Unmanned Systems, Vol. 0, No. 0 (2023) 1–14

On Active Disturbance Rejection Control for Unmanned Tracked Ground Vehicles with Non-smooth Disturbances

Mingliang Liu[a] , Yangmengfei Xu[a], Xuteng Lin[a], Ying Tan[a], Ye Pu[a], Wen Li[a], Denny Oetomo[a]

[a] *School of Electrical, Mechanical and Infrastructure Engineering, The University of Melbourne, VIC 3010, Australia*[*]

This paper proposes robust controllers for a class of unmanned tracked ground vehicles (UTGVs), which are built to autonomously clean carryback or spillage from the conveyor belts used in the mining industry. The UTGV, a nonholonomic system in its nature, needs to follow a given path in a harsh environment with large uncertainties due to the time-varying mass and inertia when the UTGV loads and unloads as well as unknown frictions and flatness of the ground. Moreover, the input constraints coming from motors do exist. It is usually hard to design robust controllers for such complex systems. By utilizing the available autonomous driving system, which is designed to be compatible with the existing remote motion controller in unmanned systems to generate autonomous ability, this paper uses the off-the-shelf motion planner to calculate desired linear and angular velocities based on the given path and sensor perceptions. Consequently, the control design can be simplified as two decoupled linear time-invariant scalar dynamic systems with uncertainties, making the active disturbance rejection controller (ADRC) applicable. By carefully designing the parameters of ADRC with the help of an extended state observer (ESO), it is shown that the proposed ADRC and ESO can achieve good tracking performance in the presence of input saturation and can handle non-smooth disturbances. The proposed simulation results and experimental results support the theoretical findings.

Keywords: Unmanned tracked ground vehicles; Active disturbance rejection controller; Extended state observer; Input saturation.

1. Introduction

Tracked ground vehicles are widely used in industrial applications, such as agriculture, nuclear sites [1], and the mining industry [2] because the large contact area of the continuous tracks with the ground helps increase traction and helps prevent the vehicle from sinking into soft terrain. The developments in robotic systems enabled tracked ground vehicles to complete complex tasks in an unmanned manner. An example of such an unmanned tracked ground vehicle (UTGV) is a cleaning robot that removes spillage from conveyor systems efficiently, as shown in Figure 1.

The majority of the UTGV uses a remote controller to finish the desired task. That is, the remote controller is manipulated by an operator, generating **reference velocities** for the low-level controller or the motion controller in the embedded system. In this remote control setting, the experienced operator has the ability to handle various uncertainties during the operations of UGTVs. If a fully autonomous UTGV is developed, designing its control algorithms becomes challenging due to the following factors. First of all, the dynamics of the UTGVs are so-called nonholonomic, whose state depends on the path taken in order to achieve it. It is well-known that non-smooth control laws are needed to stabilize nonholonomic systems, as indicated in [3]. Secondly, the UTGV may have time-varying mass and inertia when it loads and unloads, as shown in the example of the cleaning robot in Figure 1. Thirdly, the friction of the ground could change from place to place as the UTGVs normally work on unconstructed terrain. Different materials have different friction coefficients. Last but not least, the UTGVs also exhibit complex behaviours due to the nonlinear undercarriage and transmission design. This work will focus on developing control algorithms to handle these uncertainties for nonholonomic systems by using the existing autonomous driving system (ADS).

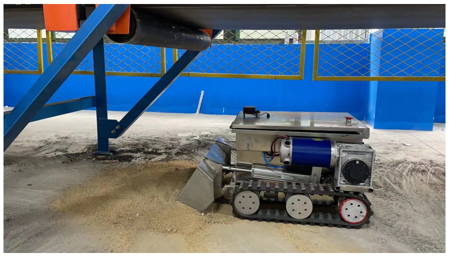


Fig. 1. An UTGV designed for cleaning purposes in mining.

---

[*] E-mail: {mingliangl, yangmengfeix}@student.unimelb.edu.au, {peter.lin, yingt, ye.pu, wen.li3, doetomo}@unimelb.edu.au

Different ways to equip the existing unmanned systems with autonomy ability have been proposed in ADS [4]. In particular, to accommodate the existing motion controller setting in the majority of UTGV, one of the typical structures of the ADS is shown in Figure 2. The sensors, such as light detection and ranging (LiDAR) sensors or cameras, are equipped on UTGV to provide localization. The role of the path planner is to generate a feasible or optimal global path with respect to the cost, such as minimal distance, connecting the robot's current localization to the final destination. Once the global path is obtained, the motion planner will calculate desired reference velocities based on the given path and sensor perceptions. The motion controller will then track these reference velocities and send the commands (*i.e.* torques) to the actuators. In such a way, the ADS can be added to the existing UTGV by only replacing the existing remote controller and the operator.

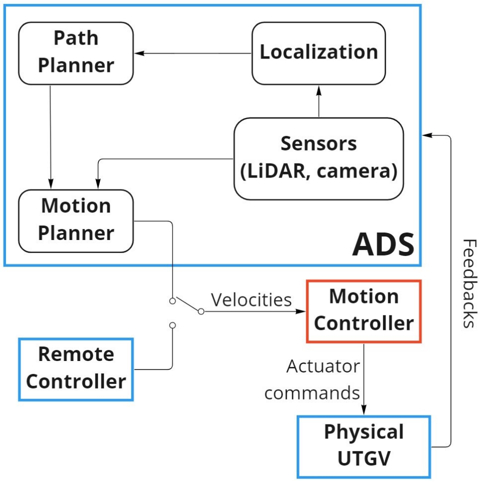


Fig. 2. Incorporating autonomous driving system (ADS) with the remote control setting.

If a motion controller is used, with the help of the off-the-shelf motion planner coming from ADS, the control design for autonomous UTGV can be simplified as two independent low-level loops, which are designed to track the given reference linear velocity and angular velocity, respectively. Each loop will decide a control law to track the desired reference velocity of UTGV from the motion planner based on a scalar linear time-invariant (LTI) dynamic system affected by unknown lumped modelling uncertainties. Using this simplification, the widely used active disturbance rejection control (ADRC) algorithm is directly applicable, while the nonholonomic system with lumped uncertainties is not in standard ADRC form (see [5] for more detailed discussion).

ADRC algorithms have been used in many engineering applications to cancel the effect of the unknown time-varying modelling uncertainties by estimating these uncertainties as external state information, see [5–9] and references therein. It always works together with the extended state observer (ESO), which aims to estimate the unknown time-varying uncertainties, as in [10–13].

In the existing ESO, as a standing assumption, it is always required that the uncertainties are smooth. i.e., their time derivatives exist and are uniformly bounded, see, for example, [10–13] and references therein. However, due to the existence of rocks or bumps on unconstructed terrain, jumps in terms of friction exist. These jumps, when treated as uncertainties, do not have well-defined time derivatives. Hence, the existing ESO cannot be directly applied. In this work, by using upper Dini derivatives, we can relax the smooth requirement uncertainties so that the ADRC and ESO can be adapted to our application to handle non-smooth uncertainties. Moreover, the proposed method can handle unbounded disturbances such as ramp signals.

In this work, input constraints, which always exist in engineering applications due to the limited capacity of actuators, are also considered. It is noted that the anti-windup technique has been widely used in industrial applications as an **additional** block to drive the input of the actuator away from the saturation bound when the saturation happens. For an LTI system, linear matrix inequalities (LMIs) have been used to find an appropriate feedback gain to avoid saturation from happening in [14]. By introducing the concept of compatible disturbance with input saturation bound, the analysis of linear LTI systems with input saturation becomes much simpler as it does not require an extra anti-windup block or solving LMIs.

By utilizing the off-the-shelf ADS and dual-loop design, it is possible to coordinate the parameters of ESO and ADRC for scalar linear time-invariant (LTI)to avoid possible input saturation under some weak conditions. More precisely, our main results show that if there is no input saturation, it is always possible to tune the parameters of the ADRC and ESO so that the solutions of the tracking errors are uniformly ultimately bounded, which can be made **arbitrarily small** (see Proposition 3.2). When input saturation exists, if the uncertainties are compatible with the saturation bound, it is always possible to tune the parameters of ADRC and ESO so that the solutions of the tracking errors can achieve some ultimate bound (see Proposition 3.5). The proposed technique is also applicable to various forms of vehicle chassis like skid drive and differential drive, as well as applications when human operators are driving a vehicle using velocity commands.

The effectiveness of the proposed algorithms is validated via both simulations and experiments. MATLAB simulations show how the selection of parameters of ADRC and ESO affects the overall tracking performance. A Unity-ROS simulation using a skid drive with six wheels (three wheels on each side) is also presented to compare the path tracking performance between a path follower with a unicycle model [15] and the motion planner using a simplified model, showing the effectiveness of the proposed method. Other than demonstrating how robust the proposed algorithms are in different ground situations, our experimental results also compare with the standard proportional-integral (PI) controllers, which are frequently used in tracking reference velocities, showing good performance in terms of tracking a given path.

2. Problem Formulation

Let $\mathbb{R}$ and $\mathbb{R}^n$ denote the set of real numbers and an n-dimensional Euclidean space respectively. For an n-dimensional vector $x \in \mathbb{R}^n$, its Euclidean norm is defined as $|x| = \sqrt{x^T x}$, where x^T denotes the matrix transpose. The notion of $diag(\cdot)$ represents a diagonal matrix with an appropriate dimension. For any matrix $A \in \mathbb{R}^{n\times m}$, its norm is the induced Euclidean norm.

The upper Dini derivatives are used to represent a class of generalizations of the derivative, that is, for a semi-continuous function $f(t)$ [a]:

$$D^+ f(t) = \limsup_{h\to 0^+} \frac{f(t+h)-f(t)}{h}.$$

A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be class $\mathcal{K}$ if its initial condition $\alpha(0) = 0$ and strictly increasing. The function α is a class $\mathcal{K}_\infty$ if it is class $\mathcal{K}$ and $\lim_{a\to\infty} \alpha(a) \to \infty$. A continuous function $\beta : [0,\ a) \times [0,\infty) \to [0,\infty)$ is said to belong to class $\mathcal{KL}$ if, for each fixed s, the mapping $\beta_a(r,s)$ belongs to class $\mathcal{K}$ with respect to r and for each fixed r, the mapping $\beta(r,s)$ is decreasing with respect to s and $\beta(r,s) \to 0$ as $s \to \infty$ [16].

The set containing all absolute-value integrable signals is denoted as $\mathcal{L}_1$. For any $w(\cdot) \in \mathcal{L}_1$, its norm is defined as $\|w\|_1 := \int_0^\infty |w(t)|\,dt$. The set containing all essentially bounded measurable functions is denoted as $\mathcal{L}_\infty$. For any $w(\cdot) \in \mathcal{L}_\infty$, its norm is defined as $\|w\|_\infty := \text{esssup}_{t\geq 0}|w(t)|$.

2.1. *Problem Setting*

This paper considers a UTGV to finish a given task, such as cleaning carryback or spillage underneath the conveyor belt, as shown in Figure 1. To navigate the robot in such an environment, two frames are defined as shown in Figure 3. The local frame or body frame is denoted as $\{R\}$. Assume that the v and ω are the linear and angular velocities of UTGV in this local frame respectively. Here x_r and y_r are the robot position in the x-y plane of the global frame $\{G\}$ and θ_r is the heading of the UTGV in the global frame $\{G\}$.

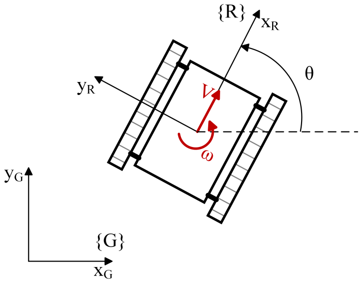


Fig. 3. An UTGV in the body frame.

Many models have been used to characterize UTGV dynamics and kinematics. For example, the different variants of nonlinear unicycle models [15, 17] have been proposed. The basic form of the nonlinear unicycle model is

$$\begin{aligned}\dot{x}_r &= v\cos(\theta_r)\\ \dot{y}_r &= v\sin(\theta_r)\\ \dot{\theta}_r &= \omega,\end{aligned} \tag{1}$$

with (x_r, y_r, θ_r) defined before. Here (v,ω) are control input signals to be designed to follow a given reference trajectory $(x_{r,ref}(t), y_{r,ref}(t), \theta_{r,ref}(t))$.

Remark 2.1. In [15], the unicycle model (1) considered the existences of disturbances in a 2D coordinate (x, y, θ). The dynamics become

$$\begin{aligned}\dot{x}_r &= \alpha_1 v\cos(\theta_r) + \beta_1\\ \dot{y}_r &= \alpha_2 v\sin(\theta_r) + \beta_2\\ \dot{\theta}_r &= \alpha_3\omega + \beta_3,\end{aligned} \tag{2}$$

where α_i and β_i parameters are added to represent the effect of possible slips from tracks. The disturbance parameters α_i and β_i, $i = 1, 2, 3$, are then estimated by Extended Kalman Filter (EKF) as explained in [15]. Using the idea similar to ADRC, after estimating the unknown parameters, the nonlinear controller was proposed to cancel the effect of uncertainties in [15]. Although the experimental results showed promising results, there was no theoretical analysis to show the boundedness of tracking errors in [15]. It is highlighted that in their design, the concept of ADRC is used to correct the reference velocities generated by the motion planner from ADS. While in our design, the motion planner from ADS is used to generate reference velocities (see Figure 2). These two designs use ADS in different ways, and our simulation results presented in Section 4.2 compare these differences. ○

Such a unicycle model is hard to use. In this work, we assume the UTGV is equipped with ADS, which provides global pose information to the robot and sends reference velocities (v_{ref}, ω_{ref}) to the motion controller using the

---

[a] The function is semi-continuous if it is continuous almost everywhere, except at certain points at which it is either upper semi-continuous or lower semi-continuous.

algorithms compatible with aforementioned ADS framework [18,19]. A more detailed explanation of the reason why the unicycle model is used is presented in Remark 2.2. This research focuses on tracking reference velocities, which are both in the local frame. Therefore, no global pose information is needed in this research, and only IMU and encoders are used to provide feedback in the control loop. Two motors are used to drive two tracks. Here (τ_l, τ_r) represents the torque applied to the left and right track respectively.

The control objective of this work is to design an appropriate low-level control loop to drive (τ_l, τ_r) so that the UTGV will follow the desired reference linear velocity v_{ref} and angular velocity ω_{ref} respectively, computed from the path planning and motion planning.

2.2. *A Simplified Model*

In this paper, a simple dynamic model for body frame $\{R\}$ is used for the design of the low-level control. It has the following form:

$$\dot{\omega} = \frac{1}{J}(\tau + d_\tau) \tag{3}$$

$$\dot{v} = \frac{1}{m}(F + d_F), \tag{4}$$

where J is the inertia of the UTGV, m is the mass of the UTGV, τ is the overall torque applied to the UTGV by the actuator, and d_τ represents the lumped unknown uncertain torque generated by unknown disturbances coming from the modelling uncertainties, the variation in the velocities per track, pose change due to the problem of slippage, as well as terrain compression as indicated in [17]. Similarly, the notion of F is the overall force applied by the actuator while d_F represents the unknown lumped uncertainties in terms of forces. These uncertainties, d_τ and d_F, are usually called total disturbance in ADRC and ESO.

It is noted that the motion planner normally runs at a lower frequency than the motion controller, therefore, the reference velocities of the motion controller from the motion planner can be treated as a constant. Thus the control objective is, for a given reference position trajectory, to design the control input (τ, F) to track constant reference velocities computed from the motion planner.

By using IMU and encoder information, the output of the UTGV is

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \omega \\ v \end{bmatrix} \in \mathbb{R}^2, \tag{5}$$

while the control input becomes

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tau \\ F \end{bmatrix} \in \mathbb{R}^2. \tag{6}$$

Remark 2.2. The reason for using this simple model is three-fold. Firstly of all, different from the unicycle model (1), this linear time-invariant (LTI) model consisting of (3) and (4) decouples the design of the linear velocity $v(t)$ and angular velocity $\omega(t)$. This greatly simplifies the design of control laws. Secondly, this simple LTI model (3) (4) considers unmodelled uncertainties d_τ and d_F, which can compensated the mismatches between dynamic (3) (4) and (1). Finally, this model only requires the information measured from the body frame without the need of (x, y, θ), which is needed in our problem setting. ○

It is assumed that the lumped uncertainty torque $d_\tau(t)$ and the lumped uncertainty force $d_F(t)$ satisfy the following assumption.

Assumption 2.3. The Dini derivatives of lumped uncertainty torque $d_\tau(\cdot)$ and the lumped uncertainty force $d_F(\cdot)$ exist almost everywhere. Moreover, they are in $\mathcal{L}_1 \cup \mathcal{L}_\infty$. That is, there exist two unknown positive constants $\bar{M}_\tau$, and $\bar{M}_F$ such that the following inequalities hold

$$\min\left\{\left\|D^+ d_\tau\right\|_\infty, \left\|D^+ d_\tau\right\|_1\right\} \leq \bar{M}_\tau \tag{7}$$

$$\min\left\{\left\|D^+ d_F\right\|_\infty, \left\|D^+ d_F\right\|_1\right\} \leq \bar{M}_F. \tag{8}$$

Remark 2.4. Assumption 2.3 includes a very large class of disturbances, including step disturbances, ramp disturbances (unbounded), pulses disturbances, and many non-smooth disturbances. In ADRC and ESO design, one of the standing assumptions for the total disturbances is that it is differentiable and its time derivative is bounded, for example, see [13,14] and reference therein. Although in [20, Hypothesis A4], the disturbances with jumps were considered, they are requested to be bounded. In this paper, the disturbances can be unbounded. To the best of the authors' knowledge, this assumption is the weakest possible assumption for total disturbances. As disturbances considered here might not be bounded, the widely used robust control techniques such as sliding mode control [21] cannot be directly applied either. ○

2.3. *Input Constraints*

In this work, $F(t)$ and $\tau(t)$ are the equivalent force and torque at the centre of gravity of the UTGV. These signals are used to compute the traction torques τ_r and τ_l to drive the right track and the left track respectively. The side view and the top view of the UTGV are shown in Figure 4 where (F_l, F_r) represents the equivalent force for the left and right tracks respectively. The notion of d is the height of the track while D is the distance between two tracks on each side.

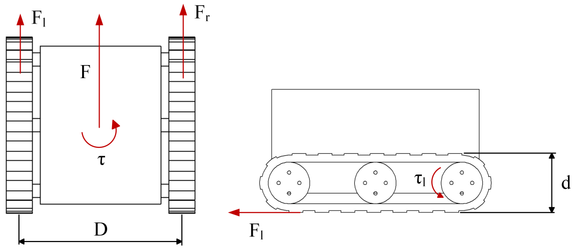


Fig. 4. UTGV Geometry - Left: top view; Right: side view.

When the tracks are not slipping, it has

$$\tau_r = F_r \cdot \frac{d}{2}, \qquad \tau_l = F_l \cdot \frac{d}{2}. \tag{9}$$

Assuming that the equivalent force satisfies

$$F = F_r + F_l,$$

with the equivalent torque τ satisfying

$$\tau = \frac{D}{2}(F_r - F_l),$$

leading to the following applied torque to the left and right track (τ_l, τ_r) respectively

$$\tau_r(t) = sat\left(\tfrac{1}{2}\left(F(t)\tfrac{d}{2} + \tau(t)\tfrac{d}{D}\right), \tau_{max}\right) \tag{10}$$

$$\tau_l(t) = sat\left(\tfrac{1}{2}\left(F(t)\tfrac{d}{2} - \tau(t)\tfrac{d}{D}\right), \tau_{max}\right), \tag{11}$$

where $\tau_{max} > 0$ is the maximum absolute torque that two actuators can produce. Here the saturation function $sat(\cdot,\cdot)$ is defined as

$$sat(\tau, \tau_{max}) = \begin{cases} \tau_{max} & \text{if } \tau \geq \tau_{max} \\ \tau & \text{if } -\tau_{max} \leq \tau \leq \tau_{max} \\ -\tau_{max} & \text{if } \tau \leq -\tau_{max} \end{cases}. \tag{12}$$

Remark 2.5. It is noted that (10) and (11) are obtained under the assumption that the tracks are not slipping. Such an assumption does not always hold when the UTGV is performing its tasks in engineering applications. One of the key advantages of this problem formulation is that the UTGV is modelled by two decoupled scalar systems. All the mismatches between the model, which is characterized by (10) and (11), and the UTGV can be incorporated into the total disturbances for the scalar dynamics. These unknown, possibly time-varying total disturbances will be estimated by ESO while ADRC will cancel the effect of the total disturbances, leading to robust performance. ○

Remark 2.6. From the (10) and (11), it follows that

$$|F(t)| \leq \frac{4\tau_{max}}{d}, \tag{13}$$

$$|\tau(t)| \leq \frac{2D \cdot \tau_{max}}{d}, \tag{14}$$

which are two constraints for input signal $\omega(t)$ and $v(t)$. Consequently, the dynamics (3) and (4) can be re-written as

$$\dot{\omega} = \tfrac{1}{J}\left(sat(\tau, \bar{\tau}_{max}) + d_\tau\right) \tag{15}$$

$$\dot{v} = \tfrac{1}{m}\left(sat(F, F_{max}) + d_F\right), \tag{16}$$

where $\bar{\tau}_{max} = \frac{2D \cdot \tau_{max}}{d}$ and $F_{max} = \frac{4\tau_{max}}{d}$. ○

With consideration of the simplified model and the input constraints, the control objective is modified as designing the actuation command $F(t)$ and $\tau(t), \forall t \geq 0$ with actuation constraints to track the desired reference linear velocity v_{ref} and angular velocity ω_{ref} respectively, computed from the path planning, in the presence of lumped uncertain torque d_τ and lumped uncertain force d_F, which satisfy Assumption 2.3.

Next will show how to use the ADRC and the ESO to achieve this modified control objective.

3. Main Results

Robust control methods are natural choices to handle the unknown lumped uncertainties d_τ and d_F that satisfy Assumption 2.3. Among them, the active disturbance rejection control (ADRC) algorithms (see, for example, [5–9] and references therein), working together with the extended state observer (ESO) (see, for example, [10–13] and references therein) have gained a lot of attention recently. The key ideas of using ADRC and ESO are quite simple. The role of ESO is to online estimate the lumped uncertainties, which are treated as the extended state, while ADRC will cancel the influences of lumped uncertainties or the extended state and stabilize the system or track the reference signal.

Let $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \omega \\ d_\tau \\ v \\ d_F \end{bmatrix}$ be the extended state of system characterized by (3) and (4). The proposed ESO takes the following form:

$$\Sigma_O : \begin{cases} \dot{\hat{x}}_1 = \frac{1}{J}\hat{x}_2 + \frac{1}{J}u_1 + l_1(y_1 - \hat{x}_1) \\ \dot{\hat{x}}_2 = \eta_1(y_1 - \hat{x}_1) \\ \dot{\hat{x}}_3 = \frac{1}{m}\hat{x}_4 + \frac{1}{m}u_2 + l_2(y_2 - \hat{x}_3) \\ \dot{\hat{x}}_4 = \eta_2(y_2 - \hat{x}_3) \end{cases}, \tag{17}$$

where y_1 and y_2 are defined in (5) and are measured from IMU and encoder. For a given desired velocities (v_{ref}, ω_{ref}), the control input signals u_1 and u_2 are defined in (6). Besides canceling the effect of the lumped uncertainties d_τ and d_F, a standard PI controller is also used to track the reference, thus, the ADRC takes the form of (18) and (20). That is,

$$u_1(t) = -\hat{x}_2 + k_{P,1}e_1 + k_{I,1}\int_0^t e_1(\tau)d\tau \tag{18}$$

where

$$e_1(t) = \omega_{ref} - y_1(t). \tag{19}$$

Similarly, $u_2(t) = F(t)$ is computed as

$$u_2(t) = -\hat{x}_4 + k_{P,2}e_2 + k_{I,2}\int_0^t e_2(\tau)d\tau, \tag{20}$$

where

$$e_2(t) = v_{ref} - y_2(t). \tag{21}$$

The parameters $l_1, \eta_1, l_2, \eta_2, k_{P,1}, k_{I,1}, k_{P,2}, k_{I,2}$ are tuning parameters to be designed.

Remark 3.1. It is noted that for a scalar dynamic system, a proportional controller can stabilize the system. An integration control is added to ensure zero steady-state error when tracking set-points. It can also reject constant disturbances as discussed in [22]. ◦

Before stating the main result, we introduce the closed-loop system in the state-space, which consists of the estimation error $\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{bmatrix} = \begin{bmatrix} x_1 - \hat{x}_1 \\ x_2 - \hat{x}_2 \\ x_3 - \hat{x}_3 \\ x_4 - \hat{x}_4 \end{bmatrix}$ and tracking error $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} \int_0^t e_1(\tau)d\tau \\ e_1 \\ \int_0^t e_2(\tau)d\tau \\ e_2 \end{bmatrix}$. They can be decoupled into two subsystems Σ_ω and Σ_v.

$$\Sigma_\omega : \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_{I,1}}{J} & -\frac{k_{P,1}}{J} & 0 & -\frac{1}{J} \\ 0 & 0 & -l_1 & \frac{1}{J} \\ 0 & 0 & -\eta_1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ D^+ d_\tau(t) \end{bmatrix},$$

$$\Sigma_v : \begin{bmatrix} \dot{z}_3 \\ \dot{z}_4 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_{I,2}}{m} & -\frac{k_{P,2}}{m} & 0 & -\frac{1}{m} \\ 0 & 0 & -l_2 & \frac{1}{m} \\ 0 & 0 & -\eta_2 & 0 \end{bmatrix} \begin{bmatrix} z_3 \\ z_4 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ D^+ d_F(t) \end{bmatrix}.$$

For the convenience of notation, we denote

$$A_1 = \begin{bmatrix} 0 & 1 \\ -\frac{k_{I,1}}{J} & -\frac{k_{P,1}}{J} \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 \\ -\frac{1}{J} & 0 \end{bmatrix}, A_3 = \begin{bmatrix} -l_1 & \frac{1}{J} \\ -\eta_1 & 0 \end{bmatrix},$$

$$A_4 = \begin{bmatrix} 0 & 1 \\ -\frac{k_{I,2}}{m} & -\frac{k_{P,2}}{m} \end{bmatrix}, A_5 = \begin{bmatrix} 0 & 0 \\ -\frac{1}{m} & 0 \end{bmatrix}, A_6 = \begin{bmatrix} -l_2 & \frac{1}{m} \\ -\eta_2 & 0 \end{bmatrix},$$

$$\Phi_1 = \begin{bmatrix} A_1 & A_2 \\ 0_{2\times 2} & A_3 \end{bmatrix}, \Phi_2 = \begin{bmatrix} A_4 & A_5 \\ 0_{2\times 2} & A_6 \end{bmatrix},$$

and $B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$. It is assumed that by selecting parameter pairs (l_1, η_1) and (l_2, η_2) appropriately, the matrices A_3 and A_6 are Hurwitz so that $\lim_{t\to\infty} \left|e^{A_3 t}\right| = 0$ and $\lim_{t\to\infty} \left|e^{A_6 t}\right| = 0$

Consequently, the following two closed-loop systems are obtained:

$$\Sigma_\omega : \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \Phi_1 \begin{bmatrix} z_1 \\ z_2 \\ \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} + BD^+ d_\tau, \qquad (22)$$

$$\Sigma_v : \begin{bmatrix} \dot{z}_3 \\ \dot{z}_4 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \end{bmatrix} = \Phi_2 \begin{bmatrix} z_3 \\ z_4 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{bmatrix} + BD^+ d_F. \qquad (23)$$

As Φ_1 (Φ_2) is in an upper-triangular form, its eigenvalues are determined by A_1 (A_4) and A_3 (A_6) respectively. With the consideration of the saturation function in computed τ_r and τ_l (see Equations (10) and (11)), the following results are obtained for Σ_ω. Similar results can be obtained for Σ_v as Σ_ω and Σ_v have a similar structure. Due to the space limitation, we only present the results for Σ_ω

We present two different sets of results for the closed-loop system (22). One result does not consider the input saturation, i.e., $\bar{\tau}_{max} = +\infty$, while the other result considers the input saturation.

Proposition 3.2. Let $\bar{\tau}_{max} = +\infty$. Assume that the uncertainty d_τ in the system (3) satisfies Assumption 2.3 with the control input u_1 taking the form of (18) with the ESO parameters l_1, η_1 are selected such that the matrix A_3 is Hurwitz. For the given reference velocity ω_{ref}, any disturbance bound $\bar{M}_\tau$, and any positive constant ν_1, by tuning parameters $k_{P,1}$, $k_{I,1}$, l_1 and η_1 appropriately, there exists a positive pair (M_1, λ_1) such that the solutions of the closed-loop system Σ_ω (22) satisfy the following inequality

$$|e_1(t)| \leq M_1 e^{-\lambda_1 t} |e_1(0)| + \nu_1, \qquad (24)$$

Moreover, no input saturation will happen and all the state trajectories of Σ_ω are uniformly bounded.

Proof: Due to space limitation, only the sketch of proof is provided.

In (22), if Assumption 2.3 is satisfied and the matrix A_3 is Hurwitz, we can show that two trajectories $(\tilde{x}_1(t), \tilde{x}_2(t))$ are uniformly bounded. The bound is related to $\bar{M}_r$ and is denoted as $M_{ESO,1} = M_{ESO,1}(\bar{M}_r)$.

It is noted that the dynamics of $z_1(t), z_2(t)$ are dominated by A_1 perturbed by A_2 with uniformly bounded solutions. The solutions converge to a neighbourhood of the origin ν_1, called ultimate bound [16]. For a given $M_{ESO,1}$ and ν_1, it is always possible to tune the parameters of the PI controller such that the inequality (24) holds for the given ν_1. Moreover, the state trajectories of Σ_ω are uniformly bounded. This completes the proof. □

Remark 3.3. Although Proposition 3.2 is long, its statement reflects the tuning process of the proposed ESO and ADRC. The tuning of PI parameters and parameters of ESO need to be coordinated in order to avoid input saturation. Based on the knowledge of disturbances, the system, the reference velocity and the desired ultimate bound ν_1, we can tune PI parameters and parameters of ESO to achieve the desired ultimate bound ν_1 without input saturation. It is noted that two parameters (M_1, λ_1) are dependent on the choice of PI parameters and cannot be arbitrarily selected. ◦

Remark 3.4. Proposition 3.2 indicates that when there is no input saturation, by tuning PI parameters appropriately, the tracking error can converge to an arbitrarily small ultimate bound if uncertainties satisfy Assumption 2.3. This includes two different cases. The first case is $D^+ d_\tau \in \mathcal{L}_\infty$. For a given $\bar{M}_r$, the bound $M_{ESO,1}$ can be made arbitrarily small by tuning l_1 and η_1. Under such a situation, tuning PI parameters becomes easier in order to achieve (24) for the arbitrarily small ν_1. The second case is $D^+ d_\tau \in \mathcal{L}_1$. In this case, for a given $\bar{M}_r$, the $M_{ESO,1}$ is bounded, but it cannot be tuned arbitrarily small. Consequently, the larger PI parameters are needed in order to achieve (24) for the arbitrarily small ν_1. ○

Next proposition considers how to tune the parameters of the PI controller in the presence of input saturation. A simple principle of design is to prevent saturation from happening by using smaller PI parameters. Such a principle can greatly simplify the stability analysis as the closed-loop system is still linear time-invariant when saturation does not happen. But this design will lead to weak stability results, such as a slower convergence speed, a smaller domain of attraction, and a larger ultimate bound.

It is worthwhile to highlight that the control law (18) will cancel the effect of the disturbance d_τ. If a time-varying disturbance $d_{tau}(t)$ satisfies Assumption 2.3 but unbounded at some time instant, by applying (18) with a perfect estimation $\hat{x}_2 = x_2 = d_\tau$, the computed control input becomes unbounded even though trajectories $z_1(t), z_2(t), \tilde{x}_1(t), \tilde{x}_2(t)$ are uniformly bounded. A bounded (or saturated input) is not able to reject an unbounded disturbance.

Next, the concept of compatibility between the disturbance and the saturation is proposed. We called the disturbance d_τ is compatible with input saturation bound $\bar{\tau}_{max}$ if it satisfies the following inequality:

$$\|d_\tau\|_\infty + M_d + M_{ESO,1}(\bar{M}_r) < \bar{\tau}_{max}. \tag{25}$$

This leads to the following result.

Proposition 3.5. Assume that the uncertainty d_τ in the system (3) satisfies Assumption 2.3 and is compatible with input saturation bound $\bar{\tau}_{max}$. Assume that the control input u_1 takes the form of (18) with the ESO parameters l_1, η_1 selected such that the matrix A_3 is Hurwitz. For the given reference velocity ω_{ref} and the given disturbance bound $\bar{M}_\tau$, there exists a positive pair (Δ_1, ν_2) such that by tuning parameters $k_{P,1}$, $k_{I,1}$, l_1, and η_1 appropriately, there exists a positive pair (M_1, λ_1) such that the solutions of the closed-loop system Σ_ω (22) satisfy the following inequality

$$|e_1(t)| \le M_1 e^{-\lambda_1 t} |e_1(0)| + \nu_2, \tag{26}$$

for all $|e_1(0)| \le \Delta_1$. Moreover, the input saturation will not happen and all the state trajectories of Σ_ω are uniformly bounded.

Proof:

For the given $\bar{\tau}_{max}$, we can select some domain of attraction $\Delta_1 < \bar{\tau}_{max}$ such that

$$\|d_\tau\|_\infty + \Delta_1 + M_{ESO,1} \le \bar{\tau}_{max}, \forall t \ge 0.$$

Moreover, for the given Δ_1, we can select PI parameter pair $(k_{P,1}, k_{I,1})$ to satisfy the following inequality:

$$\left| \begin{bmatrix} k_{I,1} & 0 \\ 0 & k_{P,1} \end{bmatrix} e^{A_1 t} \right| \Delta_1 + M_{ESO,1} \le \bar{\tau}_{max}, \forall t \ge 0,$$

so that no saturation will happen. Then, there exists an ultimate bound $\nu_2 < \Delta_1$ such that the inequality (24) holds. The uniform boundedness of the state trajectories of Σ_ω can be guaranteed. This completes the proof. ☐

Remark 3.6. When $D^+ d_\tau$ is in $\mathcal{L}_\infty$, we can select the parameters of ESO to achieve better performance. For example, we can choose the domain of the state estimation error sufficiently small so that if $\|d_\tau\|_\infty < \bar{\tau}_{max}$ is satisfied, the disturbance d_τ is always compatible with $\bar{\tau}_{max}$ by selecting l_1 and η_1 appropriately, Thus the compatibility assumption can be relaxed. When input saturation is considered, the parameters $(k_{P,1}, k_{I,1})$ need to be carefully tuned to prevent saturation from happening. It also suggests that we can not make ultimate bound ν_2 arbitrarily small. ○

4. Simulation Validation

Two types of simulations are presented to illustrate the effectiveness of the proposed method. The first type is to simulate how the selection of parameters of ADRC and ESO will affect the overall tracking performance. More precisely, the role of this simulation is to illustrate how two propositions work in terms of the selection of tuning parameters.

The second type of simulation is a Unity-ROS simulation, which has been widely used in robotics to evaluate robot performance in terms of localization, motion planning and control before implementing them in a robot. This simulation uses a skid drive with six wheels (three wheels on each side). The purpose of this simulation is to compare two different ways to track a given path: one uses a motion planner along with the control algorithm based on the unicycle model, and the other uses the motion planner and our proposed method. The performance comparison with the baseline PI controller is also presented.

4.1. *MATLAB Simulations*

As two subsystems (3) and (4) have similar structures, the focus in simulation is the subsystem (3) in order to demonstrate Propositions 3.2 and 3.5. For this subsystem, the parameters include PI parameters in (18), the domain of attraction Δ_1, and the ultimate bound ν_1 for a given positive constant $\bar{\tau}_{max}$ and $\bar{M}_r$.

Two different types of uncertainties will be considered.

D_1 : $d_\tau = \sin(10t) + t$. This disturbance satisfies $D^+ d_\tau \in \mathcal{L}_\infty$ with $\bar{M}_r = 11$.

D_2 : $d_\tau = \sin(10t) + pulse(t)$, where $pulse(t)$ is a periodic pulse signal with the period 2π. This disturbance satisfies $D^+ d_\tau \in \mathcal{L}_\infty \cap \mathcal{L}_1$ with $\bar{M}_r = 10$.

We also consider three cases: Case 1: there is no input saturation, i.e., $\bar{\tau}_{max} = +\infty$, Case 2: $\bar{\tau}_{max} = 5$, and Case 3: $\bar{\tau}_{max} = 1$.

It is worthwhile to highlight that the role of MATLAB simulations is to show how to tune the parameters of the proposed ESO and ADRC to achieve the desired performance when different disturbances are selected. The purpose of these simulations is to validate the main results

(Proposition 3.2 and Proposition 3.5). In these simulations, these disturbances are not requested to be practical as long as they satisfy Assumption 2.3. Moreover, these two disturbances are selected to emulate two possible practical scenarios. More precisely, D_1 can represent the condition when the UGTV is loading or unloading materials with the existence of some mechanical failure points in the continuous track. Similarly, D_2 emulates the simulation when there are bumps on the ground with the existence of some mechanical failure points in the continuous track. Moreover, the effectiveness of the proposed ADRC and ESO are tested by using Unity-ROS simulation and experiments with more practical disturbances in the following subsection.

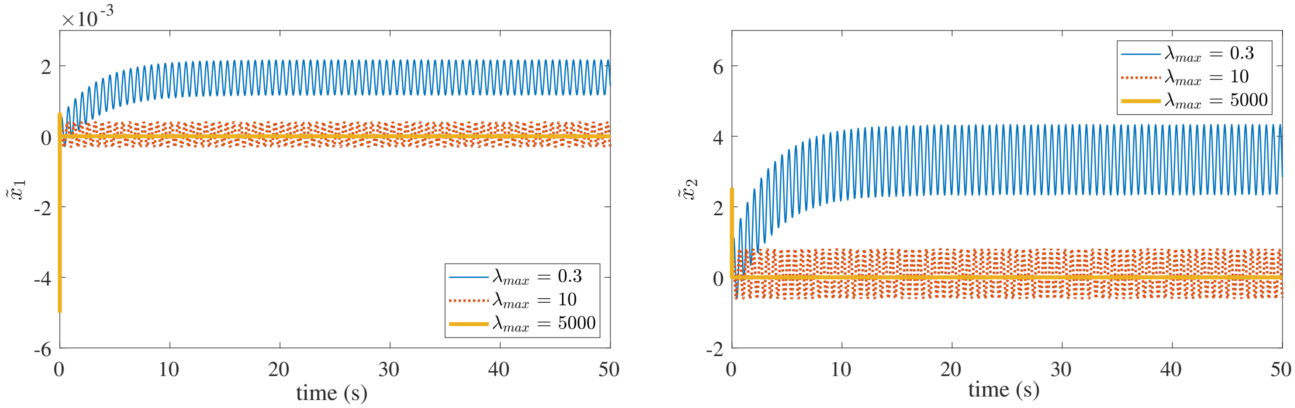
The simulation is performed using MATLAB R2021b (The MathWorks Inc., USA) and Simulink (The MathWorks Inc., USA). In all simulations, parameters m, J are fixed as $m = 7.7kg$ and $J = 0.2kg/m^2$, which are the values of m and J of the experiment platform that will be introduced in the next section.

Effect of ESO Parameters

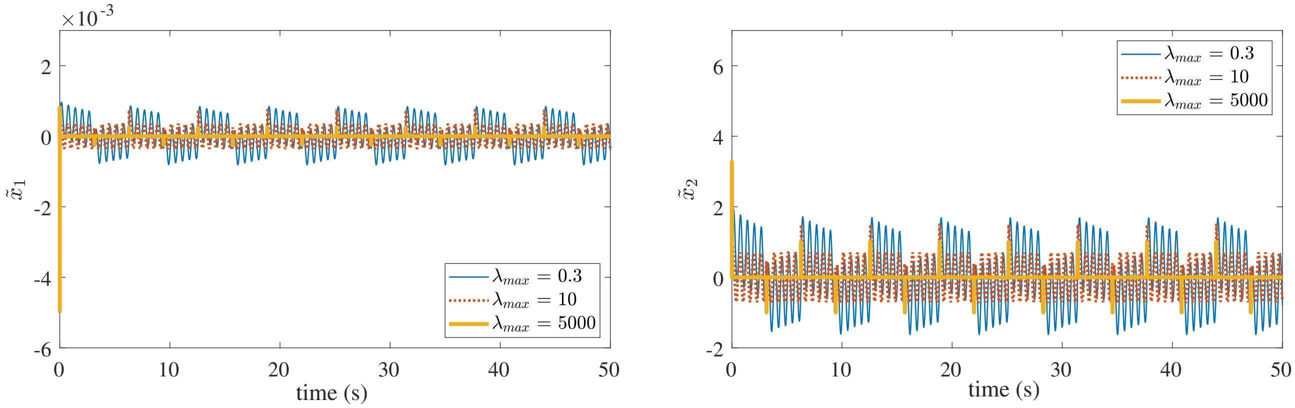
Next discuss how the choice of ESO parameters l_1, η_1 affects the tracking performance for different cases under two disturbances. It is noted that the eigenvalues of A_3 are determined by l_1 and η_1 when J is fixed. This matrix is selected to be a Hurwitz. We denote $\lambda_{max} = |\max\{\lambda_1, \lambda_2\}|$.

When D_1 exists, as shown in Figure 5(a)(b), the larger λ_{max}, the faster the estimation error converges with a smaller ultimate bound. By selecting appropriate PI parameters, as shown in Figure 6(a), we can obtain an arbitrarily smaller ultimate bound for $e_1(t)$, which is consistent with the results in Proposition 3.2. For the different choices of λ_{max}, we fix $\begin{bmatrix} \tilde{x}_1(0) \\ \tilde{x}_2(0) \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix}$. For example, three different A_3 matrices are selected as $A_{3,1} = \begin{bmatrix} -10000 & 5 \\ -600 & 0 \end{bmatrix}$, $\lambda_{max,1} = 0.3$, $A_{3,2} = \begin{bmatrix} -10010 & 5 \\ -20000 & 0 \end{bmatrix}$, $\lambda_{max,2} = 10$ and $A_{3,3} = \begin{bmatrix} -15000 & 5 \\ -10^7 & 0 \end{bmatrix}$, $\lambda_{max,3} = 5000$.

When D_2 exists, as shown in Figure 5(c)(d), the larger λ_{max}, the faster convergence speed of the estimation error with a smaller ultimate bound, but the ultimate bound can



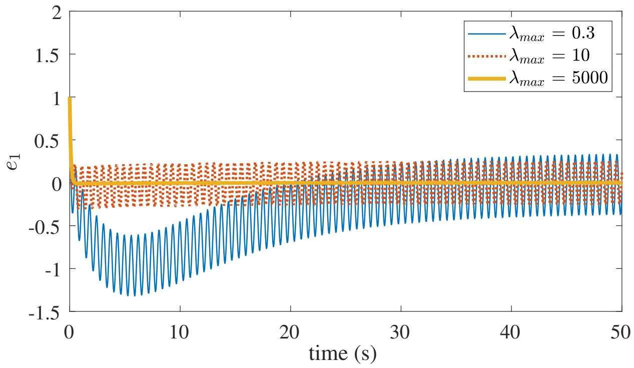
(a) $\tilde{x}_1$ with uncertainty D_1 (b) $\tilde{x}_2$ with uncertainty D_1



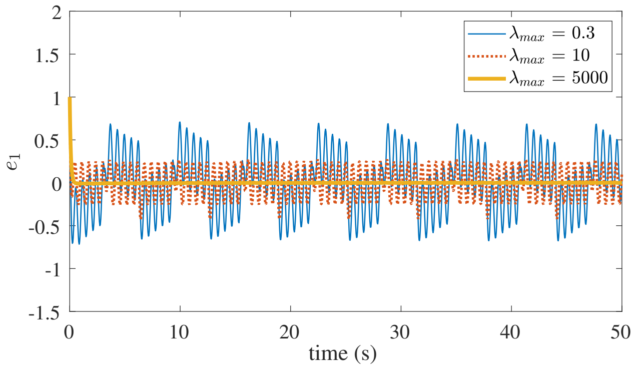
(c) $\tilde{x}_1$ with uncertainty D_2 (d) $\tilde{x}_2$ with uncertainty D_2

Fig. 5. Effect of ESO pole position on ultimate bound $M_{ESO,1}$ with Uncertainty D_1 and D_2.

not be arbitrarily small. Nevertheless, the ultimate bound for $e_1(t)$ can still be made arbitrarily small by tuning PI parameters appropriately, as shown in Figure 6(b).



(a) Uncertainty D_1



(b) Uncertainty D_2

Fig. 6. Given the same PI parameters and $\bar{\tau}_{max} = \infty$, different choices of parameters in ESO will affect the ultimate bound ν_1 for Uncertainties D_1 and D_2.

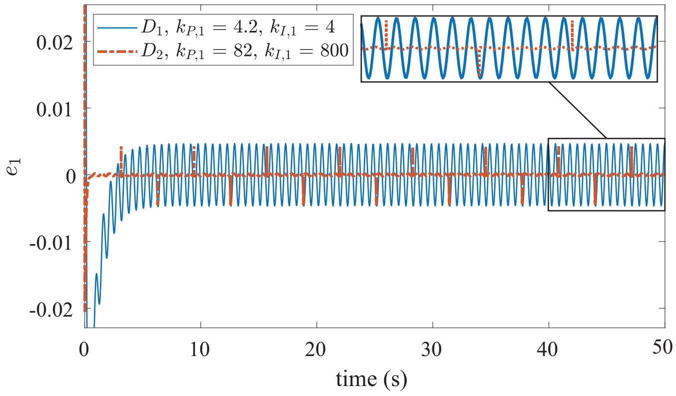


Fig. 7. Achieving the desired ultimate bound for two different types of disturbances by tuning parameters appropriately when there is no input saturation.

In order to demonstrate how to tune such PI parameters, we fix A_3 as $\begin{bmatrix} -10500 & 5 \\ -10^6 & 0 \end{bmatrix}$, $\lambda_{max,3} = 500$ as well as the desired ultimate bound $\nu = 0.01$. For two disturbances, two different sets of PI parameters are used to achieve this ultimate bound. For D_1, PI parameters are selected as $k_{P,1} = 4.2$, $k_{I,1} = 4$ while for D_2, PI parameters are selected as $k_{P,1} = 82$, $k_{I,1} = 800$. The performance is shown in Figure 7, which is consistent with the results in Proposition 3.2.

Effect of Input Saturation

When saturation exists, the first thing we need to check is whether the disturbances are compatible with the saturation bound in order to apply Proposition 3.5. Obviously, for any bounded input, it is not possible to cancel the effect of D_1 as it will approach ∞. The unbounded disturbance with input saturation might lead to unstable performance, as shown in Figure 8 when $\bar{\tau}_{max} = 5$ (Case 2) is used for the different choices of λ_{max}, and various choices of PI parameters.

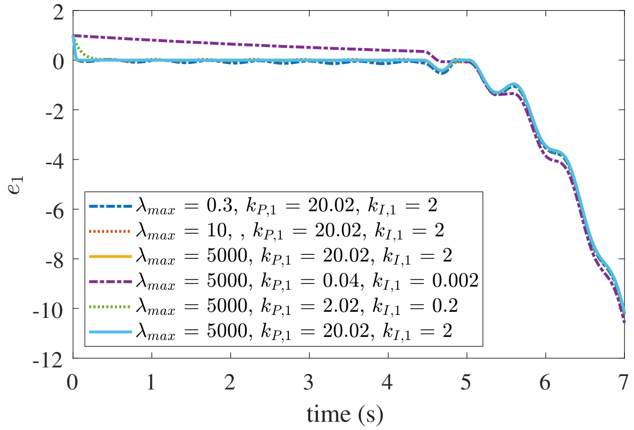


Fig. 8. The actuator bound is incompatible with Uncertainty D_1.

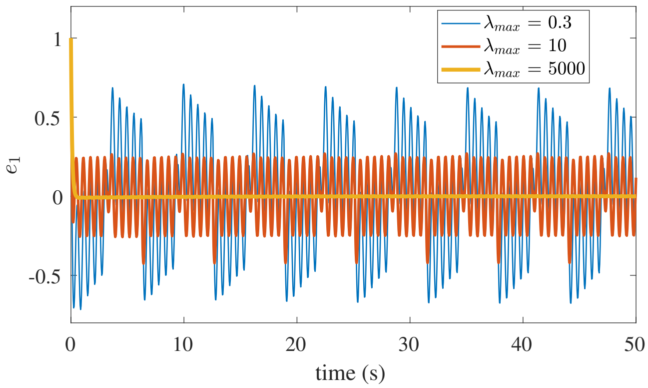


Fig. 9. The actuator bound is compatible with Uncertainty D_2.

For disturbances D_2, as stated in (25), it is compatible with $\bar{\tau}_{max} = 5$, but it is not compatible with $\bar{\tau}_{max} = 1$. Thus only Case 2 is simulated.

In Case 2, we can select small initial estimation error $\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.01 \end{bmatrix}$ and initial tracking error $e_1(0) = 1$ and fix PI parameters $k_{P,1} = 2.02, k_{I,1} = 0.2$. For three different choices of ESO, the three tracking errors converge to some ultimate bounds, as shown in Figure 9. This is consistent with the result in Proposition 3.5.

4.2. *Unity-ROS Simulations*

The simulation of a robot equipped with ADS is presented with Robot Operating System (ROS) distribution Noetic and Unity (Unity Software Inc., Denmark). The role of unity-ROS simulation is to compare two different ways to use ADS in autonomous systems and the baseline PI controller.

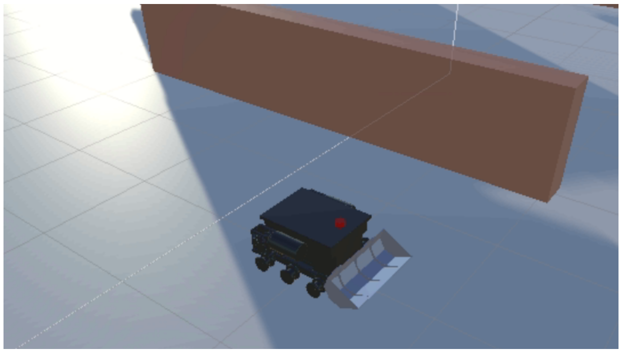


Fig. 10. Screen shot of the simulation scene in Unity.

Figure 10 shows the simulation environment in Unity. The robot model in the simulation has three wheels on each side. Two perceptions, 2D LiDAR scan and robot odometry, are used to provide information for the robot. The localization algorithm called Adaptive Monte-Carlo Localization (AMCL) [23] is used to provide global pose feedback. In these simulations, the robot is commanded to follow a round-corner-L-shaped trajectory. The straight lines of the trajectory are both 1.5m long, and the round corner has a radius of 1.5m. The motion planner is implemented via a PID controller [24], which runs at $10Hz$. Parameters of the motion planner is set as $target\_x\_vel = 1.5$, $target\_x\_acc = 10.0$, $target\_yaw\_vel = 4.0$, $target\_yaw\_acc = 10.0$, $Ki\_long = 0.2$, $Kp\_long = 5.0$, $Kd\_long = 0.5$, $Ki\_lat = 0.1$, $Kp\_lat = 10.0$, $Kp\_lat = 1.0$, $Ki\_ang = 0.4$, $Kp\_ang = 7.0$, $Kd\_ang = 0.8$. The definition of the parameters above is shown in [24]. All the parameters of them motion planner is set empirically.

An unicycle model + robust controller

The robust controller, mentioned in Remark 2.1, is used based on the given path and the unicycle model (see more details in [15]). It uses EKF based on (2) to estimate the disturbances online. The estimated disturbance terms are then used to correct the motion planner. The EKF runs at $50Hz$. To test the robustness of [15] over noises, uniformed noise whose probability density function (PDF) satisfies $U(-0.05, 0.05)$ is added to each of the three degrees of freedom of the pose measurements, and uniformed noise with PDF satisfying $U(-0.03, 0.03)$ is added to angular and linear velocity measurements. According to the pose noise and velocity noise, the process noise covariance matrix Q and measurement noise covariance matrix R are designed as

$$Q = 10^{-4} \times \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, R = 10^{-4} \times \begin{bmatrix} 8.3 & 0 & 0 \\ 0 & 8.3 & 0 \\ 0 & 0 & 8.3 \end{bmatrix}.$$

Motion planner + the proposed method

For a given path, the proposed method uses the motion planner, the simplified model, and the proposed ADRC and ESO. The selected parameters are $m = 150$, $J = 10$; and the values of the matrices are

$$A_1 = \begin{bmatrix} 0 & 1 \\ 0.3 & 10 \end{bmatrix}, A_3 = \begin{bmatrix} -60 & 0.1 \\ -900 & 0 \end{bmatrix}$$
$$A_4 = \begin{bmatrix} 0 & 1 \\ 0.0667 & 6.67 \end{bmatrix}, A_6 = \begin{bmatrix} -60 & 0.0067 \\ -900 & 0 \end{bmatrix}$$

To be tested in the Unity-ROS simulation, the proposed method is discretised with Euler method, and runs at $50Hz$.

PI controller

The widely used PI controller served as a baseline here, with the parameters $k_{I,\omega} = 3$, $k_{P,\omega} = 100$, $k_{I,v} = 10$, and $k_{P,v} = 1000$. The PI controller also runs at $50Hz$.

Two cases of wheel-terrain contact are simulated:

Case 1 Wheels on each side have the same friction rate (0.7) with the terrain;
Case 2 Friction rate (0.08) of the wheels on the right side is significantly lower than the left side (0.7).

The terrain is flat in both cases.

The three controllers are implemented using a sampling data structure, *i.e.* ,

$$u_1(t) = u_1(kT), u_2(t) = u_2(kT), \forall t \in [kT, (k+1)T), \tag{27}$$

where T is the sampling period.

As the reference trajectory has n sampling points, the position tracking error at the i^{th} sampling point $t = t_i$ is defined as

$$\epsilon_{x,i} = x_{r,ref}(t_i) - x_i,$$
$$\epsilon_{y,i} = y_{r,ref}(t_i) - y_i,$$

where (x_i, y_i) is the robot position in the global frame $\{G\}$ gat the i^{th} sampling instant. To evaluate the effect of the control law, the position mean absolute error (PMAE) is thus defined:

$$\epsilon = \frac{1}{n} \sum_{i=1}^{n} \sqrt{{\epsilon_x}_i^2 + {\epsilon_y}_i^2}. \tag{28}$$

To demonstrate the performance of ADRC and PI controllers in general cases, the averages of the PMAE in both cases are also calculated and presented in Table 1. The method proposed in this paper is denoted as $ADRC_v$, whereas $ADRC_p$ denotes the method proposed in [15] hereafter.

The results of the simulation are shown in Table 1. These results show that the proposed ADRC and ESO can work much better than the baseline PI controller and the robust control using the unicycle model.

Table 1. Tracking error of navigation system. (Unit: m)

	Case 1	Case 2	Average
ϵ_{PI}	0.236	0.507	0.372
ϵ_{ADRC_v}	0.127	0.254	0.191
ϵ_{ADRC_p}	0.195	0.386	0.291

5. Experimental Validation

An experiment is designed and tested on a lab-made experimental platform, most of whose parts are made by 3D printing, including the continuous tracks and track wheels as shown in Figure 11. The platform has similar track wheels as the prototype built in Figure 1. All the pieces of the track are bolted together. The chassis of the robot is a 6mm medium-density fibreboard (MDF) by laser cutting. Two L-shaped aluminum beams are installed laterally to reinforce the robot chassis. Compared to actual UTGVs like the one in Figure 1, the lab-made platform is smaller and lighter, but it shares the same dynamics principles as the actual ones since they both rely on the friction between the continuous tracks and the ground for movement.

The robot is driven by two motors, one on each side. They are connected to the back wheel to drive the continuous tracks. There are three track wheels on each side, only the back one is driven by the motor. In the experiments, based on the motor information, the values of F_{max} and $\bar{\tau}_{max}$ are $83.33N$ and $24.17Nm$. The onboard sensors are a nine-axis IMU and encoders that are preinstalled on the motor.

The angular velocity of the robot $\omega(t)$ is measured directly by the onboard IMU with the Root Mean Square (RMS) noise of $0.07°/s$. On the other hand, the linear velocity $v(t)$ is estimated with the Kalman Filter with the information of wheel speed measured by encoders and robot acceleration measured by the IMU with the RMS noise of $0.0098m/s^2$. These noise values are from the user manual of the sensor. The control laws are implemented by using an embedded system running on STM32F107ZET6 (STMicroelectronics, Switzerland). More precisely, two controllers (18) and (20) are implemented where two state signals $\hat{x}_2$ and $\hat{x}_4$ are obtained from the observer (17).

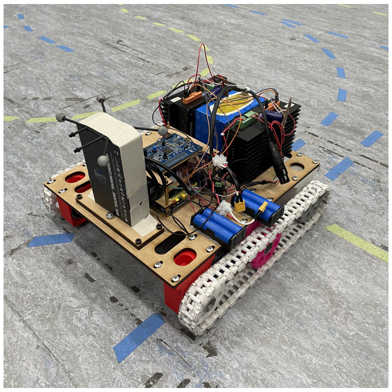


Fig. 11. Lab-made experimental platform.

Although the controllers (18) and (20) are designed in continuous time, it is implemented using a sampling data structure as shown in (27) where T is the sampling period. In the experiments, the sampling frequency is selected as $100Hz$ with $T = 0.01s$.

A camera-based motion capture system – OptiTrack (Tracklab, Australia), is used to measure the ground truth trajectory of the robot with the RMS error of $0.2mm$ at a $120Hz$ sampling rate. The positions are recorded during the experiment for post-processing.

It is noted that in the implementation of the control laws, no position information is used.

5.1. *Experiment Protocol*

The experiment includes two parts: the first part is to design reference velocities $v_r(t)$ and $\omega_r(t)$ for a given path. This is consistent with the self-driving framework mentioned in Section 1 and the control objective listed in Section 2.1.

As shown in Section 2, the first step of this work is to generate a reference trajectory in the global coordinate (see Equation 1 and Figure 3). Consequently, we use $x_{ref,R}(t), y_{ref,R}(t)$ to represent a 2-D reference trajectory for a fixed time $t \in [0, T_{final}]$. The robot is commanded to follow a pre-defined trajectory which requires the robot to run over a $4m$ rounded corner L-shaped path in 13 seconds. Each of the two algorithms is tested 3 times using the same trajectory to validate the performance. The trajectory is divided into three pieces, a 1.5-meter-long straight line, followed by a quarter of a circle with a radius of $0.5m$, and the final piece is also a straight line of $1.5m$. As shown in Figure 12, on each piece of the trajectory, reference velocities v_{ref}, ω_{ref} are given as the high-level navigation algorithms send step signals to the motion controller [19]. On the straight lines, the reference linear velocity is set to $0.3m/s$ for 5 seconds on each piece. Reference angular and linear velocities are set to $0.628rad/s$ and $0.314m/s$ on the curve.

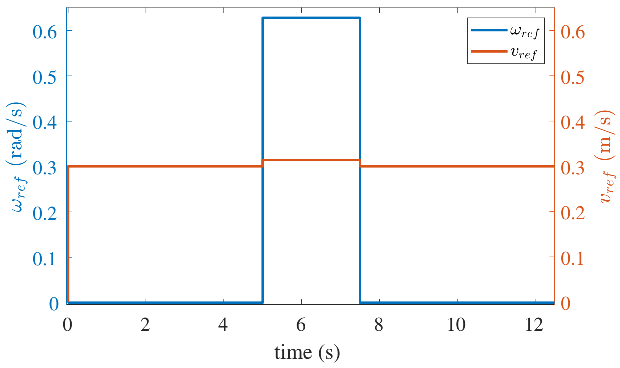


Fig. 12. Reference velocities for trajectory tracking.

Note that the initial position error of the robot does propagate along the trajectory since there is no global positioning system in the experiment platform. In our experiments, we fix the initial position and heading with the help of the camera-based motion capture system, OptiTrack, to keep the effect of initial position error to a minimum. In the future, we will discuss how to deal with initial position errors without OptiTrack.

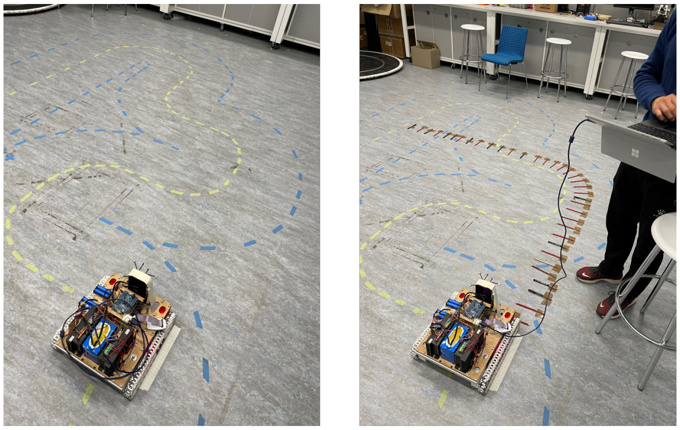
In order to test the robustness of the proposed algorithm, three different ground conditions are considered:

Case A Smooth ground (see Figure 13(a)). The tuning parameters of PI parameters and the proposed method (ADRC and ESO) are designed by trial-and-error for the smooth ground to avoid input saturation from happening. We test the robustness of two methods using Case B and Case C using the same parameters of two controllers tuned for Case A.

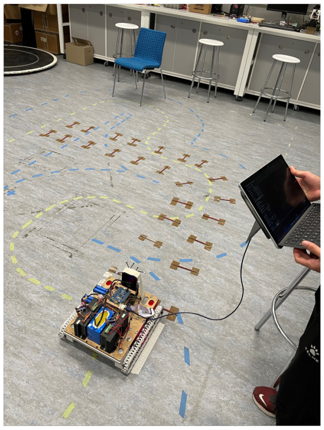
Case B Ground with two different friction coefficients (see Figure 13(b)). Some small obstacles are taped to the ground at random intervals and in a way that only the track on one side makes contact with them. These small obstacles are taped loosely so that some movements of the obstacles relative to the ground are allowed, which simulates a slippery ground condition when the track on either side contacts them.

Case C Ground with scattered small obstacles (see Figure 13(c)). The obstacles are taped firmly to the ground, and both tracks are in contact with them.

In Case B, different ground conditions are created for the tracks on different sides, leading to some discontinuous friction when the UTGV passes it. Such uncertainty is in $\mathcal{L}_1 \cup \mathcal{L}_\infty$. Thus our results are applicable. In Case C, both tracks can have discontinuous frictions. It is noted that in both Case B and Case C, the uncertainties are randomly set, and the averaged error is used to evaluate the performance of two control laws (PI and ADRC+ESO).



(a) Case A setup. (b) Case B setup.



(c) Case C setup.

Fig. 13. Experiment setup of Case A, B and C.

The same as the Unity-ROS simulation introduced in Section 4.2, PMAE is also used to evaluate the path tracking performance in the experiments. Besides, since the simplified models (3) and (4) track velocities, the velocity tracking performance is also evaluated. As these obstacles are randomly distributed on the ground, it is reasonable to use average tracking error when tracking the velocities. Thus given the reference linear velocity and reference angular velocity, the Mean Error (ME) is chosen as the metric for evaluating the tracking performance on velocities. They are thus defined as

$$\bar{e}_1 = \frac{1}{n}\sum_{i=1}^{n} e_1(t_i), \tag{29}$$

$$\bar{e}_2 = \frac{1}{n}\sum_{i=1}^{n} e_2(t_i), \tag{30}$$

where e_1 and e_2 are defined in (19) and (21), t_i is each sampled time instant during the measurement. Each experiment is repeated three times. The average performance is presented.

For each case, we will compare the performance between the proposed algorithm and the standard proportional-integral (PI) controller in terms of ϵ, $\bar{e}_1$, and $\bar{e}_2$. The parameters of PI are tuned to be $k_{P,\omega} = 12$, $k_{I,\omega} = 6$, $k_{P,v} = 200$, $k_{I,v} = 100$, where $k_{P,\omega}$ and $k_{I,\omega}$ are proportional gain and integral gain for the dynamic system in (3), $k_{P,v}$ and $k_{I,v}$ are proportional gain and integral gain for the dynamic system in (4).

The result is presented in Table 3 and Table 4.

5.2. *Results and Discussion*

The comparison between the proposed algorithm and the PI controller is presented in the following tables. Here the computation of ϵ, $\bar{e}_1$, and $\bar{e}_2$ come from (28), (29), and (30) respectively.

Table 2 shows that the average PMAE of the three cases of ADRC $\epsilon_{ADRC} = 0.316m$, which is 17.17% better than that of PI $\epsilon_{PI} = 0.382m$, showing the effectiveness of the proposed algorithm.

Table 2. Tracking PMAE (Unit: m)

	Case A	Case B	Case C	Average
ϵ_{PI}	0.313	0.431	0.402	0.382
ϵ_{ADRC}	0.238	0.365	0.346	0.316

Next will show the tracking performance for the planned reference linear velocity and angular velocity. The linear and angular velocity of the UTGV during the experiment is obtained by using the sliding window approach on the numerical differentiation of OptiTrack position and orientation measurements. The step size of the sliding window approach is chosen to be 10. For three cases, we have verified that the disturbances are compatible with the saturation bound aforementioned.

One of the design principles is to tune the parameters of ADRC and ESO so that for a given set of initial conditions, the input saturation will not happen. By selecting the parameters of ADRC and ESO carefully as $k_{I,1} = 0.1$, $k_{P,1} = 0.85$, $l_1 = 71.34$, $\eta_1 = 254.47$; $k_{I,2} = 0.24$, $k_{P,2} = 30.09$, $l_2 = 71.34$, $\eta_2 = 254.47$ based on the guess of initial values, no saturation happens.

Table 3. Angular velocity tracking ME. (Unit: rad/s)

	Case A	Case B	Case C	Average
$\bar{e}_{1,PI}$	0.0038	0.0260	0.0060	0.0119
$\bar{e}_{1,ADRC}$	-0.0043	0.0087	0.0064	0.0064

Table 4. Linear velocity tracking ME. (Unit: m/s)

	Case A	Case B	Case C	Average
$\bar{e}_{2,PI}$	0.0233	0.0404	0.0335	0.0321
$\bar{e}_{2,ADRC}$	0.0048	0.0250	0.0193	0.0163

The results in Table 3 show that, in terms of tracking the angular velocities, PI and ADRC have similar performances in Case A and C, while ADRC has better performance in rejecting the unexpected slippery in Case B. On the other hand, the results in Table 4 illustrate that ADRC demonstrates less error than the PI controller in tracking the linear velocities in all cases. On average, ADRC could reduce almost 50% (49.1% and 46.1%) of error in tracking linear and angular velocities. These results suggested that ADRC has better robustness properties as indicated in Proposition 3.5.

It is highlighted that in this simplified model, two input signals, ω and v, are designed independently. They are physically dependent on each other, as discussed in [25]. The coupling between them becomes a part of the total disturbance. The effect of this coupling is thus cancelled by using ESO and ADRC. Intuitively such a cancellation might cause unnecessary high gain control input signals. From our Unity-ROS simulation and experiments results, it was shown that the proposed method can achieve better performance compared with the robust controller similar to ADRC when using a unicycle model [15] (see Table 1) and PI controller (see Tables 2-4).

6. Conclusion

This work shows that by decoupling the path tracking problem of unmanned tracked ground vehicles (UTGVs) into path/motion planning and tracking of reference linear and angular velocity, the complicated dynamics of UTGVs can be simplified by two decoupled linear dynamics with unknown uncertainties. Such a problem formulation makes the active disturbance rejection control (ADRC) algorithm applicable, along with extended state observer (ESO). By introducing the Dini derivative, the proposed ADRC can handle non-smooth disturbances, which are commonly encountered in the mining industry. When the input constraints are considered, if the uncertainties are compatible with the input constraint, our main results show that the tracking error converges to an ultimate bound. Simulation results and experimental results demonstrate the effectiveness of the proposed algorithm.

References

[1] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima and S. Kawatsuma, Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile

rescue robots, *Journal of Field Robotics* **30**(1) (2013) 44–63.

[2] R. R. Murphy, J. Kravitz, S. L. Stover and R. Shoureshi, Mobile robots in mine rescue and recovery, *IEEE Robotics & Automation Magazine* **16**(2) (2009) 91–103.

[3] R. W. Brockett, Control theory and singular riemannian geometry, *New directions in applied mathematics*, eds. P. J. Hilton and G. S. Young (Springer, 1982), pp. 11–27.

[4] V. Shreyas, S. N. Bharadwaj, S. Srinidhi, K. U. Ankith and A. B. Rajendra, Self-driving cars: An overview of various autonomous driving systems, *Advances in Data and Information Sciences*, eds. M. L. Kolhe, S. Tiwari, M. C. Trivedi and K. K. Mishra (Springer Singapore, Singapore, 2020), pp. 361–371.

[5] B. Z. Guo and Z. L. Zhao, On convergence of the nonlinear active disturbance rejection control for mimo systems, *SIAM Journal on Control and Optimization* **51**(2) (2013) 1727–1757.

[6] Z. Gao, Active disturbance rejection control: a paradigm shift in feedback control system design, *2006 American control conference*, IEEE (2006), pp. 7–pp.

[7] J. Han, From pid to active disturbance rejection control, *IEEE transactions on Industrial Electronics* **56**(3) (2009) 900–906.

[8] Y. Huang and W. Xue, Active disturbance rejection control: Methodology and theoretical analysis, *ISA transactions* **53**(4) (2014) 963–976.

[9] Z. L. Zhao and B. Z. Guo, On convergence of nonlinear active disturbance rejection control for siso nonlinear systems, *Journal of Dynamical and Control Systems* **22**(2) (2016) 385–412.

[10] B. Z. Guo and Z. L. Zhao, On the convergence of an extended state observer for nonlinear systems with uncertainty, *Systems & Control Letters* **60**(6) (2011) 420–430.

[11] Q. Zheng, L. Q. Gao and Z. Gao, On validation of extended state observer through analysis and experimentation, *Journal of Dynamic Systems, Measurement, and Control* **134**(2) (2012).

[12] B. Z. Guo and Z. L. Zhao, On convergence of nonlinear extended state observer for multi-input multi-output systems with uncertainty, *IET Control Theory & Applications* **6**(15) (2012) 2375–2386.

[13] Z. L. Zhao and B. Z. Guo, Extended state observer for uncertain lower triangular nonlinear systems, *Systems & Control Letters* **85** (2015) 100–108.

[14] Y. Yuan, Z. Wang, Y. Yu, L. Guo and H. Yang, Active disturbance rejection control for a pneumatic motion platform subject to actuator saturation: An extended state observer approach, *Automatica* **107** (2019) 353–361.

[15] B. Sebastian and P. Ben-Tzvi, Active disturbance rejection control for handling slip in tracked vehicle locomotion, *Journal of Mechanisms and Robotics* **11**(2) (2019) p. 021003.

[16] H. K. Khalil, *Nonlinear system* (Prentice Hall, New Jersey, 2002).

[17] T. Zou, J. Angeles and F. Hassani, Dynamic modeling and trajectory tracking control of unmanned tracked vehicles, *Robotics and Autonomous Systems* **110** (2018) 102–111.

[18] P. E. Hart, N. J. Nilsson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE transactions on Systems Science and Cybernetics* **4**(2) (1968) 100–107.

[19] D. Fox, W. Burgard and S. Thrun, The dynamic window approach to collision avoidance, *IEEE Robotics & Automation Magazine* **4**(1) (1997) 23–33.

[20] W. Xue and Y. Huang, Performance analysis of 2-dof tracking control for a class of nonlinear uncertain systems with discontinuous disturbances, *International Journal of Robust and Nonlinear Control* **28**(4) (2018) 1456–1473.

[21] H. Oh, S. Kim, A. Tsourdos and B. A. White, Decentralised standoff tracking of moving targets using adaptive sliding mode control for uavs, *Journal of Intelligent & Robotic Systems* **76**(1) (2014) 169–183.

[22] N. S. Nise, *Control systems engineering* (John Wiley & Sons, 2020).

[23] P. Pfaff, W. Burgard and D. Fox, Robust monte-carlo localization using adaptive likelihood models, *European robotics symposium 2006*, Springer (2006), pp. 181–194.

[24] C. L. Tim Clephas, Rokus Ottervanger, tracking_pid.

[25] R. L. Huston, Unicycle dynamics and stability, tech. rep., SAE Technical Paper (1984).