Motion Control of a Differential Drive Mobile Robot Considering Voltage and Current Limits

Waleed Hilal
Department of Mechanical Engineering
McMaster University
Hamilton, Canada
hilalw@mcmaster.ca

Gary M. Bone
Department of Mechanical Engineering
McMaster University
Hamilton, Canada
gary@mcmaster.ca

S. Andrew Gadsden
Department of Mechanical Engineering
McMaster University
Hamilton, Canada
gadsden@mcmaster.ca

Mohammad AlShabi
Department of Mechanical and Nuclear Engineering
University of Sharjah
Sharjah, UAE
malshabi@sharjah.ac.ae

2023 Advances in Science and Engineering Technology International Conferences (ASET) | 978-1-6654-5474-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASET56582.2023.10180654

***Abstract*—The mobile robot trajectory tracking problem, whereby a controller is responsible for ensuring a robot follows a predetermined trajectory is investigated in this work. Several different algorithms are implemented as the controller for a differential drive wheeled robot in this study, and their performances are examined across different operating conditions using several performance measures. Specifically, we implement proportional, integral, and derivative controls, as well as sliding-mode control and model predictive control, and observe their control performance in ideal, tuned operating conditions, as well as in the face of varying levels of sensor noise, actuator saturation due to voltage and current constraints, or wheel slippage in one or both wheels. Background on the kinematic model of the differential drive wheeled robot as well as the implementation and tuning of the controllers are included in this work. Furthermore, we present a discussion of the advantages and limitations of each controller in the face of varying circumstances for the task of controlling a differential drive wheeled robot.**

Keywords—Differential drive robot, robot control, sliding mode control, PID control, model predictive control

I. Introduction

A differential drive wheeled robot (DDWR) is a mobile robot that achieves movement by two separately driven wheels located on either side of the robot's chassis. The way that a DDWR can change its direction is through varying the rotational velocity of either of its wheels, and therefore it does not require a separate or additional steering mechanism. For example, forward motion is achieved by maintaining both wheels at an angular velocity that is equal in magnitude and direction, while rotating in place requires having the angular velocities equal in magnitude but opposite direction of each other. Lastly, to turn or followed a curved trajectory, then one of the wheels must have a greater angular velocity than the other.

II. System Modelling

The most commonly used type of DDWR model is known as the unicycle model, which consists of two wheels and casters on the front and rear to support half of the robot's weight in total [1]. The geometry of such robots is shown in Fig. 1, which has a wheelbase length L, and is characterized by a linear velocity $\dot{y}_m$ and angular velocity $\dot{\varphi}$. It is assumed that the ground is level, and that the robot's path is either straight or moderately curved such that no sliding occurs in the Y_m direction.

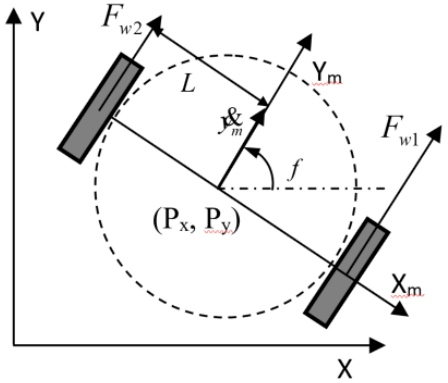

Fig. 1 Geometry of differential drive robot involved in the study.

The unicycle model provides the following relationship between the linear and angular velocities in the robot's moving coordinate frame (X_m, Y_m) and the fixed coordinate frame (X, Y) [2]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos\,\varphi & 0 \\ sin\,\varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{y}_m \\ \dot{\varphi} \end{bmatrix} \tag{1}$$

The linear velocities, v_{w1} and v_{w2}, of the right and left wheel centers, respectively, can then be determined as follows:

$$v_{w1} = \dot{y}_m + L\dot{\varphi} \tag{2}$$

$$v_{w2} = \dot{y}_m - L\dot{\varphi} \tag{3}$$

The net force and torque applied to the robot by its wheels are:

$$F_y = F_{w1} + F_{w2} \tag{4}$$

$$\tau_z = L(F_{w1} - F_{w2}) \tag{5}$$

By assuming that the robot can be modelled as a uniform cylinder of mass m_r its resultant linear and angular accelerations are:

$$\ddot{y}_m = \frac{F_y}{m_r} \tag{6}$$

$$\ddot{\phi} = \frac{\tau_z}{\frac{1}{2} m_r {r_c}^2} \tag{7}$$

A. Modelling of Rolling Wheels

The robot's wheels can be modelled according to different two cases or conditions: whether they are rolling or slipping. When the robot's wheels are currently rolling, the force applied by each of the wheels from their respective wheel centers are:

$$F_{w1} = \left(\frac{m_r r}{2R_g}\right) \ddot{\theta}_{m1} \tag{8}$$

$$F_{w2} = \left(\frac{m_r r}{2R_g}\right) \ddot{\theta}_{m2} \tag{9}$$

where R_g is the gear ratio, r is the radius of the wheel and $\ddot{\theta}_m$ is its motor's angular acceleration, which is given by:

$$\ddot{\theta}_{m1} = \left(\frac{K_t R_g^2}{J_m R_g^2 + \frac{1}{2} m_r r^2}\right) i_1 \tag{10}$$

$$\ddot{\theta}_{m2} = \left(\frac{K_t R_g^2}{J_m R_g^2 + \frac{1}{2} m_r r^2}\right) i_2 \tag{11}$$

where J_m is the motor inertia, and K_t is the motor torque constant. From (10) and (11), it is clear that the angular acceleration of each wheel is dependent on the current supplied by its motor.

If a wheel is currently rolling, it will begin to slip when the following condition is satisfied in each respective wheel:

$$\left|\left(\frac{K_t R_g}{J_m R_g^2 + \frac{1}{2} m_r r^2}\right) i_1\right| > \frac{g \mu_s}{4r} \tag{12}$$

$$\left|\left(\frac{K_t R_g}{J_m R_g^2 + \frac{1}{2} m_r r^2}\right) i_2\right| > \frac{g \mu_s}{4r} \tag{13}$$

where g is the gravitational constant and μ_s is the static coefficient of friction.

B. Modelling of Slipping Wheels

When the conditions in (12) or (13) are satisfied, the respective wheel will begin to slip, and the force it applies to the wheel center becomes [3]:

$$F_{w1} = \frac{1}{4} \mu_k m_r g \, sign\left(\frac{\dot{\theta}_{m1} r}{R_g} - v_{w1}\right) \tag{14}$$

$$F_{w2} = \frac{1}{4} \mu_k m_r g \, sign\left(\frac{\dot{\theta}_{m2} r}{R_g} - v_{w2}\right) \tag{15}$$

where μ_k is the coefficient of kinetic friction and $\mu_k < \mu_s$. Then, the motor's angular acceleration is given by:

$$\ddot{\theta}_{m1} = -\frac{F_{w1} r}{J_m R_g} + \left(\frac{K_t}{J_m}\right) i_1 \tag{16}$$

$$\ddot{\theta}_{m2} = -\frac{F_{w2} r}{J_m R_g} + \left(\frac{K_t}{J_m}\right) i_2 \tag{17}$$

If a wheel is currently slipping, then it will begin to roll when (12) and (13) are no longer true, and $v_w = \frac{\dot{\theta}_m r}{R_g}$ is within an accepted, arbitrarily threshold. The conditions to check in a numerical simulation are:

$$\left|\left(\frac{K_t R_g}{J_m R_g^2 + \frac{1}{2} m_r r^2}\right) i_1\right| \leq \frac{g \mu_s}{4r} \tag{18}$$

$$\left|\left(\frac{K_t R_g}{J_m R_g^2 + \frac{1}{2} m_r r^2}\right) i_2\right| \leq \frac{g \mu_s}{4r} \tag{19}$$

$$\left|v_{w1} - \frac{\dot{\theta}_{m1} r}{R_g}\right| < v_{tolerance} \tag{20}$$

$$\left|v_{w2} - \frac{\dot{\theta}_{m2} r}{R_g}\right| < v_{tolerance} \tag{21}$$

C. Electrical Constraints

Since the dynamic response of a motor's electric elements is much faster than its mechanical ones, we can neglect them. However, there still exists significant constraints on the current and voltage due to battery limits, as well as other constraints [4]. If the effect of inductance is assumed it be negligible, the electrical dynamic equation for each motor is simplified to:

$$V = Ri + K_b \dot{\theta}_m \tag{22}$$

where K_b is the back electromotive force (EMF) constant and R is the winding resistance. Therefore, if the battery's voltage limit is V_{bat}, then a constraint exists for each i which varies with $\dot{\theta}_m$ as follows:

$$i \leq \frac{V_{bat} - K_b \left|\dot{\theta}_m\right|}{R} \tag{23}$$

III. Proposed Objectives

The purpose of this research is to investigate the effectiveness of different control algorithms when applied to the task of controlling a DDWR, which is carried out through MATLAB simulations. Specifically, a simulated DDWR is tasked with maintaining a reference trajectory for its wheels' angular position, velocity and acceleration.

Measurements of the robot's linear and angular velocity are extracted through simulated sensors, which can then be translated into a linear velocity for each wheel using (2) and (3). The effective angular velocity of each wheel is then determined by:

$$\dot{\theta}_{m_{sensed}} = \frac{v_{w_{sensed}}}{R_g} \tag{24}$$

The derivative of the error $\dot{e}$ can then be computed, which comprises of the difference between the measured and reference angular velocity of each wheel. The integral of this results in the error e between the measured and reference angular position of the wheels. These two terms, $\dot{e}$ and e, constitute the feedback information that is passed through the control algorithm.

Upon receiving the feedback information, the controller outputs control signals u_1 and u_2 which represent the current inputs (i_1 and i_2) to each motor, with the goal of correcting the wheels' angular accelerations in order to maintain their course along the desired angular velocity trajectory. Before they are used as input for the plant model, the control signals are subject to constraints by (18) and (19) to account for slipping, as well as current restraints from the electrical elements by (23). This process is repeated until the end of the simulation.

There are many factors, however, that may affect the performance of a control algorithm in effectively ensuring that the desired trajectory is accurately followed. Among the first that a practitioner may encounter, is the need to thoroughly tune the parameters of the implemented control algorithm to ensure it is well-conditioned for the task at hand and satisfies the desired performance criteria [5]. Tuning of a control algorithm involves the use of a performance metric, or several, as a benchmark for how well the control algorithm performs and whether the desired criteria are met.

In this paper, one of the primary indicators of an algorithm's performance that is used is the steady-state error (e_{ss}), which can be computed using the measured states and the desired setpoints as follows:

$$e_{ss} = \theta_d - \theta_N \tag{25}$$

where θ_N is the measured state at time step N or the end of the simulation, and θ_d is the setpoint at that time step. In essence, the e_{ss} provides information on how close the robot's angular positions are to the desired setpoint at the end of the simulation. Several other performance metrics are also examined alongside the steady-state error, each of which give indication to different aspects of the control algorithm's performance. For instance, the mean absolute error (MAE) is another performance metric utilized which is an indicator of the average magnitude of the errors between the measured states and desired setpoint at each sampling period across the entire simulation:

$$MAE = \frac{\sum_{i=1}^{N} |\theta_{d,i} - \theta_i|}{N} \tag{26}$$

Similarly, the maximum absolute error (MAXAE) is another useful measure which provides information as to how much the desired state differs from the reference trajectory at the worst instant of the whole simulation.

$$MAXAE = max(|\theta_d - \theta_N|) \tag{27}$$

Finally, the maximum absolute value of the control signals (MAU) is another metric that is examined throughout the course of this study. This maximum absolute control signal value is useful in relaying information regarding the effort that is perceived to be required by the controller for each motor in order to correct the robot's trajectory:

$$MAU = max(|u|) \tag{28}$$

Upon successful tuning of a control algorithm using the above performance metrics as a benchmark, other factors that can impact a controller's effectiveness can be studied by using the tuned controller as a baseline. One such factor includes the presence of sensor noise. In feedback control systems, the controller relies on the measurements of the plant output from sensor measurements. Sensors, however, are often associated with noise that result in random variations of the sensor's output which are not related to the variations in the sensor's input. As a result of the random variations, the controller's inputs from the sensor outputs can be unreliable depending on the degree of the sensor's noise [6, 7]. To simulate noise and investigate its effects on the control algorithms studied, sensor measurements are augmented with uniformly distributed random numbers in the interval (-1, 1). Two scenarios, which are the presence of high or low sensor noise, are investigated. In the former, the interval of the uniformly distributed random numbers is scaled by a factor of ten percent of the measured signal's maximum magnitude, whereas in the latter, they are scaled by one percent of that maximum magnitude.

Another factor known to affect the performance of a controller are the physical limits of the plant's actuators, which are the robot's wheel motors in this case, resulting in actuator saturation. The nature of the electrical elements in the robot's motors from (23) limit the maximum possible current that can be supplied to each respective motor depending on their current angular velocity, and robot battery's voltage limit V_{bat}. As such, actuator saturation occurs when the control input exceeds the current limitations imposed by (23) and is therefore limited by the plant to the maximum allowed threshold. Therefore, any further increase in the control input will have no effect on the actuators' outputs and may even result in deterioration of the plant's dynamic performance [8]. The effects of actuator saturation are investigated by forcing the system to follow the desired trajectory in a shorter period

of time, meaning that more rapid acceleration of the robot's wheels is necessary, and consequently, greater control inputs.

Another final but critical factor that must be considered is the effects of wheel slipping on the controller's performance. As shown in the previous section of this study, the robot's wheels will begin to slip when (12) or (13) are true, resulting in a change in the force applied by the wheels as in (14) and (15). The limiting factors for slipping and rolling conditions are dependent on the static and kinetic friction coefficients, μ_s and μ_k respectively, between the robot wheels and the ground. Wheel slipping in DDWR is an issue that affects the ability to track a desired motion trajectory [9]. As such, another saturation constraint is imposed on the control signal based on (12) and (13) to prevent wheel slipping in the plant model. Wheel slippage is subsequently induced within the experimental simulations by the introduction of a modelling uncertainty, specifically in the form of a reduction of the coefficient of static friction on one of the robot's wheels compared to that of the plant model. Two scenarios are investigated in this paper, the first of which is the effects of slipping on just one of the wheels, and secondly, the effects of slipping in both wheels.

IV. Experimental Simulations

Prior to evaluating the dynamics and behaviour of the plant and model in study, it is necessary that the desired trajectory or setpoints for each of the motors be introduced. As such, the desired setpoints for each motor's angular position, angular velocity and angular acceleration are shown in Fig. 2. The angular position setpoint is based on a cubic polynomial trajectory with waypoints at desired angular positions of 0, 2 and 4 radians. The angular velocity setpoint is determined by taking the derivative of the angular position setpoint, which can subsequently be differentiated again to find the angular acceleration setpoint.

A sampling time of 0.001 seconds was chosen, and the simulations are run for a total of 5000 sampling periods, representing a total duration of 5 seconds. The values for the model's parameters used in the simulations are summarized in Table I.

TABLE I
SUMMARY OF MODEL PARAMETERS AND THEIR ASSOCIATED VALUES

Parameter (notation)	Value
Wheel radius (R)	0.02 m
Wheelbase length (L)	0.1 m
Robot mass (m_r)	0.5 kg
Static friction coefficient ($\boldsymbol{\mu_s}$)	0.7
Kinetic friction coefficient ($\boldsymbol{\mu_k}$)	0.239
1.2315e-1	1.5289e-1
Back EMF constant (K_b)	2 kg.m^2.s^{-2}.A^{-1}
Motor torque constant (K_t)	4 kg.m^2.s^{-2}.A^{-1}
Gear ratio (R_g)	4
Winding resistant (R_w)	5 Ω
Motor inertia (J_m)	2 kg.m^2
Battery voltage limit (V_{bat})	12 V

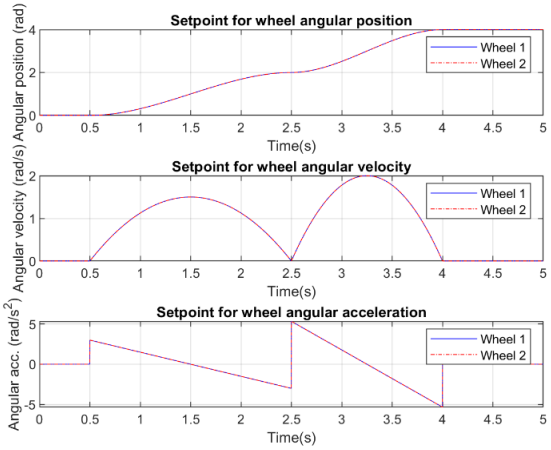

Fig. 2 Desired setpoint for the angular position, velocity and acceleration of each of the robot's wheels.

A. PID Control

The proportional-integral-derivative (PID) controller is one of the most popular and commonly used control algorithms due to its relatively effective performance compared to its simplicity [10]. As suggested in the name, a PID controller consists of three gains: the proportional, integral and derivative gains, which are tuned according to desired criteria to result in a satisfactory response.

The PID controller is based on the principle of minimizing the error between the desired setpoint and the measured plant output. Thus, a feedback loop is required to relay the measured output and respective error back to the controller to serve as its next input. A PID controller can be formulated as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{dt} \qquad (29)$$

where K_p, K_i, and K_d are the proportional, integral and derivative gains, respectively. It is evident from this equation that the derivative and integral of the errors must be determined for the implementation of the controller. The controller's output, *u(t)* is the control signal to a robot's wheel motor, prior to being subject to the necessary system constraints. Since a DDWR consists of two motor-driven wheels, each wheel requires its own PID controller.

Upon implementing the PID controller to track the desired trajectory profile of each wheel, it was necessary to tune the P, I and D gains. The three gains were tuned with the consideration of the performance metrics previously discussed, which include the e_{ss}, MAE, MAXAE and MAU, and with the objective in minimizing each of the four metrics. Of course, it is not possible to minimize all four metrics at once, therefore it was desired to achieve as balanced as a result as possible while considering whether certain decreases of each metric outweighed the increase of another . Ultimately, the tuned PID controller was comprised of K_p of 50, a K_i of 0.001 and a K_d of 12.

B. Sliding Mode Control

Sliding mode control (SMC) is a non-linear control algorithm which is a form of variable structure control [11, 12]. The premise of SMC is that it alters the dynamics of a system by applying a discontinuous control signal which forces the system's state to first reach a surface in the state space, known as the sliding surface, and then slide along this surface to reach the desired state [13, 14]. The sliding surface is defined by the equation $s(x;t) = 0$ as follows [15]:

$$s(x;t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} \tag{30}$$

where $\tilde{x}$ is simply the tracking error. When on the surface, the system's behaviour is called the 'sliding regime', and can be described by:

$$\left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} = 0 \tag{31}$$

During the sliding mode, the system dynamics can be written simply as:

$$\dot{s} = 0 \tag{32}$$

and when n=2, this equation can be solved for *u* by:

$$s = \frac{d}{dt}\tilde{x} + \lambda\tilde{x} \tag{33}$$

$$\dot{s} = \frac{d^2}{dt^2}\tilde{x} + \lambda\frac{d}{dt}\tilde{x} \tag{34}$$

$$\dot{s} = \ddot{x} - \ddot{x}_d + \lambda\frac{d}{dt}\tilde{x} \tag{35}$$

$$\dot{s} = \hat{f}(x) + \hat{b}(x)u - \ddot{x}_d + \lambda\frac{d}{dt}\tilde{x} = 0 \tag{36}$$

$$u_{eq} = \frac{\ddot{x}_d - \lambda\frac{d}{dt}\tilde{x} - \hat{f}(x)}{\hat{b}(x)} \tag{37}$$

where $\hat{f}(x)$ is the model of $f(x)$, $\hat{b}(x)$ is the model of $b(x)$ and u_{eq} is known as the equivalent control signal. However, this signal works under the assumption of a perfect plant model and no disturbance. Thus, another *u* term must be added to account for this since both of those assumptions do not hold in practice, and thus our resultant is:

$$u = u_{eq} - k\hat{b}^{-1}sgn(s) \tag{38}$$

The parameter *k* can be derived from uncertainty bounds on $f(x)$ and $b(x)$ if they are known. For instance, $\hat{b}$ is given by:

$$\hat{b} = \sqrt{b_{min}b_{max}} \tag{39}$$

where b_{min} and b_{max} represent the uncertainty bounds on $b(x)$. Then:

$$\beta = \sqrt{\frac{b_{bmax}}{b_{min}}} \tag{40}$$

Furthermore, given that η is a positive constant, the sliding conditions are satisfied when:

$$k \geq \beta(F + \eta) + (\beta - 1)\hat{b}|u_{eq}| \tag{41}$$

Upon implementation of SMC, it was determined through manual tuning that the most optimal parameters are $\lambda = 25$ and $\eta = 1$, and the uncertainty bounds were calculated as follows:

$$b_{min} = \frac{K_{t_{min}}{R_g}^2}{J_{m_{max}}{R_g}^2 + \frac{1}{2}m_r R^2} \tag{42}$$

$$b_{max} = \frac{K_{t_{max}}{R_g}^2}{J_{m_{min}}{R_g}^2 + \frac{1}{2}m_r R^2} \tag{43}$$

where the uncertainty bounds on K_t were from [2,6], and for J_m they are [1,3]. Due to the nature of the plant model, f was set equal to 0.

An anti-windup scheme is implemented into SMC to reduce the effects of chattering, which involves replacing the *sgn* function in (38) with $sat(s/\Phi)$ where Φ defines the thickness of the boundary layer, and:

$$sat(a) = \begin{cases} sgn(a) & if\ |a| > 1 \\ a & if\ |a| \leq 1 \end{cases} \tag{44}$$

The value of Φ is tuned such that it is maintained to the lowest possible value that will significantly reduce most of the chatter. In this study, this value was determined to be $\Phi = 0.1$.

C. Model Predictive Control

The idea of model predictive control (MPC) is to base the control signal input on the long-range prediction of the output. At each sample, the plant output is predicted based on a plant model, the past output values, and past and projected values of the control signal input [16] [17]. Of the main advantages of MPC is that it allows for the optimization of the current time sample, while also keeping into account future time samples [18]. This is carried out through the optimization of a finite time-horizon, but only enacting the current time sample before proceeding to optimize again [17]. Thus, MPC can anticipate future events, and act proactively by adjusting its course of action accordingly.

The implementation of MPC consists of tuning a prediction horizon N_p, which signifies the number of time periods that that controller will lookahead for. The objective function of MPC involves minimizing the norm of the reference trajectory and the predicted measurement output in the N_p lookahead horizon. It was determined that choice of lookahead horizon negligibly impacted the performance of the

TABLE II
PERFORMANCE OF CONTROLLERS IN IDEAL CASE

	Motor	e_{ss}	MAE	MAXAE	MAU
PID	1	4.0787e-5	8.3504e-4	2.7216e-3	1.6688e-1
	2	4.0787e-5	8.3504e-4	2.7216e-3	1.6688e-1
SMC	1	1.5136e-6	2.3856e-3	6.9581e-2	4.8267
	2	1.5136e-6	2.3856e-3	6.9581e-2	4.8267
MPC	1	1.3908e-8	4.9601e-2	1.2315e-1	1.5289e-1
	2	1.3908e-8	4.9601e-2	1.2315e-1	1.5289e-1

controller, which was assumed to be based on the simplicity of the desired reference trajectory. Thus, it was arbitrarily set at 20.

V. RESULTS AND DISCUSSION

A. Ideal Case

This scenario investigates the performance of the controllers under ideal, tuned conditions, and without the presence of any effects from noise, actuator saturation, or wheel slippage. Each controller's performance in this scenario can be seen in Table II.

Through tuning of the PID controller, it was observed that an increase in the proportional gain generally corresponded to a decrease in the steady-state error and increase in the MAE and MAXAE. Increasing the derivative gain results in a compensation for the increase in MAE and MAXAE at the cost of a negligible increase in steady-state error and a non-negligible increase in the MAU. Surprisingly, the integral gain did not result in significantly noticeable results, but only in negligible decreases of the steady-state error in return for a greater MAE and MAXAE.

However, the steady-state error of SMC is demonstrated to be superior to the PID controller by an order of 1 magnitude. However, the PID controller is still considered more favorable in terms of the MAE, MAXAE and MAU. Interestingly, the MAU is still at a level significantly higher than the PID controller, and even alludes to the presence of actuator saturation.

When compared to the PID and SMC, MPC's performance in terms of steady-state error is much more favourable, representing an improvement close to an order of 3 and 2 magnitudes over each respective algorithm. Furthermore, the MAU of MPC is slightly lower than that of the PID controller, by approximately 8 percent. As for the MAE and MAXAE, however, they are relatively better than that of SMC's, but are still considered inferior to the PID controller.

B. Effects of Actuator Saturation

In this scenario, the subject of interest is actuator saturation with the different controllers and how they perform in the presence of such conditions. Actuator saturation is induced in the simulations by reducing the time in which the plant's states much track the desired trajectory to less than a quarter of the original time. The idea behind this is that the robot's actuators, the wheel motors in this case, must accelerate at a greater rate in a shorter amount of time, and thus will be more likely to face the physical limits of the system. The results and performance of each controller in the presence of actuator saturation is presented in Table III.

When compared to the results of the ideal case just discussed in the previous subsection, several observations can be made about the PID controller's performance and the effects of actuator saturation. The most obvious observation from Table III is that the MAU is 42.919, which is more than 250 times greater than that of the PID controller under the baseline operating conditions. Although the steady-state error is improved by tenfold in this scenario, it can be seen that the rise in MAE and MAXAE compared to previous case is rather significant, both increasing by approximately 100 times. Furthermore, it is immediately clear to see that regardless of the anti-windup scheme, SMC completely fails to perform when actuator saturation is encountered. In fact, with SMC controller, the plant's behaviour is completely unbounded, acting similarly to the model's open loop response.

Most noticeably, compared to the previous PID and SMC methods under the same conditions, it can be seen that MPC can handle the accelerated trajectory with ease without actually encountering any saturation constraints. This is apparent from the MAU of 3.6881, compared to the maximum values attained of 42.919 by SMC and PID, which is the maximum MAU achievable under complete current saturation. As for the steady-state error, it is an increase of approximately 4 orders of magnitude, however, it is a worthwhile compromise when considering that MPC is less likely to face actuator saturation. Similarly, the MAE's increase of approximately 20% compared to the baseline MPC is considered negligible in the grand scheme of things, when considering what is being asked of the controller. The same justification can be provided in the case of the MAXAE,

TABLE IV
PERFORMANCE OF CONTROLLERS IN ACTUATOR SATURATION CASE

	Motor	e_{ss}	MAE	MAXAE	MAU
PID	1	5.0383e-6	5.0428e-2	6.7845e-1	42.91
	2	5.0383e-6	5.0428e-2	6.7845e-1	42.91
SMC	1	350.94	140.32	351.04	42.91
	2	350.94	140.32	351.04	42.91
MPC	1	1.0094e-2	5.8649e-2	6.9947e-1	3.6881
	2	1.0094e-2	5.8649e-2	6.9947e-1	3.6881

TABLE III
PERFORMANCE OF CONTROLLERS IN LOW NOISE CASE

	Motor	e_{ss}	MAE	MAXAE	MAU
PID	1	5.9047e-5	8.3624e-4	2.7526e-3	5.4739e-1
	2	5.9047e-5	8.3624e-4	2.7526e-3	5.4739e-1
SMC	1	9.9175e-6	6.9514e-4	1.8198e-3	7.6479
	2	9.9175e-6	6.9514e-4	1.8198e-3	7.6479
MPC	1	1.1553e-4	4.9599e-2	1.2313e-1	1.6157e-1
	2	1.1553e-4	4.9599e-2	1.2313e-1	1.6157e-1

which is quite comparable to the of the PID controller, and an increase of less than 3 percent.

C. Effects of Sensor Noise

The effects of sensor noise on the PID controller are investigated by simulating noise to the extracted measurements, which involves adding uniformly distributed numbers based on the maximum magnitude of the measurements. For the case of high noise, the uniform distribution was scaled down to 10 percent of the maximum measurements, while the low noise case was scaled down to 1 percent of the maximum measurements. The performance of the controllers in the low and high noise cases are summarized in Table IV and Table V, respectively.

The results of the low noise case from Table IV show that the PID controller does not suffer significantly at all in the presence of low noise conditions. The MAE and the MAXAE, in fact, are almost identical to that of the baseline case, with the steady-state error increasing by 33 percent to a still negligible level. The most noticeable effect of low noise is an increase of the MAU by slightly more than 3 times than the baseline. It is interesting to observe that despite facing an increase in steady-state error by almost 3 orders of magnitude, SMC in low noise conditions flaunts an MAE and MAXAE that is much more favourable than the baseline SMC with anti-windup from the previous subsection. This improvement in MAE and MAXAE compared to the baseline SMC is a decrease of 70 and 97 percent respectively. However, the MAU continues to further increase by 58 percent, posing an even greater risk that actuator saturation may occur. Inspecting the steady-state error, it can be said that MPC performs quite well in the face of low noise, although, not quite as much so as the PID controller and SMC. This is consistent across the MAE and the MAXAE too, however, not with the MAU, which demonstrates to be approximately 4 times less than the PID controller, and 40 times less than SMC under similar conditions.

The results of the controllers in the face of high sensor noise are shown in Table V. In this scenario, it can be seen that the steady-state error of the PID controller suffered a more drastic increase than the low noise case, by almost 20 times when compared to the baseline PID model from the ideal case. The MAE also increased significantly, by more than 10 times, while the MAXAE remained relatively unchanged. The MAU, however, is 3.9955 in the case of high noise, compared to 0.54739 in the low noise case. This MAU represents an increase by 24 times the baseline case, which as expected, results in some effects of actuator saturation.

It can also be seen from this table that in the presence of high amount of sensor noise, SMC's performance begins to degrade quite considerably. The steady-state error is greater by an order of 5 magnitudes, while the MAE is just under 3 times greater than in the baseline SMC's case. The most alarming observation, however, is the significant jump in the MAU, which is increased to be 6 times that of the baseline case, indicating to actuator saturation.

TABLE V
PERFORMANCE OF CONTROLLERS IN HIGH NOISE CASE

	Motor	e_{ss}	MAE	MAXAE	MAU
PID	1	7.1062e-4	1.0026e-3	2.7314e-3	3.9955
	2	7.1062e-4	1.0026e-3	2.7314e-3	3.9955
SMC	1	5.7122e-3	6.3636e-3	1.5514e-2	30.154
	2	5.7122e-3	6.3636e-3	1.5514e-2	30.154
MPC	1	3.7496e-4	4.9667e-2	1.2459e-1	2.5608e-1
	2	3.7496e-4	4.9667e-2	1.2459e-1	2.5608e-1

In the case of high noise, the results demonstrate the MPC's robustness in such circumstances. Specifically, the steady-state error only increased by an order of two magnitudes, exceeding the performance of both the PID controller and SMC. Furthermore, it can be seen that the MAE and MAXAE are relatively unaffected compared to the baseline MPC model, while the MAU increases by almost 100 percent to a still very impressive value of 0.25608.

D. Effects of Wheel Slip

The performance of the controllers under the effect of wheel slippage are studied in this section. Two scenarios are investigated: one wheel slipping and both wheels slipping. Slipping is induced in a wheel by reducing the coefficient of static of friction for that specific wheel or can be induced in the two wheels by reducing it in both.

The value for the coefficient of static friction at which wheel slip occurs is experimentally determined to be 0.0027 in the case of the PID controller, which is drastically lower than the model's coefficient of friction of 0.7. As for the SMC, the value of the coefficient of static friction at which slipping was determined to begin is 0.035, which is greater than that of the PID controller by more than 10 times. This value is recorded to be 0.00241 for MPC, which is considered to be the most optimal case of the three controllers investigated in this paper. Comparing this value to that of the PID's and SMC's, 0.0027 and 0.035, respectively, the coefficient of static friction required to induce wheel slipping with MPC can be considered 11 and 93 percent better, respectively. Finally, the performance of the controllers in the case of one wheel or both wheels slipping is presented in Table VI and Table VII, respectively.

In the case of one wheel slipping, it can be seen from Table VI that the effects of wheel slip have proven to be the most detrimental to the performance of the PID controller when compared to the effects of noise or actuator saturation. As expected, due to slipping, the control performance of 'wheel 1' is significantly worse than that of 'wheel 2'. However, the MAUs of both were equal at 42.919, which indicates that actuator saturation likely occurred. As a result of the slipping induced in one wheel, the PID controller was not able to effectively track the desired trajectory and instead, the robot veered of the desired course. When slipping is induced in one of the wheels, the performance results of SMC controller show that the SMC completely fails to maintain effective control of the plant as the steady-state error increases drastically, even

TABLE VI
PERFORMANCE OF CONTROLLERS WITH ONE WHEEL SLIPPING

	Motor	e_{ss}	MAE	MAXAE	MAU
PID	1	12.763	3.0119	12.771	42.919
	2	9.5623e-1	4.5868e-1	1.0886	42.919
SMC	1	228.01	37.353	228.18	42.919
	2	43.789	13.567	43.794	42.919
MPC	1	9.4537e-2	6.6598e-2	1.2315e-1	5.185e-1
	2	1.4039e-6	5.018e-2	1.2315e-1	1.4778e-1

TABLE VII
PERFORMANCE OF CONTROLLERS WITH BOTH WHEELS SLIPPING

	Motor	e_{ss}	MAE	MAXAE	MAU
PID	1	17580	2921	17594	42.919
	2	18309	3041	18324	42.919
SMC	1	1.1031e-9	2.6959e-3	4.1541e-2	26.62
	2	1.1031e-9	2.6959e-3	4.1541e-2	26.62
MPC	1	6.9e-2	6.1511e-2	1.2315e-1	2.6605e-1
	2	6.9e-2	6.1511e-2	1.2315e-1	2.6605e-1

more so than the PID controller under similar conditions. The presence of actuator saturation is also confirmed by the MAU of SMC, reaching 42.919 in both wheels.

A pleasantly surprising observation can be made from Table VI, which is that MPC performs surprisingly well despite the slipping of one of the wheels. The steady-state error, MAE and MAXAE of the unaffected motor is almost identical to that of the baseline case, with the only difference being an almost fourfold increase in the MAU. As for the wheel that is slipping, MPC still manages to effectively maintain control and ensure that the trajectory is more or less tracked. It can be seen that 'wheel 1' has a steady-state error which is more than 100 times better than that of the PID controller and is not even comparable to SMC which completely fails under the same circumstances. These findings are also consistent across the MAE, MAXAE and MAU values for MPC when compared to the other methods, further corroborating the effectiveness and robustness of MPC.

In the case of both wheels slipping, it is clear from the results in Table VII that the PID completely fails to be able to effectively control the robot to follow the desired trajectory. It can be seen that the MAU reached values of 42.919 in both wheels, indicating that actuator saturation was effectively present during the simulation. Furthermore, the error values across all three measures have increasing by significant orders of magnitude. It is also evident from these results that SMC is capable of maintaining effective control of the robot when both wheels are slipping. This is apparent from the steady-state error, MAE and MAXAE values. Surprisingly, SMC seems to perform even better than the baseline SMC case when both wheels are slipping. The improvement in the steady-state error is nearly 10 times that of the baseline, whilst the MAXAE is nearly half that of the baseline's as well. This comes at a cost of a negligible decrease in performance in MAE, but a significant increase in MAU which indicates to actuator saturation occurring.

The effects of two wheels slipping on MPC's performance further demonstrate that the method is suitable for applications involving such conditions. The results shown in Table VII are far more superior than the PID controller, which fails completely at this task, but slightly less favourable than SMC under the same scenario. However, the MAU is much more desirable with the MPC at a value of 0.26605, compared to 26.62 by SMC, which was ultimately associated with actuator saturation unlike in the present controller's case.

VI. CONCLUSION

This research investigates three different control algorithms, PID, SMC and MPC, for the task of controlling a DDWR and examines the effects of various factors on the performance of the controllers when applied to this task. Specifically, the effects of actuator saturation by nature of current and voltage saturation, sensor noise, wheel slip and modelling uncertainty are all studied. Through experimental simulations carried out using MATLAB, a discussion of the results of each controller was provided, along with a detailed comparison of the controllers in each scenario through performance measures such as the steady-state error, MAE, MAXAE and the MAU.

In summary, it has been found that the MPC is considered to be the most well-rounded and effective controller overall, especially in ill-conditioned circumstances or harsh environments. Actuator saturation is rarely encountered with the MPC, as opposed to the PID and SMC controllers under similar circumstances. The SMC, in fact, completely fails under such conditions and displays horrendous performance across all measures. Furthermore, the impacts of the presence of noise have been demonstrated to be negligible in the MPC, which is also the case for the PID and SMC. However, one of the consistent findings is that even when MPC's performance was comparable to another algorithm, it almost always is superior in terms of MAU. Most importantly, it has been proven that MPC manages to maintain course and track the desired trajectory almost perfectly regardless of the effects of wheel slip, maintaining desirable performance across all performance measures. The value of the coefficient of friction that induced slipping in the control algorithms was also determined to be lowest and most desirable with MPC.

Despite its remarkable performance, the MPC can be attributed to several downsides and limitations. Of the most noticeable limitations of MPC is its execution time, which is several orders of magnitude greater than that of the PID controller and SMC. This can be associated with the greater computational overhead associated with the task of dynamic programming involved in its operation. Furthermore, it can be argued that the complexity of implementation of MPC is significantly greater than PID controls and SMC.

References

[1] F. Dušek, D. Honc and P. Rozsíval, "Mathematical model of differential steered mobile robot," in *18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 2011.

[2] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems,* vol. 55, no. 6, pp. 460-469, 2007.

[3] I. F. Okuyama, M. R. Maximo and R. J. Afonso, "Minimum-Time Trajectory Planning for a Differential Drive Mobile Robot Considering Non-slipping Constraints," *Journal of Control, Automation and Electrical Systems,* vol. 32, pp. 120-131, 2021.

[4] K. R. Sharma, D. Honc and F. Dušek, "Model Predictive Control of Trajectory Tracking of Differential Steered Mobile Robot," in *Advances in Intelligent Systems and Computing* , Springer, 2015, pp. 85-95.

[5] J. Goodman, W. Hilal, S. A. Gadsden and C. D. Eggleton, "Adaptive SVSF-KF estimation strategies based on the normalized innovation square metric and IMM strategy," *Results in Engineering,* vol. 16, p. 100785, 2022.

[6] W. Hilal, S. A. Gadsden, S. Wilkerson and M. Al-Shabi, "A square-root formulation of the sliding innovation filter for target tracking," in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXXI*, Orlando, FL, 2022.

[7] W. Hilal, S. A. Gadsden, S. Wilkerson and M. Al-Shabi, "Combined particle and smooth innovation filtering for nonlinear estimation," in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXXI*, Orlando, FL, 2022.

[8] Q. Liu and Q. Cong, "Kinematic and dynamic control model of wheeled mobile robot under internet of things and neural network," *The Journal of Supercomputing,* vol. 87, pp. 8678-8707, 2022.

[9] S. Konduri, E. O. C. Torres and P. R. Pagilla, "Dynamics and Control of a Differential Drive Robot With Wheel Slip: Application to Coordination of Multiple Robots," *Journal of Dynamic Systems, Measurement, and Control,* vol. 139, no. 1, p. 014505, 2017.

[10] R. P. Borase, D. K. Maghade, S. Y. Sondkar and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control,* vol. 9, pp. 818-827, 2021.

[11] V. Utkin, "Variable structure systems with sliding modes," *IEEE Transaction on Automatic Control,* vol. 22, no. 2, pp. 212-222, 1977.

[12] A. Filipescu, V. Minzu, B. Dumitrascu, A. Filipescu and E. Minca, "Trajectory-tracking and discrete-time sliding-mode control of wheeled mobile robots," in *IEEE International Conference on Information and Automation*, Shenzhen, China, 2011.

[13] K. D. Young, V. I. Utkin and U. Ozguner, "A control engineer's guide to sliding mode control," *IEEE Transactions on Control Systems Technology,* vol. 7, no. 3, pp. 328-342, 1999.

[14] A. Dòria-Cerezo, D. Biel, J. M. Olm and V. Repecho, "Sliding mode control of a differential-drive mobile robot following a path," in *18th European Control Conference (ECC)*, Naples, Italy, 2019.

[15] G. M. Bone, *Advanced Mechanical Controls,* 2022, pp. 140-157.

[16] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems Magazine,* vol. 20, no. 3, pp. 38-52, 2000.

[17] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized predictive control—Part I. The basic algorithm," *Automatica,* vol. 23, no. 2, pp. 137-148, 1987.

[18] W. F. Lages and J. A. V. Alves, "Real-time control of a mobile robot using linearized model predictive control," *IFAC Proceedings Volume,* vol. 39, no. 16, pp. 968-973, 2006.