Active Disturbance Rejection Control for Handling Slip in Tracked Vehicle Locomotion

Bijo Sebastian
Mem. ASME
Robotics and Mechatronics Laboratory,
Mechanical Engineering Department,
Virginia Tech,
Blacksburg, VA 24060
e-mail: bijo7@vt.edu

Pinhas Ben-Tzvi
Mem. ASME
Robotics and Mechatronics Laboratory,
Mechanical Engineering Department,
Virginia Tech,
Blacksburg, VA 24060
e-mail: bentzvi@vt.edu

This paper describes the use of an active disturbance rejection controller (ADRC) to estimate and compensate for the effect of slip in an online manner to improve the path tracking performance of autonomous ground vehicles (AGVs). AGVs with skid-steer locomotion mode are extensively used for robotic applications in the fields of agriculture, transportation, construction, warehouse maintenance, and mining. Majority of these applications such as performing reconnaissance and rescue operations in rough terrain or autonomous package delivery in urban scenarios, require the system to follow a path predetermined by a high-level planner or based on a predefined task. In the absence of effective slip estimation and compensation, the AGVs, especially tracked vehicles, can fail to follow the path as given out by the high-level planner. The proposed ADRC architecture uses a generic mathematical model that can account for the scaling and shift in the states of the system due to the effects of slip through augmented parameters. An extended Kalman filter (EKF) observer is used to estimate the varying slip parameters online. The estimated parameters are then used to compensate for the effects of slip at each iteration by modifying the control actions given by a low-level path tracking controller. The proposed approach is validated through experiments over flat and uneven terrain conditions including asphalt, vinyl flooring, artificial turf, grass, and gravel using a tracked skid-steer mobile robot. A detailed discussion on the results and directions for future research is also presented. [DOI: 10.1115/1.4042347]

1 Introduction and Motivation

The first notable tracked locomotion system was the Holt tractor [1] used during World War I by the British, French, and American armies to move heavy equipment. Since then considerable advances have been made in the design and control of tracked locomotion modules. Compared to other locomotion modes, including legged or wheeled platforms, tracked locomotion provides much larger ground contact resulting in increased traction, reduced overall pressure on the ground, and terrain adaptability. These factors make tracked locomotion particularly useful for navigation over challenging and unstructured terrains found in agricultural, transportation, construction, and mining applications. These strengths also promote wide spread use in mobile robotic applications, such as search and rescue [2–5].

A majority of the applications that involve autonomous ground vehicles (AGVs), such as performing reconnaissance and rescue operations in rough terrain or autonomous package deliveries in an urban setting, require the system to follow a path predetermined by a high-level planner or based on a predefined task [6–8]. The path, usually defined by series of waypoints, will be taken as an input to a low-level path following controller that then generates the control inputs for the robot. For applications involving differential drive or skid-steer robots, the low-level controller most often leverages a unicycle model, as given in Eq. (1), to drive the robot along the desired path. Since the control inputs for the unicycle model are the linear (V) and angular (ω) velocity of the robot, it is easier to come up with reliable and intuitive control laws using this model as compared to the differential drive or skid-steer model. The unicycle model is given in the following

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{1}$$

where (x, y) is the two-dimensional position robot fixed frame $\{R\}$ with respect to the global inertial frame $\{G\}$ and θ represents the orientation $\{R\}$ with respect to $\{G\}$ about the Z-axis, as shown in Fig. 1. The aforementioned model can be summarized into the following form:

$$\dot{\boldsymbol{X}} = \boldsymbol{C}\boldsymbol{U}$$
$$\text{where,} \quad \boldsymbol{X} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} V \\ \omega \end{bmatrix} \tag{2}$$

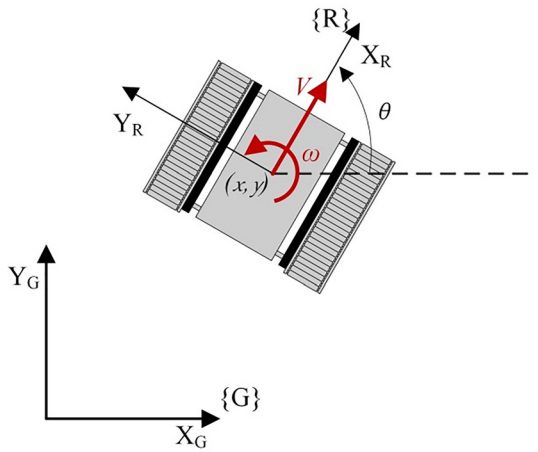

Fig. 1 Schematic representation of the kinematic model

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the Journal of Mechanisms and Robotics. Manuscript received August 3, 2018; final manuscript received December 9, 2018; published online February 27, 2019. Assoc. Editor: David J. Cappelleri.

Once the control inputs are determined based on the states (x, y, θ) of the robot, they are transformed into the left and right wheel angular velocity (ω_l, ω_r) before they can be applied on the differential drive robot, using the following equation:

$$\begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} = \begin{bmatrix} \dfrac{2V - \omega L}{D} \\ \dfrac{2V + \omega L}{D} \end{bmatrix} \tag{3}$$

where D and L are the wheel diameter and the width of the robot (axle track), respectively. The above equations are valid for a differential drive robot with two narrow wheels on both sides and a castor wheel at the base, such as the Pioneer P3DX platform, moving at a low speed without slip. For such a system, the two wheels can be assumed to make point contact with the ground and the axle track of the platform can be measured as the distance between the point contacts.

The increased ground contact offered by tracked locomotion mode becomes a disadvantage in this case. Unlike wheeled or legged locomotion, which can assume no slip under ideal conditions, the tracked locomotion requires a skidding turn, where a large portion of the track must slide against the terrain. Therefore, tracked locomotion is also called slip/skid-steer locomotion. The large amount of skidding involved during a turn makes it difficult to accurately estimate the center of rotation of the robot and thereby the future position and orientation of the robot for given left and right track velocities [6,7]. Moreover, the motor torque required to perform skid-steer varies greatly depending on the amount of friction on the terrain, geometry, material of track, etc. The aforementioned factors make it difficult to obtain a reliable model for tracked vehicle locomotion, which, in turn, affects the accuracy in position estimation, model predictive control, and motion planning. As such, model-based control techniques have limited performance when it comes to path following application for tracked robots on challenging terrain conditions. A detailed review of some of the existing methods and their limitations is given in Sec. 2. While the major factors that affect the degree of slip experienced by a system can be broadly generalized as the weight of the vehicle, nature of terrain, wheel velocity, material of wheel, etc., the exact relationship between these parameters and their effect on the motion of the robot is still unknown.

In this paper, we propose an active disturbance rejection-based method to handle the effect of slip in an online manner. The proposed approach uses an extended Kalman filter (EKF)-based observer to estimate the varying slip parameters online. The estimated parameters are then used to compensate for the effects of slip at each iteration by modifying the control actions of the low-level controller. The proposed approach is validated through experiments over the flat and uneven terrain conditions of asphalt, vinyl flooring, artificial turf, grass, and gravel using the tracked skid-steer mobile robot, STORM [9]. The remainder of this work is arranged as follows: Sec. 2 provides a brief review of existing state of the art methods, explaining limitations and persisting challenges; Sec. 3 outlines the proposed approach and how it addresses the requirements for efficient path tracking; Sec. 4 describes the experimental setup; Sec. 5 provides a detailed description of the results and the inferences drawn therein; Sec. 6 concludes the paper with directions for further research.

2 Literature Review

In the case of skid-steer robots, especially tracked platforms, the assumptions of no-slip and point contact with the ground are not valid. This, in turn, causes the linear and angular velocities, V and ω, experienced by the robot to be significantly different from the values expected by the closed loop controller. This effect, which can be represented as noise in the actuator, becomes particularly significant when the robot moves over challenging terrain. Depending on the size and inertia of the robot and the type of terrain, this can lead to cases where the effect leads to inefficient navigation profiles, jerky motion, and may even be significant enough to prevent the robot from reaching the desired goal. A recent review by Gonzalez and Iagnemma [10] states that reliable slip estimation and compensation strategies play a major role in enabling safe and efficient navigation. Even though the review focuses mainly on extraterrestrial rovers, the limitations of current techniques and the persisting challenges as reviewed by the authors are relevant to autonomous ground vehicles in general.

In general, longitudinal slip is defined as the difference between the velocity measured at the wheel and the linear velocity at the center of the wheel [10]. The velocity measured at the wheel is given by Ωr, where Ω is the rotational speed and r is the wheel radius. This is usually the desired velocity from the controller perspective. Provided there is a way to estimate the linear velocity of the wheel's center, v_{linear}, the percentage of motion lost to slip, s, can be calculated as

$$s = 100 * \begin{cases} \dfrac{\Omega r - v_{\text{linear}}}{\Omega r}, (\text{driving}) \\ \dfrac{\Omega r - v_{\text{linear}}}{v_{\text{linear}}} (\text{braking}) \end{cases} \tag{4}$$

Earlier works to improve path following in AGV's assumed that once slip is estimated, it can be easily compensated for by adjusting the control efforts, V and ω, to account for the effect of slip while using Eqs. (2)–(4). With this assumption, the focus was directed toward estimating the exact amount of slip experienced by the robot through estimation of the linear and angular velocities of the robot. The major problem with this approach is that majority of the existing platforms use inertial measurement units, which directly measure the acceleration of the robot, and therefore provide noisy velocity estimates. In this regard, an adaptive approach toward slip estimation was presented by Burke [11], to estimate slip solely based off the angular velocity of the robot, which can be measured with a high degree of accuracy from the inertial measurement unit. Other approaches [12–14] involved the use of optical flow sensors, RTK GPS, and fusion of multiple sensing modalities to accurately estimate slip. Even though some of these works succeeded in accurately estimating slip, they did not result in significant improvement in path following because a majority of these approaches relied on the traditional differential drive model to compensate for the effect of slip and assumed that the underlying path following controller would be able to handle the model mismatches as well. As mentioned in Ref. [11], model-plant mismatch is a particular concern when model predictive controllers are applied.

Recently, many research groups have tried to formulate better models to accurately predict the motion of AGVs in the presence of slip. Based on a review of the available literature, the methods can be broadly classified into two groups, one that uses the full dynamic model of the system including the robot-terrain interactions and the second that relies on the kinematic model of the robot, while ignoring the dynamic effects. The approaches that use the full dynamic model [15–18] usually start from the first principles to estimate slip based on one or more features characterizing the robot terrain interactions such as the effect of slope, tire/track forces, nature of the terrain, etc., or through physics engines that use multibody dynamics method to model the motion of the robot over challenging terrain [19]. Even though the use of physics engines allows for easy modeling while ensuring reliable estimates for the motion of the robot, this method does not directly provide an estimate for the slip experienced by the robot and is therefore not serviceable for controller design. On the other hand, using detailed analytic models to estimate slip over varying terrain conditions requires real-time estimation of terrain geometry, track tension, forces exerted by the track on the ground, and soil properties including consistency and compressibility [20].

Estimating these parameters in real time and solving the model is not feasible due to the computational limits of the systems onboard AGVs.

In the case of using the kinematic models to estimate and compensate for slip, the approach has been focused toward using additional parameters in the ideal differential drive robot model as given by Eq. (1), to account for the effects of robot-terrain interactions. The effective wheelbase model [21] and the general kinematic slip model [22] fall under this category. A recent comparison of some of these models can be found in Ref. [23]. One notable work in this direction includes a model predictive optimal control presented by Rajagopalan et al. [24]. The approach essentially used a Pure Pursuit Path Follower to generate the desired control commands along with an linear-quadratic regulator tracker to handle the effects of slip as disturbance by modifying the control inputs. The disadvantage here is that this method needs an estimated value of the slip parameters and powertrain dynamics of the robot in advance to tune the linear-quadratic regulator gains. This is infeasible for field robots specifically in cases where the terrain data is either unknown beforehand or varies drastically. The solution is to use online parameter estimation. Martinez et al. [25,26] proposed an instantaneous center of rotation (ICR)-based tracked robot model. Real-time estimation of the ICR has shown to improve robot localization [27] and trajectory tracking [28]. However, the fact that even a small amount of longitudinal slip can result in very large ICR values when the robot is moving in a straight line render it unsuitable for path following controllers. This causes the path following controller to saturate frequently leading to jerky motion of the robot. Helmick et al. proposed a Kalman Filter based method to calculate slip [29]. The method, specifically designed for the Rocky 8 Mars Rover, used the estimated slip in an inverse kinematic model to modify the wheel velocities and steering angles to better follow the path.

In addition to improvements in trajectory tracking, slip estimation has also been performed to get better estimates of the power consumption by robots. Recent work by Canfield et al. [30,31] has introduced a novel method to calculate slip parameters along with torque applied and path followed by the robot, based on an ICR based dynamic model of the robot. The estimated slip parameters were then used to evaluate the power consumption of the skid-steer mobile robot, which could in turn be used to find optimal trajectories that minimize power consumption. The proposed method was practically validated on two different manufacturing applications using a skid steer robot with magnetic-tracks. On a similar note, Gupta et al. [32] has proposed methods to perform dynamically feasible energy efficient motion planning for skid steer robots, taking into account torque limitations. The proposed method takes into account the payload of the robot along with terrain conditions and slip experienced by the robot, in order to come up with minimum turn radius constraints. These constraints are then used in a sampling-based model predictive optimization technique to generate energy efficient trajectories for skid-steered autonomous ground vehicles.

Based on a survey of existing work in this domain, the ideal approach toward handling the effect of slip in AGV path following would be to use an augmented kinematic model that can take into account the time-varying effects along with an online parameter estimation method to accurately predict the values in real time. In this regard, active disturbance rejection controller (ADRC), which is capable of handling modeling uncertainty as external disturbance, satisfies all of the requirements. The ADRC can act as a high-level controller that estimates the disturbance signal online that then modifies the control action provided by a low-level path following controller to cancel out the effects of slip.

3 Proposed Approach

This section describes the design of an ADRC to handle the effect of slip and thereby improve path following performance for ground robots. The seminal paper that introduced ADRC [33] highlighted four major contributions to tackle the shortcomings of the traditional proportional–integral–derivative framework: the use of a simple differential equation as a transient trajectory generator, a noise-tolerant tracking differentiator, nonlinear control laws, and the concept and method of total disturbance estimation and rejection. Among the improvements brought out by ADRC, its ability to estimate and reject disturbances in an online fashion has generated great interest in the method [34]. In comparison to existing classical and state-space control techniques, including model predictive control, ADRC does not require an explicit model of the process it is trying to control. Instead it can work with a simple canonical model by taking into account all of the modeling uncertainties as external disturbances. This characteristic makes ADRC an ideal choice for handling systems with process parameter variations and disturbances that cannot be easily modeled. As mentioned in Sec. 1, disturbances, modeling uncertainties, and parameter variations are the major factors that make the control of AGV on rough terrain difficult and as such, ADRC is perfectly suited for this application.

Active disturbance rejection controller is usually designed as a high-level controller acting on top of a stable low-level controller designed for an ideal, known, model of the system. The primary idea behind active disturbance rejection is to lump the modeling uncertainties and parameter variations into disturbance signals that are then formulated as additional states resulting in an augmented system. An online observer is then designed for the augmented system to estimate the additional states based on the output or feedback from the system. The estimated value of the disturbances, in turn, modifies the control action produced by the low-level controller to account for the disturbances or modeling errors present in the real system.

Assuming the estimator is designed well, ADRC effectively converts the real system into the ideal model for which the low-level controller is guaranteed to perform well. In other words, the entire ADRC architecture can be wrapped as an additional layer on top of the existing low-level controller. It is important to note the underlying assumption that the disturbance experienced by the real system can be considered to be piecewise constant. Taking into account the high update rate achievable on the computing architectures available onboard modern AGV's, this assumption holds well for practical implementation. A block diagram of the proposed ADRC architecture is given in Fig. 2. The three essential parts of the ADRC architecture are the model, estimator, and the modification to the control law.

3.1 Generalized Model. As mentioned in Sec. 2, many different models have been proposed over the years to account for the effect of slip in AGV motion. Unlike the existing models that have added parameters to account for specific factors such as the variation in the wheelbase of the robot, we propose a generic model that takes into account the scaling and shift produced in the robot states as a result of slip. The proposed generic model is given below as a modification to Eq. (1)

$$\begin{aligned} \dot{x} &= \alpha_1 V\cos\theta + \beta_1 \\ \dot{y} &= \alpha_2 V\sin\theta + \beta_2 \\ \dot{\theta} &= \alpha_3 \omega + \beta_3 \end{aligned} \tag{5}$$

The α_i parameters are added to address the effect of slip through scaling the rate of change of robot states and the β_i parameters are added to address the same through shift in the rate of change of states. Equation (5) can be summarized as

$$\dot{\boldsymbol{X}} = \boldsymbol{ACU} + \boldsymbol{B}$$
$$\text{where,} \quad \boldsymbol{A} = \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \tag{6}$$

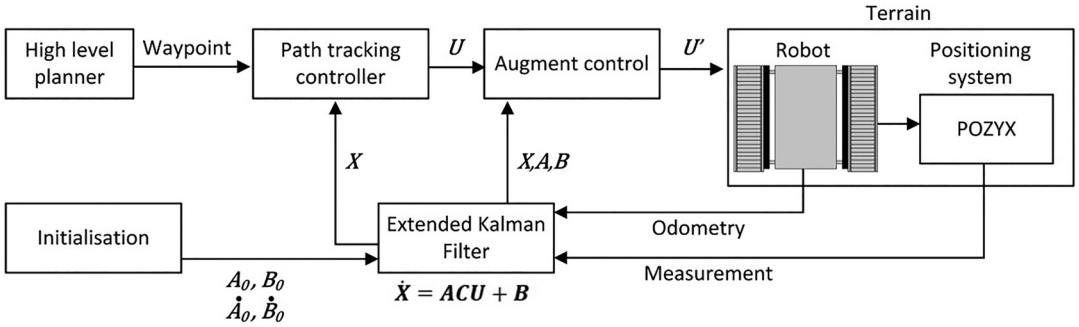

Fig. 2 Block diagram representation of the proposed ADRC for path following in AGVs

3.2 Proposed Estimation Method. The augmented model ADRC requires an observer to estimate the value of the added parameters. The proposed approach will use an EKF as the observer. The prediction model for the EKF can be obtained by discretizing the kinematic model given in Eq. (6) using the Euler method. The augmented parameters are also updated in the EKF prediction model as per the ADRC approach. The complete prediction model used in this paper is given in the following:

$$\begin{aligned}
x_k &= x_{k-1} + \left(\alpha_{1,k-1} V_{k-1} \cos\theta_{k-1} + \beta_{1,k-1}\right)\Delta t + \mu_1 \Delta t \\
y_k &= y_{k-1} + \left(\alpha_{2,k-1} V_{k-1} \sin\theta_{k-1} + \beta_{2,k-1}\right)\Delta t + \mu_2 \Delta t \\
\theta_k &= \theta_{k-1} + \left(\alpha_{3,k-1}\omega_{k-1} + \beta_{3,k-1}\right)\Delta t + \mu_3 \Delta t \\
\alpha_{i,k} &= \alpha_{i,k-1} + \dot{\alpha}_{i,k-1}\Delta t + \mu_{3+i}\Delta t \\
\beta_{i,k} &= \beta_{i,k-1} + \dot{\beta}_{i,k-1}\Delta t + \mu_{6+i}\Delta t \\
\dot{\alpha}_{i,k} &= \dot{\alpha}_{i,k-1} + \mu_{9+i}\Delta t \\
\dot{\beta}_{i,k} &= \dot{\beta}_{i,k-1} + \mu_{12+i}\Delta t
\end{aligned} \tag{7}$$

where i varies from 1 to 3, $\mu_1 - \mu_{15}$ denotes the process (additive zero-mean Gaussian) noise, k is the discrete time index, and Δt is the time-step between each update. Note that the resulting state vector has 15 elements. As denoted by the model, the order of the augmented system is higher than that of the traditional model given in Eq. (2). However, this is a necessary trade-off in the implementation of the ADRC. In accordance with the classical EKF implementation Eq. (7) can be summarized as:

$$\widehat{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{X}}_{k-1}, \boldsymbol{U}_{k-1}) \tag{8}$$

As per the classical EKF notation, variables with $\hat{}$ are predicted values and those without are corrected values. The state covariance matrix, $\boldsymbol{S}$, is 15×15 matrix that is updated as

$$\widehat{\boldsymbol{S}}_k = \boldsymbol{G}_{k-1}\boldsymbol{S}_{k-1}\boldsymbol{G}_{k-1}^T + \boldsymbol{V}_{k-1}\boldsymbol{Q}\boldsymbol{V}_{k-1}^T \tag{9}$$

where $\boldsymbol{Q}$ is the covariance matrix for process noise and $\boldsymbol{G}$ and $\boldsymbol{V}$ are the partial derivatives of $\boldsymbol{\mathcal{F}}$ with respect to $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{U}$, respectively. The sensor measurements for the EKF are obtained from an absolute positioning system that provides noisy updates of states (x_k, y_k, θ_k) collected in vector $\boldsymbol{\mathcal{Y}}_k$. The measurement equation for the EKF is, therefore, given by

$$\boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{X}}_k) = \begin{Bmatrix} x_k + \vartheta_1 \\ y_k + \vartheta_2 \\ \theta_k + \vartheta_3 \end{Bmatrix} \tag{10}$$

where $\vartheta_1 - \vartheta_3$ denote the measurement (additive zero-mean Gaussian) noise. The Kalman gain is calculated as

$$\boldsymbol{K}_k = \widehat{\boldsymbol{S}}_k \boldsymbol{M}_k^T (\boldsymbol{M}_k \widehat{\boldsymbol{S}}_k \boldsymbol{M}_k^T + \boldsymbol{R}_k)^{-1} \tag{11}$$

where $\boldsymbol{R}$ is the covariance matrix for measurement noise and $\boldsymbol{M}$ denotes the partial derivative of $\boldsymbol{\mathcal{M}}$ with respect to $\boldsymbol{\mathcal{X}}$. The state and the covariance matrices are updated using the canonical EKF update equation as given by

$$\begin{aligned}
\boldsymbol{\mathcal{X}}_k &= \widehat{\boldsymbol{\mathcal{X}}}_k + \boldsymbol{K}_k(\boldsymbol{\mathcal{Y}}_k - \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{X}}_k)) \\
\boldsymbol{S}_k &= (\boldsymbol{I}_{15x15} - \boldsymbol{K}_k \boldsymbol{M}_k)\widehat{\boldsymbol{S}}_k
\end{aligned} \tag{12}$$

The values for the $\boldsymbol{G}$, $\boldsymbol{V}$, and $\boldsymbol{M}$ for the proposed model are provided in the Appendix. The numerical values chosen for the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are given in Sec. 4. In order to improve the accuracy of the prediction step in the EKF, instead of using the V and ω provided by the controller in the calculation, the linear and angular velocity of the robot calculated from the wheel mounted encoders were used.

3.3 Low-Level Controller. A simple "Go-to-Goal" behavior is used as the low-level controller. Assuming a unicycle robot model as given by Eq. (2), this controller guides the robot to reach a given planar goal point from any given starting planar position and orientation [8]. This behavior consists of a proportional–derivative (PD) controller that determines the angular velocity, ω, of the robot with a nonlinear scaling applied to the linear velocity, V, based on the calculated ω. Assuming the robot to be at coordinates (x, y) with an orientation of θ and the goal location to be at coordinates (x_g, y_g), the desired orientation of the robot is given by:

$$\theta^* = \tan^{-1}\left(\frac{y_g - y}{x_g - x}\right) \tag{13}$$

The error in orientation is given by , $e = \theta^* \ominus \theta$, where $\ominus$ denotes difference taking into account the wraparound of angles. Based on the error, a PD controller can be written for the angular velocity of the robot as, $\omega = k_p e + k_d \dot{e}$; where k_p and k_d are the proportional and derivative gains, respectively. The linear velocity of the robot is scaled based on ω as follows:

$$V = V_{\max}\left(1 - \frac{2\tan^{-1}(|\omega|)}{\pi}\right) \tag{14}$$

where $V_{\max}$ is a tunable parameter. The aforementioned controller scales down the forward velocity of the robot from $V_{\max}$ to 0 for nonzero ω in a nonlinear fashion. This allows the robot to slow down before making sharp turns allowing for smoother navigation while ensuring that the control output stays within the actuation limits of the robot. The aforementioned controller is asymptotically stable provided k_p, k_d, and $V_{\max}$ are greater than zero.

3.4 Modification to the Control Effort. Based on the estimated value of the augmented parameters $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ and the control action, $\boldsymbol{U}$, provided by the low-level controller, the modified control actions can be obtained as follows:

$$U' = C^{-1}\hat{A}^{-1}(CU - \hat{B}) \tag{15}$$

Note that C is not a square matrix and as such a Moore–Penrose pseudo inverse is calculated to obtain C^{-1}. The updated control law, when applied to the augmented system as given by Eq. (6), converts it into the ideal system given by Eq. (2). Analogously, the ADRC takes care of the disturbances with the low-level controller essentially acting on the ideal model, which is a proven stable system.

3.5 Novelty of the Proposed Approach. Even though extended Kalman filter based methods have been used in the past to estimate the effects of slip in AGV motion toward improving localization [35], none of the prior work has used it toward estimation and compensation of the effects of slip. Additionally, in contrast with the previous methods, the proposed method can account for the effect of time varying slip as well as nonlinear actuator/drive train dynamics, the resistance offered by the terrain on the robot, and other factors that can vary with time through the use of a generalized model. Another major advantage of the proposed method is that due to its hybrid architecture the overall approach can be used with other state of the art path following or trajectory tracking controllers. Even though the proposed work uses a simple Go-to-Goal behavior as the low-level controller, any stable path following controller mentioned in Refs. [36–39] can be used instead.

4 Experimental Validation

In order to validate the performance improvement provided by the proposed ADRC architecture as compared to using the low-level controller alone, here after referred to as PD controller, path following trials were conducted. The experimental validation was performed using the hybrid mobility platform STORM, developed in the Robotics and Mechatronics Lab at Virginia Tech [9]. The system, weighing around 9 kg, has a combination of tracks and wheels that allow it to move in all directions along with ultrasonic sensors and cameras that allow for obstacle avoidance and visual odometry applications. For the experiments described in this paper, only the tracked locomotion mode was used. The robot is fitted with an ODROID XU4 single board computer with robot operating system (ROS) Indigo [40] running onboard.

For the ADRC implementation, encoders on the left and right tracks of the robot were used to calculate the forward and angular velocity executed by the robot as input for the prediction step of EKF. The state feedback, (x, y, θ), for the correction step of the EKF was obtained using the POZYX positioning system [41] fitted on the robot. POZYX is an ultrawideband based positioning system that uses four anchors placed on the perimeter of the experimental area along with a tag placed on the robot. POZYX provides three-dimensional position and orientation of the tag with an accuracy of up to 1 cm in positioning and 1 deg in orientation. Even though this paper describes the use of POZYX, the proposed architecture could work with any absolute positioning system such as the LOSA [42], VICON, or D-GPS. The STORM module mounted with the POZYX tag along with the dimensions of the robot is shown in Fig. 3. The entire ADRC architecture including the EKF was implemented as a series of ROS nodes running onboard the robot for the duration of the experiments. Separate ROS nodes were created for the EKF, low-level controller, ADRC, and hardware interfaces for obtaining sensor data and dissemination of motor commands. The EKF and the POZYX positioning node were run at 50 Hz, while the ADRC architecture consisting of the low-level controller along with the modification to the controls was run at 10 Hz.

The values of the Q matrix and R matrix used in the EKF are given as follows:

$$Q = \begin{bmatrix} 0.015 & 0.0 \\ 0.0 & 0.015 \end{bmatrix}, \quad R = \begin{bmatrix} 0.001 & 0.0 & 0.0 \\ 0.0 & 0.001 & 0.0 \\ 0.0 & 0.0 & 0.00012 \end{bmatrix} \tag{16}$$

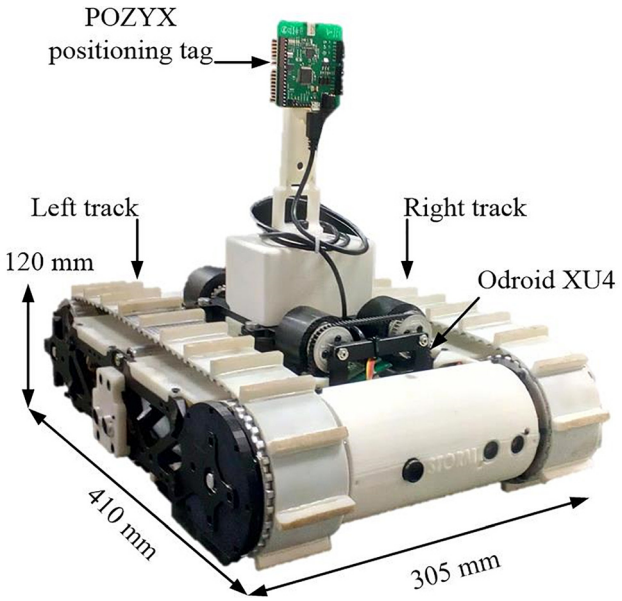

Fig. 3 Experimental setup STORM mounted with POZYX unit

Note that for the above values, Q and R are assumed to be uncorrelated. The augmented parameters in the EKF were initialized as $A_0 = 1.0, B_0 = 0.0, \dot{A}_0 = 0.0$, and $\dot{B}_0 = 0.0$, as shown in Fig. 2. The initial value of the covariance matrix was set to be a nonzero diagonal matrix with 0.01 for the diagonal elements corresponding to the states and 1.0 corresponding to each of the augmented parameters. All of the aforementioned values were chosen heuristically.

The experimental trials were performed over four different terrain conditions: vinyl flooring, asphalt, artificial turf, grass, and gravel terrain as shown in Fig. 4. The experiments over vinyl flooring and artificial turf were conducted indoors whereas asphalt, grass and gravel terrains were tested outdoors. For the grass and gravel case, the terrain was uneven and the ground sloped in both X- and Y-directions. In addition, the terrain varied from grass to gravel over the length of the path followed by the robot. On each terrain, five trials with the proposed ADRC architecture and five trials with the simple PD control were performed. The PD controller implementation is illustrated in Fig. 5. It is important to note that the low-level controller and the waypoints are kept the same for both the cases. The EKF for the PD implementation used the ideal model given in Eq. (2). The ADRC, on the other hand, used the augmented model and the estimated

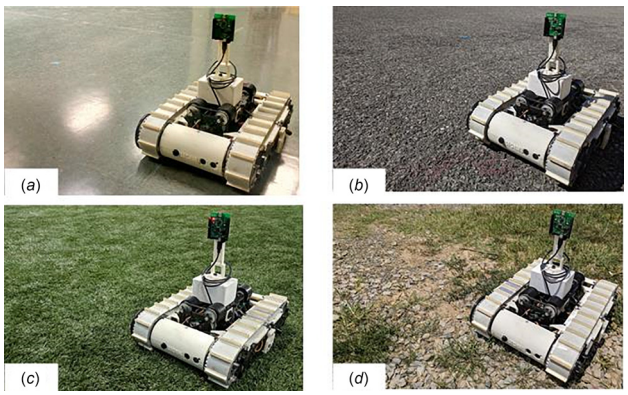

Fig. 4 Experimental validation STORM on different terrains: (*a*) vinyl flooring, (*b*) asphalt, (*c*) artificial turf, and (*d*) grass and gravel

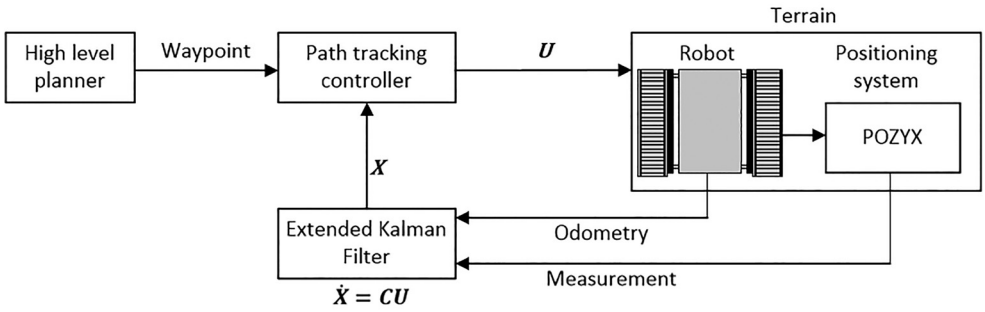

Fig. 5 Block diagram representation of the PD controller implementation

values were used to modify the control inputs. For each terrain, four waypoints laid out as a square are given to the low-level controller. The size of the square is different in case of each terrain based on the amount of space available. In addition, the POZYX data proved to be too noisy for a square perimeter larger than 10 m on each side.

5 Results and Inference

For the square waypoints given in each trial, the Go-to-goal behavior used as the low-level controller should cause the robot to make a perfect square, intersecting each of the waypoints, provided there is no slip. The desired perfect square path along with the actual path followed by the robot under the action of the ADRC and simple PD control is shown for each terrain case in Figs. 6–9. As shown by the results, the varying amount of slip on each of the terrains causes the robot to deviate from the square path. For each of the terrain conditions, the variation in orientation of the robot with respect to time and the control actions, V and ω, applied to the robot under the ADRC and simple PD are also shown in Figs. 6–9. The position and orientation of the robot as demonstrated in the plots are from the EKF running on board the robot.

5.1 Improvement in Path Following. As mentioned in the introduction, the performance of the skid-steer approach over different terrain conditions is greatly dependent on the amount of resistance offered by the terrain. The STORM module that was used for testing is fitted with tracks that have protruding treads as was shown in Fig. 3. When the platform moves over a nondeformable flat terrain such as the case of the vinyl flooring, the tips of the treads are the only part of the robot that makes contact with the ground. This causes excessive slipping of the robot, often resulting in the controller over correcting for the errors. In the case of artificial turf and grass or gravel on the uneven terrain, the treads dig into the terrain due to the weight of the robot and thereby offer additional resistance to the motion, especially while turning. This is true even in case of asphalt where the small cracks in the surface act as gripping points for the treads. The difference is that the terrain is deformable in case of artificial turf, grass and gravel as compared to the case of asphalt. As such, asphalt offers the maximum resistance to the skid-steer motion of the robot. The effect of the varying amount of resistance with the fixed maximum torque on the motors results in varied path following performance for the robot on each terrain, particularly under the action of the PD controller.

The ADRC, on the other hand, captures the varying nature of the terrain in the augmented parameters and thereby provides similar path following performance on different terrain conditions. This is evident from the smoother path tracked by the ADRC for each of the terrain conditions. Since the low-level controller used here adjusts the ω based on the error in orientation, the jerky motion produced by the PD control is particularly evident from the oscillatory nature of the plots showing the orientation of the robot. This is also observable in the plot of commanded velocity in each case. The ADRC produces relatively smooth motion command as compared to PD in each case.

Similar performance may be obtained by a properly tuned proportional–integral–derivative controller. For instance, in Figs. 6(*a*) and 7(*a*) the deviation of the PD controller from the desired path especially at the corners could be due to the overshoot of the controller rather than slip. This could be avoided by retuning the controller, but it will have to be retuned for each new terrain condition or whenever there is a significant change to the hardware of the robot such as the type of tracks being used. In this regard, the proposed approach can be considered to be an adaptive self-tuning controller. The major factor that distinguishes the proposed controller design from existing adaptive controllers is that the ADRC achieves self-tuning through the augmented parameter

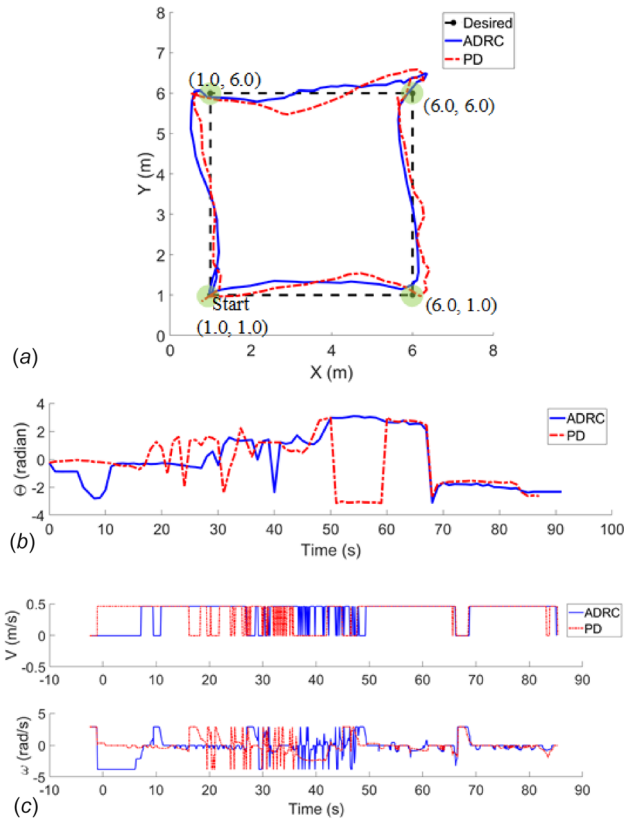

Fig. 6 Path tracking results on asphalt. (*a*) Path followed by the robot under the action of ADRC and PD. The desired path is shown in black. (*b*) Orientation of the robot while following the path under the action of ADRC and PD. (*c*) Control actions produced by ADRC and PD.

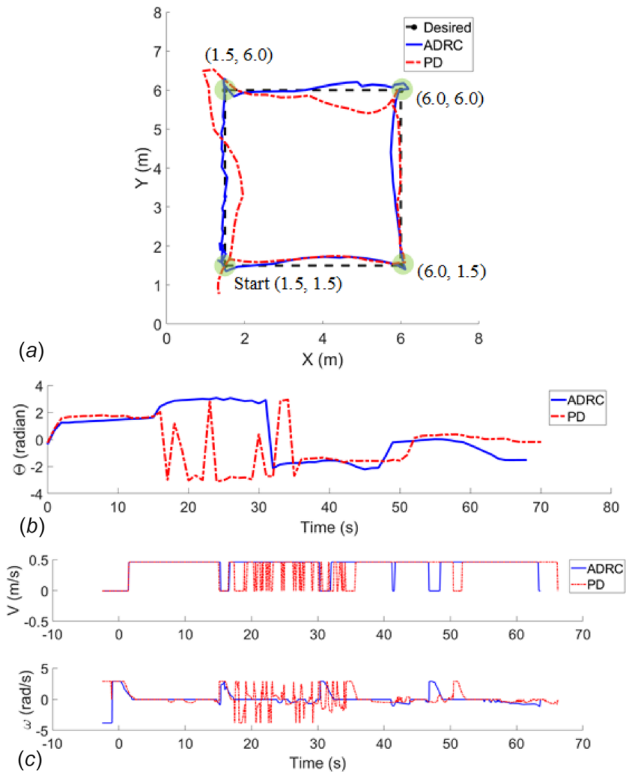

Fig. 7 Path tracking results on vinyl flooring. (*a*) Path followed by the robot on vinyl flooring under the action of ADRC and PD. (*b*) Orientation of the robot under the action of ADRC and PD. (*c*) Control actions produced by ADRC and PD.

values in the modified control law and the augmented parameters themselves are treated as a disturbance being estimated inside the EKF. This allows the entire approach to be designed without an accurate model and remain agnostic toward the noise in sensor measurements. Note that the applied low-level controller reduces the forward velocity of the robot whenever the desired ω is high as explained in Sec. 3.3. This is evident from the results as the V applied is zero whenever the ω is high.

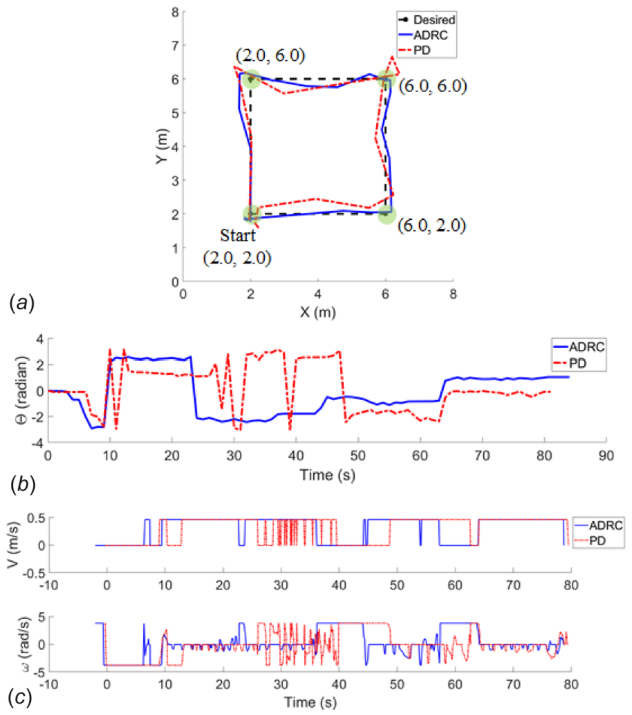

Fig. 8 Path tracking results on artificial turf. (*a*) Path followed by the robot under the action of ADRC and PD. (*b*) Orientation of the robot while following the path under the action of ADRC and PD. (*c*) Control actions produced by ADRC and PD.

5.2 Estimation of the Augmented Parameters. Figures 10–13 shows the estimated value of the augmented parameters $\boldsymbol{A}$ and $\boldsymbol{B}$, for all five trials in each terrain condition. The average nature is shown in darker shade for each parameter. As shown by the plots, the time evolution of the parameters follows a general trend. The fact that the parameters change with time for any given trial, indicates that the nature of slip varies over time even for a specific terrain. It should be noted that, even though for a given terrain condition, the overall nature of the terrain remains the same such as asphalt or artificial turf, there were variations encountered by the robot that were not uniform across the trials. Examples include but are not limited to, one of the tracks going over a large enough rock or even a patch of sand, the treads getting caught on patches of grass or turf, pebbles getting stuck between the track and the driving sprocket.

Once the robot encounters the aforementioned variations in the terrain, the overall behavior of the system changes, which causes the low-level controller to output different commands and the states of the robot, including the augmented parameters evolves differently. This is evident in the plots where for a given terrain condition some trials produced estimated values far different from the average, while still following the overall trend. In addition, the command to start the low-level controller was given manually to the robot at the start of each experimental trial. This caused the robot motion and thereby the parameter evolution, to start at different time instances across the trials, which introduced the minor shifts in the evolution of the parameters as seen from the figures. Tuning the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices further could enable the EKF to maintain the general trend better despite the variations encountered for a given terrain. The fact that the augmented parameters follow a general trend for multiple trials in a given terrain, while varying significantly across different terrain conditions, opens up interesting possibilities as discussed in Sec. 6.

The evolution of the covariance terms corresponding to the augmented parameters for each of the terrain condition is shown in Fig. 15 in the Appendix. As shown by the figure, the filter converges quickly in each case. In order to quantify the performance for each trial, we compare the energy spent, time taken by the robot to complete the path, and the mean cross-track error (MCTE) for each controller in all of the terrain cases. The mean values for each of the parameters over five trials for each terrain and their standard deviation (value given in parenthesis) are shown in Fig. 14. The total energy spent completing the path is proportional to

$$E = mV^2 + I\omega^2 \tag{17}$$

where m is the total mass of the robot and I is the rotational inertia of the main body about the axis normal to the plane of its movement. Exact measurements of the total energy spent by the robot rely on the terrain interaction forces, which in turn depend on the properties of the terrain, and individual inertia values for each of the moving components of the robot. This is beyond the scope of this paper and therefore we use the term E to refer to energy spent by the robot for comparison.

Since the onboard controller taking the measurements is discrete, the value of E for each trial was obtained by numerically integrating the value of E for each iteration of the controller and then averaging over the total time taken to complete the trial. The forward and angular velocity executed by the controller as measured by the encoders are used to calculate E.

The cross-track error is the Euclidean distance between the robot and the closest point on the path (in this case the straight line joining the previous and the next waypoint) computed during each iteration of the controller. The mean cross-track error (MCTE) is the

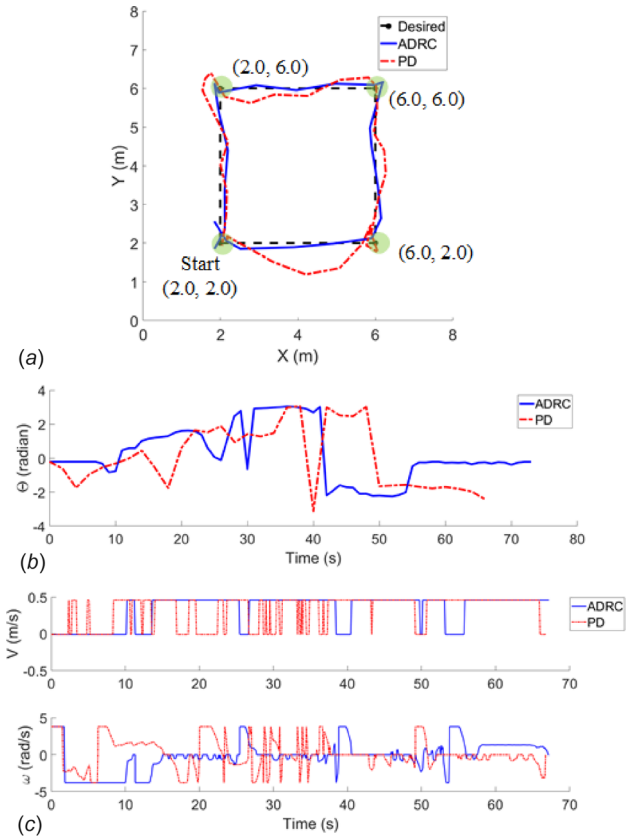

Fig. 9 Path tracking results on grass and gravel. (*a*) Path followed by the robot under the action of ADRC and PD. The desired path is shown in black. (*b*) Orientation of the robot while following the path under the action of ADRC and PD. (*c*) Control actions produced by ADRC and PD.

average of the cumulative cross track error over the total number of trials. Based on the results as shown in Fig. 14, the proposed controller performs better than PD in each case.

6 Conclusion and Future Work

This paper proposed the design of an active disturbance rejection controller to estimate and compensate for the effects of slip during path following on challenging terrain conditions using autonomous tracked vehicles. The proposed ADRC architecture uses a generic model that can account for the scaling and shift in the states of the system due to the effects of slip through augmented parameters. An EKF observer is used to estimate the value of the augmented parameters for any given terrain condition. The estimated value of the parameters is then used to modify the control outputs from a low-level controller in order to compensate for the effects of slip which result in better path following performance. Even though this paper described the use of a simple Go-to-goal behavior as the low-level controller, the proposed approach could work with a more complicated behavior that also control the heading and curvature of the path followed by the robot in relevant applications.

The performance of the proposed ADRC architecture was compared to that of the low-level PD controller through extensive experimental trials involving four different terrain conditions (vinyl flooring, asphalt, artificial turf, grass, and gravel), with ten trials on each terrain (five under ADRC and five under PD control). The path followed by the robot change in orientation with respect to time and the final control actions applied to the robot under the action of the ADRC and simple PD control were compared in each case. The time taken by the robot, energy consumed, and mean cross track error averaged over five trials under the influence of the ADRC and PD control was also compared for each of the terrains.

As shown by Figs. 6–9 and 14 the ADRC consistently shows better performance as compared to using the low-level PD controller alone. The path tracked by the robot is closer to the desired

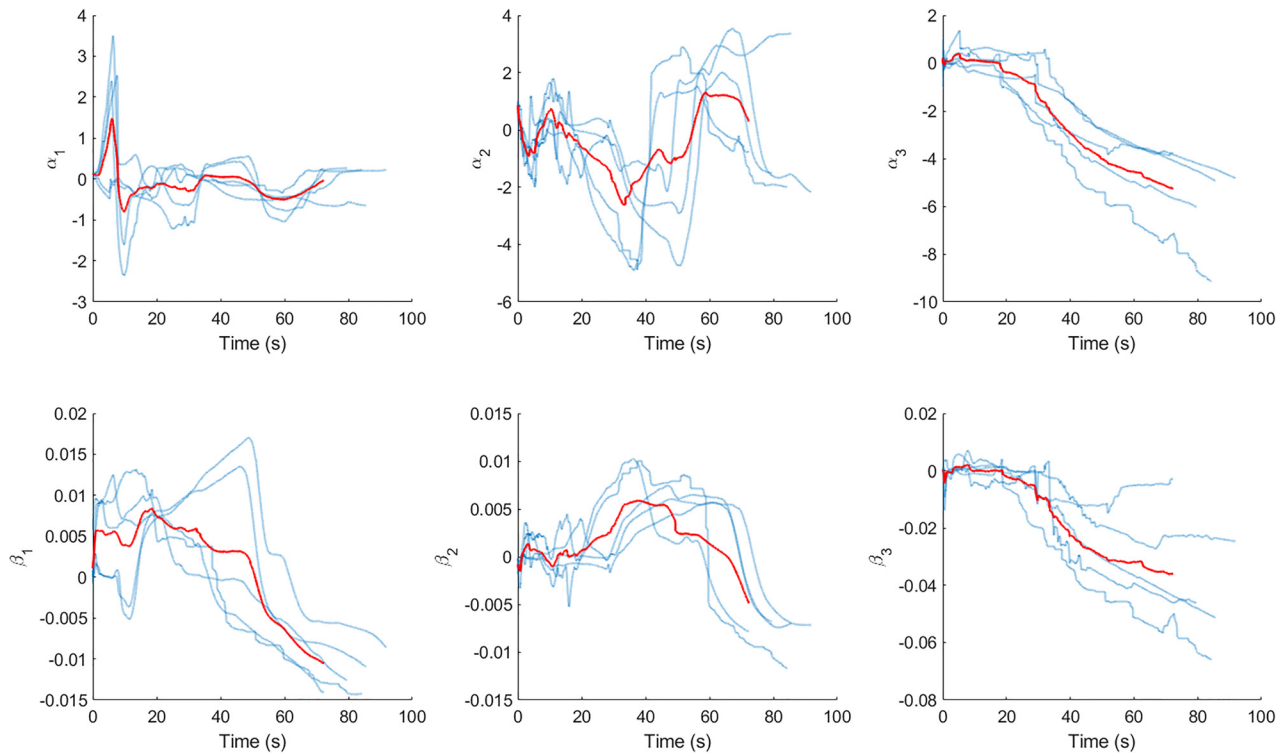

Fig. 10 Variation in value of the added parameters when the robot follows the path on asphalt under the action of the ADRC

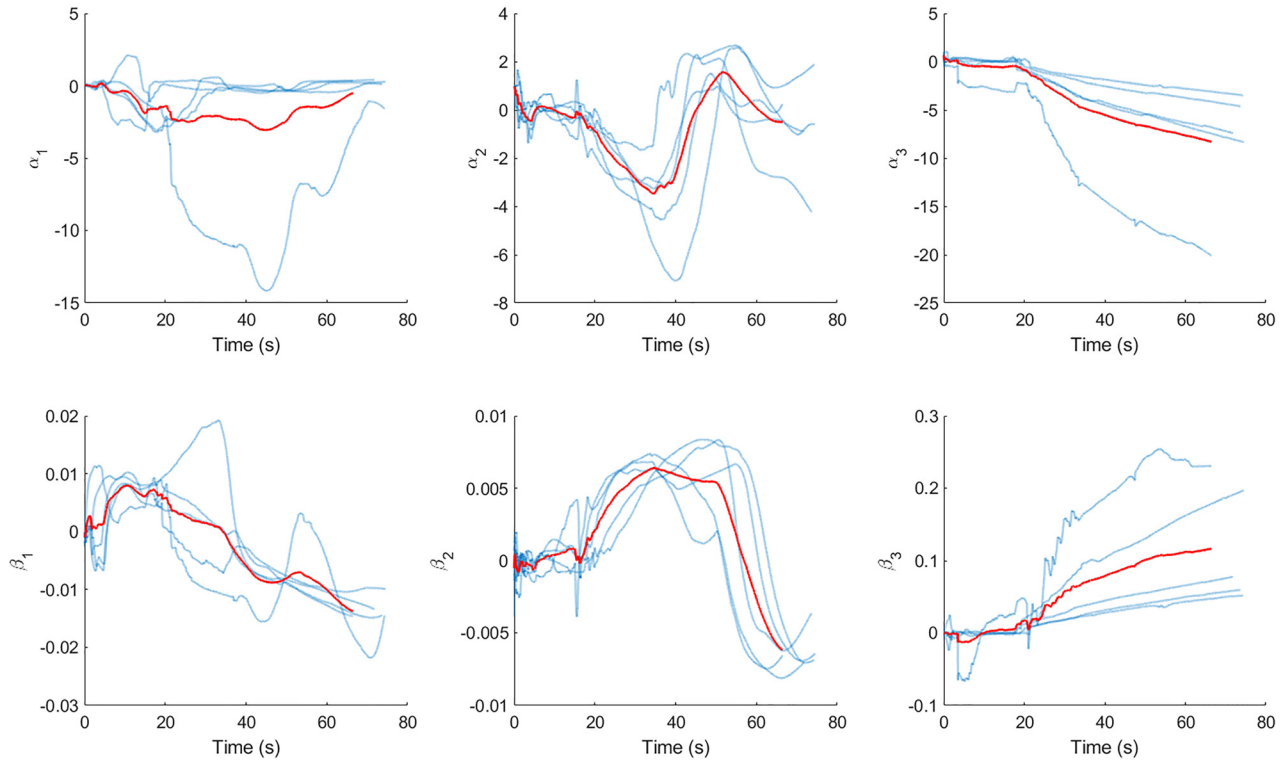

Fig. 11 Variation in value of the augmented parameters when the robot follows the path on vinyl flooring under the action of the ADRC

path in case of ADRC and the motion is smooth as shown by the variation in orientation and the applied control actions. While the ADRC provides smooth corrections to the robot, the PD often overcorrects on the account of slip and results in a jerky motion. As shown in Fig. 14, ADRC provides significant improvement in MCTE; 33.33% on asphalt, 36.36% on artificial turf and 31.25% on grass-gravel. In addition, the use of ADRC improves the energy consumption by approximately 8% on asphalt, artificial

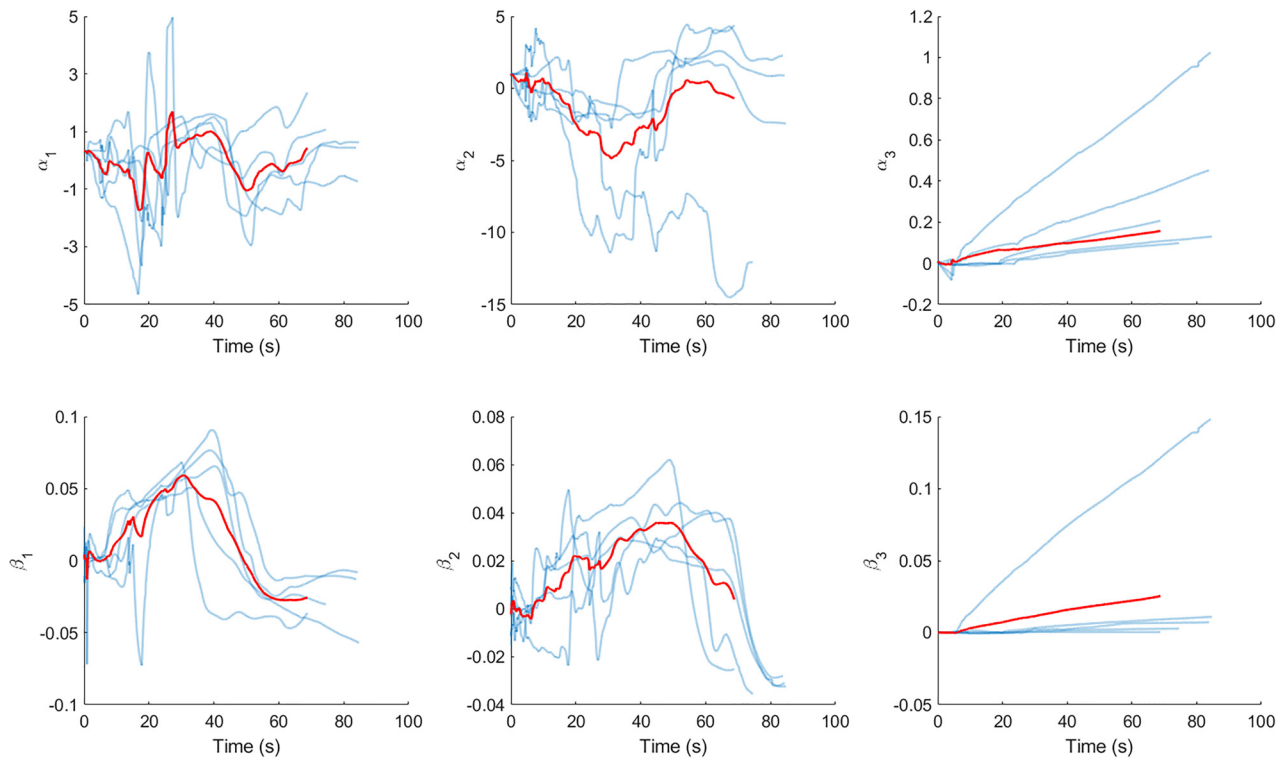

Fig. 12 Variation in value of the added parameters when the robot follows the path on artificial turf under the action of the ADRC

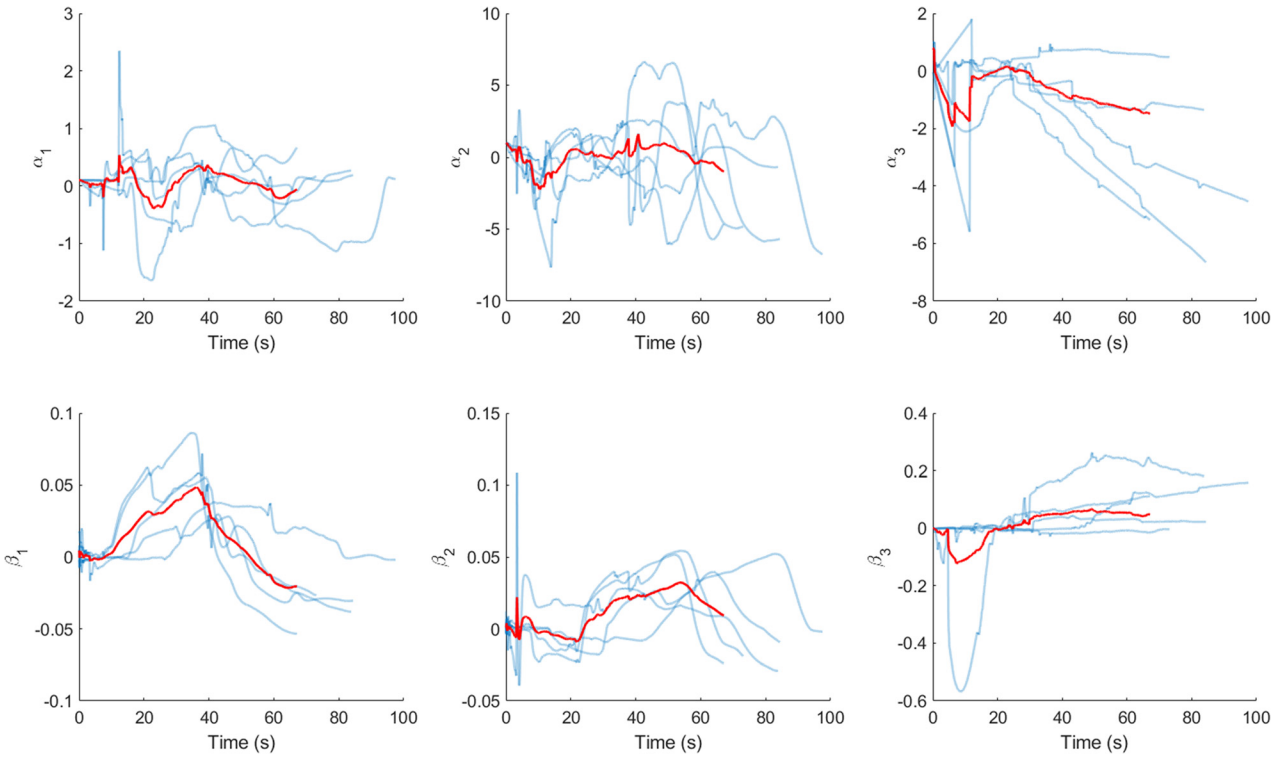

Fig. 13 Variation in value of the added parameters when the robot follows the path on grass and gravel under the action of the ADRC

turf and grass-gravel, and time taken by about 11% on asphalt. Except on vinyl flooring where the performances of both controllers are similar, experimental results on all other terrain conditions show significant improvement with the use of ADRC. It should be noted that the ADRC architecture uses the same low-level PD controller, with the only difference being that ADRC uses the augmented parameters to modify the control input that comes out of the PD controller to additionally account for the effect of slip. Taking into account the aforementioned factors, any improvement in trajectory tracking can only be attributed to the improved performance brought out by the ADRC architecture.

Even though this paper assumes slip to be the major cause of the disturbance experienced by the robot based on existing literature, it could be due to other unknown or unmodeled kinematic and dynamic properties of the system. The fact that it could be from other sources can be considered as another advantage of the proposed control method, in terms of improvement in trajectory tracking. Detailed analysis into whether the variations captured by the model is solely due to slip will require accurate slip estimation using additional sensors on the robot such as a free wheel with an encoder. Such an analysis is beyond the scope of this paper and will be a part of future work.

The performance metrics as reported in literature for any path following controller implementation depends greatly on the platform being used for testing, accuracy and precision levels of the sensors, and the type of terrain over which the system is being tested. Majority of the existing techniques for slip estimation rely on precise measurements of the velocity of the robot. As stated before, the major advantage of the proposed method is to use even noisy position estimates from the POZYX system to provide reliable path following performance. The noisy position data provided by the POZYX system onboard STORM makes reliable velocity estimation and subsequent slip estimation difficult. Even though this highlights the capability of the proposed method to provide reliable trajectory tracking performance in presence of noisy data, this prevents implementation of other existing techniques on the available platform. In this regard a detailed performance comparison between the proposed method and existing state of the art methods is beyond the scope of this paper. Future testing will rely on using different sensors including RTK GPS or vision-based tracking systems to obtain reliable slip estimations for performance comparison. In addition, the developed architecture will be further tested on heavier

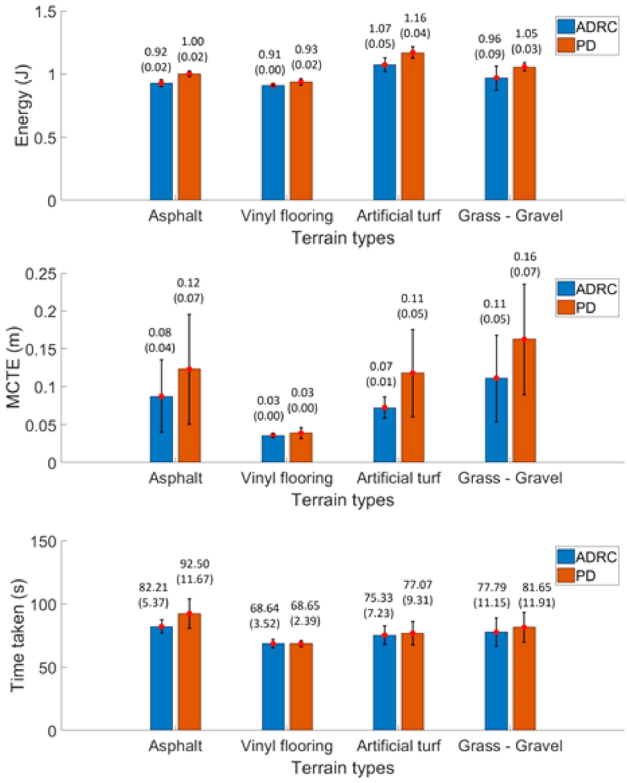

Fig. 14 Performance comparison of both controllers over different terrain conditions

robots such as the HMMR [43] on more challenging outdoor terrain conditions including sand and mud. The path following trials described in this paper was conducted over a small region due to the limited range of the POZYX positioning systems. The use of RTK GPS system will allow for longer trials in outdoor scenarios. The improvement in trajectory tracking performance brought out by the ADRC is expected to be more pronounced in the case of longer trials, which will be analyzed as a part of future work.

As a part of the results, we presented the average nature of the augmented parameters over five different trials under the action of ADRC for a given terrain condition in Figs. 10–13. It is evident from the plots that the augmented parameters follow a general trend for multiple trials in a given terrain, while varying significantly across different terrain conditions. Simple time averaging of these signals provides only an approximation of the general trend, instead a more detailed analysis using methods such as the dynamic time windowing approaches [44] to align the data collected over different trials, could help in better capturing the underlying trend specific to each terrain condition. Autonomous navigation, specifically for tracked robotic vehicle moving over challenging terrain conditions require detailed knowledge of the terrain to make high level decisions such as determining traversability of a given terrain, choosing optimal gain values for trajectory tracking controllers, etc. Existing works in this direction have focused on the use of dedicated sensing methods, such as vision, point cloud mapping, or vibrations experienced by the robot to determine the type of terrain. These methods while effective are susceptible to the failure modes of the sensor, such as dust or fog on vision and changes in the suspension system of the robot for vibration sensing. The fact that the augmented parameters contain information about the terrain allows estimation of these parameters to be used for real time terrain recognition through existing machine learning based techniques. In other words, a machine learning classifier could be trained offline using labeled data and then used to perform terrain recognition online based on the output of the EKF, without using any additional sensors. This could significantly improve autonomous decision making in systems like planetary rovers and search and rescue robots for applications like planetary terrain mapping and exploring disaster scenarios.

Acknowledgment

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision unless so designated by other documentation.

The authors would like to thank Adam Williams and Shubhdildeep Sohal of the Robotics and Mechatronics Lab at Virginia Tech for their help in this work.

Funding Data

- Medical Research and Materiel Command (W81XWH-16-C-0062, Funder ID. 10.13039/100000182).

Appendix

The analytical expression for the $\boldsymbol{G}$, $\boldsymbol{V}$, and $\boldsymbol{M}$ matrices of the EKF used in the ADRC architecture is given below

$$\boldsymbol{G}_{k-1} = \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{\mathcal{X}}}\bigg|_{\boldsymbol{\mathcal{X}}_{k-1}} = \begin{bmatrix} 1 & 0 & a & b & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & c & 0 & d & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & e & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & f & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & f & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & f & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & f & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & f \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

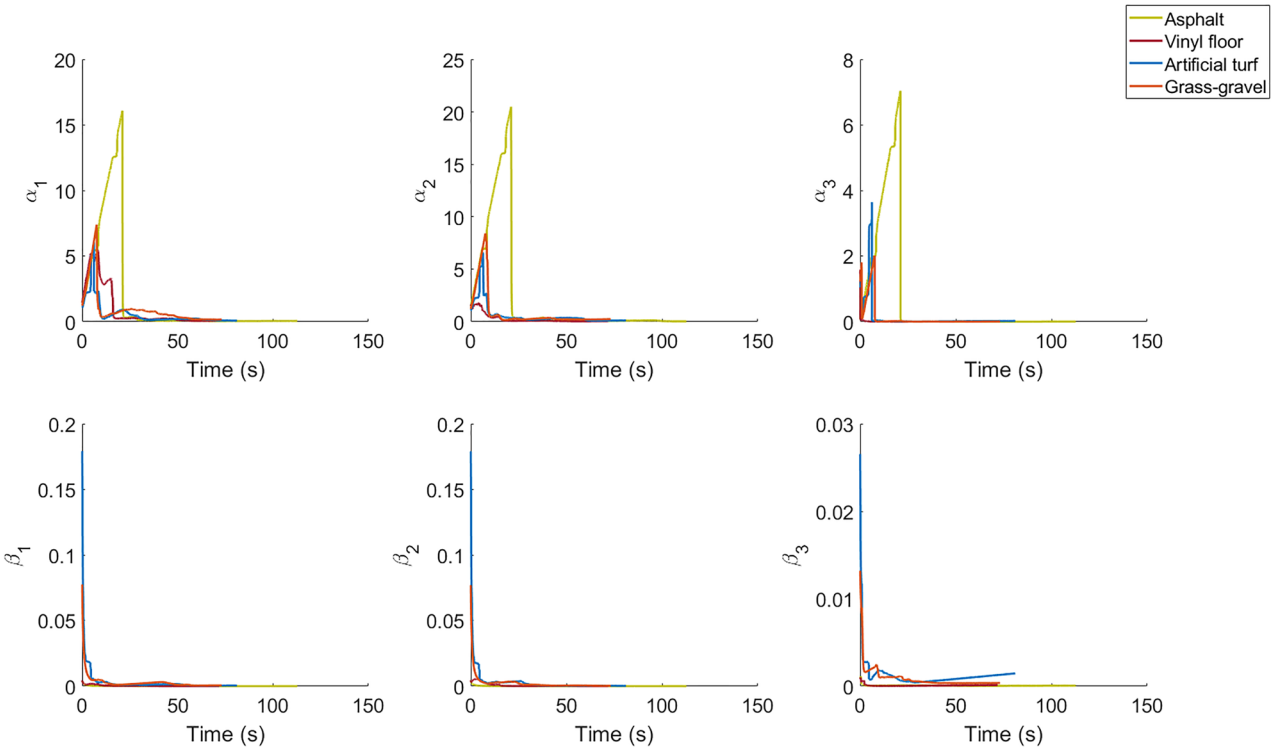

Fig. 15 Covariance of the added parameters as the robot follows the path under the action of ADRC

where $a = -\alpha_1 V \sin\theta\Delta t$, $b = V\cos\theta\Delta t$, $c = \alpha_2 V\cos\theta\Delta t$, $d = V\sin\theta\Delta t$, $e = \omega\Delta t$, and $f = \Delta t$

$$\mathbf{V}_{k-1} = \left.\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{U}}\right|_{\boldsymbol{\mathcal{X}}_{k-1}} = \begin{bmatrix} a & 0 \\ b & 0 \\ 0 & c \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where $a = \alpha_1\cos\theta\Delta t$, $b = \alpha_2\sin\theta\Delta t$, and $c = \alpha_3\Delta t$

$$\mathbf{M}_k = \left.\frac{\partial \boldsymbol{\mathcal{M}}}{\partial \boldsymbol{\mathcal{X}}}\right|_{\boldsymbol{\mathcal{X}}_k} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 15 shows the time evolution of the augmented parameter covariance inside the EKF. The value of the diagonal elements in the covariance matrix that corresponds to each of the added parameter is plotted with respect to time. Covariance values for one trial in each of the four terrain conditions are shown. As demonstrated by the plot, the EKF converges rapidly in each case.

References

[1] Bekker, M. G., 1962, *Theory of Land Locomotion: The Mechanics of Vehicle Mobility*, University of Michigan Press, Ann Arbor, MI.

[2] Murphy, R. R., 2014, *Disaster Robotics*, MIT Press, Cambridge, MA.

[3] Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., and Kawatsuma, S., 2013, "Emergency Response to the Nuclear Accident at the Fukushima Daiichi Nuclear Power Plants Using Mobile Rescue Robots," J. Feld Rob., **30**(1), pp. 44–63.

[4] Murphy, R. R., Kravitz, J., Stover, S. L., and Shoureshi, R., 2009, "Mobile Robots in Mine Rescue and Recovery," IEEE Robot. Autom. Mag., **16**(2), pp. 91–103.

[5] Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M., 2008, "Search and Rescue Robotics," *Springer Handbook of Robotics*, Springer, Berlin, pp. 1151–1173.

[6] Rodríguez, F., 2014, *Autonomous Tracked Robots in Planar Off-Road Conditions*, Springer International Publishing, Berlin.

[7] Siegwart, R., 2004, *Introduction to Autonomous Mobile Robots*, MIT Press, Cambridge, MA.

[8] Corke, P., 2011, *Robotics, Vision and Control*, Springer International Publishing, Berlin.

[9] Kumar, P., Saab, W., and Ben-Tzvi, P., 2017, "Design of a Multi-Directional Hybrid-Locomotion Modular Robot With Feedforward Stability Control," ASME Paper No. DETC2017-67436.

[10] Gonzalez, R., and Iagnemma, K., 2018, "Slippage Estimation and Compensation for Planetary Exploration Rovers. State of the Art and Future Challenges," J. Field Rob., **35**(4), pp. 564–577.

[11] Burke, M., 2012, "Path-Following Control of a Velocity Constrained Tracked Vehicle Incorporating Adaptive Slip Estimation," IEEE International Conference on Robotics and Automation, Saint Paul, MN, May 14–18, pp. 97–102.

[12] Ding, L., Gao, H., Deng, Z., and Liu, Z., 2010, "Slip-Ratio-Coordinated Control of Planetary Exploration Robots Traversing Over Deformable Rough Terrain," IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, Oct. 18–22, pp. 4958–4963.

[13] Kim, J., and Lee, J., 2016, "A Kinematic-Based Rough Terrain Control for Traction and Energy Saving of an Exploration Rover," IEEE International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, Oct. 9–14, pp. 3595–3600.

[14] Terry, J. D., and Minor, M. A., 2008, "Traction Estimation and Control for Mobile Robots Using the Wheel Slip Velocity," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, Sept. 22–26, pp. 2003–2009.

[15] Ward, C. C., and Iagnemma, K., 2008, "A Dynamic-Model-Based Wheel Slip Detector for Mobile Robots on Outdoor Terrain," IEEE Trans. Rob., **24**(4), pp. 821–831.

[16] Howard, T. M., and Kelly, A., 2007, "Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots," Int. J. Rob. Res., **26**(2), pp. 141–166.

[17] Naveed, K., Khan, Z. H., and Hussain, A., 2014, "Adaptive Trajectory Tracking of Wheeled Mobile Robot With Uncertain Parameters," *Computational Intelligence for Decision Support in Cyber-Physical Systems* (Studies in Computational Intelligence), Springer, Singapore, pp. 237–262.

[18] Taheri-Kalani, J., and Khosrowjerdi, M. J., 2014, "Adaptive Trajectory Tracking Control of Wheeled Mobile Robots With Disturbance Observer," Int. J. Adapt. Control Signal Process., **28**(1), pp. 14–27.

[19] Sebastian, B., and Ben-Tzvi, P., 2018, "Physics Based Path Planning for Autonomous Tracked Vehicle in Challenging Terrain," J. Intell. Rob. Syst. (epub).

[20] Auersch, L., 1998, "Vehicle-Track-Interaction and Soil Dynamics," Veh. Syst. Dyn., **29**(Suppl. 1), pp. 553–558.

[21] Yu, W., Chuy, O. Y., Collins, E. G., and Hollis, P., 2010, "Analysis and Experimental Verification for Dynamic Modeling of a Skid-Steered Wheeled Vehicle," IEEE Trans. Rob., **26**(2), pp. 340–353.

[22] Seegmiller, N., Rogers-Marcovitz, F., Miller, G., and Kelly, A., 2011, "A Unified Perturbative Dynamics Approach to Online Vehicle Model Identification," International Symposium on Robotics Research, Flagstaff, AZ, Dec. 9–12, pp. 1–16.

[23] Fink, J. R., and Stump, E. A., 2014, "Experimental Analysis of Models for Trajectory Generation on Tracked Vehicles," IEEE International Conference on Intelligent Robots and Systems, Chicago, IL, Sept. 14–18, pp. 1970–1977.

[24] Rajagopalan, V., Mericli, C., and Kelly, A., 2016, "Slip-Aware Model Predictive Optimal Control for Path Following," IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, May 16–21, pp. 4585–4590.

[25] Martínez, J. L., Mandow, A., Morales, J., Pedraza, S., and García-Cerezo, A., 2005, "Approximating Kinematics for Tracked Mobile Robots," Int. J. Rob. Res., **24**(10), pp. 867–878.

[26] Mandow, A., Martínez, J. L., Morales, J., Blanco, J. L., García-Cerezo, A., and González, J., 2007, "Experimental Kinematics for Wheeled Skid-Steer Mobile Robots," IEEE International Conference on Intelligent Robots and Systems, San Diego, CA, Oct. 29–Nov. 2, pp. 1222–1227.

[27] Pentzer, J., Brennan, S., and Reichard, K., 2014, "Model-Based Prediction of Skid-Steer Robot Kinematics Using Online Estimation of Track Instantaneous Centers of Rotation," J. Field Rob., **31**(3), pp. 455–476.

[28] Pentzer, J., Brennan, S., and Reichard, K., 2014, "The Use of Unicycle Robot Control Strategies for Skid-Steer Robots Through the ICR Kinematic Mapping," IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, Sept. 14–18, pp. 3201–3206.

[29] Helmick, D. M., Roumeliotis, S. I., Cheng, Y., Clouse, D. S., Bajracharya, M., and Matthies, L. H., 2006, "Slip-Compensated Path Following for Planetary Exploration Rovers," Adv. Robot., **20**(11), pp. 1257–1280.

[30] Canfield, S. L., Hill, T. W., and Zuccaro, S. G., 2018, "Prediction and Experimental Validation of Power Consumption of Skid-Steer Mobile Robots in Manufacturing Environments," J. Intell. Rob. Syst. Theory Appl. (epub).

[31] Canfield, S. L., Hill, T. W., and Zuccaro, S. G., 2016, "Modeling Power Requirements for Skid-Steer Mobile Robots in Manufacturing Environments," ASME Paper No. DETC2016-60152.

[32] Gupta, N., Ordonez, C., and Collins, E. G., 2017, "Dynamically Feasible, Energy Efficient Motion Planning for Skid-Steered Vehicles," Auton. Robots, **41**(2), pp. 453–471.

[33] Han, J., 2009, "From PID to Active Disturbance Rejection Control," IEEE Trans. Ind. Electron., **56**(3), pp. 900–906.

[34] Herbst, G., 2013, "A Simulative Study on Active Disturbance Rejection Control (ADRC) as a Control Tool for Practitioners," Electronics, **2**(4), pp. 246–279.

[35] Seegmiller, N., Rogers-Marcovitz, F., Miller, G., and Kelly, A., 2013, "Vehicle Model Identification by Integrated Prediction Error Minimization," Int. J. Rob. Res., **32**(8), pp. 912–931.

[36] Sebastian, B., and Williams, A., 2018, "Gaussian Kernel Controller for Path Tracking in Mobile Robots," ASME Paper No. DETC2018-85641.

[37] Park, J. J., and Kuipers, B., 2011, "A Smooth Control Law for Graceful Motion of Differential Wheeled Mobile Robots in 2D Environment," IEEE International Conference on Robotics and Automation, Shanghai, China, May 9–13, pp. 4896–4902.

[38] Wit, J., Crane, C. D., and Armstrong, D., 2004, "Autonomous Ground Vehicle Path Tracking," J. Rob. Syst., **21**(8), pp. 439–449.

[39] Coulter, R. C., 1992, "Implementation of the Pure Pursuit Path Tracking Algorithm," Robotics Institute, Carnegie Mellon University, Technical Report No. CMU-RI-TR-92-01.

[40] Open Source Robotics Foundation, 2014, "Indigo—ROS Wiki," Open Source Robotics Foundation, Mountain View, CA, accessed July 7, 2018, http://wiki.ros.org/indigo

[41] Pozyx Labs, 2015, "Pozyx Accurate Positioning," Pozyx Labs, Ghent, Belgium, accessed June 25, 2018, https://www.pozyx.io/

[42] Kumar, A., and Ben-Tzvi, P., 2016, "Spatial Object Tracking System Based on Linear Optical Sensor Arrays," IEEE Sens. J., **16**(22), pp. 7933–7940.

[43] Ben-Tzvi, P., Goldenberg, A. A., and Zu, J. W., 2010, "Articulated Hybrid Mobile Robot Mechanism With Compounded Mobility and Manipulation and On-Board Wireless Sensor/Actuator Control Interfaces," Mechatronics, **20**(6), pp. 627–639.

[44] Paliwal, K., Agarwal, A., and Sinha, S., 1982, "A Modification Over Sakoe and Chiba's Dynamic Time Warping Algorithm for Isolated Word Recognition," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '82), Paris, France, May 3–5, pp. 1259–1261.