Heading prediction in unmanned ground vehicles by laser compass.

Mey Khalili, Richard Weatherly, Robert Bolling, Keven Ring, Bob Grabowski, Kevin Forbes, Cindy Cicalese

MITRE corporation

7515 Colshire Drive

McLean, VA 22102-7539

Email: {mkhalili, weather, rbolling, keven, rgrabowski, forbes, cicalese} @mitre.org

Abstract— **This paper presents an on-line algorithm which provides high signal to noise ratio heading predictions for unmanned ground vehicles. The algorithm uses cross correlation of SICK laser scans to improve the heading predictions from GPS. It is tested on our ground vehicles in outdoor urban environment and verified to provide accurate smooth heading predictions which help with the accurate localization of the vehicle.**

I. INTRODUCTION

Unmanned ground vehicles (UGVs) determine their position by three parameters: altitude, longitude and heading (i.e. yaw, referred here as η). Heading is the clockwise deviation from true North on the X-Y plane The accurate knowledge of heading is essential for autonomous localization of robots in an unknown environment. UGVs can obtain the heading from a variety of sources such as a magnetic compass, Global Positioning System (GPS) or Inertial Measurement Unit (IMU). A magnetic compass is sensitive to iron content in the environment and requires suspension which causes inaccuracy. IMU provides an estimate of the heading through double integration of sensed acceleration and suffers from integration drift caused by skidding and measurement errors. Usually GPS is the most accurate vehicle localization method. However, it can produce spikes (on the order of 10s of degrees) and requires line of sight to the satellite which rules out many indoor environments. It also produces no reliable heading at low speeds. Our vehicles operate mostly at speeds which produce low signal to noise heading from GPS due to their large size and safety issues. There has been extensive work using cameras as a vision compass to detect the rotational motion of a robot.[1]–[3] However, these algorithms usually require camera calibration parameters and in general are not suitable for real-time implementation. The objective of this paper is to design a fast and inexpensive on-line algorithm to obtain the vehicle heading from the cross-correlation of SICK LMS sweeps when GPS is unreliable.

The outline of the paper is as follows. The subsequent section introduces MITRE UGVs. Section III presents the algorithm's derivation and parameterization. section IV presents the tests of the algorithm and syncing it with GPS, section V concludes the paper.

II. ROBOTS

The MITRE Corporation has a number of UGVs. Probably the best known is a 2004 Ford Explorer Sport Trac, named Meteor.[4],[5] Meteor was designed to participate in the Defense Advanced Research Projects Agency (DARPA) Grand Challenge, the purpose of which was to design autonomous vehicles capable of completing a 132-mile, off-road course. Meteor was among 23 vehicles that qualified to compete in the final race.

FIG. 1: Centaur with its sensors.

MITRE also has a pair of Ontario Drive and Gear (ODG) DM950 Centaur Turbo diesel off-road utility vehicles (Figure 1). Since they are skid-steer, they are better suited as models of vehicles used in combat. Currently, the Centaurs have one GPS each and two SICK LMS, one scanning horizontally, and the other vertically. If GPS is unavailable or noisy, the vehicles have no other means of predicting heading. The Laser Compass algorithm was designed to mitigate this problem. The specifications of the horizontal SICK LMS are provided in Table I.

TABLE I: Relevant specifications of horizontal SICK LMS. Translation and rotation are with respect to vehicle's origin.

Translation (m)	Pitch, yaw, Roll (◦)	Range (m)	FOV (◦)	f (Hz)
1.6, 0.0, -0.8	0.0, 0.0, 0.0	80	180	20

All three UGVs have the same operating system architecture called wombat. The details of the design and implementation of wombat are presented elsewhere.[6]–[8] Wombat is a distributed system comprising the robots and the robot command center. The software architectures consist of agents which receive state information, change it, and pass it to other parts of the system. Messages from different sensors

and components regardless of the programming language, are broadcast via the UDP protocol on a given subnet. This scheme facilitates the use of multiple programming languages. Safety-critical and real-time parts of the system are written in the Ada. Navigation, maneuvering, user interaction, and visualization software are written in Java. The system runs on GNU/Linux and Windows.

III. METHODS

A. Cross-Correlation function

The cross-correlation of two signals $f[m]$ and $g[m]$ in an arbitrary dimension measures the degree of similarity between them as a function of the lag n in that dimension. The cross-correlation function is calculated as follows:

$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f[m] \, g[n+m]. \qquad (1)$$

$\max(f \star g)$ corresponds to the best fit between f and g, $\arg\max(f \star g)$ corresponds to n that will produce the best fit.

B. Transformation for Distance Correction

When Centaurs move forward, objects get larger from one sweep to another. To correct for this effect, we transformed the sweep at time t, $\boldsymbol{X}(t)$, to time $t-1$. Figure 2 shows the trigonometric relations of the transformation.
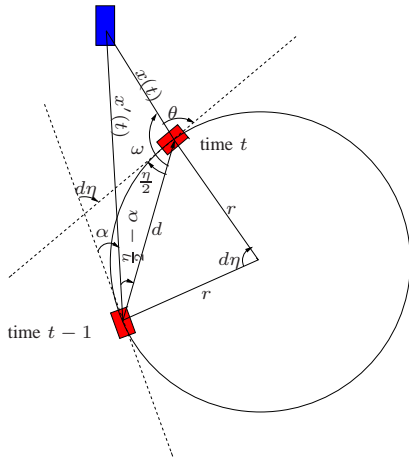

FIG. 2: Trigonometric relations for the transformation of one ray, $x(t)$, to $t-1$ ($x'(t)$). The red rectangle is the vehicle, the blue rectangle is an object, d is the distance traveled, $d\eta$ is the heading change

When the vehicle turns, we assume that it moves on an arc of a circle with radius $r = \frac{d}{2\sin(d\eta/2)}$, where $d \simeq v.dt$ (assuming dt is small enough so that the arc can be approximated by a straight line). $d\eta$ is the heading change during dt; dt is the time difference between the two consecutive sweeps $\boldsymbol{X}(t-1)$ and $\boldsymbol{X}(t)$. dt is obtained from the encoding time of laser messages; v is the speed of the vehicle and is obtained from the average speed of the left and right tracks of Centaur. When the vehicle moves on a straight line, $d\eta$ is

zero. In figure 2,

$$\omega = \pi - \theta + d\eta/2. \qquad (2)$$

$$x'_t = \sqrt{x_t^2 + d^2 - 2dx_t\cos(\omega)} \qquad (3)$$

$$d\eta/2 - \alpha = \sin^{-1}\left(\frac{x_t \sin(\omega)}{x'_t}\right) \qquad (4)$$

The formulas transform $\boldsymbol{X}(t)$ on $\boldsymbol{A}$ in figure 2. $\boldsymbol{A}$ is non-uniform and it does not correspond to the original vector of sweeps, $\boldsymbol{\Theta}$. To obtain the transformation on $\boldsymbol{\Theta}$, polynomial interpolation is employed to interpolate $\boldsymbol{X}'$ on $\boldsymbol{\Theta}$ from $\boldsymbol{A}$. The interpolation yields the new sweep vector $\boldsymbol{\Theta}'(t)$.

This transformation is non-linear when $d\eta$ is non zero. After the transformation and interpolation, it is possible for the range of $\boldsymbol{\Theta}'(t)$ to be different from $\boldsymbol{\Theta}(t-1)$. Only segments of $\boldsymbol{X}'(t)$ and $\boldsymbol{X}(t-1)$ corresponding to $\boldsymbol{\Theta}(t-1) \cap \boldsymbol{\Theta}'(t)$ are used in the cross correlation.

The transformation depends on $d\eta$ which is unknown. Initially $d\eta$ is set to zero, and the algorithm minimizes the root mean square deviation (RMSD) between $\boldsymbol{X}'(t)$ and lag-shifted $\boldsymbol{X}(t-1)$ with respect to $d\eta$.

C. Processing laser scans

Centaurs operate outdoors in a dynamic environment. LMS record the environment in polar coordinates, and objects that are at the same distance from the vehicles register on a parabola. An exception is a no-return. When the laser ray is lost due to hitting a black, shiny object or the horizon, the recorder registers the maximum possible distance, which is 81.9 m regardless of the angular orientation of the laser beam. We call these values no-returns.

No-returns throw off the correlation sum. Because a no-return is the largest value in the distance space of the SICK lasers, when there is a significant number of them present, the correlation function will correlate the no-returns as opposed to everything else. Depending on the environment, a great number of recordings can be no-returns as shown in a sweep histogram in figure 3B. Removing no-returns from the sweeps produced low-quality algorithm performance, as did removing them and interpolating between the remaining points with a number of different interpolation methods. We believe this is because no-returns provide information about the environment. If they correspond to the horizon or a large black car, they keep their relative location on two consecutive scans.

The following procedure proved successful. First, we removed all isolated no-returns. We calculated the histogram of the sweep and set all the no-returns to the median of the second largest histogram bin. An example is shown in figure 3.

D. Implementation notes

When the laser scans an obstacle at $t-1$, it has more information about the environment behind the obstacle than it does at time t. When the sweep at time t is transformed to $t-1$ it will have a wider field of view around obstacles, but no information about this environment. These blind spots
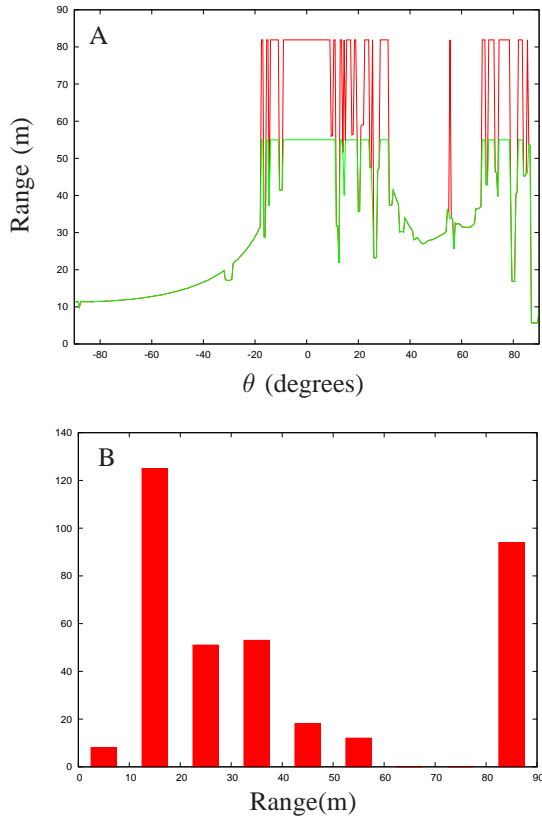
FIG. 3: A) An example of a laser sweep before and after scaling of no-returns. Red line is the original laser sweep. Green line is same sweep after cleanup. B) The histogram of the original laser sweep. The median of the largest bin after no-returns is 55 m.

were treated as no-returns and scaled. We also interpolated the laser sweeps to increase the resolution of the heading prediction. The default bin size is 0.5 degrees. We interpolated the sweep ten times using polynomial interpolation to reduce the bin size ($\Delta\theta_{int}$) to 0.05 degrees.

The cross correlation formula (Equation 1) is quadratic ($O(N^2)$), where $N = |\boldsymbol{X}(t)|$). The cross-correlation can be obtained from the Fourier transform which can be implemented in $N \log N$ time. We experimented with two Fourier transform implementations. The Java Fourier transform library of Apache turned out to be slower than the quadratic implementation which does not use Fourier transform for small arrays because of its extensive use of memory heap (Table II). We used an efficient implementation which does not use memory heap.[9] Table II shows the run times of the three implementations. The run times were obtained by generating arrays of 400, 4000 and 40,000 random numbers and cross correlating them 100 times. The outline of the algorithm is presented in Algorithm 1.

IV. RESULTS

A. Performance evaluation

The laser compass is a feed-forward algorithm and it does not have any mechanism for correcting mistakes, and it suffers from integration drift from ground truth over time. If

TABLE II: Running times (in seconds) of different Cross-correlation algorithms.

Array size	400	4000	40000
Quadratic algorithm	75	7389	734780
Apache Fourier	159	2992	90637
Fourier w/o memory heap	36	380	12070

Algorithm 1: Layout of laser compass algorithm

Data: $\boldsymbol{X}(t-1), \boldsymbol{X}(t), dt, v$
Result: Differential heading $\Delta_{\eta_{LC}}(t)$
begin
    Remove no-returns from $\boldsymbol{X}(t-1)$ and $\boldsymbol{X}(t)$;
    $error = \infty$;
    $\Delta_{\eta_{LC}}(t) = 0$;
    while $\Delta_{error} \neq 0$ **do**
        $(\boldsymbol{X}'(t), \boldsymbol{\Theta}'(t))$=transform$(\boldsymbol{X}(t), dt,$
        $\Delta_{\eta_{LC}}(t), v)$;
        Fill blind spots of $\boldsymbol{X}'(t)$;
        $[R_{min}, R_{max}]= \boldsymbol{\Theta}(t-1) \cap \boldsymbol{\Theta}'(t)$;
        $\boldsymbol{X}^I(t-1)$=Interpolate$(x(t-1, R_{min}:R_{max}))$;
        $\boldsymbol{X}^I(t)$=Interpolate$(x'(t, R_{min}:R_{max}))$;
        $C = (\boldsymbol{X}^I(t-1) \star \boldsymbol{X}^I(t))$;
        $j = \arg\max C$;
        $RMSD = \sqrt{\sum_i (x^I(t-1, i+j) - x^I(t,i))^2}$;
        $\Delta = j * \Delta\theta_{int}$;
        $\Delta_{error} = RMSD - error$;
        if $\Delta_{error} < 0$ **then**
            $error = RMSD$;
            $\Delta_{\eta_{LC}}(t) = \Delta$;
        end
    end
    return $\Delta_{\eta_{LC}}(t)$
end

GPS signal is available and reliable, it is used to correct for the drift. Initially, we evaluated the algorithm's performance without any GPS corrections. We collected two short log files on Centaur 1 operating in the MITRE parking lot. We compared the GPS heading with the laser compass headings. The graphs are shown in figure 4A and B. As expected, the drift between laser compass and ground truth increases with time. However, the compass can reproduce ground truth's features remarkably well.

To determine the performance of the algorithm for more dynamic environments, we performed a long run with Centaur 2 where we started the robot in the MITRE parking lot and then drove it for 20 minutes on the MITRE campus, as well as nearby streets and highways (Figure 4C). In figure 4C, we reset the laser compass heading twice to GPS at 410 s and 500 s to make comparisons of the two graphs easier. From figure 4C, it is evident that the laser compass at times underestimates the turns significantly. For example, at 550 s, it underestimates the turn by 125 degrees. The reason for the poor performance for this segment is due to a lack of
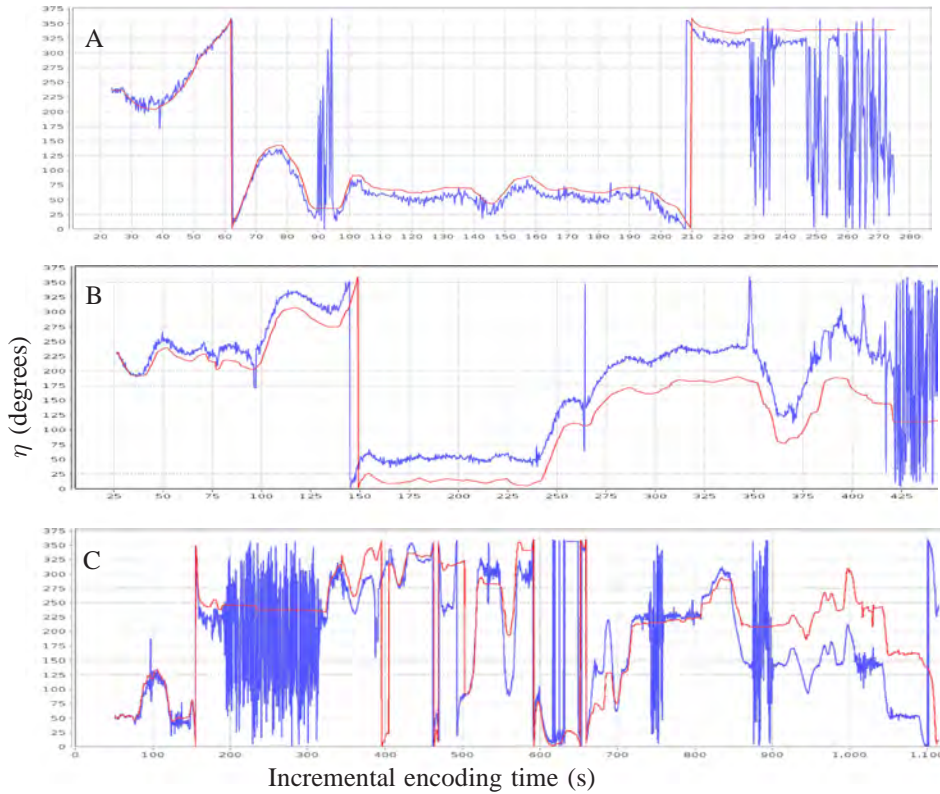
FIG. 4: (A and B) Two short runs in the MITRE parking lot comparing the performance of Laser compass (red line) and GPS (blue line). At the end of both runs, Centaur is parked. The drift gets larger over time. C) A 20 minute run comparing the performance of the algorithm with GPS. Laser compass was reset to GPS twice at 410 s and 500 s to make comparison easier. The algorithm suffers from lack of features in the laser sweeps especially when the robot is taking sharp turns (for example at 550 s).

features across consecutive laser sweeps. During this time, the robot is passing an intersection to cross a highway and there are few features within the laser's recording range. A camera snapshot of Centaur's field of view during this time (figure 5A) shows that the vehicle is taking a sharp turn in an environment where there is nothing within the recording range. The lack of features means that consecutive laser sweeps are very similar although the environment changes significantly (figure 5B).

Another problem for the algorithm are large dynamic features which register significantly on SICK lasers, especially when Centaurs themselves are stationary. For example, when Centaur is stopped at the red light in front of a busy highway, large trucks and buses result in heading change (data not shown). To alleviate this effect, we employed two measures (see section C for implementation details). We do not update the heading when the average speed of the vehicle is zero, and we put a maximum threshold ($\Delta_{\eta_{LC}}(t) < 10^o$) on the amount of heading change we allow. Usually a fast moving object results large heading derivative (on the order of 20°) in two consecutive sweeps (in about 0.2 s). We discard these values and use the last best known heading estimate (Δ_{good}, see section C). With a current single SICK laser, the laser compass is data limited. This algorithm should work well when there are multiple laser sweeps per laser (e.g. Velodyne lidar). We are currently in the process of
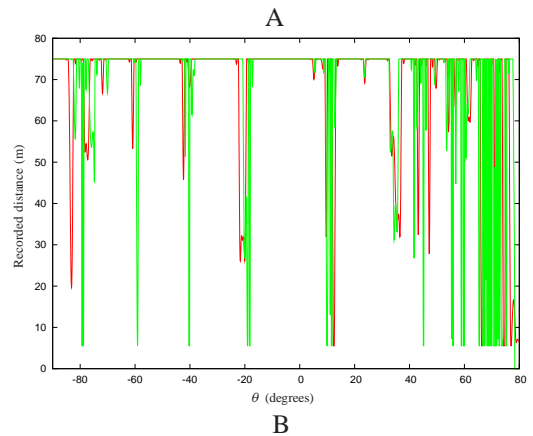

FIG. 5: A) A camera snapshot of Centaur around $t = 550$ s in figure 4C. B) Two consecutive laser sweeps during the same time.

implementing a Velodyne lidar on our Centaurs. However, for the current instantiation, the laser compass needs periodic re-synchronization with a GPS when it is available and reliable.

B. Statistics

The algorithm uses a number of parameters to calculate the heading. We performed a statistics-based study of these parameters to make sure that they are reliable. The statistics were calculated from two hours of combined runs from both centaurs.

The laser compass uses the speed of the Centaur's tracks to calculate the average and assumes that the speed of the vehicle is equal to this average. For this assumption to be valid, the speed of the two tracks have to be almost the same. Figure 6 shows the two tracks plotted against each other. The coefficient of correlation between them is more than 0.9.
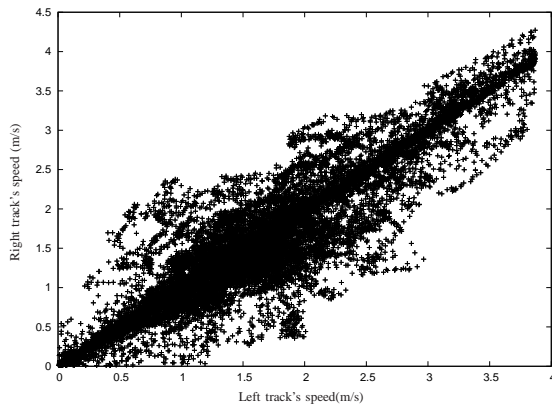

FIG. 6: Comparison of Centaur's left track speed with right track speed. There is very good correlation between the speed of the two tracks with the coefficient of correlation of 0.9.

To determine how accurate the average speed is compared to ground truth, we compared it to GPS speed. In general, we found a very good correlation between GPS and the average track speed (data not shown), except at points where the GPS speed made large excursions. As expected, most of these excursions happened at low speeds. Once GPS outliers were removed, we found very good agreement between GPS speed and average track speed with the correlation coefficient more than 0.9. It is interesting to note that the magnitude of the errors in the GPS speed are similar on the same days.

GPS heading does not depreciate linearly with the decrease in speed of the vehicle. Due to its internal correction models, GPS produces relatively small noise in the heading magnitude before suddenly breaking down. To determine this break-point, we calculated the mean and standard deviation of GPS heading and GPS speed over a 20 point data window. Since we needed the speed to be very accurate, we removed all the data points from our samples where the standard deviation of speed was more than 0.1 m/s. Figure 7 shows that once the speed falls below 0.5 m/s, heading is not reliable. We did not merge GPS with laser compass when speed was less than 1 m/s (see section *C*).
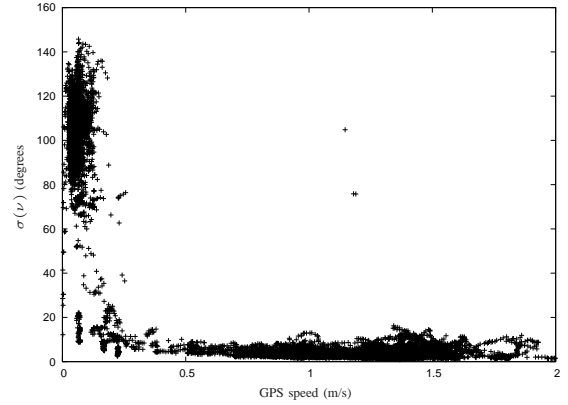

FIG. 7: Heading from GPS becomes unreliable once the speed of vehicle falls below 0.5 m/s. There are a few outliers on the graph around 1.25 m/s.

C. Merge with GPS

The laser compass produces much less noisy heading predictions than GPS. When merging the two, it would be desirable to keep this feature of the laser compass in merged heading. To sync the two, we used an α-β tracker.[10] We first calculated the derivative of GPS

$$\Delta_{GPS}(t) = \eta_{GPS}(t) - \eta_{GPS}(t-1) \qquad (5)$$

and used $\sigma(\Delta_{GPS})$ (i.e. the standard deviation of the derivative of GPS) as the criteria for merge. From our statistics-based study, we chose the following parameters for our filter

$$\alpha = \begin{cases} 0 & \sigma(\Delta_{GPS}) > 5° \text{ or } v < 1m/s \\ \frac{1}{20} & \text{otherwise} \end{cases}$$

and

$$\beta dt = \begin{cases} 0 & v = 0 \\ \Delta_{\eta_{LC}}(t) & \Delta_{\eta_{LC}}(t) < 10° \\ \Delta_{good} & \text{otherwise} \end{cases}$$

we calculated the merged heading ($\hat{\eta}$) by the three iterative equations of α-β filter

$$\hat{\eta}(t) = \hat{\eta}(t-1) + \beta dt \qquad (6)$$

$$r(t) = \eta_{GPS}(t) - \hat{\eta}(t) \qquad (7)$$

$$\hat{\eta}(t) = \hat{\eta}(t) + \alpha r(t) \qquad (8)$$

Figure 8 shows to relative long runs comparing merged heading with GPS heading. The first row corresponds to the run of Figure 4C. The spike in the GPS heading at t=370 s is due to GPS error. GPS heading has a large standard deviation at this point and does not get merged. The second row corresponds to a run of Centaur 1 in MITRE parking lot.
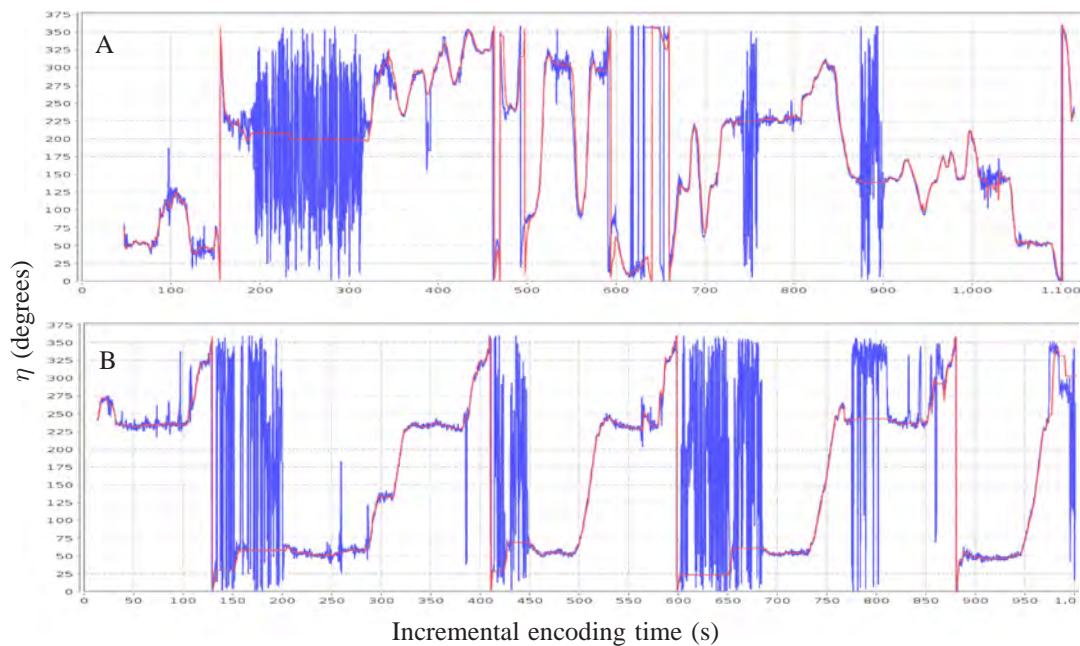
η (degrees)

Incremental encoding time (s)

FIG. 8: Merged heading for two runs. A) long run of Centaur 2 in figure 4C. The GPS spike at $t = 370\,$s has a large σ and it does not get incorporated to the merged heading. B) A run of Centaur 1 on MITRE parking lot.

V. CONCLUSIONS

This paper presents an on-line algorithm that uses SICK laser scans and cross-correlation function to improve the heading predictions of GPS in UGVs. The algorithm produces heading that has higher signal to noise ratio than that of GPS alone.

There has been extensive work in the field of simultaneous localization and mapping (SLAM).[11], [12] Algorithms that solve SLAM, provide not only heading, but also the displacement along the x and y direction, as well as the map of the environment. However, SLAM algorithms are usually computationally expensive and have various shortcomings. If GPS is available and reliable, it out-performs SLAM algorithms in localization. However, GPS suffers from outages and high signal to noise ratio at low speeds or spikes. In this paper, we have addressed these issues.

Our algorithm can also be used to improve the performance of motion models. In most motion models, the new x and y displacement are determined from heading by projecting the heading onto the x and y coordinates. If there is no means of heading prediction available, the heading is drawn from a random distribution. This ties the motion model to the type of the distribution picked and whether the assumption is valid. The error in heading can propagate to x and y displacement and result in poor localization. Using laser compass as a source of heading as opposed a randomly sampling it from a probability distribution can improve the quality of motion models significantly. We are in the process of implementing this concept for Centaurs.

REFERENCES

[1] J. Kosecka and W. Zhang, "Video compass," *7th European Conference on Computer Vision*, pp. 476–491, 2002.

[2] A. Makadia, C. Geyer, S. S. Sastry, and K. Danillidis, "Radon-based structure from motion without correspondences." *IEEE Conference on Computer Vision and Pattern Recognition.*, pp. 796–803, 2005.

[3] J. M. M. Montiel and A. J. Davison, "A visual compass based on slam," *IEEE international Conference on Robotics and Automation*, pp. 2403–2408, 2007.

[4] R. J. Garbowski, R. M. Weatherly, R. H. Bolling, D. Seidel, M. Shadid, and A. Jones, "Mitre meteor: An off-road autonomous vehicle for darpa's grand challenge." *Journal of field Robotics, special issue on the DARPA grand challenge.*

[5] R. Bolling, R. Garbowski, A. Jones, M. Shadid, and R. Weatherly, "The mitre meteorites 2005 darpa grand challenge entry," *http://www.darpa.mil/grandchallenge05/TechPapers/MITREMeteorites.pdf.*

[6] C. Cicalese, R. M. Weatherly, J. Sherrill, R. H. Bolling, K. Forbes, R. J. Grabowski, K. Ring, and S. Seidel, "A distributed, multi-language architecture for large unmanned ground vehicles." *Proceedings of the 2008 ACM annual international conference on SIGAda annual international conference.*, pp. 133–138, 2008.

[7] R. M. Weatherly, F. S. Kuhl, R. H. Bolling, and R. J. Grabowski, "The mitre meteor robot control software: Simulate as you operate." *Proceedings of the 38th Winter Simulation Conference Dec. 3-6*, pp. 1294–1298, 2006.

[8] R. M. Weatherly and E. H. Page, "Efficient process interaction simulation in java: Implementing co-routines within a single java thread." *Proceedings of the 36th Winter Simulation Conference Dec 5-8*, pp. 1437–1443, 2004.

[9] Http://cnx.rice.edu/content/m12016/latest/.

[10] T. L. Ogle and W. D. Blair, "Fixed-lag alpha-beta filter for target trajectory smoothing," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1417 – 1421, 2004.

[11] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 13, p. 99110, 2006.

[12] ——, "Simultaneous localisation and mapping (slam):part ii state of the art," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 108–117, 2006.