Rut Detection and Following for Autonomous Ground Vehicles

Camilo Ordonez [*], Oscar Y. Chuy Jr. [†], Emmanuel G. Collins Jr. [‡], and Xiuwen Liu [§]
Center for Intelligent Systems, Control and Robotics (CISCOR)
[*] [†] [‡] Department of Mechanical Engineering
Florida A&M - Florida State University
[§] Department of Computer Science
Florida State University
Tallahassee, FL 32310, United States
Email: camilor@eng.fsu.edu chuy@eng.fsu.edu ecollins@eng.fsu.edu liux@cs.fsu.edu

Abstract— **Expert off road drivers have found through experience that ruts formed on soft terrains as a result of vehicular transit can be used to improve vehicle safety and performance. Rut following improves vehicle performance by reducing the energy wasted on compacting the ground as the vehicle traverses over the terrain. Furthermore, proper rut following can improve vehicle safety on turns and slopes by utilizing the extra lateral force provided by the ruts to reduce lateral slippage and guide the vehicle through its path. This paper presents a set of field experiments to show the relevance of rut following for autonomous ground vehicles and proposes a reactive based approach based on knowledge of the width of the tires and the vehicle body clearance to provide mobile robots with rut detection and following abilities. Experimental results on a Pioneer 3AT robot show that the proposed system was able to detect and follow S-shaped ruts, and ruts that are not directly in front of the robot.**

I. Introduction

Autonomous ground vehicles (AGVs) are increasingly being considered and used for challenging outdoor applications. These tasks include fire fighting, agricultural applications, search and rescue, as well as military missions. In these outdoor applications, ruts are usually formed in soft terrains like mud, sand, and snow as a result of habitual passage of wheeled vehicles over the same area. Fig. 1 shows a typical set of ruts formed by the traversal of manned vehicles on off road trails.

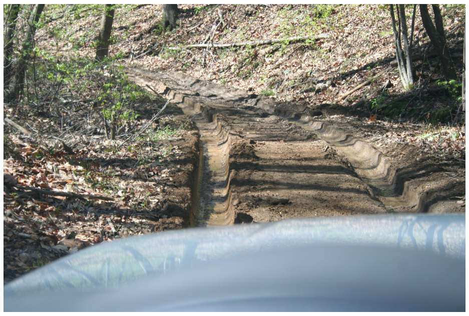


Fig. 1. Typical Off Road Ruts Created by Manned Vehicles

Expert off road drivers have realized through experience that ruts can offer both great help and great danger to a vehicle [1]. On soft terrains ruts improve vehicle performance by reducing the energy wasted on compacting the ground as the vehicle traverses over the terrain [2]. Furthermore, when traversing soft and slippery terrains, proper rut following can improve vehicle safety on turns and slopes by utilizing the extra lateral force provided by the ruts to reduce lateral slippage and guide the vehicle through the desired path [1, 3, 4, 5, 6]. On the other hand, a vehicle moving at high speed that hits a rut involuntarily can lose control and tip over. An AGV provided with rut detection and rut following abilities can benefit from the correct application of this off road driving rule, and thereby improve its efficiency and safety in challenging missions.

Besides the benefits of rut following already explained, proper rut detection and following can be applied in diverse applications. Rut detection can signal the presence of vehicles in the area, and also can help in the guidance of loose convoy operations. In planetary exploration, ruts can play an important role; due to the high cost of these missions, it is desirable, in some situations, for a rover to retrace its path after a successful exploration mission and minimize the risk of getting stuck in terrain that is difficult to traverse. A rut detection system can be used as a robot sinkage measurement system, which is key in the prediction of center high situations. Automatic rut detection can also be employed to determine the coefficient of rolling resistance [7] (a vital parameter in robot dynamic models), and in general can be used to learn different properties of the terrain being traversed.

Prior to the research in [8], work on rut detection focused exclusively on paved surfaces in a road surface inspection application [9, 10]. However, these approaches are not concerned with the continuity of the ruts, something achieved in the proposed approach by using local rut models in the vicinity of the vehicle. In contrast to the method presented in [8], the rut detection method presented here incorporates domain knowledge regarding tire width and vehicle body clearance into the rut detection problem. By doing so, the detection process becomes more efficient because the search for ruts can be performed on specific candidate points over the laser scan instead of at every point as in [8]. Besides that, by incorporating geometric constraints on the rut depth and width, center high situations can be reduced and ruts that are too wide or too narrow can be avoided. Another important difference between the current rut detection implementation and the one of [8] is that this new approach uses a polynomial representation of the left and right ruts in the local vicinity of the robot. By doing this, the robot can differentiate between the left and the right rut, which is necessary for waypoint assignment during rut following.

Additional research that is related to rut detection is the development of a seed row localization method using machine vision to assist in the guidance of a seed drill [11]. This system was limited to straight seed rows and was tested in agricultural type environments, which are relatively structured. The work presented on [12], presents a vision-based estimation system for slip angle based on the visual observation of the trace produced by the wheel of the robot. However, it only detects the wheel traces being created by the robot. An important result was shown in [7], where a correlation between the rut depth and the rolling resistance was presented. However, this work did not deal with the rut detection problem. As previously mentioned, a rut detection method for mobile robots was developed in [8]. However, that paper did not present any approach to the rut following problem. Two successful systems of road lane detection and tracking are presented in [13, 14]. However, these approaches are tested on flat ground and are mainly concerned with keeping the vehicle inside the road and not with keeping the wheels inside specific regions of the terrain as is the case for rut following.

The main contributions of this paper are the conception, design and performance of field experiments to show the relevance of rut detection and following for autonomous vehicles. In addition, the paper proposes, implements and performs an experimental validation of a solution to provide mobile robots with rut detection and following capabilities.

The remainder of the paper is organized as follows. Section II presents a series of motivational experiments with two different robotic platforms and two different terrains. Section III describes the proposed approach to rut detection and following. Section IV introduces the experimental setup and shows experimental results. Section V provides a set of improvements to the proposed approach. Finally, Section VI presents concluding remarks, including a discussion of future research.

II. MOTIVATIONAL EXPERIMENTS

To show some of the benefits of rut following three controlled experiments were performed using two different robotic platforms on two different soft terrains. It is important to note that the motivational experiments do not use the proposed algorithm, but do experimentally show the relevance of rut following for off road robot navigation.

During the motivational experiments, the robot stays in the ruts by following a set of preassigned waypoints. In the case of the Pioneer 3-AT, which has less accurate localization capabilities than the XUV, the runs were performed on short and straight ruts and the vehicle was carefully placed and aligned at the starting point of the ruts. In the case of the XUV the experiments were performed over longer ruts because it counts with a localization system comprised of a differential GPS and a high cost IMU. Fig. 2(a) shows the Pioneer 3-AT robot following ruts in sandy terrain, and Fig. 2(b) shows the XUV robot following ruts in muddy terrain.

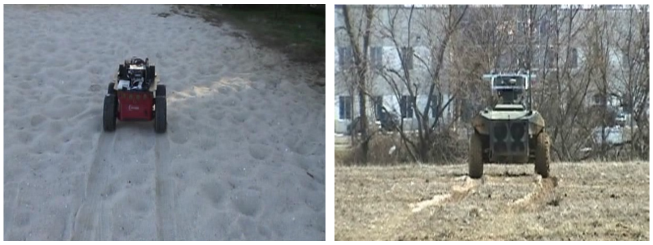

(a) (b)

Fig. 2. (a) Pioneer 3-AT Robotic Platform on Sand. (b) XUV Robotic Platform on Mud

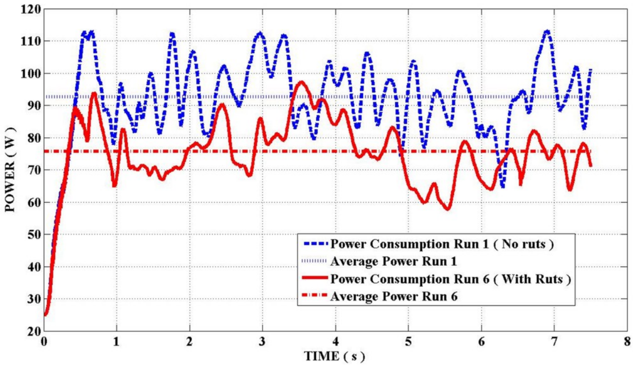


Fig. 3. Decrease in Power Consumption by Following Ruts (Pioneer 3 on Sand)

In the motivational experiments, power consumption and velocity tracking are used as performance metrics. The power consumption (P_c) is computed as the RMS value of $F_r V_r$, where F_r is the force required to overcome the rolling resistance when the vehicles is moving at constant velocity V_r. The velocity tracking performance is computed as the RMS value of the velocity error $E_v(t) = V_r(t) - V_c(t)$, where V_r is the robot velocity and V_c is the commanded velocity.

First, a Pioneer 3-AT robotic platform was commanded to follow a set of ruts over sandy terrain at 0.8 m/s. Six trials were performed; the first run was used as a baseline because it corresponds to the no-rut case (i.e., the robot is beginning the first creation of ruts). Fig. 3 shows a comparison of the power consumption for the first (no ruts) pass and the sixth pass. Notice that by following the ruts, there is an average reduction in power consumption of 18.3%. Furthermore, the experiments revealed that as early as the second pass, there is an average reduction in power consumption of 17.9%.

A second experiment was performed on mud with the XUV robotic platform. The robot was commanded to follow a set of waypoints along a straight line at a speed of 5 mph. Fig. 4 shows the reduction of the rolling resistance coefficient μ_ρ and power consumption for 4 successive trials. Notice that in the second pass, there is a reduction in power consumption of 12.6%.

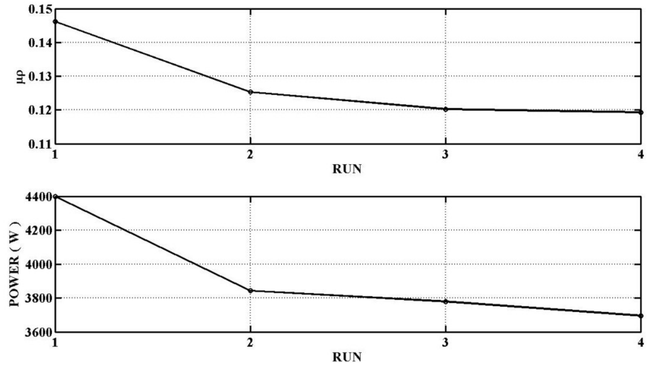


Fig. 4. Decrease in Power Consumption by Following Ruts (XUV on Mud)

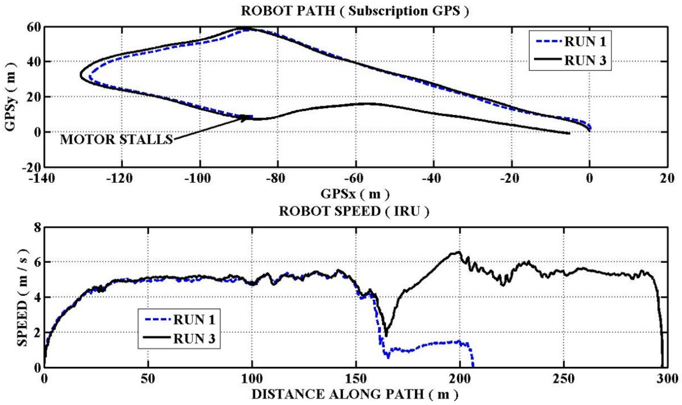


Fig. 5. Velocity Tracking Improvement by Following Ruts (XUV on Mud)

A third experiment was performed on mud with the XUV robotic platform. The robot was commanded to follow a set of waypoints along a curved path at 11 mph and three trials were performed. Fig. 5 shows the robot path and velocity profiles for the first and third run. Notice that on the first run, when there were no ruts, the vehicle was not capable of generating enough torque to track the commanded speed. This caused the motor to stall and the vehicle was not able to complete its mission. On the contrary, in the 3rd trial the robot was able to complete its mission by using the ruts created during the first two passes. The velocity tracking error reduced from 46.2% for the first run to 19.3% for the third run.

It is also worth mentioning that the robot finished the mission successfully on the second pass and exhibited a velocity tracking error of 20%. In the above experimental results it is clear that rut following improved the vehicle performance. This is important from a practical stand point because it means that a robot in the field can benefit from detecting and following ruts, even those that are freshly formed.

III. PROPOSED APPROACH TO RUT DETECTION AND FOLLOWING

The proposed approach assumes that the AGV is equipped with a laser range finder that observes the terrain in front of the vehicle. The proposed approach is divided into two subsystems as shown in Fig. 6: 1) a reactive system in charge of generating fine control commands to place the robot wheels in the ruts, and 2) a local planning system conceived to select the best rut to follow among a set of possible candidates based on a predefined cost function. Once the planner has selected a rut to follow, the reactive system is engaged. This paper focuses on the reactive system, which is a very important component of the proposed approach because it allows precise rut following. A reactive system is selected because it can handle situations for which a system based only on global information would fail. As shown in Fig. 6, the reactive system is composed of the stages described below.

A. Rut Detection

The rut detection stage is in charge of analyzing the laser scans to find a set of possible rut candidates. These rut candidates are then passed through a two stage validation process, which efficiently removes the candidates that don't satisfy the validation criteria. First, the candidate ruts are validated both in depth and width using important domain knowledge regarding the width of the tires and the vehicle body clearance. Second, the candidate ruts are validated using a set of current local models of the ruts in the vicinity of the vehicle.

1) Rut Center Candidate Generation: Fig. 7 illustrates all the relevant coordinate systems used in this work: the inertial system N, the sensor frame S, the sensor head frame H and the vehicle frame B. This stage starts by transforming the laser scan from sensor coordinates to the B_p frame coordinates, which is coincident with the the vehicle kinematic center (B) and has the X_{bp} axis oriented with the robot and the Z_{bp} axis perpendicular to the terrain. This is a convenient transformation because it compensates for the vehicle roll and pitch.

The rut candidates are the local minima of the function $Z(\theta)$, where θ is the angle of the laser beam with respect to the X_s axis, and $Z(\theta)$ is the elevation of a laser point in the B_p frame. The current laser has a coverage of $\approx 140^\circ$ and an angular resolution $\theta_{res} \approx 0.3515^\circ$. Therefore, θ is given by

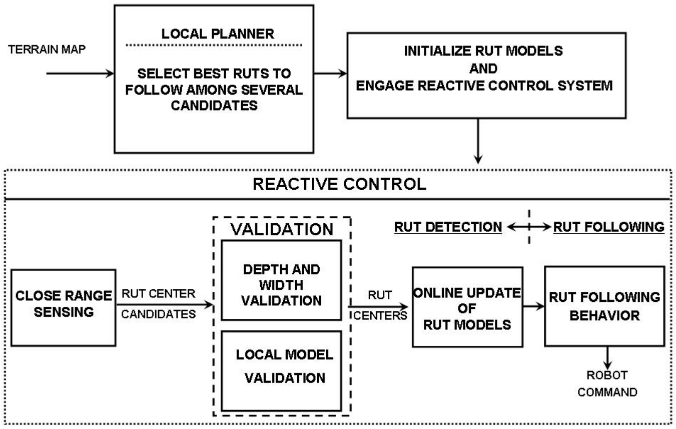


Fig. 6. Schematic of the Proposed Approach to Rut Detection and Following

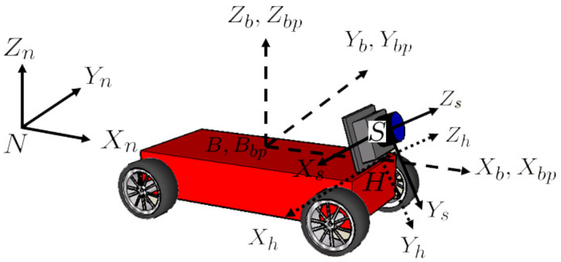


Fig. 7. Coordinate Systems

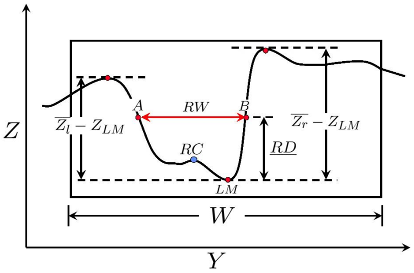


Fig. 8. Cross Sectional View of a Rut in the Rut Frame

$$\theta = 20^\circ + (i-1)\theta_{res},\ i = 1,2,...,399. \tag{1}$$

Given two laser beams with angles θ_a and θ_b with respect to the X_s axis, $\theta_i \in [\theta_a, \theta_b]$[1], is a local minima of $Z(\theta)$ if the following three conditions are satisfied:

1) $Z(\theta_i) < Z(\theta)\ \forall\ \theta \in [\theta_a, \theta_b]$,
2) $Z(\theta_i) \leq Z(\theta_a) - \underline{RD}$,
3) $Z(\theta_i) \leq Z(\theta_b) - \underline{RD}$,

where $\underline{\text{RD}}$ is the minimum depth that a rut should have to be considered a rut. As explained before, these local minima are only rut candidates, which need to be validated in two stages as now discussed. Note that the validation stages are implemented in cascade and therefore if a rut candidate doesn't pass the first stage is immediately removed from the candidate list and doesn't have to go through the second stage.

2) Depth and Width Validation: Once a set of rut candidates has been selected as described in III-A.1, a local window W is constructed around each candidate. The size of this window is a design parameter. In the proposed approach W is selected so that the widest ruts to be detected are covered by W when the relative orientation between the vehicle and the rut is 30°.

As explained in III-A, it is important to verify that the rut does not violate the vehicle body clearance. This constraint is checked by using the following two rules:

$$max(\overline{Z_r} - Z_{LM}, \overline{Z_l} - Z_{LM}) \leq \overline{RD}, \tag{2}$$
$$min(\overline{Z_r} - Z_{LM}, \overline{Z_l} - Z_{LM}) \geq \underline{RD}, \tag{3}$$

where as shown in Fig. 8, Z_{LM} is the elevation of the local minima, $\overline{Z_r}$ and $\overline{Z_l}$ are respectively the points with maximum elevation inside the window W to the right and to the left of the local minima, and $\underline{RD}$ and $\overline{RD}$ are respectively the minimum and maximum rut depths that do not violate the body clearance constraint.

It is not desirable to follow ruts that are either too narrow or two wide. This constraint is posed in terms of the width of the tire (TW) and is verified using

$$TW \leq RW \leq 1.5\ TW, \tag{4}$$

where RW is an estimate of the rut width at a depth of $\underline{\text{RD}}$. Once RW has been estimated, the rut center (RC) can be obtained. All the rut centers are then passed to the Local Model Validation stage.

3) Local Model Validation: The robot keeps local models of the right and left ruts in the vicinity of the vehicle. As illustrated in Fig. 9, the ruts are modeled locally as second order polynomials of the form,

$$y(x) = \sum_{k=1}^{3} a_k x^{k-1}. \tag{5}$$

The rut centers with coordinates (x_i, y_i) that passed the depth and width validation stage are then validated against the local models by computing the model prediction error $e_i = y(x_i) - y_i$. The rut centers that yield the minimum prediction error are used as the new rut centers to update the local rut models.

Note that polynomial modeling of the ruts is just one option. For example, a clothoid model can be used.

4) Online Update of Rut Models: The rut centers that passed the two stage validation process are then used to update the rut local models given by (5). The model parameters a_k are found using a least squares minimization approach.

In the current implementation the laser has a fixed pitch, and therefore the robot has to move to initialize the models. It does so by looking for 10 pairs of ruts centers that have a separation similar to the track width of the robot. However, in the future implementation, this constraint will be removed by the inclusion of a tilt platform. In addition, the model initialization will be performed by the local path planning subsystem, which uses a predefined cost function to select the best rut to follow among several possible candidates. By doing this, the possibility of following random tracks can be minimized.

B. Rut Following

The rut that exhibits the minimum prediction error is used to generate a new waypoint for the robot as shown in Fig. 10.

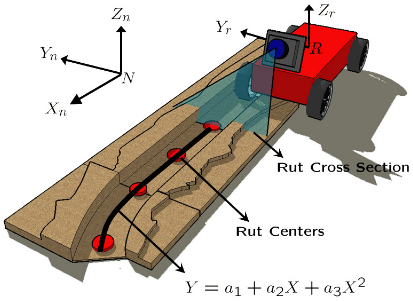


Fig. 9. Rut Model

[1] Note that θ takes on multiple values of the angular resolution of the sensor (0.3515°) in [θ_a,θ_b].

This waypoint takes into consideration the vehicle geometry so that the wheels of the robot can be placed in the rut. Assume that the right rut presents the minimum model prediction error. Then, the waypoint $W_p = (X_w, Y_w)$ is chosen using the rut center $R_{c1} = (X_{c1}, Y_{c1})$ as follows:

$$X_w = X_{c1}, \tag{6}$$

$$Y_w = Y_{c1} + RobW/2, \tag{7}$$

where $RobW$ is the vehicle track width and W_p and R_{c1} are expressed in the body frame B. It is important to clarify that R_{c1} is located at the intersection of the laser plane with the rut. In the current implementation R_{c1} is located at ($\approx 15cm$) from the front of the robot to allow a maximum traversal speed of $75cm/s$.

After a waypoint has been generated, a command for the angular velocity ω is generated using

$$l^2 = X_w^2 + Y_w^2, \; r^2 = (r - Y_w)^2 + X_w^2, \tag{8}$$

$$r = \frac{l^2}{2\, Y_w}, \tag{9}$$

$$\omega = \frac{v}{r}, \tag{10}$$

where r is the turning radius and v is the linear velocity of the robot, which is kept low and constant as is recommended for off road driving [4]. Equations (8)-(10) define an algorithm similar to the Pure Pursuit algorithm [15].

IV. EXPERIMENTAL SETUP AND RESULTS

The experiments were conducted on a Pionner 3-AT robotic platform. It was equipped with a laser range finder URG-04LX [16]. This laser has an angular resolution of 0.36°, a scanning angle of 240°, and a detection range of 0.02m-4m. In the current implementation, the laser readings were taken at 5 Hz. In addition, a tilt sensor was employed to obtain pitch and roll information with an accuracy of ± 0.2° (static measurements) and a sampling rate of 8 Hz [17].

The experimental evaluation was performed on soft dirt. It is important to note that the ruts created in this terrain type are structured similarly to the ruts typically encountered in off road trails as illustrated in Fig. 1. The evaluation of the algorithm on less structured ruts and different terrains is part of our current research. The depth of the ruts was in the range of $3-6cm$ which is comparable to the changes in elevation of the non-compacted terrain (i.e., the terrain that is not part of the ruts.)

A. Rut Following of an S-shaped Rut.

An S-shaped rut, shown in Fig. 11, was chosen to evaluate the tracking performance of the algorithm. This particular shape was chosen because it includes both straight and curved regions.

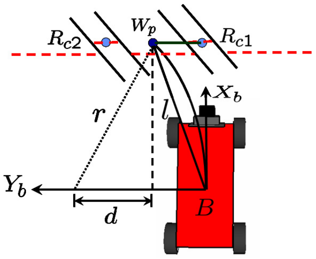


Fig. 10. Waypoint Assignment

TABLE I

RUT DETECTION PERFORMANCE S-SHAPED RUT

No of Rut Cross Sections	Detection Rate	False Alarm Rate
612	89%	16.67%

Fig. 12 shows the raw laser readings corresponding to the scenario with the S-shaped rut. The figure also shows the rut detection results (filled circles), false alarms (filled stars) and two regions were the algorithm fails to detect the ruts. These false negatives can be caused by occlusions, excessive pitch of the robot, and in some situations (see, for example, Region 2 in Fig. 12) they are mainly caused by the relative orientation of the robot and the rut. However, it is important to notice that due to the online models of the left and right ruts, the robot was able to remain inside the ruts, despite the missed detections and the false alarms. Table I summarizes the rut detection results.

In order to quantify the tracking performance, define the cross-track error as $e_{ct}(l_p) \triangleq y(l_p) - y_{des}(l_p)$, where y_{des} is the desired path for the rear right wheel as a function of the path length (l_p) and y corresponds to the actual path followed by the rear right wheel. The RMS value of the cross-track error computed for two different trials was approximately $2cm$. The actual path followed by the wheel was manually measured by using a distinct mark left by the rear right wheel.

B. Rut Following with an Initial Position Offset

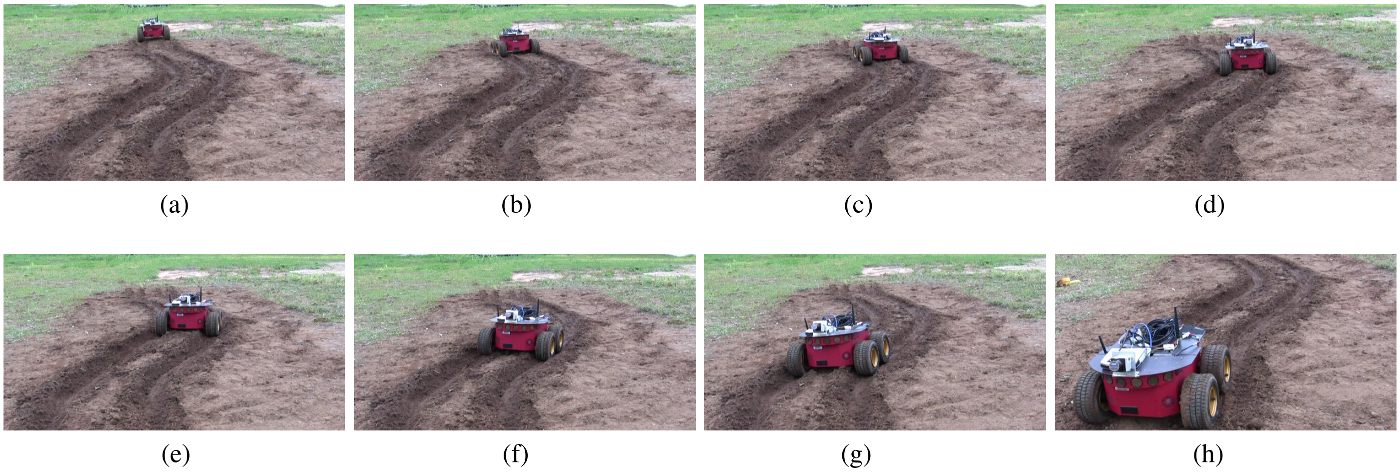
To test the ability of the proposed approach to track ruts that are not directly in front of the robot, the following experiment was performed. As shown in Fig. 13, the robot started its mission with an offset. This offset is a non dimensional quantity computed as the distance from the center of the right rut to the center of the front right wheel and normalized by the track width of the vehicle. Three experiments were conducted for offsets of 0.5, 1.0, and 1.5.

Fig. 13 shows the trajectory followed by the rear right wheel for the three different offsets. In all the trials the robot was able to find the ruts and position itself in the right location to follow the ruts. Table II summarizes the rut detection results for this experiment.

TABLE II

RUT DETECTION PERFORMANCE UNDER INITIAL POSITION OFFSET

No of Rut Cross Sections	Detection Rate	False Alarm Rate
328	82.9%	1.83%



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11. Pioneer 3-AT Following an S-Shaped Rut

V. IMPROVEMENTS TO THE PROPOSED APPROACH

This section presents a brief description of a set of improvements to the rut detection and following approach. The new features of the approach are introduced with the objective of improving the robustness of the algorithm by using a probabilistic framework to perform the rut detection and a tracking module, based on an Extended Kalman Filter (EKF), that exploits the spatio-temporal coherence that exists between consecutive rut detections and generates state estimates that directly feed a steering control system to follow the ruts.

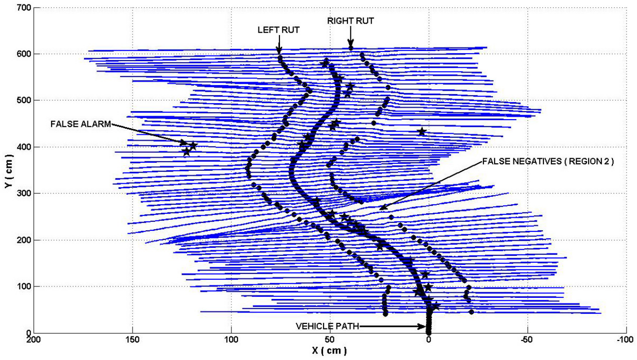


Fig. 12. Terrain Map and Rut Detection Results

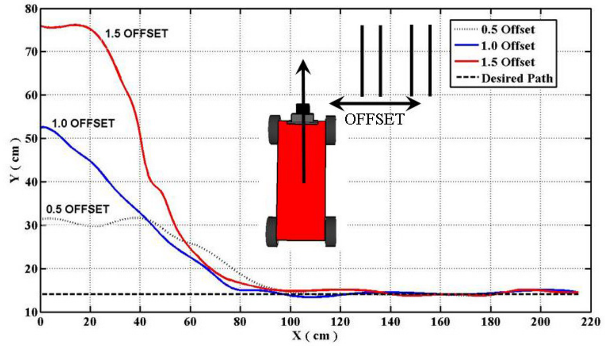


Fig. 13. Wheel Path for Different Initial Position Offsets

A. Probabilistic Based Rut Detection

A probabilistic framework is selected because it provides the means to account for the uncertainty that arises in the rut detection process due to sensor noise and modeling approximations. Since ruts are expected to vary in shape depending on the terrain and vehicle, we propose to experimentally generate a set of rut templates obtained using rut samples from the range of traversable ruts. That is, ruts with a width in the range $[TW, 1.5TW]$ and with a depth in the range $[0.5BC, 0.8BC]$, where TW is the tire width and BC represents the body clearance. To improve computational efficiency, only 4 rut templates were used in the current implementation.

Once a laser scan is obtained, the rut templates are passed point by point through a search region (designated by the EKF, see subsection V-B) of the laser scan and the sum of squared errors between each of the templates and the laser points are computed for each position. Then, the minimum of these errors e_{min} is used as the feature to estimate the probability of the laser point being a rut center. These probabilities are computed using Bayes' theorem as follows:

$$p(w_j/e_{min}) = \frac{p(e_{min}/w_j)p(w_j)}{\sum_{j=1}^{2} p(e_{min}/w_j)p(w_j)}, \qquad (11)$$

where w_j represents the class of the measurement (rut or not rut) and $p(w_j)$ are the prior probabilities of each class, which are assumed equal to 0.5. The likelihoods $(p(e_{min}/w_j))$ are estimated using a maximum likelihood approach [18] and a training set which contains 100 rut samples. Fig. 14 illustrates the posterior probability estimates $p(Rut/e_{min})$ for each point of a laser scan that contains two ruts.

B. Rut Tracking

The rut tracking relies on an EKF that recursively estimates the lateral offset (y_{off}), the relative angle between the vehicle and the rut (θ_{vr}), and the parameters of the rut, which motivated by the work of [19] is here modeled locally as a curve of constant curvature (κ). The rut is modeled using frame R as illustrated in Fig. 15, which makes an angle θ_r

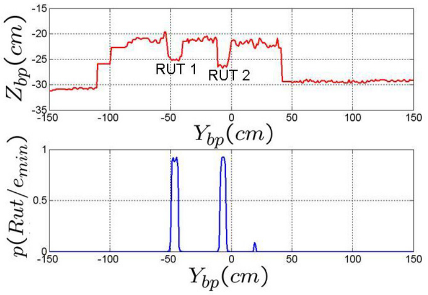


Fig. 14. Laser Data Containing Two Ruts (top) and Corresponding Probability Estimates of $p(RUT/e_{min})$ (bottom)

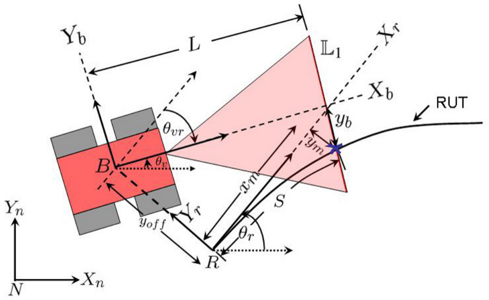


Fig. 15. Rut Frame Coordinates used by the Process and Measurement Models

with the inertial frame N and moves with the vehicle having the X_r axis tangent to the rut at all times.

Assuming that the vehicle moves with forward velocity v and angular velocity $\omega = \frac{d\theta_v}{dt}$, the evolution of θ_r, θ_{vr}, and y_{off} are computed using

$$\dot{\theta}_r = vcos(\theta_{vr})\kappa, \quad (12)$$

$$\dot{\theta_{vr}} = \omega - vsin(\theta_v r)\kappa, \quad (13)$$

$$y_{o\dot{f}f} = vsin(\theta_{vr}). \quad (14)$$

Using the backward Euler rule with sampling time δ_t and assuming that the evolution of the curvature is driven by white and Gaussian noise, it is possible to express the process model as

$$\begin{bmatrix} \theta_{vr_k} \\ \kappa_k \\ y_{off_k} \end{bmatrix} = \begin{bmatrix} \theta_{vr_{k-1}} - \kappa_{k-1} vcos(\theta_{vr_{k-1}})\delta_t \\ \kappa_{k-1} \\ y_{off_{k-1}} + vsin(\theta_{vr_{k-1}})\delta_t \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \delta\theta_{v_{k-1}} + w_{k-1}, \quad (15)$$

where $\delta\theta_{v_{k-1}}$ is the model input (the commanded change in vehicle heading and w represents the process noise, which is assumed white and with normal probability distribution with zero mean, and covariance Q $(p(w) \sim N(0,Q))$.

The measurement model corresponds to the lateral distance y_b from the vehicle X_b axis to the rut center, which is located at the intersection of the laser and the rut (see Fig. 15). Using geometry, it is possible to express y_b as

$$y_{b_k} = -sin(\theta_{vr_k})x_m + \frac{1}{2}\kappa x_m^2 cos(\theta_{vr}) - y_{off_k} cos(\theta_{vr}) + \nu_k, \quad (16)$$

where ν is a white noise with normal probability distribution $(p(\nu) \sim N(0,R))$. As shown in Fig. 15, x_m is a function of the state $x_k = [\theta_{vr_k}, \kappa_k, y_{off_k}]^T$ and the lookahead distance (L) of the laser and satisfies

$$\frac{1}{2}x_m^2 \kappa_k sin(\theta_{vr_k}) + cos(\theta_{vr_k})x_m - (L + y_{off_k} sin(\theta_{vr_k})) = 0, \quad (17)$$

where (17) is obtained as a result of a coordinate transformation from the rut frame R to the vehicle frame B. As mentioned in subsection V-A, the measurement model of the EKF (16) is used to generate a prediction of the rut location for the next iteration and therefore allows the rut detection module to limit the search to a small region around the predicted value.

C. Steering Control for Rut Following

The state estimates generated by the EKF are then used by a nonlinear steering control law, which is an adaptation of the controller proposed in [20], which was designed for an Ackerman steered vehicle. In this work, we approximate the vehicle kinematics using a differential drive model.

The main objective of the controller is to drive the relative angle between the vehicle and the rut θ_{vr} to zero and the lateral offset y_{off} to a desired offset $y_{off_{des}} = \frac{RobW+TW}{2}$, where $RobW$ is the width of the robot and TW is the width of the tire. To achieve this, a desired angle for the vehicle $\theta_{v_{des}}$ is computed using a nonlinear steering control law as follows

$$\theta_{v_{des}} = \theta_r + \arctan(\frac{k_1(y_{off_{des}} - y_{off})}{v}), \quad (18)$$

where θ_r is the angle of the rut with respect to the global frame N, v is the robot velocity, and k_1 is a gain that controls the rate of convergence towards the desired offset. The desired angle $(\theta_{v_{des}})$ is then tracked using the proportional control law

$$\omega = k_2(\theta_{v_{des}} - \theta_v) = k_2(\theta_{vr} - \arctan(\frac{k_1(y_{off_{des}} - y_{off})}{v}), \quad (19)$$

where ω is the commanded angular velocity for the robot. Notice that (19) takes as inputs the state estimates generated by the EKF.

D. Simulation Evaluation

To test the proposed approach, a computer simulation using Matlab was developed. A theoretical rut was simulated using a curved path with constant curvature $\kappa = 0.25m^{-1}$. The sensor measurements were simulated by finding the intersection of the laser L_1 with the rut as illustrated in Fig. 16. The lookahead distance was set to $45cm$ and the robot linear velocity was maintained constant at $20cm/s$. The process noise covariance Q was set to $Q = diag(1e-5, 2e-4, 1e-5)$ and the measurement noise covariance was set to $R = 1e-3$, which is 10 times larger than the typical variance for a laser sensor. The initial covariance estimate P_o was set equal to Q and the robot was originally placed parallel to the rut but with a lateral offset of $1m$. Notice that the desired offset is $y_{off_{des}} = \frac{RobW+TireW}{2} = 25cm$.

The first performance metric RMS_{TvsE} is the RMS error between the true and estimated offsets, where the true offset is defined as the distance between the kinematic center of the vehicle B and the closest point on the rut and the estimated offset is the one estimated by the EKF. The second performance metric RMS_{EvsD} is the RMS error between the estimated offset and the desired offset at steady state . The average RMS values for 10 runs were $RMS_{TvsE} = 0.33cm$ and $RMS_{EvsD} = 0.9cm$. Notice that both of the RMS errors are very small. However, these errors are expected to increase in the physical experiments because the curvature of actual ruts changes continuously and there will be more uncertainty in the initial state estimates of the filter $x_0 = [\theta_{vr_0}, \kappa_0, y_{off_0}]^T$.

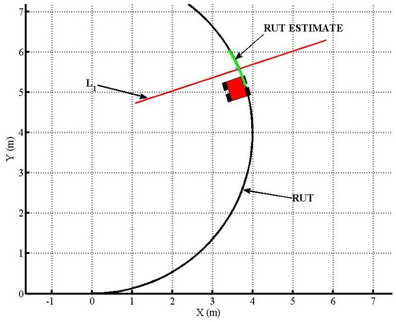


Fig. 16. Robot Following a Rut of Constant Curvature Using an EKF and the Proposed Steering Control

VI. CONCLUSIONS AND FUTURE WORK

A set of experiments on different robotic platforms and terrains were conducted to show the value of rut following for off road navigation. Then, the first stage of a rut detection and following system was designed, implemented and experimentally evaluated. The experimental results showed that the proposed system was able to detect and follow S-shaped ruts and it also showed its ability to follow ruts that have a lateral offset with respect to the robot. To increase the robustness of the proposed reactive system, a set of improvements including a probabilistic based rut detection approach and a tracking module based on an EKF were suggested and tested in simulation with promising results for future implementation.

A planner based subsystem needs to be developed to select the best rut to follow among several candidates and to provide a mechanism to initialize the EKF (i.e., provide the initial state values). In addition, a vision based approach to rut detection should be investigated because it would provide long range information to complement the current local information obtained with the laser range finder and open the possibility of detecting ruts based on different features (e.g., texture) and not only range. Therefore, shallower ruts could be detected.

ACKNOWLEDGMENT

Prepared through collaborative participation in the Robotics Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD 19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

REFERENCES

[1] W. Blevins. Land rover experience driving school. Class notes for Land Rover experience day, Biltmore, NC, 2007.

[2] T. Muro and J. O'Brien. *Terramechanics*. A.A. Balkema Publishers, 2004.

[3] Land rover lr3 overview mud and ruts.
Available: http://www.landroverusa.com/us/en/Vehicles/LR3/
Overview.htm,[Accesed: Aug. 12 2008].

[4] J. Allen. *Four-Wheeler's Bible*. Motorbooks, 2002.

[5] N. Baker. Hazards of mud driving.
Available: http://www.overland4WD.com/PDFs/Techno/muddriving.pdf,
[Accesed: Aug. 12 2008].

[6] 4x4 driving techniques. Available: http://www.ukoffroad.com
/tech/driving.html [Accesed: Aug. 12 2008].

[7] M. Saarilahti and T. Anttila. Rut depth model for timber transport on moraine soils. In *Proceedings of the 9th International Conference of International Society for Terrain-Vehicle Systems, Munich, Germany*, 1999.

[8] C. Ordonez and E. Collins. Rut Detection for Mobile Robots. In *Proceedings of the IEEE 40th Southeastern Symposium on System Theory*, pages 334–337, 2008.

[9] J. Laurent, M. Talbot, and M. Doucet. Road Surface Inspection Using Laser Scanners Adapted for the High Precision 3D Measurements on Large Flat Surfaces. In *Proceedings of the IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling*, 1997.

[10] W. Ping, Z. Yang, L. Gan, and B. Dietrich. A Computarized Procedure for Segmentation of Pavement Management Data. In *Proceedings of Transp2000, Transportation Conference*, 2000.

[11] V. Leemand and M. F. Destain. Application of the hough transform for seed row localization using machine vision. *Biosystems Engineering*, 94:325–336, 2006.

[12] K. Nagatani G. Reina, G. Ishigami and K. Yoshida. Vision-based Estimation of Slip Angle for Mobile Robots and Planetary Rovers. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008.

[13] Z. Kim. Realtime Lane Tracking of Curved Local Road. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2006.

[14] X. Hu Y. Zhou, R. Xu and Q. Ye. A robust lane detection and tracking method based on computer vision. *Measurement Science and Technology*, 17:736–745, 2006.

[15] R. C. Coulter. Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, 1992.

[16] LTD Hokuyo Automatic Co. *Range Finder Type Laser Scanner URG-04LX Specifications*.

[17] PNI Corporation. *TCM2 Electronic Compass Module User's Manual*.

[18] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, INC., 2001.

[19] L.B. Cremean and R.M. Murray. Model-based estimation of off-highway road geometry using single-axis ladar and inertial sensing. pages 1661–1666, May 2006.

[20] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, 23(9):661–692, 2006.