Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Rut Tracking and Steering Control for Autonomous Rut Following

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

***Abstract*—Ruts formed as a result of vehicle traversal on soft ground are used by expert off road drivers because they can improve vehicle safety on turns and slopes thanks to the extra lateral force they provide to the vehicle. In this paper we propose a rut detection and tracking algorithm for Autonomous Ground Vehicles (AGVs) equipped with a laser range finder. The proposed algorithm utilizes an Extended Kalman Filter (EKF) to recursively estimate the parameters of the rut and the relative position and orientation of the vehicle with respect to the ruts. Simulation results show that the approach is promising for future implementation.**

***Index Terms*—Rut detection, rut estimation, rut following, robot trace, robot track.**

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.

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], [3]. 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], [4], [5], [6], [7]. Besides the benefits of rut following already explained, proper rut detection and following can be applied in diverse applications like guidance of loose convoy operations, robot sinkage measurement, and estimation of the coefficient of rolling resistance [8]. On the other hand, a vehicle moving at high speed that hits a rut involuntarily can lose control and tip over. Therefore, an AGV provided with rut detection and rut following abilities can benefit from the correct application of this off road driving rule, thereby improving its efficiency and safety in challenging missions.

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

Initial research on rut detection [9], [10], [11] focused on the detection of ruts using single laser scans and did not considered the spatial continuity of the ruts. An improved rut detection and following system that incorporated the spatial continuity of the ruts by modeling the ruts locally using second order polynomials was then presented in [3]. The approach of [3] was experimentally evaluated on a Pionner 3AT robot on an S-shaped rut as illustrated in Fig. 2.

However, the approach of [3] did not make use of the spatio-temporal coherence that exists between the detections from consecutive laser scans while the vehicle is in motion. The integration of this spatio-temporal coherence, which is the main focus of the present work, is a crucial step towards a robust rut detection and following algorithm because it provides the robot with the ability of making predictions about the rut location relative to itself while in motion. In addition, by predicting the rut center locations before taking a measurement, it is possible not only to limit the search for ruts to a small region but also to handle more challenging rut following scenarios (e.g. scenarios with discontinuous ruts). This paper incorporates the spatio-temporal coherence between measurements by using a rut detection and tracking module based on an Extended Kalman Filter (EKF) that recursively estimates the parameters

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

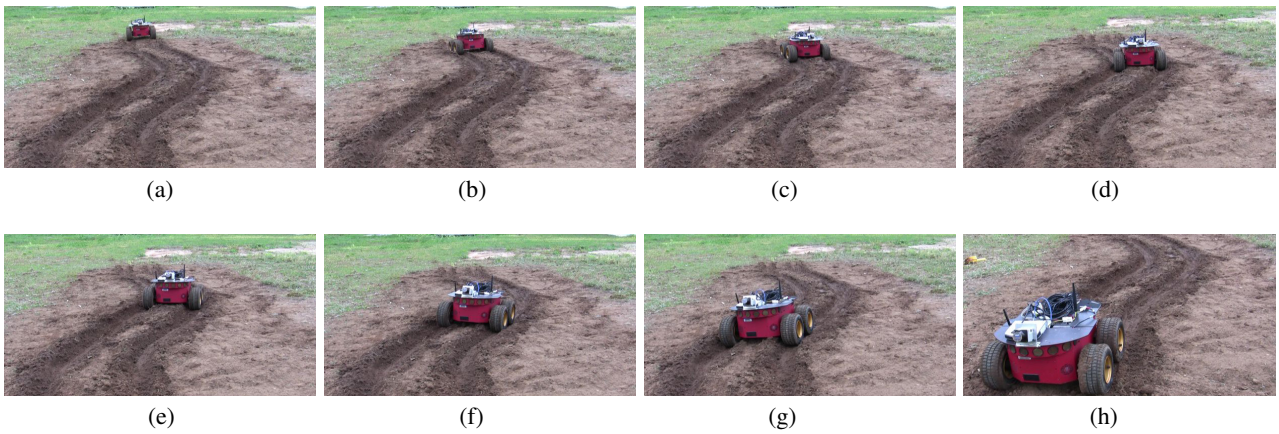

Fig. 2. Pioneer 3-AT Following an S-Shaped Rut

of the ruts (tracking) and then uses these estimates to improve the detection of the ruts for subsequent laser scans. In addition, the Kalman filter also generates smooth state estimates of the relative position and orientation (Ego-state) of the vehicle with respect to the ruts, which are the inputs to the steering control system used to follow the ruts.

Work related to the recursive estimation of the rut model parameters deals with the estimation of paved road center lines and road lanes. The vast majority of road tracking approaches rely on vision systems that detect the lane markings in the roads and then utilize filtering techniques such as Kalman or particle filters to recursively estimate the parameters of the road center lines, [12], [13], [14], [15], [16]. Since vision based systems are easily affected by illumination changes and road appearance (e.g. changes from paved surfaces to dirt), other researchers have used laser range finders as the main sensor to detect and track road boundaries [17], [18], [19].

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 [20]. This system was limited to straight seed rows and was tested in agricultural type environments, which are relatively structured. The work presented in [21] proposes 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 [8], where a correlation between the rut depth and the rolling resistance was presented. However, this work did not deal with the rut detection problem.

The main contributions of this paper are the development of a novel rut detection approach using a probabilistic framework and the incorporation and simulation evaluation of an EKF to simultaneously achieve smooth estimation of the rut parameters and the states required by the steering control system to follow the detected ruts.

The remainder of the paper is organized as follows. Section II describes the proposed approach to rut detection, tracking and following. Section III presents simulation results of the proposed approach using the EKF. Finally, Section IV presents concluding remarks, including a discussion of future research.

II. RUT DETECTION/TRACKING AND FOLLOWING

In the following discussion, it is assumed that the AGV is equipped with a laser range finder that observes the terrain in front of the vehicle. In addition, we assume that there are ruts in front of the vehicle. It is also important to clarify that in this work we follow ruts that were created by a vehicle with a similar track width. Although a desirable characteristic of the ruts this is not a requirement. In fact, in many situations, provided the vehicle has a good traction control system, off road drivers use ruts that do not match the track width of the vehicle [1].

The proposed approach to rut tracking and steering control is divided into two subsystems as shown in Fig. 3: 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 where a system based on only global information would fail. As shown in Fig. 3, the reactive system is composed of the stages described below.

A. Rut Detection and Tracking

The rut detection module analyzes the laser scans in the small search regions recommended by the tracking module, described in subsection II-A3. The rut detection module starts with the generation of rut templates as described in subsection II-A1. Then, rut candidates are generated using a probabilistic approach based on Bayes' theorem [22] as described in subsection II-A2.

Fig. 4 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.

1) Rut Templates Generation: Since ruts shapes are terrain and vehicle dependent, we propose to develop experimental models of traversable ruts (ruts that do not violate body clearance and have a similar width to the vehicle tire) that can be used for rut detection. Notice that these models can be generated offline on the terrains of interest and/or they can be updated online by having a laser looking at the ruts being created by the robot at run time. In this paper, we obtain the experimental models offline, using laser data from ruts created by a vehicle prior to the mission. The vehicle used to conduct experiments in this research is a Pioneer 3AT robot.

A set $\mathbb{S}_1$ of fifty rut cross sections was manually selected with ruts depths ranging from (3-6 cm) and rut widths ranging from (10-12 cm). The rut samples in this data set were obtained with the vehicle having relative orientations with respect to the ruts of (-20°, -10°, 0°, 10°, and 20°). Different orientations were used because the shape of the rut changes depending on the relative angle between the vehicle and the rut. $\mathbb{S}_1$ contains 10 rut samples for each orientation. Each rut sample contains 31 points equally spaced in their horizontal coordinates with 1 cm resolution and is chosen such that the center of the rut corresponds with the center of the sample.

One could use all of the rut samples in $\mathbb{S}_1$ as the rut templates for rut detection. However, this number is large, which can lead to slow rut detection, and may not properly cover the widths and depths of the ruts that the robot should identify. Hence, we used this data set to construct a small set of rut templates for rut detection. The first step was to use

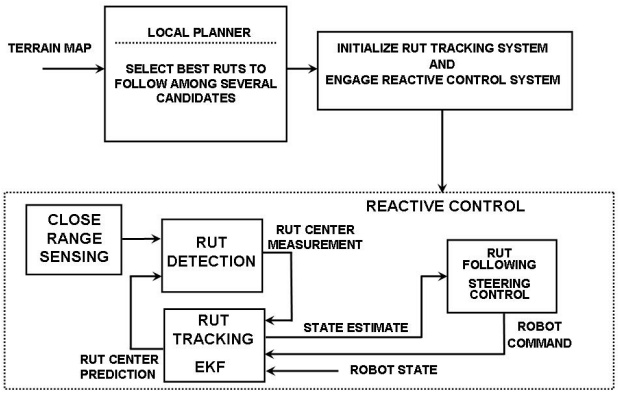

Fig. 3. Schematic of the Proposed Approach to Rut Tracking and Steering Control

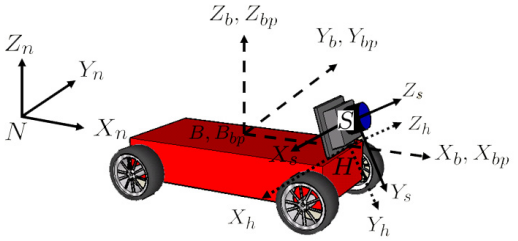

Fig. 4. Coordinate Systems

all of the samples in $\mathbb{S}_1$ to construct an average rut template $\overline{T}$ (see Fig. 5. In this case $\overline{T} = T_1$) with a width of 31 cm. From $\overline{T}$ we constructed, as detailed below, four rut templates to cover the desired range of rut widths and depths.

Let BC denote the vehicle body clearance and TW the width of any one of the tires. We assumed that ruts with depths in the range [0.5BC, 0.8BC] and with widths in the range [TW, 1.5TW] are traversable (see Fig. 6). In order to generate templates that cover this range we stretched (in the horizontal direction) and scaled (in the vertical direction) the average template $\overline{T}$ using a stretching factor (α_1) and a scaling factor (α_2). The stretching and scaling process is carried out using the linear transformation,

$$\begin{bmatrix} X_{T_j} \\ Z_{T_j} \end{bmatrix} = \begin{bmatrix} \alpha_{1_j} & 0 \\ 0 & \alpha_{2_j} \end{bmatrix} \begin{bmatrix} X_{\overline{T}} \\ Y_{\overline{T}} \end{bmatrix}, \tag{1}$$

where $[X_{\overline{T}}, Y_{\overline{T}}]^T$ are the coordinates of a point in the average template $\overline{T}$, and $[X_{T_j}, Y_{T_j}]^T$ are the corresponding coordinates of template T_j with $j = 1,2,3,4$. Four combinations of (α_1, α_2) were chosen so that each of the four resulting rut templates had rut width and depth corresponding to one of the four corners of the *Acceptable Region* of Fig. 6.

For the vehicle geometry of a Pioneer 3AT robot with a body clearance (BC) of 8 cm and tire width (TW) of 10 cm the above procedure generated the four templates shown in Fig. 5. These templates yielded accurate rut detection in our limited experiments. However, more experiments are needed to more rigorously evaluate their efficacy. Also, it is expected that a larger vehicle may encounter a wider variety of rut shapes, which may require a larger number of templates, which can be generated by using scalings corresponding to the interior of the *Acceptable Region* of Fig. 6.

2) Feature Selection and Rut Candidates: In the rut detection process, the closeness between the laser points in a window of $31cm$ (which is the width of the templates) and every rut template $\overline{\mathbb{T}}_i$ is determined by computing the sum of squared error (SSE). Then the minimum of these errors e_{min} is used as the feature to estimate the probability of a laser point being a rut center $P(Rut/e_{min})$. The posterior probabilities are computed using Bayes' theorem as follows:

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

TABLE I
RUT DETECTION RESULTS ON $\mathbb{S}_3$

Detection Rate	False Alarm Rate
96%	3%

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 [22] and a training set $\mathbb{S}_2$, which contains 100 rut samples with depths ranging from (5 - 6 cm) and rut widths ranging from (11-13 cm).

A separate testing set $\mathbb{S}_3$ containing 100 ruts samples with depths ranging from (4 - 6 cm) and widths ranging from (10-15 cm) was used to test the rut detection approach and the results are presented in Table I.

To better illustrate the rut detection process. Fig 7 shows the posterior probability $P(Rut/e_{min})$ estimation for each point of a laser scan obtained from a set of ruts in front of the vehicle. Notice that the two probability peaks coincide with the location of the ruts.

3) Proposed Approach to Rut Tracking: In this paper we propose an improvement to the previous approach to rut tracking presented in [3] by incorporating an EKF that recursively estimates the parameters of the ruts (curvature) and also generates smooth estimates of the vehicle state with respect to the rut, which is advantageous compared to the approach of [3] because these states can be used directly for the steering control as shown in subsection II-B. In particular, the EKF generates estimates of the rut curvature and the orientation

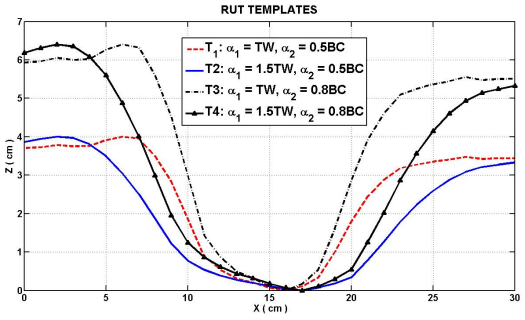

Fig. 5. Rut Templates

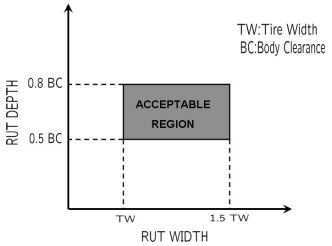

Fig. 6. Assumed Range of Rut Widths and Depths for Traversable Ruts

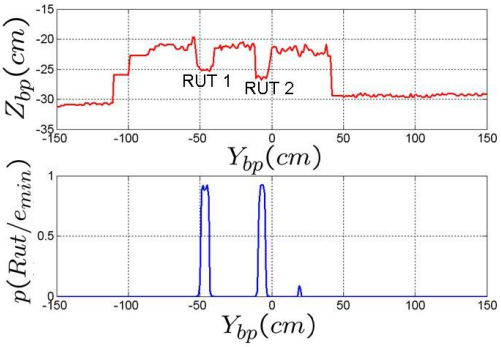

Fig. 7. Laser Data Containing Two Ruts (top) and Corresponding Probability Estimates of $P(RUT/e_{min})$ (bottom)

and lateral offset of the vehicle with respect to the rut.

3.1) Rut Model and Extended Kalman Filter

Motivated by the work of [18], which models the road center line using heading and curvature, we model the rut locally as a curve of constant curvature κ using frame R, which moves with the vehicle and is illustrated in Fig. 8. In addition, the X_r axis is always tangent to the rut and the Y_r axis passes at each instant through the kinematic center of the vehicle B. In frame R, the position of a point P in the rut as a function of arc-length (s) is given by

$$x_r(s) = \int_0^s cos(\theta(\tau))\, d\tau,$$
$$y_r(s) = \int_0^s sin(\theta(\tau))\, d\tau, \qquad (3)$$
$$\theta_r(s) = \kappa s$$

where θ is the orientation relative to frame R of the tangent vector to the curve at point P. Let us also define θ_r as the orientation of frame R relative to the inertial frame N. As the vehicle moves with linear velocity v and angular velocity $w = \frac{d\theta_v}{dt}$, the evolution of θ_r, θ_{vr} (the relative orientation between the vehicle and the rut), and y_{off} (the lateral offset between the robot and the rut) are computed using

$$\dot{\theta}_r = vcos(\theta_{vr})\kappa, \qquad (4)$$
$$\dot{\theta_{vr}} = w - vsin(\theta_v r)\kappa, \qquad (5)$$
$$\dot{y_{off}} = vsin(\theta_{vr}). \qquad (6)$$

Assuming that the evolution of the curvature is driven by white and Gaussian noise and after discretizing (5) and (6), using the backward Euler rule and sampling time δ_t, the process model can be expressed as

$$\begin{bmatrix}\theta_{vr_k}\\ \kappa_k\\ y_{off_k}\end{bmatrix}=\begin{bmatrix}\theta_{vr_{k-1}}-\kappa_{k-1}vcos(\theta_{vr_{k-1}})\delta_t\\ \kappa_{k-1}\\ y_{off_{k-1}}+vsin(\theta_{vr_{k-1}})\delta_t\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}\delta\theta_{v_{k-1}}+\omega_{\mathbf{k-1}}, \tag{7}$$

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

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 plane Π_1 and the rut (see Fig. 8). Using geometry, it is possible to express y_b as

$$y_{b_k}=-sin(\theta_{vr_k})x_m+\frac{1}{2}\kappa x_m^2cos(\theta_{vr})-y_{off_k}cos(\theta_{vr})+\nu_k, \tag{8}$$

where ν is a white noise with normal probability distribution ($p(\nu)\sim N(0,R)$). As shown in Fig. 8, 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

$$\frac{1}{2}x_m^2\kappa_k sin(\theta_{vr_k})+cos(\theta_{vr_k})x_m-(L+y_{off_k}sin(\theta_{vr_k}))=0, \tag{9}$$

where (9) is obtained as a result of a coordinate transformation from the rut frame R to the vehicle frame B. In compact form, we can rewrite (7) and (8) as

$$\begin{aligned}x_k&=f(x_{k-1},\delta\theta_{v_{k-1}})+\omega_{k-1},\\ y_{b_k}&=h(x_k)+\nu_k\end{aligned} \tag{10}$$

where x_k is the state of the process to be estimated, and $f(\cdot)$ and $h(\cdot)$ are nonlinear functions of the states and the model input and are given by

$$\begin{aligned}f(x_{k-1},\delta\theta_{v_{k-1}}) &= [f_1(x_{k-1},\delta\theta_{v_{k-1}}),\ f_2(x_{k-1},\delta\theta_{v_{k-1}}),\\ &\quad f_3(x_{k-1},\delta\theta_{v_{k-1}})]^T,\end{aligned} \tag{11}$$

$$h(x_k)=-sin(\theta_{vr_k})x_m+\frac{1}{2}\kappa_k x_m^2cos(\theta_{vr_k})-y_{off_k}cos(\theta_{vr_k}), \tag{12}$$

where

$$\begin{aligned}f_1(x_{k-1},\delta\theta_{v_{k-1}})&=\theta_{vr_{k-1}}-\kappa_{k-1}vcos(\theta_{vr_{k-1}})\delta_t+\delta\theta_{v_{k-1}},\\ f_2(x_{k-1},\delta\theta_{v_{k-1}})&=\kappa_{k-1},\\ f_3(x_{k-1},\delta\theta_{v_{k-1}})&=y_{off_{k-1}}+vsin(\theta_{vr_{k-1}})\delta_t.\end{aligned} \tag{13}$$

In the following discussion, we adopt the notation of [23], where $\hat{x}_k^-$ is our a priori state estimate at step k given knowledge of the process prior to step k. P_k^- is the a priori estimate error covariance and P_k is the a posteriori estimate error covariance.

The time update equations of the EKF are then given by

$$\hat{x}_k^-=f(\hat{x}_{k-1},\delta_{\theta_{v_{k-1}}}), \tag{14}$$

$$P_k^-=A_kP_{k-1}A_k^T+Q, \tag{15}$$

Equations (14) and (15) project the state and covariance estimates from the previous time step k-1 to the current time step k, $f(\cdot)$ comes from (11), Q is the process noise covariance and A_k is the process Jacobian at step k, which is computed using

$$A_{k_{[i,j]}}=\frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1},\delta_{\theta_{v_{k-1}}}). \tag{16}$$

Once a measurement y_{b_k} is obtained, the state and the covariance estimates are corrected using

$$K_k=P_k^-H_k^T(H_kP_k^-H_k^T+R)^{-1}, \tag{17}$$

$$\hat{x}_k=\hat{x}_k^-+K_k(y_{b_k}-h(\hat{x}_k^-)), \tag{18}$$

$$P_k=(I-K_kH_k)P_k^-, \tag{19}$$

where $h(\cdot)$ comes from (12), R is the measurement covariance, K_k is the Kalman gain and H_k is the measurement Jacobian, which is computed as

$$H_{k_{[i,j]}}=\frac{\partial h_{[i]}}{\partial x_{[j]}}(\hat{x}_k^-). \tag{20}$$

3.2) Data Association

In order to improve the efficiency and robustness of the rut detection and tracking algorithm, only a small region of the laser scan is analyzed for ruts. The search region is selected based on the distribution of the measurement prediction, which we assume follows a Gaussian distribution after the linearization process [24] and is given by

$$p(y_{b_k})=N(h(\hat{x_k}^-),H_kP_k^-H_k^T+R). \tag{21}$$

Any confidence interval can then be defined around the predicted measurement $h(\hat{x_k}^-)$. In this work, we use a search region of two standard deviations, which results in the following search region:

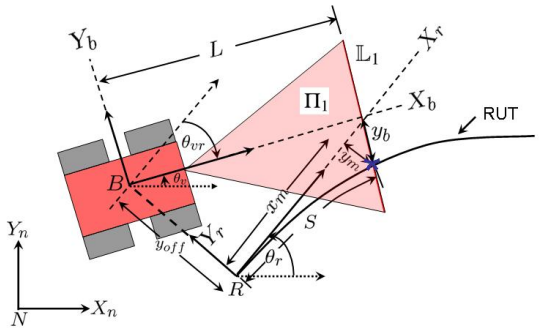

Fig. 8. Rut Frame Coordinates used for Rut Local Modeling

$$\mathbb{S}_r = \begin{cases} [\ h(\hat{x}_k^-) - 2\sigma,\ h(\hat{x}_k^-) + 2\sigma\ ] & \sigma \leq \overline{\sigma} \\ [\ h(\hat{x}_k^-) - 2\overline{\sigma},\ h(\hat{x}_k^-) + 2\overline{\sigma}\ ] & \sigma \geq \overline{\sigma} \end{cases} \quad (22)$$

where $\sigma = (H_k P_k^- H_k^T + R)^{0.5}$ and $\overline{\sigma}$ is a predefined threshold, which is set to 10 cm, such that $2\overline{\sigma}$ corresponds to half the width of the vehicle.

Once the search region $\mathbb{S}_r$ has been determined, the rut detection algorithm, explained in subsection II-A, is applied pointwise to each laser point within the region, which generates estimates of $P(Rut/y_b)$ the probability of having a rut given a measurement y_b. Recall that y_b is the distance between the X_b axis of the vehicle and the rut center (see Fig. 8).

Finally, we select as measurement y_{bo} such that:

$$y_{bo} = \arg \max_{y_b \in \mathbb{S}_r} p(Rut/y_b) P(y_b) \quad (23)$$

This sensor measurement is then used to compute the filter innovation (18). In case that $P(Rut/y_{bo}) \leq P_{th}$, where P_{th} is a predefined threshold, then no measurement is used and the filter uses the prediction without the update step.

B. Rut Following (Steering Control)

The proposed steering control is an adaptation of the controller proposed in [25]. The controller of [25] was developed for an Ackerman steered vehicle. Here, we approximate the vehicle kinematics using a differential drive model.

The EKF continuously generates estimates of the lateral offset y_{off} and the relative angle between the vehicle and the rut θ_{vr}, which are used to drive the vehicle towards the rut, Fig. 9.

In order for the robot to follow the ruts, θ_{vr} should be driven to zero and the lateral offset y_{off} should be driven to a desired offset $y_{off_{des}} = \frac{RobW + TireW}{2}$, where $RobW$ is the width of the robot and $TireW$ 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

$$\theta_{v_{des}} = \theta_r + \arctan\left(\frac{k_1 (y_{off_{des}} - y_{off})}{v}\right), \quad (24)$$

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 a proportional control law as follows

$$w = k_2(\theta_{v_{des}} - \theta_v) = k_2\left(\theta_{vr} - \arctan\left(\frac{k_1 (y_{off_{des}} - y_{off})}{v}\right)\right), \quad (25)$$

where w is the commanded angular velocity for the robot. Notice that (25) takes as inputs the state estimates generated by the EKF.

In order to avoid abrupt changes in the heading of the vehicle, saturation limits are imposed on (25), which leads to the final steering control scheme

$$w_c = \begin{cases} w & -w_{max} \leq w \leq w_{max} \\ w_{max} & w \geq w_{max} \\ -w_{max} & w \leq -w_{max} \end{cases} \quad (26)$$

where w_{max} is the maximum angular rate that would be commanded to the vehicle, and w_c represents the actual commanded angular velocity.

III. Simulation Results

To test the proposed approach, a computer simulation using Matlab was developed. A theoretical rut was simulated using a piecewise clothoid model similar to [18]. In the simulated rut, for each rut segment i which covers the arc-length interval $[s_i, s_{i+1})$, the curvature (κ_i) is modeled as $\kappa_i(s - s_i) = \kappa_{0,i} + (s - s_i)\kappa_{1,i}$, where $\kappa_{0,i}$ is the curvature at the beginning of the segment and $\kappa_{1,i}$ is the rate of change of curvature for the segment. In particular, for this simulation an S-shaped rut of 10 segments and maximum curvature of $0.5m^{-1}$ is employed.

The sensor measurements are simulated by finding the intersection of the laser L_1 with the rut as illustrated in Fig. 10. The lookahead distance is set to $45cm$ and the robot linear velocity is 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.

Fig. 10 illustrates the robot at the initial position with an initial offset of 1m and being parallel to the rut. Notice that the desired offset is $y_{off_{des}} = \frac{RobW + TireW}{2} = 25cm$.

As an evaluation metric we employ the RMS error between the true and estimated offsets RMS_{TvsE}, where the true offset is defined as the distance between the kinematic center of the vehicle B and the closest point on the rut. On the other hand, the estimated offset is the one estimated by the EKF.

Fig. 11 shows average values for the true, estimated and desired offsets for 10 simulation runs as a function of arc length. A second performance metric is the RMS error between the estimated offset an the desired offset on steady state RMS_{EvsD}. That is, neglecting the initial approach to the rut. The average RMS values for the 10 runs are $RMS_{TvsE} = 1.47cm$ and $RMS_{EvsD} = 1.52cm$.

IV. Conclusions and Future Work

A rut detection algorithm based on a probabilistic framework was designed and tested on experimental data collected

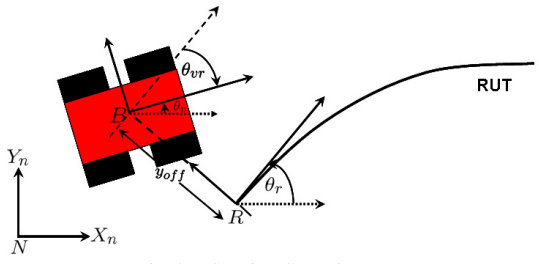

Fig. 9. Steering Control Parameters

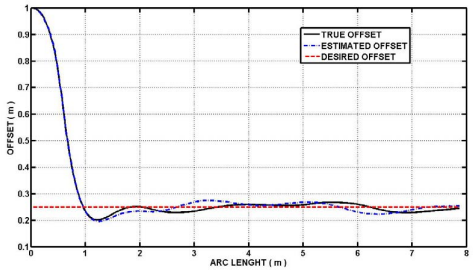

Fig. 11. Simulation Results on a S-shaped Rut

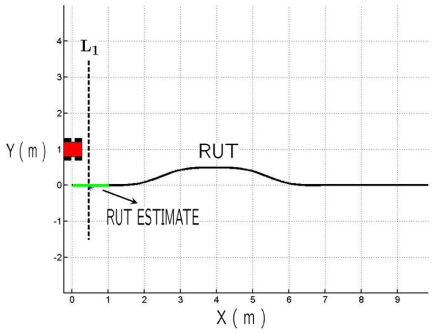

Fig. 10. Initial Configuration of the Robot with a Lateral Offset of 1m and Parallel to the Rut

using a laser range finder. Experimental results on real ruts showed good detection rates with a low false alarm rate. In addition, a rut tracking algorithm was proposed and tested in simulation using an EKF. Initial simulation results showed that the state estimates from the filter are good enough to perform rut following using the proposed steering control mechanism. The rut following simulations showed tracking errors of the lateral offset (Estimated vs Desired) of $RMS_{EvsD} = 1.52cm$ for steady state operation.

Future work will involve the experimental verification of the proposed system and the development of a planning based subsystem to select the best rut to follow among several candidates and initialize the EKF (i.e. provide the filter with the initial state estimate).

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] C. Ordonez, O. Chuy, and E. Collins. Rut detection and following for autonomous ground vehicles. In *Proceedings of Robotics: Science and Systems V*, Seatle, USA, June 2009.

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

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

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

[7] 4x4 driving techniques. Available: http://www.ukoffroad.com /tech/driving.html [Accesed: Aug. 12 2008].

[8] 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.

[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] 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.

[12] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):199–213, 1992.

[13] K.A. Redmill, S. Upadhya, A. Krishnamurthy, and U. Ozguner. A lane tracking system for intelligent vehicle applications. pages 273–279, 2001.

[14] D. Khosla. Accurate estimation of forward path geometry using two-clothoid road model. volume 1, pages 154–159 vol.1, June 2002.

[15] Yong Zhou, Rong Xu, Xiaofeng Hu, and Qingtai Ye. A robust lane detection and tracking method based on computer vision. *Measurement Science and Technology*, 17(4):736–745, 2006.

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

[17] W.S. Wijesoma, K.R.S. Kodagoda, and A.P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *Robotics and Automation, IEEE Transactions on*, 20(3):456–464, June 2004.

[18] 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.

[19] A. Kirchner and Th. Heinrich. Model based detection of road boundaries with a laser scanner. In *IEEE International Conference on Intelligent Vehiclesn*, pages 93–98, 1998.

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

[21] 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.

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

[23] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report TR 95-041, 2004.

[24] R. Negenborn. Robot Localization and Kalman Filters. Master's thesis, Utrecht University, Utrecht, The Netherlands, 2003.

[25] 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.