# Application of Sliding Mode Trajectory Tracking Control Design for Two-Wheeled Mobile Robots

Yankun Yang
*University of Kent*
Canterbury, Kent
yy237@kent.ac.uk

Xinggang Yan
*University of Kent*
Canterbury, Kent
x.yan@kent.ac.uk

Konstantinos Sirlantzis
*University of Kent*
Canterbury, Kent
k.sirlantzis@kent.ac.uk

Gareth Howells
*University of Kent*
Canterbury, Kent
w.g.j.howells@kent.ac.uk

***Abstract*—A trajectory tracking controller is proposed to drive the wheeled mobile robot (WMR) to follow a predefined trajectory robustly within a finite time under the presence of uncertainties. The two-wheeled mobile robot and tracking error system are modelled by kinematic equations, the stability and reachability of the sliding mode controller are analysed based on the system models. A two-wheeled mobile robot is built by using the STM32F407 (ARM Cortex-M4 microcontroller) board and a MATLAB GUI, and a cooperative real-time operating system are implemented by using C programming language in order to provide convenient system configuration and improve the overall tracking performance. It is demonstrated that the line and circular trajectories are well tracked in simulation and experiment.**

***Index Terms*—trajectory tracking, wheeled mobile robot, sliding mode control, nonlinear systems, robotics, embedded system.**

## I. Introduction

Two-wheeled differential drive mobile robots (TWMR) have been widely used in industrial and civilian life. As wheeled mobile robot (WMR) is a highly non-linear non-holonomic mechanical system. This constraint system is introduced by a Germany scholar Heinrich Hertz in 1894 [1], which dividing into two categories, the former is the holonomic constraint system and the latter is the non-holonomic constraint system. The holonomic constraint system imposes restrictions on the position and orientation of WMRs, whereas the non-holonomic constraint system restricts the motion of a WMR. In details, the nature of non-holonomic constraint is that no motion occurs in the axial direction of the wheel of WMR, there is no sliding but rolling motion between the ground surface and the wheels. This assumption implies that there exists a set of non-integral first-order differential constraints [2]. Moreover, Brockett theorem indicates that the utilisation of differentiable or continuous time-invariant feedback control law which is unable to stabilise the non-holonomic system [3].

Due to the difficulty of stabilising a non-holonomic, non-linear, strong coupling, time-variant system of wheeled mobile robots, a number of related papers [4], [5] have been proposed to resolve this problem. In general, there are primarily two control techniques to solve trajectory tracking control of a non-holonomic wheeled mobile robot, which are back-stepping method [6], [7] and sliding mode method [9].

Back-stepping method is based on the adjustment of each step related to the combination of Lyapunov function and virtual control systems so as to design a controller to stabilise the system. The advantage of using this method is to divide the whole system into small sub-systems and conquer each part individually. The disadvantages, however, are the difficulties to design Lyapunov function and sensitive to parameter variation. For the non-linear sliding mode control method, which consisting of two different phases, firstly, the system reaches to the sliding surface in a finite amount of time, once the system is on the sliding surface, the sliding phase takes place and slides toward to the equilibrium point. The advantages of using sliding mode method are applicable for non-linear systems even for higher order systems, easy to combine other control methods, such as PID, fuzzy controls, insensitive to parameter variation and robust to matched uncertainties. The disadvantage is chattering effect due to the fast discontinuous switching motions. In addition, another common control method widely used in industry is PID control which has better performance to resolve linear system, due to the non-linear characteristics of the wheeled mobile robots and the tracking error system, it is difficult to design PID controllers to achieve better performance of the trajectory tracking control [10].

Sliding mode technique is a special non-linear control of the variable structure control [8], its non-linearity is mainly represented by the characteristics of discontinuity, the difference between sliding mode control and other control methods is the flexible structure of the system, which can be varied constantly based on the current states, such as errors in order to force the system to perform the tracking motion according to the predefined ”sliding mode” [9]. The design of sliding mode technique contains two sections, one is the stability analysis, the other is the reachability analysis. The sliding mode control design guarantees that the wheeled mobile robot will track on the predefined trajectory in a limited amount of time and stay on the path robustly under the presence of uncertainties [11]–[15].

## II. System modelling

The modelling of two-wheeled differential drive mobile robots can be defined in two different coordinate systems, i.e., Inertial Reference Coordinate Frame System ($X^I$-$Y^I$) and

Wheeled Mobile Robot Body Coordinate Frame System ($X^B$-$Y^B$) as illustrated in Fig 1.

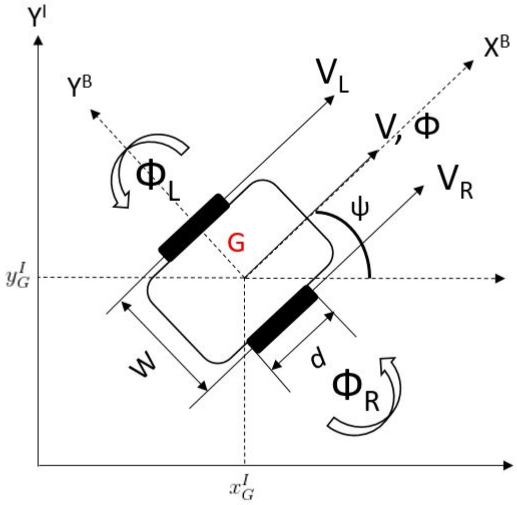


Figure 1: Modelling of two-wheeled differential drive mobile robot

where $\Psi$ is the yaw (heading) angle, $d$ is the diameter of each wheel, $W$ is the width of TWMR, $\Phi_L$ and $\Phi_R$ are angular velocities of left and right wheels respectively, $V$, $\Phi$ are the linear and angular velocities accordingly, $V_L$ is the linear velocity of the left wheel, $V_R$ is the linear velocity of the right wheel.

### A. Kinematic Constraints of TWMR

Consider $P^B_G = [x^B_G, \, y^B_G, \, \Psi^B]^T$ and $P^I_G = [x^I_G, \, y^I_G, \, \Psi^I]^T$ be the TWMR postures of body and inertial coordinate frames. The relationship between these two coordinate systems are defined as follows.

$$P^I_G = R(\Psi) P^B_G \tag{1}$$

where $P^I_G$ and $P^B_G$ are the postures of inertial and body frames, $R(\Psi) = \begin{bmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is the rotation matrix along the z-axis. Equation (1) is also satisfied for derivative, which describes the relationship between velocities in the inertial and body coordinate frames.

$$\dot{P}^I_G = R(\Psi) \dot{P}^B_G \tag{2}$$

The two-wheeled mobile robot is characterised by three non-holonomic constraints, which are acquired by two primary assumptions.

- Slip-free lateral movement
  This constraint indicates that the steering of the wheeled mobile robot can only be driven in a curved motion rather than lateral movement, which means that the velocity of the centre gravity $G$ is zero along the lateral axis ($Y^B$).
  $$\dot{y}^B_G = 0 \tag{3}$$
  The first constraint is derived by using (2) and (3)
  $$-\dot{x}^I_G sin\Psi + \dot{y}^I_G cos\Psi = 0 \tag{4}$$
- Slip-free pure rolling movement
  Pure rolling represents no slippage and skidding motion along the longitudinal and orthogonal axes of the body frame $X^B$ and $Y^B$
  $$\dot{x}^I_G cos\Psi + \dot{y}^I_G sin\Psi + \frac{W}{2}\dot{\Psi} - \frac{d}{2}\dot{\Phi}_R = 0 \tag{5}$$
  $$\dot{x}^I_G cos\Psi + \dot{y}^I_G sin\Psi - \frac{W}{2}\dot{\Psi} - \frac{d}{2}\dot{\Phi}_R = 0 \tag{6}$$

The general form of constraint equations of the two-wheeled mobile robot is

$$\Lambda(P^I)\dot{P^I} = 0 \tag{7}$$

where $\Lambda(P^I) = \begin{bmatrix} -sin\Psi & cos\Psi & 0 & 0 & 0 \\ cos\Psi & sin\Psi & \frac{W}{2} & -\frac{d}{2} & 0 \\ cos\Psi & sin\Psi & -\frac{W}{2} & 0 & -\frac{d}{2} \end{bmatrix}$ and $P^I = [\dot{x}^I_G, \, \dot{y}^I_G, \, \dot{\Psi}^I, \, \dot{\Phi}_R, \, \dot{\Phi}_L]^T$.

### B. Kinematic Modelling of TWMR

The linear and angular velocities of the control input ($\mu$) in terms of the angular rate of each motor wheel are as follows.

$$V = \frac{V_R + V_L}{2} = \frac{\frac{d}{2}\Phi_R + \frac{d}{2}\Phi_L}{2} = \frac{d(\Phi_R + \Phi_L)}{4} \tag{8}$$

$$\Phi = \frac{V_R - V_L}{W} = \frac{\frac{d}{2}\Phi_R - \frac{d}{2}\Phi_L}{W} = \frac{d(\Phi_R - \Phi_L)}{2W} \tag{9}$$

which yields

$$\mu = \begin{bmatrix} V \\ \Phi \end{bmatrix} = \begin{bmatrix} \frac{d}{4} & \frac{d}{4} \\ \frac{d}{2W} & -\frac{d}{2W} \end{bmatrix} \begin{bmatrix} \Phi_R \\ \Phi_L \end{bmatrix} \tag{10}$$

The trajectory tracking system satisfies the following conditions,

$$P_e = \lim_{x \to \infty} \|P_B - P_A\| = 0 \tag{11}$$

where $P_e = [X_e, \, Y_e, \, \Psi_e]$ is the tracking error from position A to position B in Fig 2.

As the majority of the rotation of the TWMR encountered under Euclidean 3D space with the fixed Z-axis, the only consideration is the rotation about the centre of gravity $G$ around Z-axis, which utilising the rotation matrix described in (1). Thus, the tracking error system can be represented as follows

$$P_e = R(\Psi)(P_B - P_A) \tag{12}$$

The general attitude vector of the TWMR can be represented as $P = [X, Y, \Psi]^T$ and $\mu = [V, \Phi]^T$ is the control input, where $[X, Y], \Psi$ are the position and heading angle of the TWMR with respect to the horizontal axis, $V$ and $\Phi$ are the linear and angular velocities respectively.

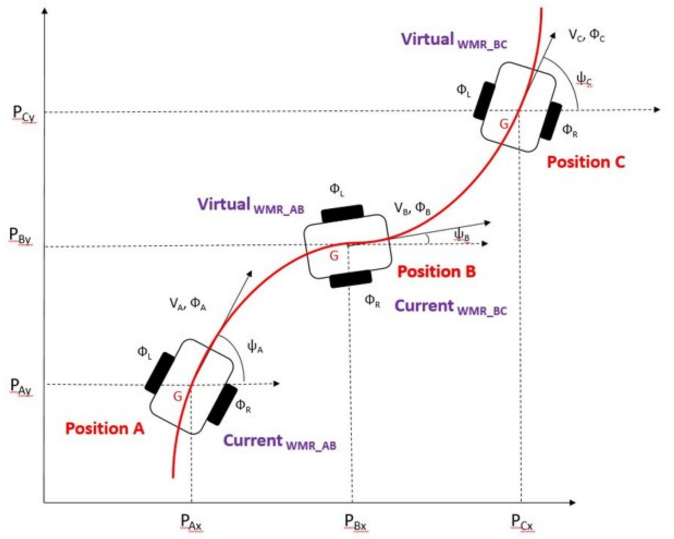


Figure 2: Motion of two-wheeled differential drive mobile robot

The asymptotically convergent rate of the tracking error system with respect to time t from (12) is derived in the following.

$$\dot{X}_e(t) = V_B(t)cos\Psi_e(t) - V_A(t) + Y_e(t)\Phi_A(t) \tag{13}$$

$$\dot{Y}_e(t) = V_B(t)sin\Psi_e(t) - \Phi_A(t)X_e(t) \tag{14}$$

$$\dot{\Psi}_e(t) = \Phi_B(t) - \Phi_A(t) \tag{15}$$

Put into matrix form yields,

$$\dot{P}_e(t) = \begin{bmatrix}\dot{X}_e\\ \dot{Y}_e\\ \dot{\Psi}_e\end{bmatrix} = \underbrace{\begin{bmatrix}V_B cos\Psi_e\\ V_B sin\Psi_e\\ \Phi_B\end{bmatrix}}_{M(t,P_e)} + \underbrace{\begin{bmatrix}-1 & Y_e\\ 0 & -X_e\\ 0 & -1\end{bmatrix}}_{N(t,P_e)} \underbrace{\begin{bmatrix}V_A\\ \Phi_A\end{bmatrix}}_{\mu(t)} \tag{16}$$

The objective is to design a robust controller to minimise the tracking error system (16) to zero whilst maintaining the stability of the overall system, which will be investigate in the following section.

## III. Control strategies

### A. Stability analysis of sliding mode

Equation (14) can be rearranged to $\dot{Y}_e = V_B sin\Psi_e$ when $X_e = 0$, the Lyapunov function can be generated as follows.

$$V_Y = \frac{1}{2}Y_e^2 \tag{17}$$

Suppose $\Psi_e = -\frac{K_2}{K_1}arctan(Y_e)$, where $K_1$ and $K_2$ are two positive numbers.
The rate of change of $V_Y$ is as follows.

$$\dot{V_Y} = Y_e\dot{Y_e} \tag{18}$$

Substituting (14) into (18) gives,

$$\dot{V_Y} = -V_B sin(\frac{K_2}{K_1}arctan(Y_e))Y_e - X_eY_e\Phi_A <= 0$$

It can be easily to prove that the selected Lyapunov function is negative definite, and the equality holds if and only if $Y_e = 0$ which can be achieved by making $X_e$ converge to zero, $\Psi_e$ converges to $-\frac{K_2}{K_1}arctan(Y_e)$.

### B. Design of sliding surface

The non-linear sliding surfaces can be generated as follows.

$$\alpha = \begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} = \begin{bmatrix}X_e\\ K_1\Psi_e + K_2 arctan(Y_e)\end{bmatrix} \tag{19}$$

where $K_1$ and $K_2$ are positive numbers.

The rate of change of the non-linear sliding surfaces gives,

$$\dot{\alpha} = \begin{bmatrix}\dot{\alpha_1}\\ \dot{\alpha_2}\end{bmatrix} = A(P_e)\begin{bmatrix}V_B cos\Psi_e\\ V_B sin\Psi_e\\ \Phi_B\end{bmatrix} + B(P_e)\mu_{NL} \tag{20}$$

where

$$A(P_e) = \begin{bmatrix}1 & 0 & 0\\ 0 & \frac{K_2}{1+Y_e^2} & K_1\end{bmatrix}, B(P_e) = \begin{bmatrix}-1 & Y_e\\ 0 & -K_1 - \frac{K_2X_e}{1+Y_e^2}\end{bmatrix}$$

$$\mu_{NL} = \begin{bmatrix}V_A\\ \Phi_A\end{bmatrix}, X = \begin{bmatrix}V_B cos\Psi_e\\ V_B sin\Psi_e\\ \Phi_B\end{bmatrix}$$

Rearranging (20) gives the control input $\mu$ in terms of controllable states and the non-linear sliding surfaces.

$$\mu_{NL} = -B(P_e)^{-1}\left(A(P_e)\begin{bmatrix}V_B cos\Psi_e\\ V_B sin\Psi_e\\ \Phi_B\end{bmatrix} + \underbrace{\begin{bmatrix}\Gamma_1 sgn(\alpha_1)\\ \Gamma_2 sgn(\alpha_2)\end{bmatrix}}_{discontinuity}\right) \tag{21}$$

The derivative of the non-linear sliding surfaces can be easily derived by substituting (21) into (20).

$$\dot{\alpha} = \begin{bmatrix}-\Gamma_1 sgn(\alpha_1)\\ -\Gamma_2 sgn(\alpha_2)\end{bmatrix} \tag{22}$$

where $\Gamma_1$ and $\Gamma_2$ are reaching gains coefficients of non-linear sliding surfaces.

To alleviate the chattering effect from the control input $\mu$, (22) is redesigned using saturation functions,

$$\dot{\alpha} = \begin{bmatrix}\dot{\alpha_1}\\ \dot{\alpha_2}\end{bmatrix} = \begin{bmatrix}-\Gamma_1\frac{\alpha_1}{\|\alpha_1\|+\rho_1}\\ -\Gamma_2\frac{\alpha_2}{\|\alpha_2\|+\rho_2}\end{bmatrix} \tag{23}$$

where $\rho_1$ and $\rho_2$ are boundary layer parameters.

The general control law can be further developed from (21) using the saturation function as follows,

$$\mu_{NL} = -B(P_e)^{-1}\left(A(P_e)\begin{bmatrix}V_B cos\Psi_e\\ V_B sin\Psi_e\\ \Phi_B\end{bmatrix} + \underbrace{\begin{bmatrix}\Gamma_1\frac{\alpha_1}{\|\alpha_1\|+\rho_1}\\ \Gamma_2\frac{\alpha_2}{\|\alpha_2\|+\rho_2}\end{bmatrix}}_{saturation\ \ function}\right) \tag{24}$$

**Remark.** *The non-linear control law (24) is applicable if and only if the following conditions are hold,*

$$X_e \neq -\frac{K_1}{K_2}(1+Y_e^2),\ \ Y_e \in \Re,\ |\Psi_e| \leq \pi \tag{25}$$

*For $X_e \to -\frac{K_1}{K_2}(1+Y_e^2)$, the control law will suffer from the local minimum issue as the matrix $B(P_e)$ becomes singular. Nevertheless, the actual two-wheeled mobile robot will mainly keep track of the reference trajectory at all time without affecting the tracking performance.*

## IV. Application to WMR

### A. Hardware implementation

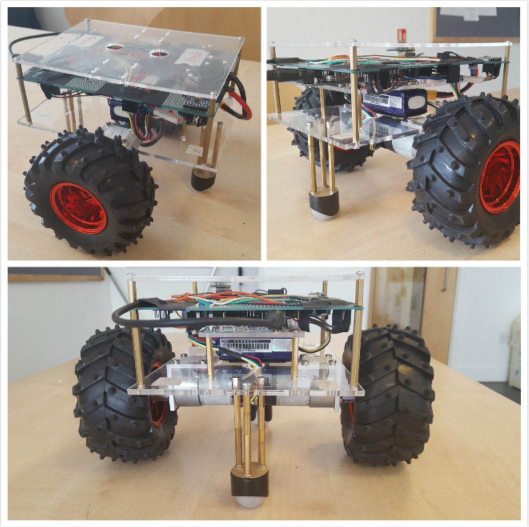


Figure 3: Application of two-wheeled mobile robot

The two-wheeled differential drive mobile robot is built and assembled to demonstrate the feasibility and performance of the trajectory tracking control in a practical manner, which is illustrated in Fig 3. The robot consists of four different sub-modules which are execution unit, acquisition unit, actuation unit and communication/data-logging unit.

- The execution unit is mainly the microcontroller board (STM32F407VGT6 in Fig 4) which collecting and processing the sensor data and executing the trajectory tracking control algorithms to achieve the objective. The built-in microcontroller chip (ARM Cortex-M4) is operated under the maximum 168 MHz frequency and up to 82 General Purpose Input and Output (GPIO) ports available for connecting external devices, and up to 12 16-bit Timers for hardware pulse width modulation (PWM) signal generation, signal input capture and encoder data acquisition via the dedicated interfaces. In addition, 6 USART interfaces are supported for data communication and programs debugging purposes, and 3 SPIs interfaces for high speed data transmission.
- An Inertial Measurement Unit (IMU) is utilised to measure the orientation of the robot accurately in Fig 5. This IMU (MPU6050) contains 3 axes of gyroscope which can be configured to measure the angular velocities of 3-axis up to ±2000°/sec.

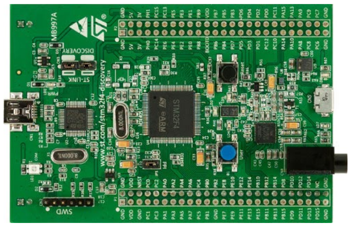


Figure 4: Main microcontroller board

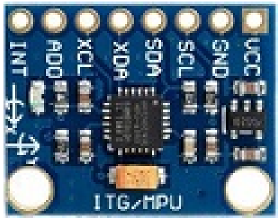


Figure 5: Inertial Measurement Unit module

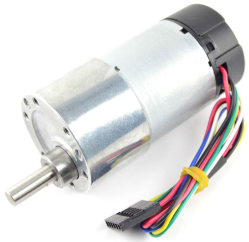


Figure 6: DC brushed motor with encoder

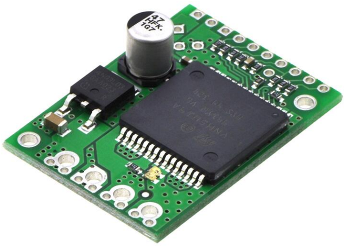


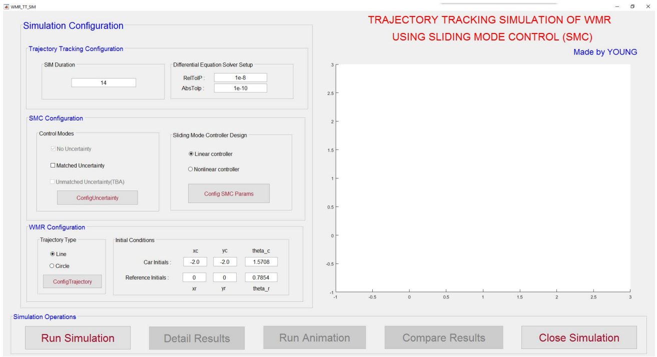
Figure 7: DC brushed motor driver

- Two 12V DC brushed actuators (Pololu 30:1 motor gearbox with encoder in Fig 6) are mounted on the left and right sides of the TWMR respectively for differential driving purposes. The quadrature incremental encoder (with resolution of 64 counts per revolution) is attached on the motor shaft to measure the angular positions and velocities of the motor. Moreover, two independent fully integrated H-bridge high current motor drivers (VNH5019), illustrated in Fig 7, are used for bidirectional speed control of the DC brushed motors along with the hall effect current sensing module (140 mV/A) for motor current measurement. In addition, motors are controlled by the PWM signals which are generated by the microcontroller board (STM32F407).
- Communication module utilises bluetooth device to send the real-time data to the MATLAB graphical user interface (GUI) software on PC for further data analysis, and the data are also logged into the on-board SD card module simultaneously.

### B. Software implementation

A MATLAB GUI simulator and a light-weighted Cooperative Real-time Operating System (RTOS) are implemented using C programming language for simulation and system performance improvement.

The MATLAB GUI simulator, illustrated in Fig 8, is a convenient software to configure the parameters of trajectory tracking controller, types of trajectories, initial TWMR and reference attitudes. Moreover, the static and animated tracking results are displayed along with the detailed information of the control related outputs, such as the stability of the sliding surfaces, tracking error system, linear and angular velocities and so on.



SIM Duration: 14

Differential Equation Solver Setup
RelTolP : 1e-8
AbsTolp : 1e-10

☑ No Uncertainty
☐ Matched Uncertainty
☐ Unmatched Uncertainty(TBA)

● Linear controller
○ Nonlinear controller

● Line
○ Circle

|  | xc | yc | theta_c |
|---|---|---|---|
| Car Initials : | -2.0 | -2.0 | 1.5708 |
| Reference Initials : | 0 | 0 | 0.7854 |
|  | xr | yr | theta_r |

Figure 8: MATLAB GUI simulator of trajectory tracking control

The cooperative RTOS, implemented on STM32F407 microcontroller board, consists of a scheduler and seven different tasks which are able to execute simultaneously using different time periods in Fig 9.

1) System task, running at 10 Hz, calculates the CPU and tasks utilisations of the overall system.
2) Gyroscope and accelerometer tasks, both running at 1000 Hz, use the I2C interface to reach the maximum sampling frequency for better data accuracy.
3) Attitude task, running at 100 Hz, utilises the Madgwick's data fusion algorithm [16] to estimate the Euler angles (Roll, Pitch and Yaw) for orientation usages of the WMR.
4) Motor encoder task, running at 100 Hz, reads the motor angular velocities from the encoder module and feeds the data into two separate PI controllers along with the desired angular velocities generated from (10) to precisely control two DC brushed motors.
5) Trajectory tracking control task, running at 100 Hz, calculates the attitudes of the actual WMR, reference WMR and tracking error system. Furthermore, applying the non-linear sliding mode algorithm to generate the desired linear and angular velocities of the robot, the reference angular velocities of the motors are further calculated and sent to the motor encoder task.
6) Data communication and logging task, running at 20 Hz, send the data to MATLAB GUI for real-time visualisation of trajectory tracking control whilst logging the data to on-board SD card for detailed off-line data analysis thereafter.

## V. Simulation and Experimental Results

The sliding mode trajectory tracking control was demonstrated both under simulation and actual experiment of the WMR. This paper presents line and circular trajectory tracking and the corresponding results of simulation and experiment are illustrated as follows.

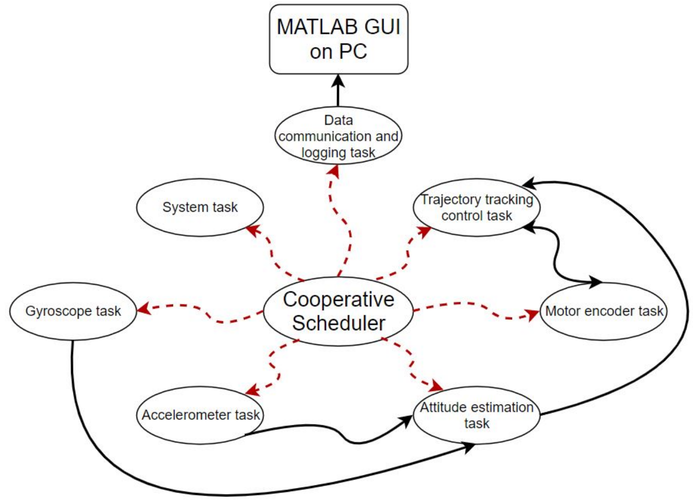


Figure 9: Architecture of cooperative real-time operating system

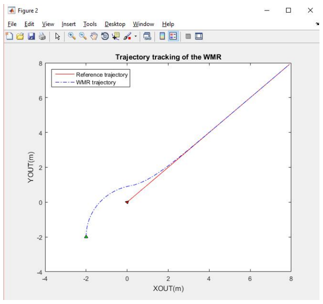


Figure 10: Simulation result of line trajectory tracking

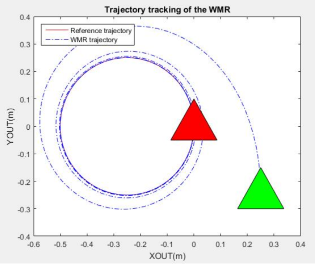


Figure 11: Simulation result of circular trajectory tracking

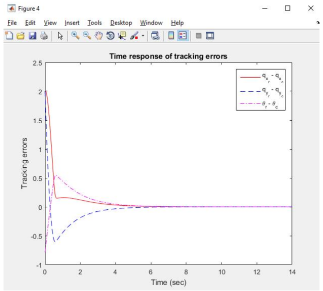


Figure 12: Line tracking error results

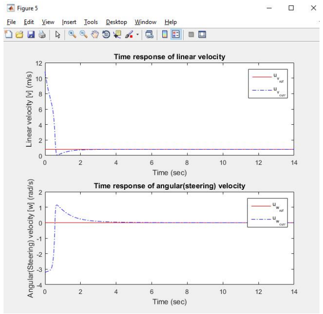


Figure 13: Linear and angular velocities results

### A. Line tracking

The configuration of line tracking consists of the initial attitude of actual WMR $P_{WMR} = [-0.2, -0.2, 1.5708]^T$, the attitude of reference WMR $P_{ref} = [0, 0, 0.7854]^T$ and the reference of linear and angular velocities $[V_{ref}, \Phi_{ref}] = [0.2, 0.0]$.

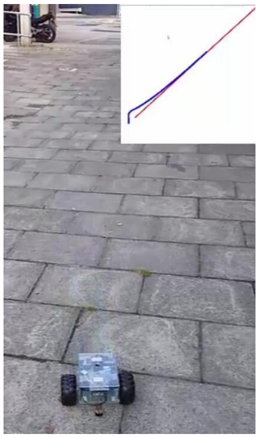


Figure 14: Line trajectory tracking using WMR

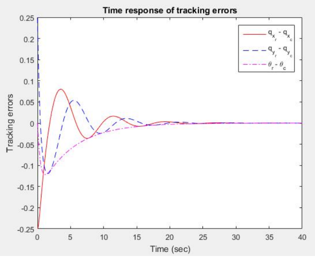


Figure 15: Circular tracking error results

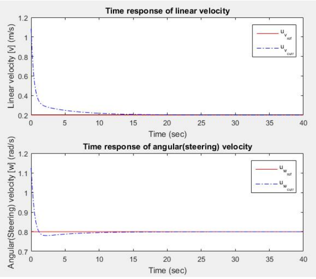


Figure 16: Linear and angular velocities results

### B. *Circular tracking*

The configuration of circular tracking consists of the initial attitude of actual WMR $P_{WMR} = [0.25, -0.25, 1.5708]^T$, the attitude of reference WMR $P_{ref} = [0, 0, 1.5708]^T$ and the reference of linear and angular velocities $[V_{ref}, \Phi_{ref}] = [0.2, 0.8]$.

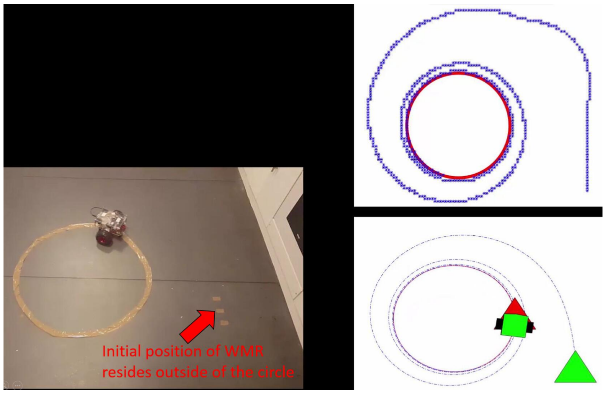


Figure 17: Circular trajectory tracking using WMR

The top right images of Fig 14 and Fig 17 are the real-time trajectory tracking results and the compared simulation result on the bottom right of Fig 17. Based on the simulation and experimental results, it can be demonstrated that the tracking error system is asymptotically stable within a finite amount of time in spite of matched uncertainties from the motor input channels and unmatched uncertainties due to the wheel slippage of high angular velocity at the first steering turn in Fig 17.

## VI. Conclusions

This paper presents a sliding mode trajectory tracking controller design. A two-wheeled mobile robot is built and the tracking performance is achieved both in simulation and actual wheeled mobile robot. It is also showed that the matched uncertainties from the motor controlled input channel do not affect the tracking performance as the sliding motion is completely independent of the matched uncertainties once it has been attained onto the sliding surfaces.

## References

[1] Berry, and Michael, “Anticipations of Geometric Phase,” Physics Today, vol. 43, pp. 34–40, 1990.

[2] D. Z. Cheng, “Controllability of Switched Bilinear Systems,” IEEE Trans. on Automatic Control. vol. 50, pp. 511–515, 2005.

[3] R.W. Brockett, R.S. Millman and H.J. Sussman, Eds, “Asymptotic Stability and Feedback Stabilization, Differential Geometric Control Theory,” in Birkhauser, Boston, USA, pp. 181–191, 1983.

[4] Kim Duyen Ha Thi, Manh Cuong Nguyen, Hoang Thuat Vo, Van Manh Tran, Duc Dinh Nguyen, Anh Dung Bui, “Trajectory tracking control for four-wheeled omnidirectional mobile robot using Backstepping technique aggregated with sliding mode control,” 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), pp. 131–134, 2019.

[5] Samia Sahloul, Donia Benhalima, Chokri Rekik, “Tracking Trajectory of a Mobile Robot Using Sliding Mode Control,” 2018 15th International Multi-Conference on Systems Signals and Devices (SSD), pp. 1386–1390, 2018.

[6] P.V. Kokotovic, “The joy of feedback: nonlinear and adaptive,” IEEE Control Systems. vol. 12, pp. 7–17, 1992.

[7] Rogelio Lozano and Bernard Brogliato, “Adaptive control of robot manipulators with flexible joints,” IEEE Transactions on Automatic Control. vol. 37, pp. 174–181, 1992.

[8] Xing-Gang Yan, Sarah Spurgeon and Edwards, Variable Structure Control of Complex Systems: Analysis and Design. Springer, 2017.

[9] V. I. Utkin, “Variable Structure Systems with Sliding Modes,” IEEE Transactions Automatic Control. vol. 2, pp. 212–222, 1977.

[10] Peter Gruber and Silvano Balemi, “Overview of non-linear control methods,” Swiss Society for Automatic Control. 2010.

[11] X. G. Yan and C. Edwards and S. K. Spurgeon and J. A. M. Bleijs, “Decentralised sliding-mode control for multimachine power systems using only output information,” IEEE Proceedings - Control Theory and Applications. vol. 151, no. 4, pp. 627–635, 2004.

[12] X. G. Yan and S. K. Spurgeon and C. Edwards, “Global decentralised static output feedback slidingmode control for interconnected time-delay systems,” IET Control Theory and Applications. vol. 6, no. 2, pp. 192–202, 2012.

[13] X. G. Yan and S. K. Spurgeon and C. Edwards, “Global time-delay dependent decentralised sliding mode control using only output information,” Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on. pp. 6709–6714, 2009.

[14] X. G. Yan and S. K. Spurgeon and C. Edwards, “State and Parameter Estimation for Nonlinear Delay Systems Using Sliding Mode Techniques,” IEEE Transactions on Automatic Control. vol. 58, no. 4, pp. 1023–1029, 2013.

[15] X. G. Yan and S. K. Spurgeon and C. Edwards, “Robust sliding mode observer with parameters estimation for a class of nonlinear time-delay systems,” 49th IEEE Conference on Decision and Control (CDC). pp. 96–101, 2010.

[16] Sebastian O. H. Madgwick, Andrew J. L. Harrison, Ravi Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm ,” 2011 IEEE International Conference on Rehabilitation Robotics. pp. 1–7, 2011.