PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

How extending the kinematic model affects path following in off-road terrain for differential drive UGVs

Andreas Tisland, Magnus Baksaas, Kim Mathiassen

Andreas Tisland, Magnus Baksaas, Kim Mathiassen, "How extending the kinematic model affects path following in off-road terrain for differential drive UGVs," Proc. SPIE 12124, Unmanned Systems Technology XXIV, 1212406 (31 May 2022); doi: 10.1117/12.2618731

SPIE.

Event: SPIE Defense + Commercial Sensing, 2022, Orlando, Florida, United States

How extending the Kinematic Model affects Path Following in Off-Road Terrain for Differential Drive UGVs

Andreas Tisland[a,b], Magnus Baksaas[a], and Kim Mathiassen[a,b]

[a]Norwegian Defence Research Establishment, Instituttveien 20, Norway
[b]Department of Technology Systems, University of Oslo, Gunnar Randers vei 19, 2007 Kjeller

ABSTRACT

Traversal in off-road conditions for Unmanned Ground Vehicles is highly relevant for defence applications, with an increasing amount of research being put into the field. A central part of the autonomous traversal is path following, where there currently exist many stable controllers. However, conventional path following controllers often relies on ideal vehicle models that make assumptions about the terrain that are no longer valid in off-road conditions. Therefore, research is needed into how conventional controllers are affected by off-road terrain and if extending the vehicle model with relevant parameters can improve the performance.

In this paper, a controller based on Active Disturbance Rejection Control and a controller based on Instantaneous Centre of Rotation are tested against a conventional controller in off-road conditions. Results from simulations illustrate how the conventional controller is affected by variations in the vehicle's rotation centre while the proposed controllers have improved performance when simulating rough terrain conditions. Real world experiments were conducted in uneven sandy terrain, where all of the controllers showed decent performance, but the proposed controllers had the lowest cross-track error.

Future Unmanned Ground Vehicle operations can improve performance by using the proposed controllers when the vehicle is experiencing rough terrain where the Instantaneous Centre of Rotation is considerably shifted from its ideal location. On the other hand, the conventional controller should produce decent performance in moderate conditions. Further research is needed to understand what types of real world conditions make the performance of the conventional controller significantly decrease, thus justifying the use of one of the proposed controllers.

Keywords: Unmanned Ground Vehicle, path following, autonomous vehicle, mobile robots, active disturbance rejection control, instantaneous centre of rotation

1. INTRODUCTION

The context of this paper is a research project on autonomy for defence applications carried out by the Norwegian Defence Research Establishment (FFI), where the project entails understanding how autonomous systems work and how they can be used by the Norwegian Armed Forces. The vehicle used for research by FFI is an Unmanned Ground Vehicle (UGV) called THeMIS, developed by Milrem Robotics. It is a tracked differential drive vehicle, specifically designed for military applications and traversal in off-road conditions. There are several possible applications of this vehicle in defence.[1–3] From unmanned reconnaissance missions and on-base logistics to transporting supplies and supporting dismounted troops, where an important requirement is the ability to traverse varying and rough terrain conditions. Initial development and testing have showcased functioning autonomous operation of the vehicle using FFIs platform, both in simulation and real world experiments.[1]

Path following with the UGV is currently performed by a controller based on the unicycle kinematic model, which assumes no slippage at the tracks and traversal on a planar surface. When driving in off-road conditions, the vehicle will be subject to slopes and also slippage caused by cohesion and compaction of the soil.[4] As a result, the assumptions in the unicycle model are no longer valid, which can lead to decreased path following

Further author information: (Send correspondence to K.M.)
K.M.: E-mail: kim.mathiassen@ffi.no

Unmanned Systems Technology XXIV, edited by Hoa G. Nguyen, Paul L. Muench, Brian K. Skibba,
Proc. of SPIE Vol. 12124, 1212406 · © 2022 SPIE · 0277-786X · doi: 10.1117/12.2618731

performance. Therefore, it is of interest to investigate path following approaches that account for the disturbances caused by off-road driving.

Many approaches exist for designing systems with better ability to handle harsh off-road conditions, regarding both vehicle models and path following controllers. The three main vehicle models in the literature[5] are: geometric model, kinematic model and dynamic model. Geometric models contain minimal amount of information, which quickly makes them inaccurate off-road conditions. On the other hand, dynamic models can be complicated and it may be difficult to obtain terrain-dependent parameters. Therefore, this paper focuses on kinematic models as they can accurately describe vehicle movement, without being subject to terrain-dependent parameters.

To account for varying friction in the environment, there have been many approaches considering slip effects in the kinematic model.[6–9] In Ref. 6 they included a simple term relating estimated slip to the kinematic model. The slip was calculated based on the difference between measured linear velocity of the vehicle and the measured angular velocity of each track. Improved tracking performance was achieved using this model with a controller based on feedback linearization.[6] In Ref. 9, the slip kinematic methodology was extensively used and incorporated with several controllers. They used a more extensive model than Ref. 6 and included a novel offset model to design controllers. Controllers were improved to better tackle uncertainties and disturbances in the terrain, where they got the best performance using Sliding Mode Control fitted with a Disturbance Observer. Computer vision[10,11] has also been used for estimating slip in terrain conditions as an alternative to comparing velocity measurements.

The Instantaneous Centre of Rotation (ICR) kinematic model describes the vehicle's movement related to the location of its rotation centres, which are known to vary in different types of terrain.[12] In Ref. 13, this was presented for tracked vehicles where the model considers the ICR of each track and the vehicle as a whole. Estimating ICR locations using an Extended Kalman Filter (EKF)[12] is a relevant approach for off-road traversal, as it obtains estimates in real-time and can therefore account for transitions between different terrain types. In Ref. 14, researchers mapped the position of a vehicle in the ideal model to a vehicle with tracks on the ICR locations. In this way, they could in each moment consider the mapped vehicle as an ideal vehicle and use an ordinary path following controller for the unicycle model. The results showed improvements compared to using an ideal unicycle model without ICR-locations. There has also been ICR-based models using nonlinear Lyapunov-based controllers,[15] where researchers experimentally tested their approach on a skid-steered wheeled robot in different terrain types and with higher speed. The controller outperformed Ref. 14 (discussed previously) and also Ref. 16 which uses a dynamic vehicle model.

Model Predictive Control (MPC) offers iterative optimisation of control objectives with the ability to satisfy several constraints. MPC has been applied to path following for UGVs operating in off-road conditions.[9,17,18] In Ref. 9, researchers incorporated the slip kinematic model with MPC where the controller was adapted and robustified to cope with disturbances, resulting in improved path following performance. The ICR kinematic model has also been used with MPC,[17] where researchers proposed a novel method for parameter estimation using an EKF in order to obtain estimates of the ICR locations. Results from experiments showed the controller outperforming normal MPC and a Levenberg-Marquardt based approach as well as a classical path follower (pure pursuit). To account for uncertainties in the system model there have been approaches trying to make MPC more robust[19,20] when applied to path following.

In recent years, Active Disturbance Rejection Control (ADRC) has increasingly been used for path following.[21–23] ADRC was applied to path following in Ref. 23 for a tracked vehicle in conditions with slippage, based on an an extended unicycle model where the total disturbance was modelled as scaling and shifting of the vehicle's kinematics, which were compensated for in the control law. A simple Proportional Derivative (PD) controller was used for path following and an EKF was used to estimate the disturbances. Experimental trials were performed in several terrain conditions showing that when using ADRC, the path following performance was improved compared to the same controller without ADRC, meaning that the controller was able to compensate for some of the disturbances from the terrain. A weakness of the testing in this particular paper was that there were no comparison against other state-of-the-art approaches to path following or terrain modelling.

In this paper two path following controllers based on extended kinematic models were evaluated, and compared against the current unicycle-based controller.[1] This was done to investigate how the ideal unicycle model is

affected by off-road traversal and if extending the kinematic model with relevant parameters can improve path following performance. The current controller (Chapter 34.4.2 in Ref. 24), referred to as Morin, is based on the ideal unicycle model and its implementation is described in Ref. 1. The first proposed controller, denoted HBZ, is the nonlinear controller from Ref. 15 using the ICR kinematic model, wherein this paper the ICR locations are estimated in real-time using an EKF.[12] The second proposed controller, denoted ADRC, is based on Active Disturbance Rejection Control using an extended unicycle model[23] where the Morin controller is used for path following.

The reasons for choosing these particular approaches are that they extend the kinematic model with relevant parameters to increase the accuracy of the vehicle model and additionally have shown improved performance in their respective papers. Table 1 lists an overview of the controllers, details are presented in Section 2.

Controller	Description	Source
Morin	Nonlinear control based on the ideal unicycle model.	Ref. 1, 24
HBZ	Nonlinear control using ICR model with real time estimates.	Ref. 12, 15
ADRC	Extended unicycle model where disturbances are compensated from the Morin controller.	Ref. 23

Table 1: Controllers with name and description

In the following, Section 2 presents the proposed approaches in detail with theory and equations. Section 3 details the simulation and real world experiments. Then, Section 4 presents results from the experiments, followed by Section 5 where the results are discussed and interpreted. Finally, Section 6 will contain concluding remarks from the results in this paper.

2. SIMULATION MODEL AND CONTROLLERS

This chapter starts with presenting the simulation model and follows with each controller and EKF, explaining their equations and parameters.

2.1 Simulation model

The simulation model is based on the ICR kinematic model,[13] which can simulate the vehicle's motion related to its rotation centres. These will ideally lie at the centre of each track, but these can move when traversing varying terrain. By using this model, the purpose is to explicitly vary the rotation centres of the vehicle laterally and longitudinally to simulate terrain traversal.

Each controller generates control inputs v and ω which are distributed to each track based on ideal ICR-locations. For the vehicle in this paper, with distance between tracks $L = 2.0$m, the ideal ICR-locations are located at $y_L = 1.0$, $y_R = -1.0$ and $x_c = 0.0$. The resulting velocity is at the left track $v_L = v - \frac{\omega L}{2}$ and the right track $v_R = v + \frac{\omega L}{2}$. These can then be sent to the ICR-kinematic model where the ICR-locations of the vehicle are set to simulate motion in varying terrain.

With the ICR-locations (y_L, y_R, x_c) set, see Figure 1, the motion of the vehicle can be simulated from controller inputs v_L and v_R

$$v_x = \frac{v_R y_L - v_L y_R}{y_L - y_R} \qquad v_y = \frac{v_L - v_R}{y_L - y_R} x_c \qquad \omega = -\frac{v_L - v_R}{y_L - y_R} \tag{1}$$

Giving motion in the global frame

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{2}$$

Using this relation, the vehicle's motion can be simulated from generated control inputs v_L and v_R resulting in movement $\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$

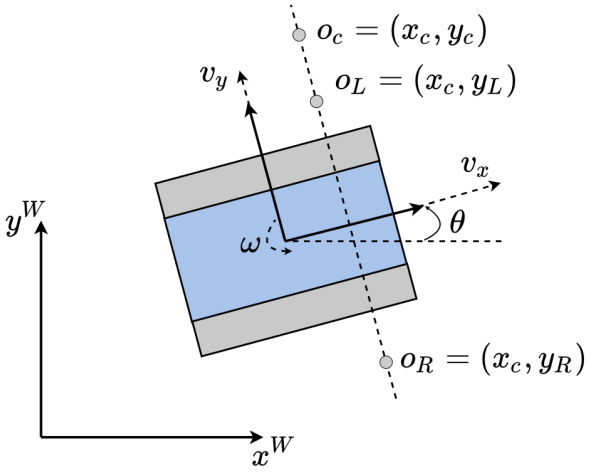


Figure 1: ICR model for a differential drive vehicle

2.2 Controllers

Here we present the three controllers used in this paper. First is the Morin controller, then the HBZ controller, and at last the ADRC controller.

2.2.1 Morin

The controller currently used on the UGV is found in Chapter 34.4.2 in Handbook of Robotics,[24] where the nonlinear controller is expressed by

$$w_2 = -k_2 v_r y_e - k_3 |v_r| \tan(\theta_e) \quad (k_2, k_3 > 0) \tag{3}$$

$$\omega = w_2 \cos^2(\theta_e) + \omega_r \tag{4}$$

where v_r and ω_r are the nominal linear and rotational velocity of the reference vehicle and y_e and θ_e are the cross-track and heading error in the reference vehicle frame. The controller has been shown to give a globally asymptotically stable system.[24] For more details on this controller and its implementation on the UGV, see Ref. 1.

2.2.2 HBZ

The controller from Ref. 15 uses the ICR kinematic model where the vehicle model is transformed to path coordinates (Serret-Frenet frame), as seen in Figure 2, using the following transform

$$\mathbf{q}^W = \mathbf{p}^W + \mathbf{R}_F^W \mathbf{r}^F \tag{5}$$

$$\dot{\mathbf{q}}^W = \dot{\mathbf{p}}^W + \dot{\mathbf{R}}_F^W \mathbf{r}^F + \mathbf{R}_F^W \dot{\mathbf{r}}^F \tag{6}$$

where W denotes the world frame, F the Serret-Frenet frame on the path and

$$\mathbf{r}^W = \begin{bmatrix} x_e \\ y_e \end{bmatrix}, \ \mathbf{R}_F^W \dot{\mathbf{p}} = \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{q}^W = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

$$\dot{\mathbf{r}}^F = \mathbf{R}_W^F \dot{\mathbf{q}}^W - \mathbf{R}_W^F \dot{\mathbf{p}} - \mathbf{R}_W^F \mathbf{S}(\omega_F^W) \mathbf{R}_F^W \mathbf{r}^F \tag{7}$$

The error dynamics of the vehicle using path coordinates can then be expressed as

$$\dot{x}_e = v \cos\theta_e + x_c \omega \sin\theta_e - \dot{s}(1 - c(s) y_e) \tag{8}$$

$$\dot{y}_e = v \sin\theta_e - x_c \omega \cos\theta_e - \dot{s} c(s) x_e \tag{9}$$

$$\dot{\theta}_e = \omega - \dot{s} c(s) \tag{10}$$

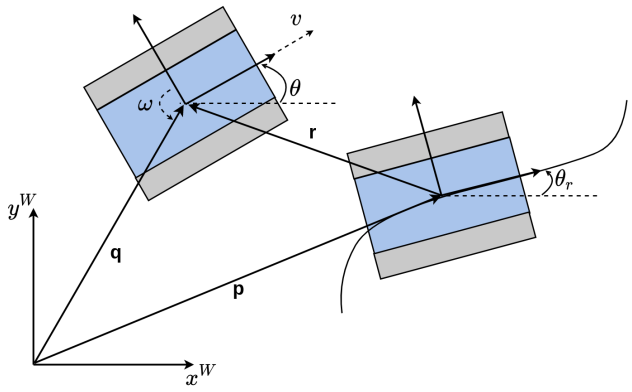


Figure 2: Geometry of vehicle with path coordinates

Where s is the arc length to the reference point and $c(s)$ is the path curvature at point s. Curvature is defined as

$$c(s) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}} \tag{11}$$

Controlling the vehicle is separated into a steering control law and a speed control law.

Steering control law The steering control law controls the change of path coordinate s and the orientation error θ_e.

$$\dot{s} = v\cos\theta_e + x_c\,\omega\sin\theta_e + k_1 x_e \tag{12}$$

$$\dot{\theta}_e = \dot{\psi}(y_e, v) + \frac{|\sin(\theta_e - \psi(y_e, v))|}{\sin(\theta_e - \psi(y_e, v))\cos(\theta_e - \psi(y_e, v))}\cdot \left(-\sigma y_e v \sin\theta_e + \sigma y_e\, x_c\,\omega\cos\theta_e - k_2(\theta_e - \psi(y_e, v))^2\right) \tag{13}$$

where

$$\psi(y_e, v) = -sgn(v)\theta_a \tanh y_e \tag{14}$$

and k_1, k_2, σ and θ_a are parameters to the controller. For more details on the parameters, see Ref 15.

From the control law it is then possible to find vehicle steering input ω from the relation $\theta = \theta_e + \theta_r$ which gives $\dot{\theta} = \dot{\theta}_e + \dot{\theta}_r \rightarrow \underline{\omega = \dot{\theta}_e + \dot{s}c(s)}$. The controller is proven Lyapunov stable.[15]

Speed control Based on a measure of deviation (V) from the path, the speed controller sets an appropriate velocity. With low deviation, the speed is set high, and when the deviation is high, the speed is set low to correct the deviation.

Using $\underline{V = \frac{1}{2}(x_e^2 + y_e^2 + \frac{1}{\sigma}|\sin(\theta_e - \psi(y_e, v))|)}$

$$\omega \geq 0 \;:\; v = \begin{cases} -\frac{\alpha_r y_L V_m}{y_R - y_L} & V \geq \epsilon \\ \frac{\alpha_r V_m}{1+|y_R \cdot c|} & V < \epsilon \end{cases} \tag{15}$$

$$\omega < 0 \;:\; v = \begin{cases} \frac{\alpha_l y_R V_m}{y_R - y_L} & V \geq \epsilon \\ \frac{\alpha_l V_m}{1+|y_L \cdot c|} & V < \epsilon \end{cases} \tag{16}$$

where V_m is the max speed of the vehicle, σ_l and σ_r are track parameters and ϵ is the deviation threshold.

2.2.3 Estimating ICR-locations with an EKF

The ICR-locations needed for the controller are the lateral ICR-coordinate of the left track y_L, the lateral ICR-coordinate of the right track y_R and the longitudinal ICR-coordinate of the vehicle x_c. In order to estimate these using an EKF as in Ref. 12, they are added to the state-vector along with vehicle configuration. Using measurements of x, y, θ to update the estimate.

Kinematic equations Using the ICR kinematic model

$$\dot{x} = v_x \cos\theta - v_y \sin\theta \tag{17}$$
$$\dot{y} = v_x \sin\theta + v_y \cos\theta \tag{18}$$
$$\dot{\theta} = \omega \tag{19}$$

where

$$v_x = \frac{v_R y_L - v_L y_R}{y_L - y_R} \tag{20}$$
$$v_y = \frac{v_L - v_R}{y_L - y_R} x_c \tag{21}$$
$$\omega_z = -\frac{v_L - v_R}{y_L - y_R} \tag{22}$$

Process and measurement equation Discretized using 2nd order Runge-Kutta for configuration and Euler's method for ICR locations. The change of ICR locations are seen as pure gaussian noise processes.

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ y_{L_k} \\ y_{R_k} \\ x_{c_k} \end{bmatrix} = \begin{bmatrix} x_{k-1} + V_{x_{k-1}}\Delta t + \mu_1 \Delta t \\ y_{k-1} + V_{y_{k-1}}\Delta t + \mu_2 \Delta t \\ \theta_{k-1} + \omega_{k-1}\Delta t + \mu_3 \Delta t \\ y_{L_{k-1}} + \mu_4 \Delta t \\ y_{R_{k-1}} + \mu_5 \Delta t \\ x_{c_{k-1}} + \mu_6 \Delta t \end{bmatrix} \tag{23}$$

$$\theta_{rk} = \theta_{k-1} + \frac{\omega_{k-1}\Delta t}{2} \tag{24}$$
$$V_{x_{k-1}} = v_{x_{k-1}} \cos\theta_{rk} - v_{y_{k-1}} \sin\theta_{rk} \tag{25}$$
$$V_{y_{k-1}} = v_{x_{k-1}} \sin\theta_{rk} + v_{y_{k-1}} \cos\theta_{rk} \tag{26}$$

The filter uses measurements of x, y, θ from navigation system with white additive measurement noise w_{1-3}.

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} x_k + w_1 \\ y_k + w_2 \\ \theta_k + w_3 \end{bmatrix} \tag{27}$$

Using

$$\boldsymbol{\mathcal{F}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) = \begin{bmatrix} x_{k-1} + V_{x_{k-1}}\Delta t \\ y_{k-1} + V_{y_{k-1}}\Delta t \\ \theta_{k-1} + \omega_{k-1}\Delta t \\ y_{L_{k-1}} \\ y_{R_{k-1}} \\ x_{c_{k-1}} \end{bmatrix}, \boldsymbol{\Gamma} = \Delta t \cdot \mathbf{I}_{6\text{x}6}, \boldsymbol{\mu}_k = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_6 \end{bmatrix}$$
$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{w}_k = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Process and measurement equation can be written in canonical form as

$$\mathbf{x}_k = \boldsymbol{\mathcal{F}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \boldsymbol{\Gamma}\boldsymbol{\mu}_k \tag{28}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \tag{29}$$

$$\mathbf{x}_0 \sim \mathcal{N}(0, \bar{\mathbf{P}}_0)\ ,\ \boldsymbol{\mu}_k \sim \mathcal{N}(0, \mathbf{Q}\delta_{kl}),\ \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}\delta_{kl})$$

The initial state $\mathbf{x}_0$, process noise $\boldsymbol{\mu}_k$ and measurement noise $\mathbf{w}_k$ are assumed to be uncorrelated.

With discrete process and measurement equations, the equations for discrete Kalman Filter are used. Since the process equation contains a nonlinear term $\boldsymbol{\mathcal{F}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$, it is linearized around the current estimate at each time step by the Jacobian

$$\boldsymbol{\Phi}_{k-1} = \left.\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{x}^T}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{x}_{k-1}} \tag{30}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & 0 & \Phi_{13} & \Phi_{14} & \Phi_{15} & \Phi_{16} \\ 0 & 1 & \Phi_{23} & \Phi_{24} & \Phi_{25} & \Phi_{26} \\ 0 & 0 & 1 & \Phi_{34} & \Phi_{35} & \Phi_{36} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

$$\Phi_{13} = (-v_x \sin\theta - v_y \cos\theta)\Delta t,\ \Phi_{23} = (v_x \cos\theta - v_y \sin\theta)\Delta t \tag{32}$$

$$\Phi_{14} = \Big(\frac{v_L y_R - v_R y_R}{(y_L - y_R)^2}\cos\theta + \frac{v_L - v_R}{(y_L - y_R)^2} x_c \sin\theta\Big)\Delta t \tag{33}$$

$$\Phi_{15} = \Big(\frac{v_R y_L - v_L y_L}{(y_L - y_R)^2}\cos\theta - \frac{v_L - v_R}{(y_L - y_R)^2} x_c \sin\theta\Big)\Delta t \tag{34}$$

$$\Phi_{16} = \Big(-\frac{v_L - v_R}{y_L - y_R}\sin\theta\Big)\Delta t \tag{35}$$

$$\Phi_{24} = \Big(\frac{v_L y_R - v_R y_R}{(y_L - y_R)^2}\sin\theta - \frac{v_L - v_R}{(y_L - y_R)^2} x_c \cos\theta\Big)\Delta t \tag{36}$$

$$\Phi_{25} = \Big(\frac{v_R y_L - v_L y_L}{(y_L - y_R)^2}\sin\theta + \frac{v_L - v_R}{(y_L - y_R)^2} x_c \cos\theta\Big)\Delta t \tag{37}$$

$$\Phi_{26} = \Big(\frac{v_L - v_R}{y_L - y_R}\cos\theta\Big)\Delta t \tag{38}$$

$$\Phi_{34} = \frac{v_L - v_R}{(y_L - y_R)^2}\Delta t,\ \Phi_{35} = -\frac{v_L - v_R}{(y_L - y_R)^2}\Delta t,\ \Phi_{36} = 0 \tag{39}$$

Prediction State transition with process noise zero

$$\bar{\mathbf{x}}_k = \boldsymbol{\mathcal{F}}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \tag{40}$$

Error covariance prediction

$$\bar{\mathbf{P}}_k = \boldsymbol{\Phi}_{k-1}\hat{\mathbf{P}}_{k-1}\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{\Gamma}_{k-1}\mathbf{Q}_{k-1}\boldsymbol{\Gamma}_{k-1}^T \tag{41}$$

Measurement update For the measurement update

$$\mathbf{K}_k = \bar{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{42}$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\bar{\mathbf{x}}_k) \tag{43}$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\bar{\mathbf{P}}_k \tag{44}$$

2.2.4 ADRC

The control law is implemented as in Ref. 23 where internal and external disturbances are modelled as scaling and shifting of the unicycle kinematic model with parameters α_{1-3} and β_{1-3}

$$\dot{x} = \alpha_1 v \cos\theta + \beta_1 \tag{45}$$
$$\dot{y} = \alpha_2 v \sin\theta + \beta_2 \tag{46}$$
$$\dot{\theta} = \alpha_3 \omega + \beta_3 \tag{47}$$

Expressed in matrix form as

$$\dot{\mathbf{X}} = \mathbf{ACU} + \mathbf{B} \tag{48}$$

where

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{U} = \begin{bmatrix} v \\ \omega \end{bmatrix}$$
$$\mathbf{A} = \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

Later in this chapter, it will be explained how estimates of the disturbances $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ can be obtained using an EKF. The idea is to control the system using the ideal unicycle kinematic model

$$\dot{\mathbf{X}} = \mathbf{CU} \tag{49}$$

with the control law currently used on the vehicle,[1] and then to compensate for the disturbances $\mathbf{A}$ and $\mathbf{B}$. This is done by transforming the extended model into an ideal model by equating them and choosing the compensating control law $\mathbf{U}'$.

$$\begin{aligned} \hat{\mathbf{A}}\mathbf{CU}' + \hat{\mathbf{B}} &= \mathbf{CU} \\ \hat{\mathbf{A}}\mathbf{CU}' &= \mathbf{CU} - \hat{\mathbf{B}} \\ \mathbf{U}' &= \mathbf{C}^{-1}\hat{\mathbf{A}}^{-1}(\mathbf{CU} - \hat{\mathbf{B}}) \end{aligned}$$

It is now possible to use a control law $\mathbf{U}$ for an ideal unicycle kinematic model and then compensate for the estimated disturbances using $\underline{\mathbf{U}' = \mathbf{C}^{-1}\hat{\mathbf{A}}^{-1}(\mathbf{CU} - \hat{\mathbf{B}})}$ which will transform the system to an ideal system. The accuracy of this transformation is highly dependent on meaningful estimates $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$.

An important note is the inversion of matrix $\mathbf{C}$ which is not a square matrix. Therefore, the implementation uses a Moore-Penrose pseudoinverse[25] $\underline{\mathbf{C}^{-1} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T}$. In this case $\mathbf{C}^{-1} = \mathbf{C}^T$.

2.2.5 Estimating disturbances using EKF

In order to estimate the disturbances α_{1-3} and β_{1-3}, they are viewed as added states to the system along with their velocities $\dot{\alpha}_{1-3}$ and $\dot{\beta}_{1-3}$. Using position and orientation measurements from the vehicles navigation system to estimate the parameters with an EKF.

Process and measurement equation Discretized using 2nd order Runge Kutta for configuration and Euler's method for disturbances (with both the disturbances α_i, β_i and their velocities $\dot{\alpha}_i, \dot{\beta}_i$ in the state vector). Where $\mu_1 - \mu_{15}$ are additive white gaussian process noise.

$$x_k = x_{k-1} + (\alpha_{1,k-1} v_{k-1} \cos\theta_{rk} + \beta_{1,k-1})\Delta t + \mu_1 \Delta t \tag{50}$$
$$y_k = y_{k-1} + (\alpha_{2,k-1} v_{k-1} \sin\theta_{rk} + \beta_{2,k-1})\Delta t + \mu_2 \Delta t \tag{51}$$
$$\theta_k = \theta_{k-1} + (\alpha_{3,k-1}\omega_{k-1} + \beta_{3,k-1})\Delta t + \mu_3 \Delta t \tag{52}$$
$$\alpha_{i,k} = \alpha_{i,k-1} + \dot{\alpha}_{i,k-1}\Delta t + \mu_{3+i}\Delta t \tag{53}$$
$$\beta i,k = \beta_{i,k-1} + \dot{\beta}_{i,k-1}\Delta t + \mu_{6+i}\Delta t \tag{54}$$
$$\dot{\alpha}_{i,k} = \dot{\alpha}_{i,k-1} + \mu_{9+i}\Delta t \tag{55}$$
$$\dot{\beta}_{i,k} = \dot{\beta}_{i,k-1} + \mu_{12+i}\Delta t \tag{56}$$

where

$$\theta_{rk} = \theta_{k-1} + \frac{\omega_{k-1}\Delta t}{2} \tag{57}$$

Discrete measurements of x, y and θ from navigation system

$$z_1 = x_k + w_1 \tag{58}$$
$$z_2 = y_k + w_2 \tag{59}$$
$$z_3 = \theta_k + w_3 \tag{60}$$

where w_1, w_2, w_3 are additive white gaussian measurement noise.
Using

$$\boldsymbol{\mathcal{F}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) = \begin{bmatrix} x_{k-1} + (\alpha_{1,k-1} v_{k-1} \cos\theta_{k-1} + \beta_{1,k-1})\Delta t \\ y_{k-1} + (\alpha_{2,k-1} v_{k-1} \sin\theta_{k-1} + \beta_{2,k-1})\Delta t \\ \theta_{k-1} + (\alpha_{3,k-1}\omega_{k-1} + \beta_{3,k-1})\Delta t \\ \alpha_{i,k-1} + \dot{\alpha}_{i,k-1}\Delta t \\ \beta_{i,k-1} + \dot{\beta}_{i,k-1}\Delta t \\ \dot{\alpha}_{i,k-1} \\ \dot{\beta}_{i,k-1} \end{bmatrix} \tag{61}$$

$$\boldsymbol{\Gamma} = \Delta t \cdot \mathbf{I}_{15\text{x}15} \ , \ \boldsymbol{\mu}_k = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{15} \end{bmatrix}$$

and

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{3\text{x}15}, \mathbf{w}_k = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

the process and measurement equations can be written in canonical form as

$$\mathbf{x}_k = \boldsymbol{\mathcal{F}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \boldsymbol{\Gamma}\boldsymbol{\mu}_{k-1} \tag{62}$$
$$\mathbf{z}_k = \mathbf{H}x_k + \mathbf{w}_k \tag{63}$$
$$\mathbf{x}_0 \sim \mathcal{N}(0, \bar{\mathbf{P}}_0) \ , \ \boldsymbol{\mu}_k \sim \mathcal{N}(0, \mathbf{Q}\delta_{kl}) \ , \ \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}\delta_{kl}) \tag{64}$$

where the initial state $\mathbf{x}_0$, process noise $\boldsymbol{\mu}_k$ and measurement noise $\mathbf{w}_k$ are assumed to be uncorrelated.

In order to use EKF, the nonlinear function $\boldsymbol{\mathcal{F}}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ is linearized around the current estimate at each time step using the Jacobian

$$\boldsymbol{\Phi}_{k-1} = \left.\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{x}^T}\right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}} \tag{65}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & a & b & 0 & 0 & \Delta t & 0 & 0 & \cdots & 0 \\ 0 & 1 & c & 0 & d & 0 & 0 & \Delta t & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \omega\Delta t & 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \Delta t \\ & & & & & & \ddots & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{15\text{x}15} \tag{66}$$

$$a = -\alpha_1 v \sin(\theta)\Delta t,\ b = v\cos(\theta)\Delta t,\ c = \alpha_2 v\cos(\theta)\Delta t,\ d = v\sin(\theta)\Delta t$$

The prediction and measurement update equations can now be written for a discrete EKF.

Prediction State transition with process noise zero

$$\bar{\mathbf{x}}_k = \boldsymbol{\mathcal{F}}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \tag{67}$$

Error covariance prediction

$$\bar{\mathbf{P}}_k = \mathbf{\Phi}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{\Gamma}_{k-1}\mathbf{Q}_{k-1}\mathbf{\Gamma}_{k-1}^T \tag{68}$$

Measurement update For the measurement update

$$\mathbf{K}_k = \bar{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{69}$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\bar{\mathbf{x}}_k) \tag{70}$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\bar{\mathbf{P}}_k \tag{71}$$

Resulting in estimate $\hat{\mathbf{x}}_k$ of the vehicle's configuration and disturbances α_{1-3} and β_{1-3}.

3. EXPERIMENTS

The controllers were tested in both simulations and real world experiments to investigate how they are affected by off-road terrain and how their performance compares. Each controller and EKF was tuned to produce optimal path following in ideal conditions. Then, the tuning parameters were held constant throughout the rest of the experiments, thus testing how each controller can account for dynamic variations of the terrain without needing to re-tune its parameters. The parameters used in the experiments can be found in Appendix A.

Relevant data from the experiments were acquired and stored, including navigation data, controller output, EKF estimates and more. The central metrics used for comparison were the cross-track error (y_e) and the orientation error (θ_e), each expressed in the reference vehicle frame. Some of the experiments were repeated several times to calculate the mean and variance of the performance over several experiments, which are represented by the mean Root Mean Square (mRMS) and mean Standard Deviation (mSTD) of cross-track error and orientation error. mRMS calculates the root mean square over k runs in each point on the vehicle's path and takes the mean over time. Thus giving a measure of the error the controller produces over several runs along the path. mSTD calculates the variation in each point over k runs and takes the mean over all points. Resulting in a measure of how the error varies over several runs and whether the behaviour is predictable. Equation 72 show the formulas used for cross-track error, where n is the number of points on the vehicle's path (for orientation error, y_e is changed to θ_e).

$$mRMS = \frac{1}{n}\sum_{j=1}^{n}\sqrt{\frac{1}{k}\sum_{i=1}^{k} y_e(i,j)^2} \quad \bar{y}_e(j) = \frac{1}{k}\sum_{i=1}^{k} y_e(i,j) \quad mSTD = \frac{1}{n}\sum_{j=1}^{n}\sqrt{\frac{1}{k}\sum_{i=1}^{k}(y_e(i,j) - \bar{y}_e(j))^2} \tag{72}$$

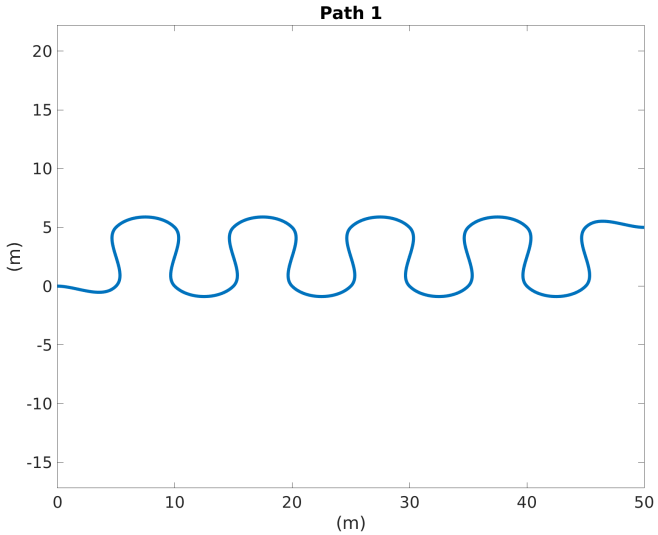


Figure 3: Path used in simulations

3.1 Simulations

Simulations were carried out using the ICR simulation model described in Section 2.1 and implemented in MATLAB.[26] The simulation model makes it possible to vary the vehicle's rotation centres to simulate effects from driving in off-road conditions. Shifting the rotation centres laterally and longitudinally can simulate longitudinal and lateral slip of the vehicle respectively, where the simulation consists of several combinations of moving the rotation centres. Figure 3 displays the path used in simulations, which is a path with large curvatures, thus increasing the difficulty for each controller and giving the EKFs relevant terrain data for the estimates.

	y_L	y_R	x_c
Ideal	1.0 m	−1.0 m	0.0 m
Outward	5.0 m	−5.0 m	0.0 m
Forward	1.0 m	−1.0 m	0.3 m
Combined	5.0 m	−3.0 m	−0.35 m

Table 2: ICR locations used in simulations

All simulated test conditions are listed in Table 2. The ICR of each track is shifted laterally and longitudinally to simulate different effects from off-road driving. For the outward shift, the vehicle behaves as if the distance between the tracks is increased from L = 2.0 m to L = 10.0 m, which is a quite large increase, and was chosen to clearly illustrate the difference between the controllers.

3.2 Real world experiments

For the real world experiments, each controller was implemented on the UGV, which is interfaced using ROS2 (more details on implementation for the UGV in Ref. 1). Testing was performed outdoors in an uneven terrain consisting of mostly sand (see Figure 4), where three different experiments were conducted. First, each controller was driven along a 30 m long straight path with an initial lateral offset of 2.0 m to obtain an approximate step response of each controller. Second, each controller was again driven along a straight path, but with a lower initial offset of 1.0 m and repeated five times in order to calculate root mean square and standard deviation. Last, the vehicle was driven along a curved path to investigate how the controllers perform on a more demanding path.

Figure 4: Terrain condition when performing real world experiments

	Ideal		Outward		Forward		Combined	
	mRMS	**mSTD**	**mRMS**	**mSTD**	**mRMS**	**mSTD**	**mRMS**	**mSTD**
Morin	0.013 m	≈ 0 m	0.317 m	≈ 0 m	0.092 m	≈ 0 m	0.191 m	≈ 0 m
ADRC	0.016 m	≈ 0 m	0.022 m	≈ 0 m	0.106 m	≈ 0 m	0.097 m	0.055 m
HBZ	0.031 m	≈ 0 m	0.19 m	≈ 0 m	0.129 m	≈ 0 m	0.112 m	≈ 0 m

Table 3: mRMS for cross-track error over $k = 5$ simulations

4. RESULTS

4.1 Simulations

The evolution of cross-track error for all simulated conditions can be seen in Figure 5. In all of the conditions, it is evident that HBZ uses longer time to complete the path. This is caused by its separate speed control, which lowers the speed control output when the vehicle has error in position and orientation.

Table 3 presents performance metrics with mRMS and mSTD of cross-track error for all conditions over 5 simulations each. For most of the conditions, the mSTD was close to zero which indicates low variance in performance between simulation runs. This predictable behaviour was expected as the simulation model is completely deterministic.

4.2 Real world experiments

4.2.1 Straight path

Results from driving a straight path with an initial lateral offset of 2.0 m can be seen in Figure 6, with related estimates from ADRC and HBZ displayed in Figure 7 and 8 respectively. In Table 4, metrics for driving a straight path with an initial offset of 1.0 m is shown.

4.2.2 Curved path

A visualisation of the controllers driving along the curved path is displayed in Figure 9, with the evolution of the error in Figure 10. Table 5 shows performance metrics when repeating the curved path $k = 3$ times. As the controllers did not start in the same initial position for the curved path, calculation of the metrics (mRMS and mSTD) was done from $x = -20.0$, which is approximately when all controllers have corrected for the initial offset from the path.

	Straight path			
	Cross-track error		Orientation error	
	mRMS	mSTD	mRMS	mSTD
Morin	0.303m	0.149 m	0.047 rad	0.033 rad
ADRC	0.255m	0.121 m	0.030 rad	0.019 rad
HBZ	0.189m	0.061 m	0.060 rad	0.023 rad

Table 4: Metrics for $k = 5$ runs on the straight path (1.0 m offset)

	Curved path			
	Cross-track error		Orientation error	
	mRMS	mSTD	mRMS	mSTD
Morin	0.230 m	0.045 m	0.122 rad	0.026 rad
ADRC	0.173 m	0.076 m	0.100 rad	0.053 rad
HBZ	0.195 m	0.046 m	0.100 rad	0.030 rad

Table 5: Metrics for $k = 3$ runs on the curved path

5. DISCUSSION

5.1 Simulations

Ideal conditions The lateral offset of all controllers quickly goes to zero and remains low throughout the path, and looking at the mRMS in Table 3, it remains low for all controllers. This was expected as the simulated vehicle is not experiencing any disturbances from the environment, thus confirming the proposed controllers' ability to follow paths in ideal conditions.

Outward shift When the ICR was shifted outward, Morin was considerably affected with deviations from the path when turning, where the cross-track error peaks at around 50 cm. The mRMS was also the highest with 0.317 m. Both ADRC and HBZ improved the performance compared to Morin, with lower deviations while turning and lower mRMS. ADRC clearly had the best performance, with minimal deviations from the path while turning and mRMS of 0.022 m. HBZ suppressed the peak deviations to around 20 cm and had a mRMS of 0.19 m, which is still a considerable improvement over Morin.

Forward shift Results from the forward shift show similar performance of all the controllers. The cross-track error peaks at around 20 cm for all controllers while turning and the mRMS is around 0.1 m for all the controllers, where Morin has the lowest with 0.091 m. In this condition, the proposed controllers were not able to improve the performance over Morin.

Combined shift Combined shift causes Morin to experience large deviations of about 60 cm, which occurs while turning left. The reason the deviation is largest while turning left is because the ICR of the left track moves further out than the ICR of the right track, making it harder to turn left. ADRC and HBZ both manage to suppress the large deviations experienced by Morin and end up with cross-track error peaking at about 20 cm, which is similar to the results from forward shift. Morin has the highest mRMS with 0.191 m while ADRC and HBZ has a mRMS of around 0.1 m which is also similar to the condition with forward shift. This can indicate that the proposed controllers are adequate at compensating for a lateral shift of the rotation centre while struggling to improve performance when the shift is longitudinal. Anyway, the presented results from outward shift and combined shift show the proposed controllers' ability to improve path following performance compared to Morin.

5.2 Real world experiments

Results from driving straight with 2.0 m offset show the error go to zero for all the controllers. Estimates of the ICR from HBZ in Figure 8 show that the rotation centre of each track stabilises at about $y_L = 1.3$ m and $y_R = -1.3$ m which means that the controller sees the vehicle as having a larger distance between its tracks - L = 2.6 m instead of L = 2.0 m. Which can be interpreted as it being harder to rotate the vehicle. A similar

interpretation can be gained from the disturbance estimates from ADRC in Figure 7, where a_3 stabilises at about 0.2, which means it interprets the rotational velocity of the vehicle as scaled by 0.2, thus making it harder to rotate.

When driving a straight path with 1.0 m initial offset, Table 4 shows ADRC and HBZ both having lower mRMS of cross-track error compared to Morin. HBZ has the lowest, with 0.189 m, it also has the lowest mSTD with 0.061 m which means cross-track error is more consistent and its behaviour more predictable when repeating the same path.

For the curved path, the controllers get some overshoot when going around the sharp corners and then corrects for the offset. After 3 repetitions Morin gets a mRMS of 0.230 m for cross-track error, while ADRC and HBZ both improve slightly with 0.173 m and 0.195 m respectively.

6. CONCLUSION

Three path following controllers for differential drive UGVs were investigated and evaluated to gain insight into their performance in off-road conditions. From simulations, it was shown that the proposed controllers were able to repress the large deviations during curves that Morin experiences when the ICR is shifted laterally, and most interesting, ADRC compensated almost completely for the outward shift and had minimal error. For the real world experiments, estimates from ADRC and HBZ indicated a terrain where the angular velocity of the vehicle is scaled down, thus making it harder to rotate, and both controllers had lower mRMS for cross-track error than Morin for straight and curved paths.

In conclusion, the results from this paper show the Morin controller as being robust and reliable in most conditions, but it is considerably affected by large shifts in the vehicle's rotation centre. In addition, the results show that ADRC and HBZ can improve path following performance against rough off-road effects where Morin's performance is degraded. They also had on average lower cross-track error in simulations and real world experiments. These results support the findings in Ref. 23, where ADRC improves path following compared to an ideal controller, and Ref. 15, where HBZ improves path following compared to two state-of-the-art controllers.

Future autonomous operation of UGVs can significantly improve robustness and performance by using ADRC or HBZ in environments where there is a large lateral ICR shift of the tracks. In less severe conditions, Morin will probably have decent performance. Further research should investigate what types of terrain are too rough for Morin and therefore justifies using ADRC or HBZ, and also investigate how the proposed controllers handle transitions between different terrain types.

REFERENCES

[1] Mathiassen, K., Baksaas, M., Græe, S. A., Mentzoni, E. A., and Nielsen, N. H., "Making the milrem themis UGV ready for autonomous operations," in [*Unmanned Systems Technology XXIII*], Muench, P. L., Nguyen, H. G., and Skibba, B. K., eds., SPIE (4 2021).

[2] Mathiassen, K., Hyndøy, J. I., Østevold, E., Valaker, S., Danielsen, T., Baksaas, M., Olsen, L. E., Thoresen, M., Ruud, E.-L., Selvåg, J., and Sandrib, J., "Base defence demonstration at trident juncture 2018 – tact unmanned systems for base and force protection," (2019). FFI-Report 19/00807, Norwegian Defence Research Establishment.

[3] Larsen, E., Mathiassen, K., Landmark, L., Larsen, M. V., and Øivind Kure, "Autonomous sensors and communications infrastructure," (2019). FFI-Report 19/00808, Norwegian Defence Research Establishment.

[4] González, R., Rodriguez, F., and Guzmán, J. L., [*Autonomous Tracked Robots in Planar Off-Road Conditions*], Springer (2014).

[5] Amer, N. H., Zamzuri, H., Hudha, K., and Kadir, Z. A., "Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges," **86** (2017).

[6] Gonzalez, R., Rodriguez, F., Guzman, J., and Berenguel, M., "Localization and control of tracked mobile robots under slip conditions," (2009).

[7] Wang, D. and Low, C. B., "Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective," **24**, 676–687 (2008).

[8] Moosavian, S. A. A. and Kalantari, A., "Experimental slip estimation for exact kinematics modeling and control of a tracked mobile robot," (2008).
[9] Taghia, J. and Katupitiya, J., [*Applied Guidance Methodologies for Off-road Vehicles*], Springer (2020).
[10] Reina, G., Ishigami, G., Nagatani, K., and Yoshida, K., "Vision-based estimation of slip angle for mobile robots and planetary rovers," (2008).
[11] Song, X., Seneviratne, L., and Althoefer, K., "A vision based wheel slip estimation technique for mining vehicles," **42**, 179–184 (2009).
[12] Pentzer, J., Brennan, S., and Reichard, K., "Model-based prediction of skid-steer robot kinematics using online estimation of track instantaneous centers of rotation," **31**, 455–476 (2014).
[13] Martínez, J. L., Mandow, A., Morales, J., Pedraza, S., and García-Cerezo, A., "Approximating kinematics for tracked mobile robots," **24**, 867–878 (2005).
[14] Pentzer, J., Brennan, S., and Reichard, K., "The use of unicycle robot control strategies for skid-steer robots through the icr kinematic mapping," (2014).
[15] Huskić, G., Huskić, G., Huskić, G., Lacroix, S., and Zell, A., "High-speed path following control of skid-steered vehicles," **38**, 1124–1148 (2019).
[16] Soetanto, D., Lapierre, L., and Pascoal, A., "Adaptive, non-singular path-following control of dynamic wheeled robots," (2003).
[17] Ziye, Z., Haiou, L., Huiyan, C., Jiaming, H., and Hongming, G., "Kinematics-aware model predictive control for autonomous high-speed tracked vehicles under the off-road conditions," **123**, 333–350 (2019).
[18] Rajagopalan, V., Çetin Meriçli, and Kelly, A., "Slip-aware model predictive optimal control for path following," (2016).
[19] Álvaro Javier Prado, Torres-Torriti, M., Yuz, J., and Cheeina, F. A., "Tube-based nonlinear model predictive control for autonomous skid-steer mobile robots with tire–terrain interactions," **101** (2020).
[20] González, R., Fiacchini, M., Guzmán, J. L., Álamo, T., and Rodríguez, F., "Robust tube-based predictive control for mobile robots in off-road conditions," **59**, 711–726 (2011).
[21] Han, W., Yuan, P., Chen, D., Li, Y., Lai, T., Lin, M., Shi, Z., and Wang, T., "A novel tracking control method based on ladrc for autonomous mobile robot," (2015).
[22] CHEN, S., XUE, W., LIN, Z., and HUANG, Y., "On active disturbance rejection control for path following of automated guided vehicle with uncertain velocities," (2019).
[23] Sebastian, B. and Ben-Tzvi, P., "Active disturbance rejection control for handling slip in tracked vehicle locomotion," **11** (2019).
[24] Siciliano, B. and Khatib, O., [*Springer Handbook of Robotics*], Springer (2008).
[25] Ben-Israel, A. and Greville, T. N., [*Generalized Inverses*], Springer (2003).
[26] MATLAB, "9.9.0.1592791 (r2020b)," (2020). The MathWorks Inc., Natick, Massachusetts.

APPENDIX A. TUNING

Simulation			**Real world experiment**		
$k_1 = 1.0$	$k_2 = 5.0$	$k_3 = 4.0$	$k_1 = 0.0$	$k_2 = 0.1$	$k_3 = 0.7$

Table 6: Tuning of the Morin controller

Simulation				**Real world experiment**			
$k_1 = 1.0$	$\sigma = 1.0$	$V_m = 5\text{m/s}$	$\alpha_l = 1.0$	$k_1 = 1.0$	$\sigma = 1.0$	$V_m = 1\text{m/s}$	$\alpha_l = 1.0$
$k_2 = 100.0$	$\theta_a = \frac{\pi}{4}$	$\epsilon = 0.15$	$\alpha_r = 1.0$	$k_2 = 1.2$	$\theta_a = \frac{\pi}{4}$	$\epsilon = 0.5$	$\alpha_r = 1.0$

Table 7: Tuning of the HBZ controller

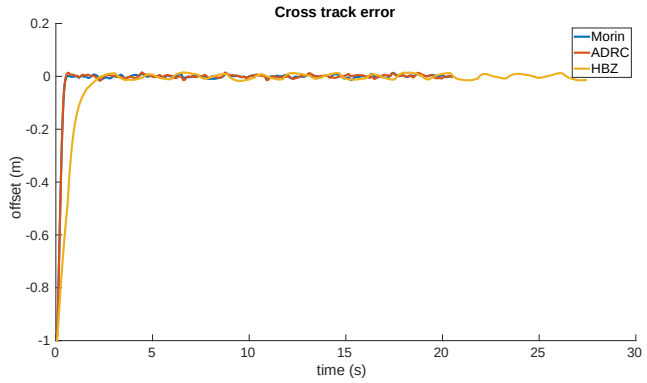
Simulation	Real world experiment
$\mathbf{R} = \text{diag}(0.01^2, 0.01^2, \left(\frac{\pi}{180}\right)^2)$	$\mathbf{R} = \text{diag}(0.01^2, 0.01^2, \left(\frac{\pi}{180}\right)^2)$
$\hat{\mathbf{P}}_0(1,1 \to 3,3) = \text{diag}(0.01^2, 0.01^2, \left(\frac{\pi}{180}\right)^2)$ $\hat{\mathbf{P}}_0(4,4 \to 6,6) = \text{diag}(0.01^2, 0.01^2, 0.01^2)$	$\hat{\mathbf{P}}_0(1,1 \to 3,3) = \text{diag}(1.0^2, 1.0^2, \left(\frac{\pi}{2}\right)^2)$ $\hat{\mathbf{P}}_0(4,4 \to 6,6) = \text{diag}(0.1^2, 0.1^2, 0.1^2)$
$\mathbf{Q}(1,1 \to 3,3) = \text{diag}(0.1^2, 0.1^2, \left(\frac{5\pi}{180}\right)^2)$ $\mathbf{Q}(4,4 \to 6,6) = \text{diag}(0.1^2, 0.1^2, 0.1^2)$	$\mathbf{Q}(1,1 \to 3,3) = \text{diag}(0.5^2, 0.5^2, \left(\frac{15\pi}{180}\right)^2)$ $\mathbf{Q}(4,4 \to 6,6) = \text{diag}(0.01^2, 0.01^2, 0.01^2)$

Table 8: Tuning of the EKF used by HBZ

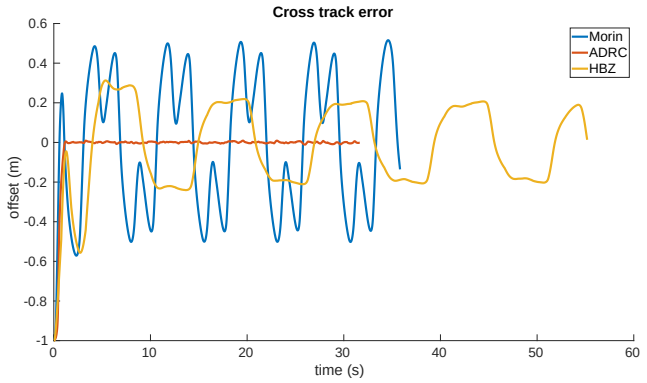
Simulation	Real world experiment
$\mathbf{R} = \text{diag}(0.01^2, 0.01^2, \left(\frac{\pi}{180}\right)^2)$	$\mathbf{R} = \text{diag}(0.01^2, 0.01^2, \left(\frac{\pi}{180}\right)^2)$
$\hat{\mathbf{P}}_0(1,1 \to 3,3) = \text{diag}(0.01^2, 0.01^2, \left(\frac{\pi}{180}\right)^2)$ $\hat{\mathbf{P}}_0(4,4 \to 15,15) = \text{diag}(0.1^2, 0.1^2, ..., 0.1^2)$	$\hat{\mathbf{P}}_0(1,1 \to 3,3) = \text{diag}(1.0^2, 1.0^2, \left(\frac{\pi}{2}\right)^2)$ $\hat{\mathbf{P}}_0(4,4 \to 5,5) = \text{diag}(0.1^2, 0.1^2)$ $\hat{\mathbf{P}}_0(6,6) = 0.2^2$ $\hat{\mathbf{P}}_0(7,7 \to 15,15) = \text{diag}(0.01^2, 0.01^2, ..., 0.01^2)$
$\mathbf{Q}(1,1 \to 3,3) = \text{diag}(0.1^2, 0.1^2, \left(\frac{5\pi}{180}\right)^2)$ $\mathbf{Q}(4,4 \to 15,15) = \text{diag}(0.1^2, 0.1^2, ..., 0.1^2)$	$\mathbf{Q}(1,1 \to 3,3) = \text{diag}(0.5^2, 0.5^2, \left(\frac{5\pi}{180}\right)^2)$ $\mathbf{Q}(4,4 \to 5,5) = \text{diag}(0.01^2, 0.01^2)$ $\mathbf{Q}(6,6) = \left(\frac{5\pi}{180}\right)^2$ $\mathbf{Q}(7,7 \to 15,15) = \text{diag}(0.01^2, 0.01^2, ..., 0.01^2)$

Table 9: Tuning of the EKF used by ADRC

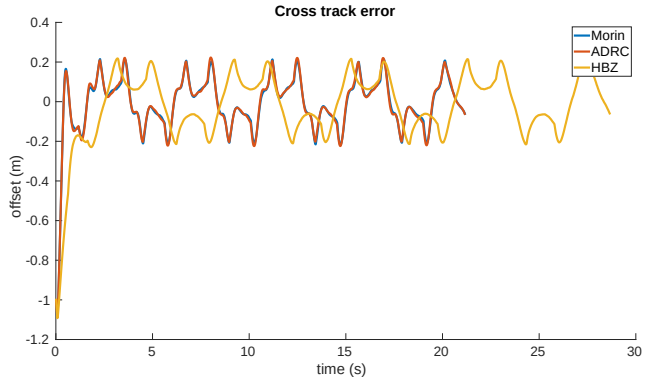
APPENDIX B. RESULT FIGURES



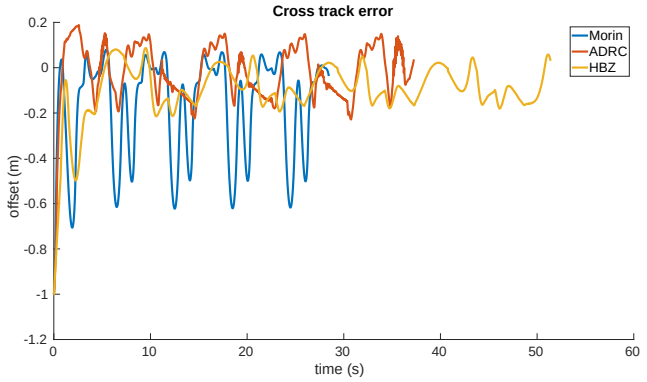
(a) Ideal conditions



(b) Outward ICR-shift

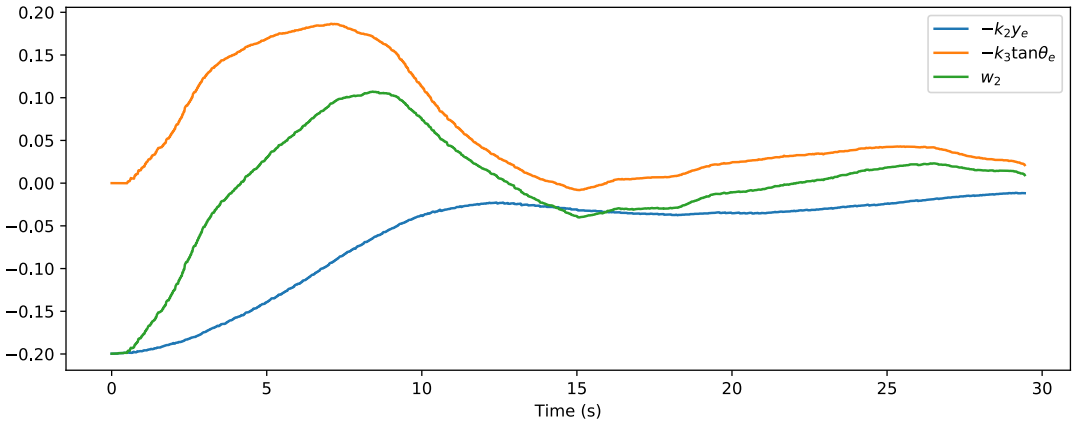


(c) Forward ICR-shift

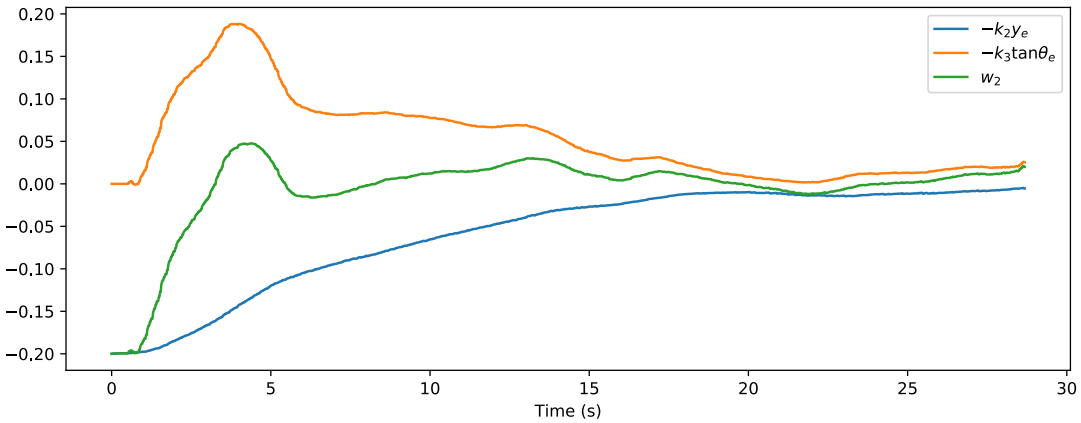


(d) Combined ICR-shift

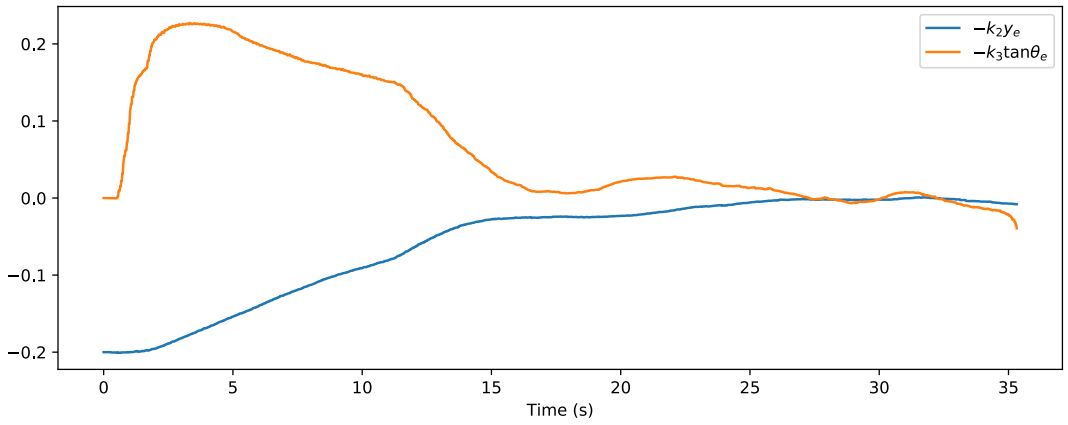
Figure 5: Cross track error from simulations



(a) Morin



(b) ADRC



(c) HBZ

Figure 6: Real world experiment results from driving a straight path with 2.0 m initial lateral offset

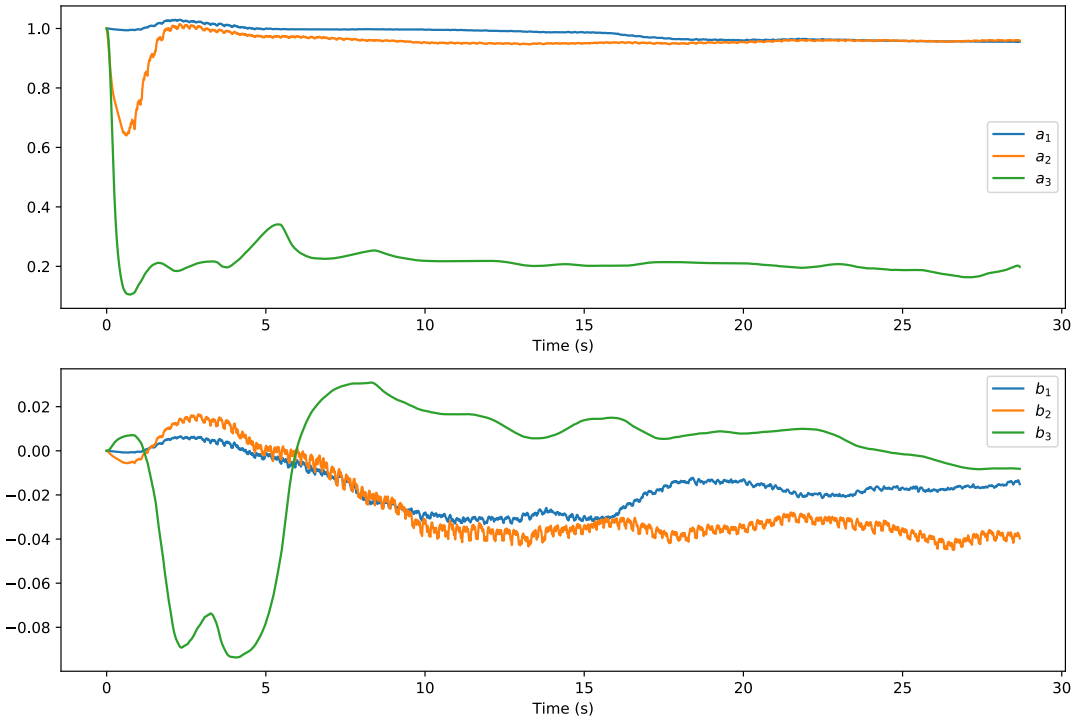


Figure 7: Disturbance estimates used by ADRC when driving a straight path with 2.0 m initial lateral offset

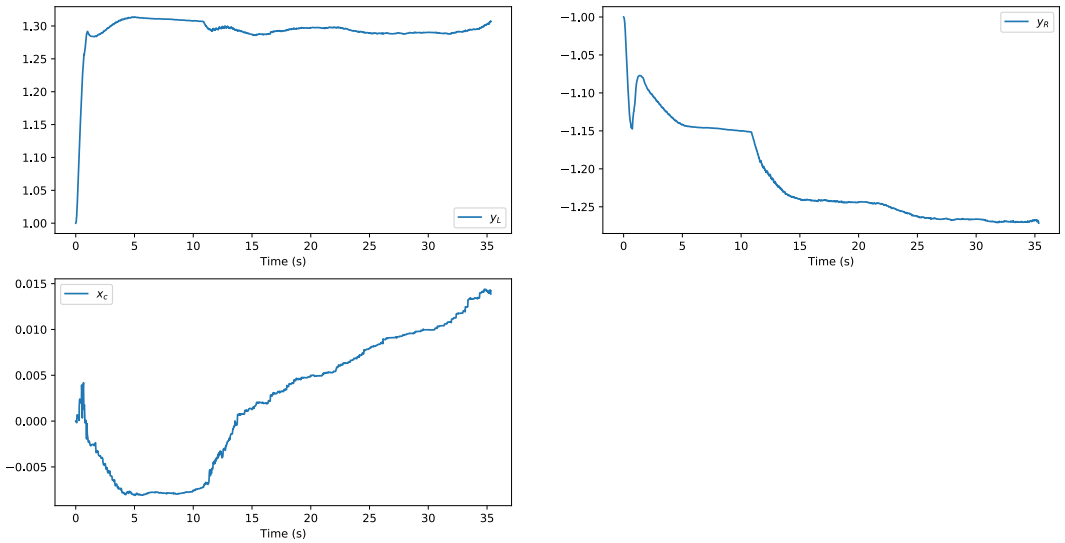


Figure 8: ICR estimates used by HBZ when driving a straight path with 2.0 m initial lateral offset

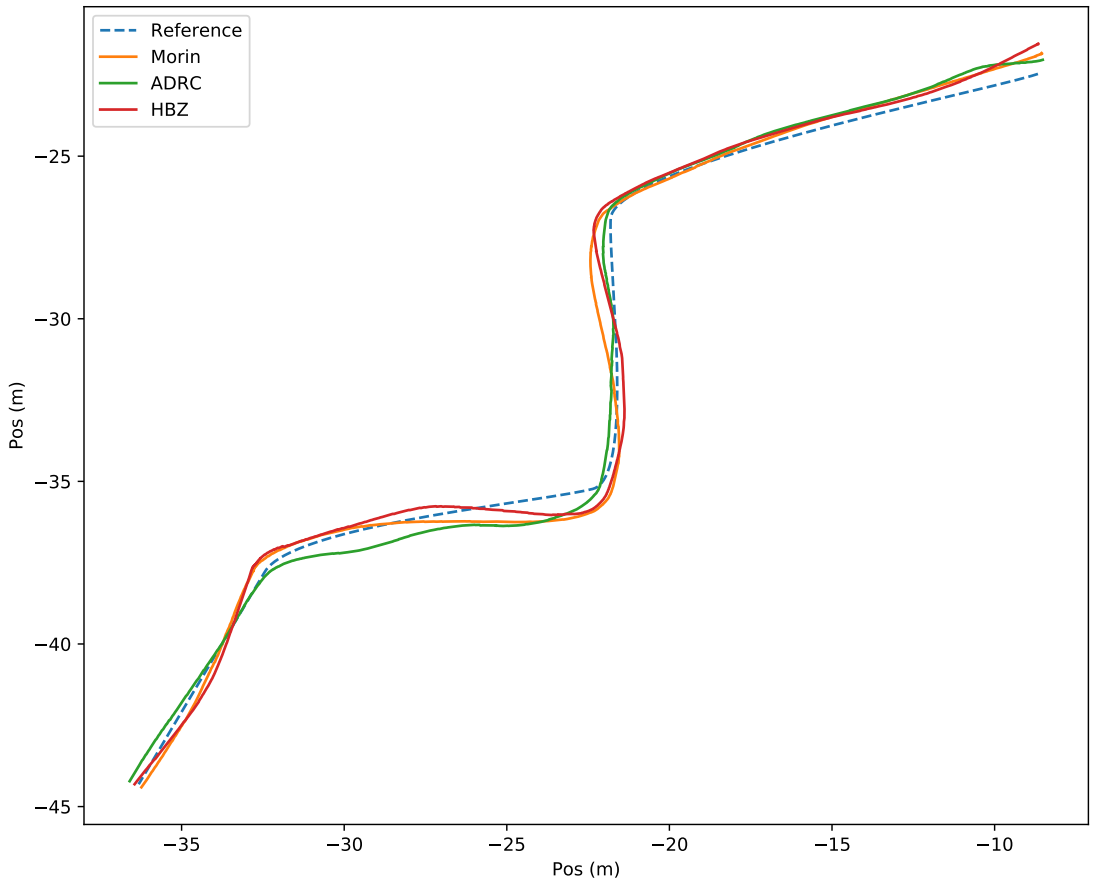
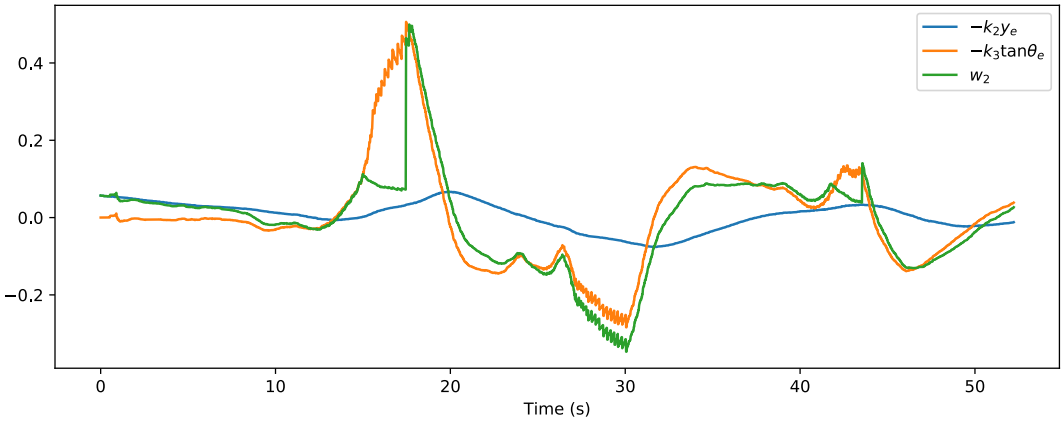
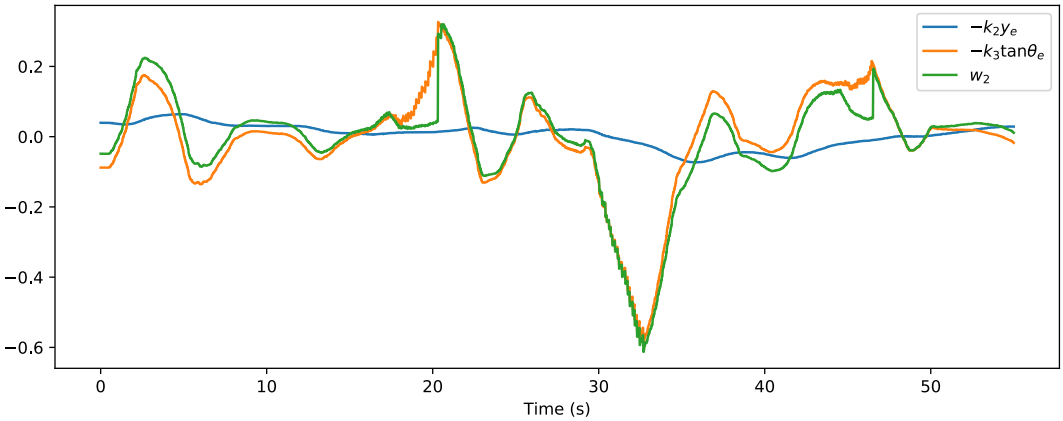


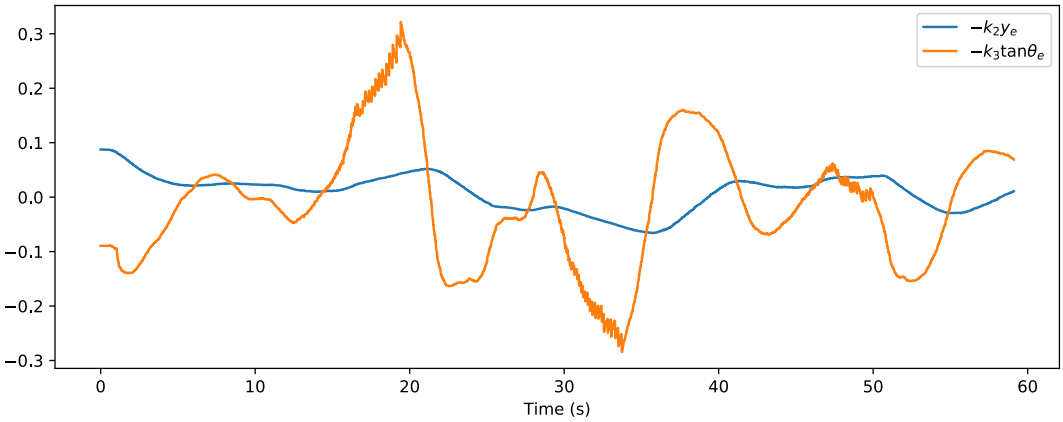
Figure 9: Real world experiment result from driving a curved path



(a) Morin



(b) ADRC



(c) HBZ

Figure 10: Real world experiment results from driving a curved path