Balance Control of Reaction Wheel Pendulum Based on Second-order Sliding Mode Control

Yusie Rizal
Robotics Lab., Dept. of Electronics Engineering
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
Email: yusie.rizal@poliban.ac.id

Ronny Mantala
Dept. of Informatics Management
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
Email: rmantala@poliban.ac.id

Syaiful Rachman
Dept. of Electronics Engineering
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
Email: saifulrachman1@poliban.ac.id

N. Nurmahaludin
Dept. of Electronics Engineering
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
Email: mahaludin@poliban.ac.id

***Abstract*—This paper presents the balance control of reaction wheel pendulum system based on second-order sliding mode control. The system consists of rotating wheel and a pendulum where the rotating wheel is controlled such that the pendulum can be stabilized on its upright position. The stabilizing controller is designed based on second-order sliding mode control using super-twisting algorithm (STA). To implement and verify the designed controller, the computer simulation based on Open Dynamic Engine (ODE) is developed using Marilou AnyKode. The system parameters (c_1, c_2) and control gains (k_1, k_2) are chosen with values of 1.5, 2.0, 3.5, and 4.8, respectively. From the simulation results, it is shown that the control scheme is effectively stabilized the system.**

I. Introduction

Reaction wheel pendulum is also known as an inertia wheel pendulum. This system consists of a rotating wheel attached on the tip of a pendulum. Although it is a simple system but it has been used in many practical applications e.g. as a mechanism for balancing a unicycle robot [1], single-wheel pendulum robot [2], or cubical robot [3]. Moreover, the system has attractive features from a pedagogical and research standpoints [4]. Similar control method for stabilization control based on second-order sliding mode control is applied to inverted pendulum system [5]. However, in [5] the twisting controller [6] and sub-optimal controller [7] are employed instead of super-twisting algorithm.

The reaction wheel pendulum is a nonlinear underactuated mechanical system with unstable equilibrium point on its upright position [8]. To control the system in this upper position, a linear control and nonlinear control methods can be employed. In literature, stabilization of reaction wheel pendulum is solved using several control methods, for examples using energy approach [9], feedback linearization [4], pole placement method [10], PD control [11], PID control [12], LQR method [13], Linear Matrix Inequality (LMI) [14], first-order sliding mode control [1], and third-order discontinuous integral sliding mode algorithm [15].

In [15], the use of discontinous function in integral action generates a continous control signal and minimize the chattering effect for stabilizing reaction wheel pendulum. In this paper, an alternative approach to stabilize a reaction wheel pendulum is proposed. It is based on second-order sliding mode control using super-twisting algorithm. This work is improvement of previous work for stabilizing controller using first-order sliding mode control that was implemented as part of unicycle robot [1]. The proposed control method is taken into account because it is one of the promising robust control methods to reject bounded match perturbation theoretically completely [16].

The control problems of reaction wheel pendulum system typically fall into two categories, i.e. stabilization control on its upright position and swing-up control to its upright position. Several control methods have been proposed to solve those problems and have been implemented in numerical-based simulations as well as in real world mechatronics setups. To the best of our knowledge, the computer simulation based on Open Dynamics Engine (ODE) has not been used that much in mechatronics setup. One example is given by Naba [17] i.e. he developed an ODE-based simulation for cart-pole of system plant to test his control method. In this paper, it is shown that this simulation framework is visually more attractive to students for educational purposes. Furthermore, the use of ODE based simulation may reduce the cost to realize the mechatronics setups.

The rest of this paper is organized as follows. In Section II, the system modeling and simulation framework are presented. Section III briefly discusses the second-order sliding mode control. It follows with the controller design in Section IV. In Section V the experimental setup is discussed. Finally, the simulation results and concluding remarks are given in VI and VII, respectively.

II. System Description

Reaction wheel pendulum is a system which consist of a pendulum and rotating wheel as described in Fig. 1. The pendulum is pivoted at its end-point to the base. The reaction wheel is actuated by dc motor such that the pendulum can be stabilized on its upright position.

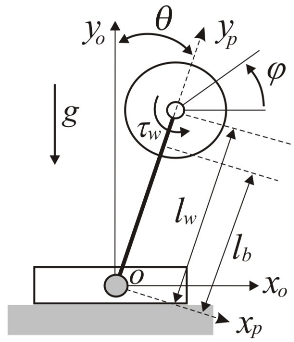

Fig. 1. Reaction wheel model

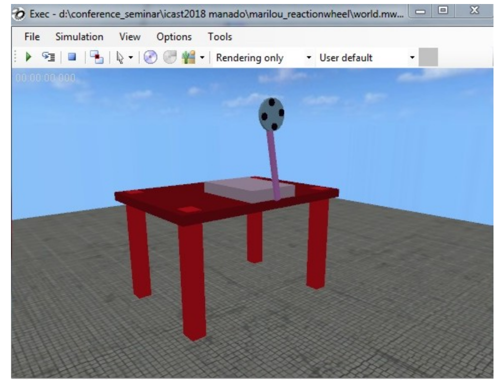

Fig. 2. Development of reaction wheel pendulum simulation using AnyKode Marilou.

A. System Modeling

The dynamic model of reaction wheel pendulum has been discussed in literature [9,18]. It is given by

$$\begin{aligned} (I_p + I_w + ml_b^2 + Ml_w^2)\,\ddot{\theta} + I_w\,\ddot{\varphi} &= (ml_b + Ml_w)\,g\sin\theta \\ I_w\,\ddot{\theta} + I_w\,\ddot{\varphi} &= \tau_w \end{aligned} \tag{1}$$

where I_p is the moment of inertia of pendulum, and I_w is the moment of inertia of reaction wheel. M is the mass of reaction wheel and m is the mass of pendulum, and τ_w is the torque input to the system.

Let τ_w as a reduced-order model of bldc motor where the model is

$$\tau_w = nK_t i. \tag{2}$$

n is gear ratio, K_t and u are the motor constant and motor current of control input, respectively. Substituting (2) into (1) yields

$$\ddot{\theta} = \frac{(ml_b+Ml_w)g\sin\theta}{ml_b^2+Ml_w^2+I_p} - \frac{nK_t i}{ml_b^2+Ml_w^2+I_p} \tag{3}$$

$$\ddot{\varphi} = -\frac{(ml_b+Ml_w)g\sin\theta}{ml_b^2+Ml_w^2+I_p} + nK_t\overline{m}_p i \tag{4}$$

where

$$\overline{m}_p = \left(\frac{1}{I_w} + \frac{1}{ml_b^2 + Ml_w^2 + I_p}\right). \tag{5}$$

The dynamic model of (3) and (4) will be employed to design controller as discuss in Section IV.

B. ODE based Simulation

Open dynamics engine (ODE) is a library for simulating articulated rigid body dynamics [19]. It was developed by Russel Smith and provided for free. Since then, many companies including Marilou Anykode [20] make use of this library for detecting collision and dynamics managements. Marilou Robotic Studio or Marilou AnyKode is known as a simulation software that can be used in robotic systems.

There are many features that have been developed and provided by Marilou AnyKode for modeling and developing simulated mobile robots or articulated robots. These features include sensors (accelerometers, force sensor, compas, encoder), actuators (dc motor and linear motor), camera, etc. To develop a robotic simulation, user needs to design the system model in Marilou Physics Editor and program it in C/C++ using Microsoft Visual Studio. Several programming languages are also provided other than C/C++ [20].

Although ODE based simulation has been associated to robotic applications for example to simulate mobile robot [21], humanoid robot [22], robot skin [23], etc, this computer simulation actually can also be used as a testbed in control system e.g. to validate the proposed control methods such as in adaptive control [17].

In this paper, Marilou AnyKode is used to model a reaction wheel pendulum i.e. the actuator and sensors are provided in this environment. Furthermore, the controller laws developed in this paper is implemented using C/C++ programming to simulate the system. The simulation of reaction wheel pendulum was developed as given in Fig. 2.

TABLE I. DEFINITIONS OF PARAMETERS OF SYSTEM

θ	Angle of pendulum with respect to vertical axis
φ	Angle of the wheel
l_w	Length of the wheel
l_b	Length of the body center
τ_w	Motor torque applied to reaction wheel
(x_o, y_o)	2D axes of space-fixed polar coordinate system
(x_p, y_p)	2D axes of body-fixed coordinate system

III. SECOND-ORDER SLIDING MODE CONTROL BASED ON SUPER-TWISTING ALGORITHM

The basic concept of sliding mode control is the switching feedback control i.e. the control law switches between different values according to switching function. The sliding controller synthesis consists of two steps: (a) design the sliding surface and (b) design the control action. When the system dynamics is constrained in sliding surface, the system dynamics is independent of the control and is governed by the chosen sliding surface's properties [24].

Despite of its advantages of simple controller and robustness, chattering phenomenon and its reduction are still challenging problem. Investigating the chattering amplitude with regard to first-order sliding mode control and super-twisting algorithm are discussed in [25,26]. In literature, choosing second-order sliding mode control instead of first-order sliding mode control may reduce the chattering phenomenon. However, Swikir [25] and Ventura [26] suggested that using super-

TABLE II. SYSTEM PARAMETERS

Symbol	Value and unit	Description
m	0.15 kg	Mass of pendulum
M	0.25 kg	Mass of reaction wheel
I_p	1.98 $\times 10^{-3}$ kg.m^2	Moment of inertia of pendulum
I_w	7.81 $\times 10^{-5}$ kg.m^2	Moment of inertia of reaction wheel
l_b	0.122 m	Length of body center
l_w	0.150 m	Length of reaction wheel
K_t	0.0375 N.m/A	Motor constant
R_a	4.7 Ω	Armature constant
g	9.8 m/s^2	Gravitational acceleration

twisting control does not necessarily reduce the chattering problem. In this paper, the second-order sliding mode control based on super-twisting algorithm is employed to stabilize the reaction wheel pendulum regardless of chattering problem.

The super-twisting algorithm [27] can be used for systems with relative degree one as the following form:

$$\dot{s} = \phi(s,t) + \gamma(s,t)u \tag{6}$$

where $\phi(\cdot)$ and $\gamma(\cdot)$ are known nonlinear functions and fulfill the conditions: $0 < |\phi(\cdot)| \leq \Phi$, and $0 \leq \Gamma_m \leq \gamma(\cdot) \leq \Gamma_M$. The super-twisting algorithm determines the control law of $u(t)$ by

$$\begin{aligned} u(t) &= u_1(t) + u_2(t) \\ \dot{u}_1 &= \begin{cases} -u, & |u| > 1 \\ -W\,\mathrm{sign}(s) & |u| \leq 1 \end{cases} \\ u_2 &= \begin{cases} -\lambda |s_0|^{\rho}\,\mathrm{sign}(s), & |s| > s_0 \\ -\lambda |s|^{\rho}\,\mathrm{sign}(s), & |s| \leq s_0 \end{cases} \end{aligned} \tag{7}$$

where $|s| < s_0$. The algorithm converges in finite time with following sufficient conditions:

$$\begin{aligned} & W > \frac{\Phi}{\Gamma_m} > 0; \;\; \lambda^2 \geq \frac{4\Phi\Gamma_M(M+\Phi)}{\Gamma_m^3(W-\Phi)}; \\ & 0 < \rho < \frac{1}{2}. \end{aligned} \tag{8}$$

The parameters of W and λ are needed to be tuned to ensure finite-time convergence and stability.

IV. DESIGN OF STABILIZING CONTROLLER

A. State-Space of Dynamic System

The system dynamics (3)-(4) can be rewritten as

$$\dot{x} = f(x) + g(x)u \tag{9}$$

where

$$f(x) = \begin{bmatrix} x_2 \\ \frac{(ml_b+Ml_w)g\sin x_1}{ml_b^2+Ml_w^2+I_p} \\ x_4 \\ -\frac{(ml_b+Ml_w)g\sin x_1}{ml_b^2+Ml_w^2+I_p} \end{bmatrix}, \tag{10}$$

and

$$g(x) = \begin{bmatrix} 0 \\ -\frac{nK_t}{ml_b^2+Ml_w^2+I_p} \\ 0 \\ nK_t\overline{m}_p \end{bmatrix}, \tag{11}$$

where $x = [x_1 \; x_2 \; x_3 \; x_4]^{\mathrm{T}}$ i.e.

$$\begin{aligned} x_1 &= \theta, \\ x_2 &= \dot{\theta}, \\ x_3 &= \varphi, \\ x_4 &= \dot{\varphi}. \end{aligned} \tag{12}$$

To find the change of variable, we want to choose

$$\phi(0) = 0, \;\; \frac{\partial \phi}{\partial x} g(x) = 0, \tag{13}$$

i.e. $\phi(x)$ is a function to be chosen such that a change of variable is a diffeomorphism on a domain $D_x \subset \Re$ [28]. Substituting eq. (11) into (13), we have

$$\begin{aligned} & \frac{\partial \phi}{\partial x_2} \cdot \left(-\frac{nK_t}{ml_b^2+Ml_w^2+I_p}\right) + \frac{\partial \phi}{\partial x_4} \cdot nK_t\overline{m}_p = 0, \\ & \phi(0) = 0. \end{aligned} \tag{14}$$

Eqs. (18) can be solved by using variable separation. One may obtain that

$$\phi(x) = x_2 + \frac{I_w}{I_p + I_w + ml_b^2 + Ml_w^2} x_4. \tag{15}$$

Let

$$y = \phi(x), \tag{16}$$

and hence, it can be rewritten as

$$y = \dot{\theta} + \frac{I_w}{I_p + I_w + ml_b^2 + Ml_w^2} \dot{\varphi}. \tag{17}$$

It follows that

$$\dot{y} = \frac{(ml_b + Ml_w)g\sin\theta}{I_p + I_w + ml_b^2 + Ml_w^2}. \tag{18}$$

Furthermore, by substituting (3) using

$$i = \frac{(ml_b + Ml_w)g\sin\theta - v(I_p + ml_b^2 + Ml_w^2)}{nK_t}, \tag{19}$$

we have

$$\ddot{\theta} = v + \rho. \tag{20}$$

where ρ is the disturbance and bounded.

In (12), φ is cyclic variable and can be ignored in the system [1,18]. Let $z = [z_1 \; z_2 \; z_3]^{\mathrm{T}}$ where z is

$$\begin{aligned} z_1 &= y, \\ z_2 &= \theta, \\ z_3 &= \dot{\theta}. \end{aligned} \tag{21}$$

Here, for simplicity, we assume that all states are available. The system dynamics in regular form can be rewritten as

$$\begin{aligned} \dot{z}_1 &= A\sin z_2, \\ \dot{z}_2 &= z_3, \\ \dot{z}_3 &= v + \rho \end{aligned} \tag{22}$$

i.e.:

$$A = \frac{(ml_b + Ml_w)\,g}{I_p + I_w + ml_b^2 + Ml_w^2}. \tag{23}$$

B. Design second-order sliding mode control

The system in (22) has relative degree three and so we need to design sliding surface such that the system has relative degree one with respect to sliding surface [29]. Let suppose the sliding surface is defined by

$$s = c_1 z_1 + c_2 z_2 + z_3, \tag{24}$$

where c_1, $c_2 > 0$. It follows that first derivative of s is

$$\dot{s} = c_1 \dot{z}_1 + c_2 \dot{z}_2 + \dot{z}_3, \tag{25}$$

and by subsituting with (22) we obtain

$$\dot{s} = c_1 A \sin z_2 + c_2 z_3 + v + \rho. \tag{26}$$

By choosing

$$v = -c_1 A \sin z_2 - c_2 z_3 + u \tag{27}$$

and substituting into (26), it follows that

$$\dot{s} = u + \rho, \tag{28}$$

where u is new control input. The controller law based on super-twisting algorithm is given

$$\begin{aligned} u &= u_1 + u_2 \\ u_1 &= -k_1 \, |s|^{\frac{1}{2}} \operatorname{sgn}(s) \\ \dot{u}_2 &= -k_2 \operatorname{sgn}(s). \end{aligned} \tag{29}$$

where k_1 and k_2 are the gain to be chosen. Substituting (29) into (28), we obtain

$$\dot{s} = u_1 + u_2 + \rho \tag{30}$$

If the gain k_1 and k_2 are chosen properly, the system can converge to zero in finite time [30].

The stability of (30) can be proven using Lyapunov approach [30,31]. Let equation (30) is rewritten in the form as follow [30]

$$\begin{aligned} \dot{x}_1 &= -k_1 |x_1|^{\frac{1}{2}} \operatorname{sgn}(x_1) + x_2 + \rho \\ \dot{x}_2 &= -k_2 \operatorname{sgn}(x_1) \end{aligned} \tag{31}$$

where we assume $\rho \leq \rho_0 |x_1|^{\frac{1}{2}}$. If the candidate Lyapunov function is chosen as

$$V = 2k_2 |x_1| + \frac{1}{2} x_2^2 + \frac{1}{2}(k_1 |x_1|^{\frac{1}{2}} \operatorname{sgn}(x_1) - x_2)^2 \tag{32}$$

Then, one may rewrite (32) in the form of

$$V = \xi^{\mathrm{T}} P \xi \tag{33}$$

where

$$\xi^{\mathrm{T}} = \begin{pmatrix} |x_1|^{\frac{1}{2}} \operatorname{sgn}(x_1) & x_2 \end{pmatrix} \tag{34}$$

and

$$P = \frac{1}{2} \begin{pmatrix} 4k_2 + k_1^2 & -k_1 \\ -k_1 & 2 \end{pmatrix}. \tag{35}$$

By calculating (33) and applying the bounds for perturbation as given in [30], it follows that

$$\dot{V} = -\frac{k_1}{2|x_1|^{\frac{1}{2}}} \xi^{\mathrm{T}} \tilde{Q} \xi \tag{36}$$

where

$$\tilde{Q} = \begin{pmatrix} 2k_2 + k_1^2 - (\frac{4k_2}{k_1})\rho_0 & -k_1 + 2\rho_0 \\ -k_1 + 2\rho_0 & 1 \end{pmatrix}. \tag{37}$$

If $\tilde{Q} > 0$, this implies that the derivative of Lyapunov function is negative definite, i.e. it is concluded that the system in (31) converges to zero. The control gains satisfy

$$k_1 > 2\rho_0, \quad k_2 > k_1 \frac{5\rho_0 k_1 + 4\rho_0^2}{2(k_1 - 2\rho_0)}. \tag{38}$$

The readers may refer [30,31] for detail proof of this stability.

C. Implementation of control law

To the super-twisting algorithm, we use (29) as the control input where v in (27) is

$$v = u - c_1 A \sin\theta - c_2 \dot{\theta}. \tag{39}$$

where

$$u = u_1 + \int u_2. \tag{40}$$

The parameters c_1 and c_2 are chosen with positive constants. The control gains k_1 and k_2 are tuned with proper values such that the system can be stabilized. The sampling time (T_s) of 10 milliseconds is chosen and the following algorithm are implemented in C/C++ programming:

Input: θ, $\dot{\theta}$, $\dot{\varphi}$
Output: v
Initialization : c_1, c_2, k_1, k_2, u, T_s, s_0
c_1, c_2, k_1, k_2
while (1) **do**
  $\theta \leftarrow$ *from sensor*
  $\dot{\theta} \leftarrow$ (*from sensor*)
  $\dot{\varphi} \leftarrow$ (*from encoder*)
  $y \leftarrow \dot{\theta} + a * \dot{\varphi}$
  $s \leftarrow c_1 y + c_2 \theta + \dot{\theta}$
  $u_a \leftarrow -c_1 A \sin\theta$
  $u_b \leftarrow -c_2 \, \dot{\theta}$
  // calculate u_1:
  if ($s > s_0$) **then**
    $u_1 \leftarrow -k_1 |s_0|^{\frac{1}{2}} \operatorname{sgn}(s)$
  else if ($s \leq s_0$) **then**
    $u_1 \leftarrow -k_1 |s|^{\frac{1}{2}} \operatorname{sgn}(s)$
  end if
  // calculate u_2:
  if ($u > 1$) **then**
    $u_2 \leftarrow (u_2 - u \times T_s)$
  else if ($u \leq 1$) **then**
    $u_2 \leftarrow (u_2 - k_2 \operatorname{sgn}(\mathrm{s}) \times T_s)$
  end if
  $u = u_1 + u_2$
  $v \leftarrow (u_a + u_b + u_1 + u_2)$
  wait (T_s)
end while
return P

V. Experimental Setup

The experimental setup of this research is carried out using ODE based simulation. The reaction wheel pendulum is developed in simulation with physical dimension as given

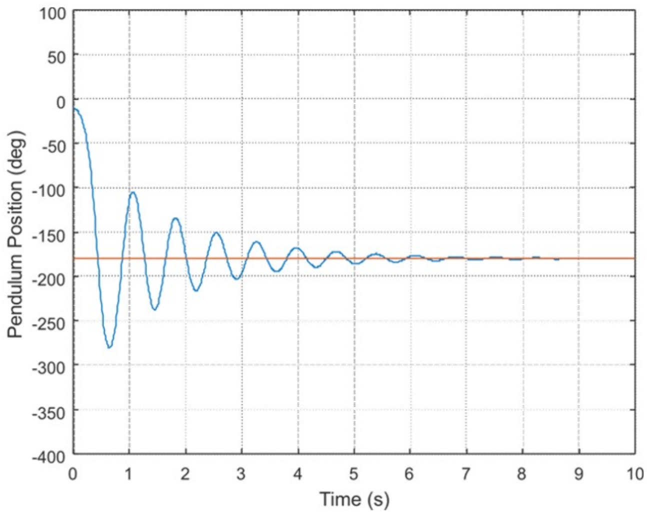

Fig. 3. Simulation result of reaction wheel pendulum with no control input.

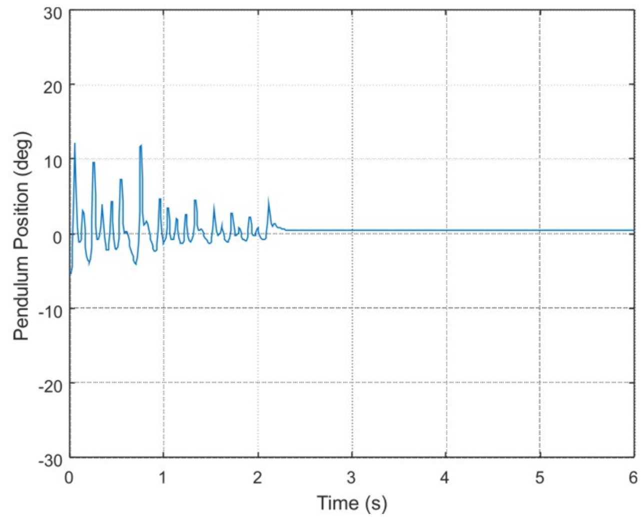

Fig. 4. Simulation result of reaction wheel pendulum with $-5°$ initial position.

in Table II. The controller is implemented in C/C++ program to control the system. The pendulum is attached to the edge of a base and the initial position of pendulum is assigned to have small deviation with respect to Z axis. Accelerometer sensor is attached to the tip of pendulum so that the angular position and rotation of pendulum can be detected. The wheel rotation can be measured by motor encoders. Thus, all of state variables are available to measure.

First, to verify the system dynamic behaviour, a simple method is applied to the system by releasing the pendulum from its upright position with small angle away from its equilibrium point. In this case, the initial position of pendulum is set to $-10°$ with no control input. As it is expected, the pendulum falls down and swings left and right and finally stops at $-180°$. The video demonstration of this simple test can be accessed in https://youtu.be/z4CesqfT8O4.

In the experiment, the scenario is to give different initial positions ranging from $0°$ to $-15°$ with respect to equilibrium point. The control gains and parameters are kept the same for all experiments, but the initial positions are not. The pendulum is released at each initial positions and observed if the pendulum can be balanced. At each initial position, the experiment is repeated 30 times. For each experiment, if the system can be balanced for more than 30 seconds then it counts as success, otherwise failure.

VI. Results and Discussion

First, the validation of a test bed is given by performing simple method without control input as discussed in previous section. It is shown that dynamic behaviour follows the law of physics i.e. the pendulum swings right and left as given in Fig. 3. To perform experiments based on given scenario, the system parameters of c_1 and c_2 are respectively assigned to 1.5 and 2.0 while the control gains for k_1 is 3.5 and k_2 is 4.8.

Here, one example of simulation performance is given in Fig. 4 where system with initial position $-5°$ can be stabilized using proposed stabilizing controller. It is shown that the pendulum converge to its equilibrium point in less than 1.5 second. This demonstrating video can be accessed in https://youtu.be/Zd5kbhnZjVo.

In Fig. 5, the initial position of pendulum is set to $-12°$ and the controller can stabilize the pendulum. However, the converging time is within 2.5 seconds i.e. longer than previous experiment. By increasing the initial positions, it may cause the control torque cannot bring the system back to its equilibrium point or otherwise the system can be balanced with longer converging time.

Experiments with different initial conditions can be summarized in Table III. It is shown that when initial position is $-15°$, the system cannot be stabilized. In fact, for initial position is $-12°$ the mean of success is 76.7% i.e. 23 times out of 30. If the smaller initial condition is given, the control scheme would effectively stabilize the pendulum on its upright position.

It is important to note that all experiments were using fixed system parameters (given in table II), and thus, the robustness of proposed controller cannot be observed. By changing these values, the same control gains can be determined if it can stabilize the pendulum.

Another aspect related to second-order sliding mode control is the chattering problem. It is commonly recognized that the sliding mode control may induce chattering effect. However, in experiments, the test bed behave as the real system and the chattering effect can not be seen obviously. At least not as clearly as using Matlab/Simulink simulation. Nonetheless,

TABLE III. Experiment with different initial conditions

Init. pos.	**Number of experiments**	**Number of success**	**Persentage**
$0°$	30 times	30 times	100 %
$-5°$	30 times	27 times	90 %
$-10°$	30 times	28 times	93.3 %
$-12°$	30 times	23 times	76.7 %
$-15°$	30 times	0 times	0 %

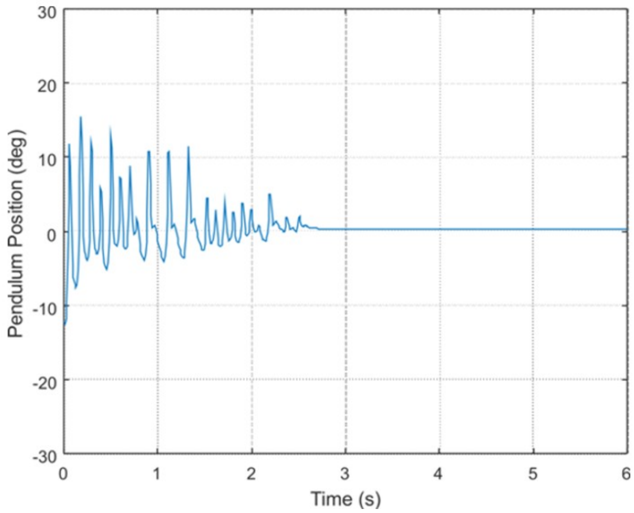

Fig. 5. Simulation result of reaction wheel pendulum with $-12°$ initial position.

as given in Fig. 4 and 5 for different initial conditions, it seems that the chattering effect were taken place.

VII. Conclusion

Second-order sliding mode control based on super-twisting algorithm was employed to stabilized a reaction wheel pendulum system. The ODE-based simulation for reaction wheel pendulum can be used as a testbed to verify the proposed control method. It is shown the efficacy of the proposed controller if the initial angular positions is on vicinity of equilibrium point with $\theta \leq -12°$.

References

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

[2] Y. Zhu, Y. Gao, C. Xu, J. Zhao, H. Jin, and J. Lee, "Adaptive control of a gyroscopically stabilized pendulum and its application to a single-wheel pendulum robot," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2095–2106, 2015.

[3] Z. Chen, X. Ruan, Y. Li, Y. Bai, and X. Zhu, "Dynamic modeling of a self-balancing cubical robot balancing on its edge," in *International Conference on Robotics and Automation Engineering*, 2017, pp. 11–15.

[4] M. W. Spong, P. Corke, and R. Lozano, "Nonlinear control of the reaction wheel pendulum," *Automatica*, vol. 37, no. 11, pp. 1845–1851, 2001.

[5] Y. Rizal, J.-W. Syu, and M.-T. Ho, "Balance control of an inverted pendulum system using second-order sliding mode control," in *CACS International Automatic Control Conference*, 2014, pp. 191–196.

[6] G. Bartolini, A. Ferrara, A. Levant, and E. Usai, "On second order sliding mode controllers," in *Variable structure systems, sliding mode and nonlinear control*. London: Springer, 1999, pp. 329–350.

[7] G. Bartolini, A. Ferrara, and E. Usai, "Applications of a sub-optimal discontinuous control algorithm for uncertain second-order systems," *International Journal of Robust and Nonlinear Control*, vol. 7, no. 4, pp. 299–319, 1997.

[8] M. Olivares and P. Albertos, "Linear control of the flywheel inverted pendulum," *ISA Transactions*, vol. 53, no. 5, pp. 1396–1403, 2014.

[9] I. Fantoni, R. Lozano, and M. W. Spong, "Stabilization of the reaction wheel pendulum using an energy approach," in *European Control Conference*, 2001, pp. 2552–2557.

[10] F. Jepsen, A. Soborg, A. R. Pedersen, and Z. Yang, "Development and control of an inverted pendulum driven by a reaction wheel," in *International Conference on Mechatronics and Automation*, 2009, pp. 2829–2834.

[11] D. J. Block, K. J. Åström, and M. W. Spong, *The reaction wheel pendulum*. Morgan & Claypool, 2007.

[12] J. Scerri, G. S. Dordević, and D. Todorović, "Modeling and control of a reaction wheel pendulum with visual feedback," in *International Conference on Control, Automation and Diagnosis*, 2017, pp. 24–29.

[13] M. T. Ho, Y. Rizal, and Y. L. Chen, "Balance control of a unicycle robot," in *IEEE International Symposium on Industrial Electronics*, 2014, pp. 1186–1191.

[14] H. Gritli and S. Belghith, "Robust feedback control of the underactuated inertia wheel inverted pendulum under parametric uncertainties and subject to external disturbances: LMI formulation," *Journal of the Franklin Institute*, pp. 1–42, 2017.

[15] D. Gutierrez-Oribio, A. Mercado-Uribe, J. A. Moreno, and L. Fridman, "Stabilization of the reaction wheel pendulum via a third order discontinuous integral sliding mode algorithm," in *International Workshop on Variable Structure Systems (VSS)*, 2018, pp. 132–137.

[16] A. Chalanga, S. Kamal, L. M. Fridman, B. Bandyopadhyay, and J. A. Moreno, "Implementation of super-twisting control : super-twisting and higher order sliding mode observer based approaches," *IEEE Transaction on Industrial Electronics*, vol. 63, no. 6, pp. 3677–3685, 2016.

[17] A. Naba, "Adaptive control based on adjustable evaluation function," 2007.

[18] I. Fantoni and L. Rogelio, "The reaction wheel pendulum," in *Nonlinear Control for Underactuated Mechanical Systems*. London: Springer, 2002, ch. 7, pp. 89–106.

[19] R. Smith, "Open dynamics engine," pp. 1–96, 2010.

[20] L. Ricatti, L. Junillon, and I. Metenier, "Marilou robotic studio," 2018. [Online]. Available: www.anykode.com

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

[22] 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," in *Workshop on Performance Metrics for Intelligent Systems*, 2008, pp. 57–61.

[23] A. Habib, I. Ranatunga, K. Shook, and D. O. Popa, "SkinSim: A simulation environment for multimodal robot skin," in *IEEE International Conference on Automation Science and Engineering*, 2014, pp. 1226–1231.

[24] M. K. Khan, "Design and application of second order," Ph.D. dissertation, University of Leicester, 2003.

[25] A. Swikir and V. Utkin, "Chattering analysis of conventional and super-twisting sliding mode control algorithm," in *International Workshop on Variable Structure Systems*, 2016, pp. 98–102.

[26] U. P. Ventura and L. Fridman, "Is it reasonable to substitute discontinuous SMC by continuous HOSMC?" *arXiv preprint arXiv:1705.09711*, pp. 1–7, 2017.

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

[28] H. K. Khalil, *Nonlinear Control*. Pearson Education Ltd, 2015.

[29] K. Kumari, A. Chalanga, and B. Bandyopadhyay, "Implementation of Super-Twisting control on higher order perturbed integrator system using higher order sliding mode observer," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 873–878, 2016.

[30] J. A. Moreno and M. Osorio, "A Lyapunov approach to second-order sliding mode controllers and observers," *Proceedings of the IEEE Conference on Decision and Control*, pp. 2856–2861, 2008.

[31] J. Rivera, L. Garcia, C. Mora, J. J. Raygoza, and S. Ortega, "Super-twisting sliding mode in motion control systems," in *Sliding Mode Control*. InTech, 2011, pp. 237–254.