g}{R_a}\sin x_1$$
$$\xi_3 = -\frac{d_{21}K_t g}{R_a}x_2 \cos x_1 \quad (36)$$

It follows that the normal form of system (2) in feedback linearization is:

$$\dot{\xi}_1 = \xi_2$$
$$\dot{\xi}_2 = \xi_3 \quad (37)$$
$$\dot{\xi}_3 = v$$

When u is chosen as:

$$u = -\frac{(b_{12}b_{21} - b_{11}b_{22})R_a^2}{b_{12}d_{21}K_t^2 g\cos x_1}\left(v + \frac{b_{22}d_{21}^2K_t g^2}{(b_{12}b_{21} - b_{11}b_{22})R_a}\right. \quad (38)$$
$$\left.\times \cos x_1 \sin x_1 - \frac{d_{21}K_t g}{R_a}x_2^2 \sin x_1\right),$$

i.e., valid for $|x_1| < \pi/2$.

We choose the sliding surface as:

$$s = c_1\xi_1 + c_2\xi_2 + \xi_3 \quad (39)$$

For c_1, $c_2 > 0$. Using (37) in the first-time derivative of (39), we have

$$\dot{s} = c_1\xi_2 + c_2\xi_3 + v. \quad (40)$$

To design controller, we choose

$$v = u_{sm} - u_{eq} \quad (41)$$

where:

$$u_{eq} = c_1\xi_2 + c_2\xi_3. \quad (42)$$

The new system (40) has relative degree one because the control input appears in first time derivative of s, and thus, the Super-Twisting Algorithm [19] can be applied. Hence, u_{sm} is given by:

$$u_{sm} = u_1 + u_2$$
$$u_1 = -\alpha\,|s|^{\frac{1}{2}}\,\mathrm{sign}(s)$$
$$\dot{u}_2 = \begin{cases} -u_{sm} & \text{if } |u_{sm}| > u_m \\ -\beta\,\mathrm{sign}(s) & \text{if } |u_{sm}| \le u_m \end{cases} \quad (43)$$

The control gains for α and β are adaptively changing by:

$$\dot{\alpha} = \begin{cases} \omega_2\sqrt{\frac{\gamma_2}{2}}\mathrm{sign}(|s| - \mu_2), & \text{if } |\alpha| > \alpha_m \\ 0 & \text{if } |\alpha| \le \alpha_m \end{cases}$$
$$\beta = 2\,\epsilon_2\,\alpha \quad (44)$$

Where $\epsilon_2, \gamma_2, \omega_2, \mu_2 > 0$ with α_m is small positive value.

C. Control Yaw Dynamics

Following Rizal et al [4], from the yaw dynamics model in (3), we have:

$$I_{fz}\ddot{\psi} + I_z\ddot{\gamma} = 0,$$
$$I_{fz}\ddot{\psi} + I_{fz}\ddot{\gamma} = \frac{K_t}{R_a}u - \frac{K_t^2}{R_a}\dot{\psi} \quad (45)$$

Where $I_z = I_{wz} + I_{fz} + I_{bz} + I_{dz}$. Simplifying and eliminating for $\ddot{\psi}$ in (49), then:

$$\ddot{\gamma} = -\frac{1}{(I_{wz} + I_{bz} + I_{dz})}\left(\frac{K_t}{R_a}u - \frac{K_t^2}{R_a}\dot{\psi}\right) \quad (46)$$

we define

$$e_\gamma = \gamma - \gamma_d$$
$$\dot{e}_\gamma = \dot{\gamma} \quad (47)$$
$$\ddot{e}_\gamma = \ddot{\gamma}$$

Let $z_1 = e_\gamma$ and $z_2 = \dot{e}_\gamma$, then we have:

$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = -\frac{1}{(I_{wz} + I_{bz} + I_{dz})}\left(\frac{K_t}{R_a}u - \frac{K_t^2}{R_a}\dot{\psi}\right) \quad (48)$$

We choose a sliding manifold as:

$$s = z_2 + c_1 z_1 \quad (49)$$

Then, if $s \to 0$ then both $z_1 \to 0$ and $z_2 \to 0$ and thus $\gamma \to \gamma_d$. This means the angle heading of unicycle γ will converge to γ_d. It follows from (49) that:

$$\dot{s} = \dot{z}_2 + c_1\,\dot{z}_1$$
$$= c_1\,\dot{z}_1 - \frac{1}{(I_{wz} + I_{bz} + I_{dz})}\left(\frac{K_t}{R_a}u - \frac{K_t^2}{R_a}\dot{\psi}\right) \quad (50)$$

By choosing:

$$u = -\frac{(I_{wz} + I_{bz} + I_{dz})R_a}{K_t}\left(v - c_1\dot{z}_1\right.$$
$$\left. - \frac{K_t^2\,\dot{\psi}}{(I_{bz} + I_{dz} + I_{wz})R_a}\right) \quad (51)$$

and substitute (51) for u in (50) then we have

$$\dot{s} = v. \quad (52)$$

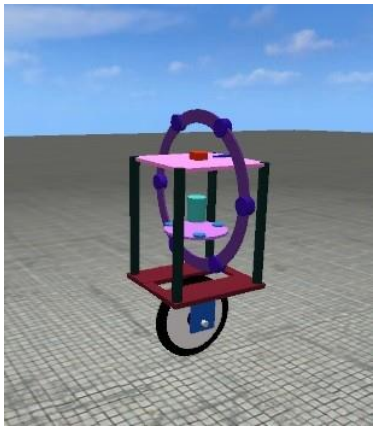

Fig. 2. Unicycle robot simulation.

The super-twisting algorithm [19] is used in (51) i.e., given by:

$$\begin{aligned} v &= u_1 + u_2 \\ u_1 &= -\alpha |s|^{\frac{1}{2}} \operatorname{sign}(s) \\ \dot{u}_2 &= \begin{cases} -v & \text{if } |v| > u_m \\ -\beta \operatorname{sign}(s) & \text{if } |v| \le u_m \end{cases} \end{aligned} \tag{53}$$

where $\alpha = \alpha(s,\dot{s})$ is adaptively changing by

$$\begin{aligned} \dot{\alpha} &= \begin{cases} \omega_3 \sqrt{\frac{\gamma_3}{2}} \operatorname{sign}(|s| - \mu_3 \\ 0, \end{cases} \\ \beta &= 2\,\epsilon_3\,\alpha \\ &\text{if } |\alpha| > \alpha_m \\ &\text{if } |\alpha| \le \alpha_m \end{aligned} \tag{54}$$

Where ω_3, γ_3, and are positive constants and α_m is small value.

IV. Results and Discussion

A. Unicycle Robot Simulation

The developed of robot simulation in Fig. 2 is inspired from the appearance and mechanical design of Yang's work in his thesis [2]. The simulation is based on Open Dynamics Engine (ODE) simulation using commercial robot software AnyKode Marilou [7]. We assume all states are available to measure and the sensors and actuators are provided by robot simulation software. The system parameters used in simulation is given in Table I.

TABLE I. Physical Parameters

Symbol	Value and unit	Description
m_b	0.500 kg	Mass of body (without other parts)
m_d	0.610 kg	Mass of reaction wheel
m_f	0.200 kg	Mass of turntable
m_w	0.453 kg	Mass of wheel
Iwx	1.34×10^{-3} kg.m^2	Moment of inertia of wheel about x_c axis
Iwy	6.71×10^{-4} kg.m^2	Moment of inertia of wheel about y_c axis
Iwz	6.71×10^{-4} kg.m^2	Moment of inertia of wheel about z_c axis
Ifx	1.68×10^{-4} kg.m^2	Moment of inertia of turntable about x_c axis
Ify	1.68×10^{-4} kg.m^2	Moment of inertia of turntable about y_c axis
Ifz	3.36×10^{-4} kg.m^2	Moment of inertia of turntable about z_c axis
Ibx	1.70×10^{-2} kg.m^2	Moment of inertia of body about x_c axis
Iby	1.70×10^{-2} kg.m^2	Moment of inertia of body about y_c axis
Ibz	3.30×10^{-5} kg.m^2	Moment of inertia of body about z_c axis
Idx	3.82×10^{-2} kg.m^2	Moment of inertia of reaction wheel about x_c
Idy	3.39×10^{-2} kg.m^2	Moment of inertia of reaction wheel about y_c
Idz	2.12×10^{-3} kg.m^2	Moment of inertia of reaction wheel about z_c
l	0.166 m	Length from the center of wheel to turntable
lr	0.077 m	Radius of the wheel
lwf	0.077 m	Length of center of reaction wheel to turntable
K_t	0.0530 N.m/A	Motor constant
R_a	1.3000 Ω	Armature constant
g	9.8 m/s^2	Gravitational acceleration

Here, we use IMU (inertia measurement unit) to detect α, β and γ angles. Furthermore, each motor has motor encoder with resolution 1024 ppr and thus, the angular position of wheel θ can be measured.

In simulation, we use equations (29)-(30), (43)-(44), and (53)-(54) to control pitch, roll, and yaw dynamics, respectively. The control parameters for pitch dynamics are $\alpha_m = 0.32$, $\omega_1 = 0.178$, $\gamma_1 = 0.1325$, $\epsilon_1 = 1.029$, $\mu_1 = 0.05$, $c_1 = 6.75$, $c_2 = 3.1 c_3 = -1.511$, $u_m = 1$. Moreover, for roll dynamics, its parameters are $\alpha_m = 0.22$, $\omega_2 = 1.115$, $\gamma_2 = 0.25$, $\epsilon_2 = 0.125$, $\mu_2 = 0.22$, $c_1 = 212$, $c_2 = 6$, $u_m = 1$ Finally, the parameters for yaw dynamics are $\alpha_m = 0.22$, $\omega_3 = 0.5$, $\gamma_3 = 1.75$, $\epsilon_3 = 1.15$, $\mu_3 = 0.121$, $c_1 = 212$, $u_m = 1$.

These control laws were implemented using C/C++ programming in Microsoft Visual C++ 2008 for controlling the virtual unicycle robot in AnyKode Marilou.

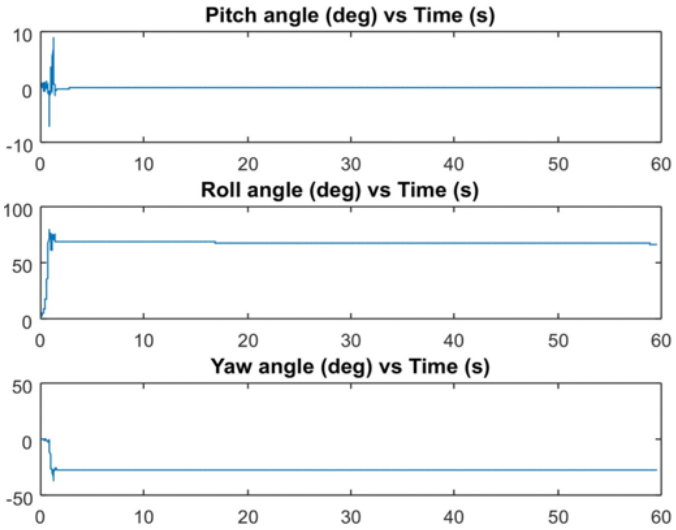

Fig. 3. Unstable unicycle robot.

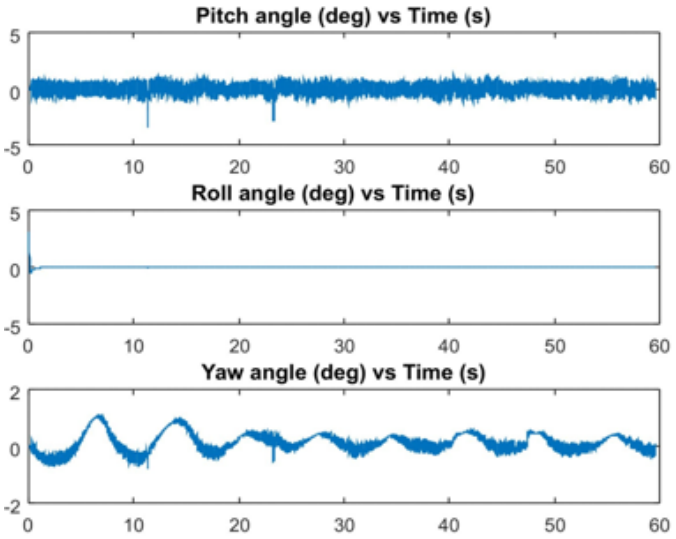

Fig. 4. Stable unicycle robot.

The simulation results are given in Figure 3 and Figure 4 for unbalanced and balanced systems, respectively. In Figure 3, the robot is released from any initial position and since the system follows the physical laws, the system falls on the ground. This means the angular position of unicycle robot in roll and yaw axes are not equal to zeros.

For the second simulation, we set the initial position of the robot has 3° in lateral direction (roll). If the control parameters are chosen properly and tuned as given parameters above, then the robot can be balanced upright. It is shown in Fig. 4, the angular position for pitch, roll, and yaw are on equilibrium points (nearly zeros). This means the robot successfully can be stabilized upright.

In different scenarios, we set different initial positions in lateral axis range from 1° ~ 10°. From this experiment, we found that the maximum angle for initial position is 6°. Other than that, the system becomes unstable, and it is harder to control. In real-world application, for example in Ho et al and Rizal et al [3,4], the initial position of unicycle robot in lateral axis was less than ±1.14° (±0.02rad), i.e., very close to equilibrium point. Thus, the efficacy of proposed balance control in this paper is better than those experimental results presented in [3,4]. Although this proposed robust control is promising, but the drawback of this controller is too many control gains and parameters compared to previous works [3,4].

V. Conclusion

In this paper, we discussed the adaptive-gain second order sliding mode control to balance the unicycle robot by using the simplified dynamics of unicycle robot model in literature. The system was decoupled into three independent subsystem and then the controller design is derived separately. To verify the effectiveness of the control law, we developed a virtual robot as a testbed to implement the controllers. It shown that the virtual robot follows the physical laws i.e., when the system is not properly controlled, the robot may fall down to ground due to gravitational force. By properly tuning the control parameters and control gains, the robot can be stabilized upright. It is found in simulation that the proposed controller can stabilize the unicycle robot upright even when the initial position in lateral axis is more than 3° as compared to existing approaches in Ho et al [3] and Rizal et al [4].

References

[1] Y.W. Chin, Balance control and path following of a unicycle robot using second-order sliding mode control. M. S. Thesis, National Cheng Kung University, Taiwan, 2016.

[2] W.H.Yang, Stabilization and waypoint following control of unicycle robot. M. S. Thesis, National Cheng Kung University, Taiwan, 2019.

[3] M.T. Ho, Y. Rizal and Y.L. Chen, "Balance control of a unicycle robot," 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), pp. 1186-1191, 2014. IEEE.

[4] Y. Rizal, C.T. Ke and M.T. Ho, "Point-to-point motion control of a unicycle robot: Design, implementation, and validation," 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 4379-4384, 2015. IEEE.

[5] G. Daoxiong, P. Qi, Z. Guoyu and L. Xinghui, "LQR control for a selfbalancing unicycle robot on inclined plane," Journal of System Design and Dynamics, vol. 6. no. 5. pp. 685–699, 2012.

[6] G. Daoxiong and L. Xinghui, "Dynamics modeling and controller design for a self-balancing unicycle robot," in Proc. 32nd Chinese Control Conference, pp. 3205–3209, 2013.

[7] L. Ricatti, Marilou robotic studio. *www.anykode.com*, [Online]. Retrieved from: http://www.anykode.com/marilousimulate.php. Accessed on Nov. 14, 2020.

[8] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale and F. Nori, "An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator," Proceedings of the 8th workshop on performance metrics for intelligent systems, pp. 57-61, 2008.

[9] Y. Rizal, "Computer Simulation of Human-Robot Collaboration in the Context of Industry Revolution 4.0," In Becoming Human with

Humanoid-From Physical Interaction to Social Intelligence. IntechOpen, 2019.

[10] K. Hauser, “Robust contact generation for robot simulation with unstructured meshes,” Robotics Research. Springer, Cham, pp. 357-373, 2016.

[11] R.R. Shamshiri, I.A. Hameed, L. Pitonakova, C. Weltzien, S.K. Balasundram, I.J. Yule, T.E. Grift and G. Chowdhary, ”Simulation software and virtual environments for acceleration of agricultural robotics: Features highlights and performance comparison,” International Journal of Agricultural and Biological Engineering, vol. 11, no. 4, pp. 15–31, 2018.

[12] O. Michel, ”Webots: professional mobile robot simulation,” International Journal of Advanced Robotic Systems, vol. 1, no. 1, pp. 39– 42, 2004.

[13] L. Hohl, R. Tellez, O. Michel, and A.J. Ijspeert, ”Aibo and Webots: Simulation, wireless remote control and controller transfer,” Robotics and Autonomous systems, vol. 54, no. 6, pp. 472–485, 2006.

[14] N. Koenig and A. Howard, ”Design and use paradigms for gazebo, an open-source multi-robot simulator,” Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2149–2154, 2004.

[15] Z. Yilmaz and L. Bayindir, “Simulation of lidar-based robot detection task using ros and gazebo,” Avrupa Bilim ve Teknoloji Dergisi, pp. 513-529, 2019.

[16] R. Olfati-Saber, ”Global configuration stabilization for the VTOL aircraft with strong input coupling,” IEEE Transactions on Automatic Control, vol. 47, no. 11, pp. 1949–1951, 2002.

[17] R. Xu and Ü. Özgüner, “Sliding mode control of a class of underactuated systems,” Automatica, vol. 44, no. (1), pp. 233-241, 2008.

[18] J. Liu, ”Sliding mode control for underactuated system with decoupling algorithm” *Sliding Mode Control Using MATLAB*, Academic Press, pp. 307–327, 2017.

[19] A. Levant, ”Sliding order and sliding accuracy in sliding mode control,” International Journal of Control, vol. 58, no. 6, pp. 1247–1263, 1993.

[20] M.W. Spong, P. Corke and R. Lozano, “Nonlinear control of the reaction wheel pendulum,” Automatica, vol. 37, no. (11), pp. 1845-1851, 2001.

[21] Y. Rizal, R. Mantala, S. Rachman and N. Nurmahaludin, “Balance control of reaction wheel pendulum based on second-order sliding mode control,” International Conference on Applied Science and Technology (iCAST), pp. 51-56, 2018. IEEE.

[22] H.K. Khalil, Nonlinear Control. London: Pearson Education, 2015.