Path Tracking and Localization in Forest Environment

Thomas Hellström, Ola Ringdahl, Arsalan Siddiqui

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

August 30, 2006

Abstract

This paper describes an ongoing design and development project of an autonomous path-tracking forest machine. The work is part of a long-term vision in the forest industry of developing an unmanned shuttle that transports timber from the felling area to the main roads for further transportation. The developed prototype system has two modes of operation: Path Learning, in which the human operator drives or remote controls the vehicle along a selected path back and forth from the area of felling to the transportation road. In this phase, position, speed, heading, and the operator's commands are recorded in the vehicle computer. When the vehicle has been loaded with timber the operator activates Path Tracking mode, which means that the vehicle autonomously drives along the recorded path to the transportation road. A new path-tracking algorithm is introduced, and is demonstrated as superior to standard algorithms, such as Follow the Carrot and Pure Pursuit. This is accomplished by using the recorded data from the path-learning phase. By using the recorded steering angle, the curvature of the path is automatically included in the final steering command. Localization is accomplished by fusing data from Real-Time Kinematic Differential GPS/GLONASS, gyro, wheel odometry, and laser odometry. The laser odometry algorithm works by using consecutive scans to estimate the pose change (position and heading). A search is conducted in pose space to find the optimal fit between the two scans. Test results for path tracking and localization accuracy from runs conducted on the full-sized forest machine are presented.

1 Introduction

The forest industry has a long-term vision of developing unmanned shuttles that transport the timber from the area of felling to the main roads for further transportation [5]. This paper describes the IFOR navigation project [6], an ongoing project of designing and developing an autonomous path-tracking forest machine as part of that vision. The main advantages to an unmanned shuttle are lower labor cost and, due to a lower weight of the vehicle, less emissions and ground damage. The resulting system has two modes of operation: *Path Learning*, in which the human operator drives or remote-controls the vehicle along a selected path back and forth from the area of felling to the transportation road, while the vehicle learns the path. In this phase, position, speed, heading and the operator's commands are recorded in the vehicle computer. When the vehicle has been loaded with timber (this subtask could also be done

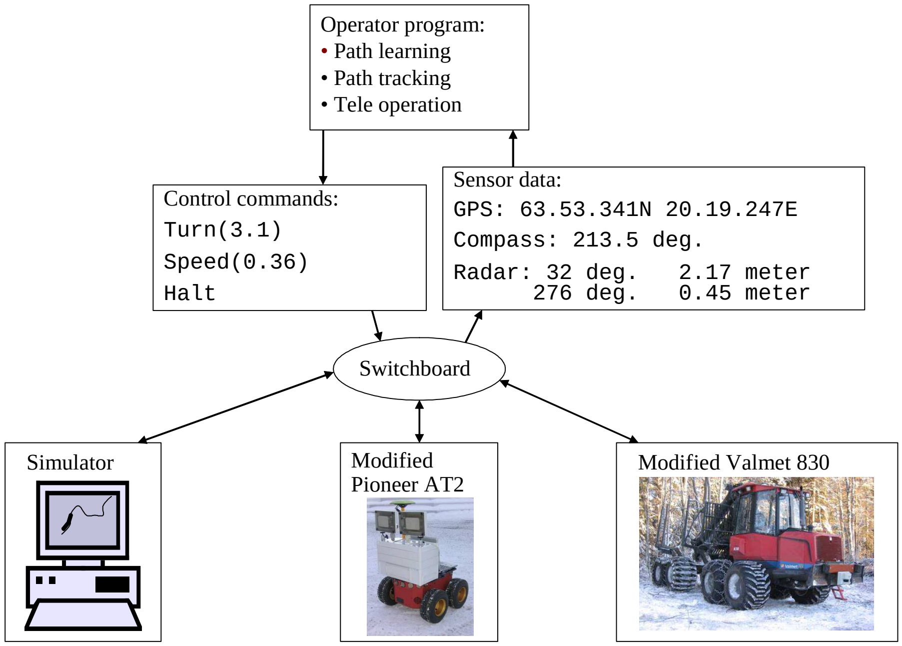


Figure 1: The work has been conducted on three different target machines, each with increased complexity. This approach greatly simplifies the research and development of both hardware and software.

autonomously, but is not considered in this project) the operator activates *Path Tracking* mode, in which the vehicle is able to autonomously track the learnt path back to the transportation road. The vehicle is also able to handle unexpected events, such as avoiding any obstacles in the way and compensate for irregularities in the terrain or noise in the positioning sensors. If the vehicle ends up at the side of the learnt path for any reason, it is able to autonomously steer back towards the path again.

Testing algorithms on the full-size forest machine is both impractical and inefficient. Therefore, the work has been conducted on three different target machines, each with increased complexity. As illustrated in Figure 1, the same main program can control any of the three target machines through a software switchboard. Likewise, sensor data passes from the target machine to the main program. In this way, high-level routines like path tracking are easily developed and implemented by the use of a simple simulator [10]. The simulator implements no sophisticated sensor models, and has a simplified kinematics model for propulsion, but serves well its purpose for debugging and testing the developed algorithms. The user interface is also easily developed using the simulator as the target machine. The infrastructure for sensors or actuators, and the modules for communication between the two main computers are most conveniently developed on the small-size Pioneer AT2 robot. Various types of sensors are also evaluated on this target machine. In the current phase of the project, the system is moved to the real forest machine, and the routines for vehicle control are fine-tuned and tested. Also, reliable sensor tests are only possible using this final target machine.

2 Localization

To navigate safely through the forest, we need to know where we are and in which direction we are heading. To do this we use several different sensors fused together. Our main sensor is a GPS that gives both accurate position and heading as described in Section 2.1. The satellite navigation technology has limitations that make a GPS system insufficient as the single position sensor for an autonomous moving vehicle. The most common problems involve [4] obstruction of line-of-sight to satellites, multi-path problems and active jamming from other RF sources. A GPS system is therefore often combined with INS or wheel odometry. In our case we use an AHRS400 gyro from Crossbow Technology to get an accurate heading when accuracy of the GPS drops. As a secondary position sensor, we use wheel odometry to estimate the vehicle's position. As described in Section 2.2, we have developed a laser scanner odometry sensor which is much more accurate than wheel odometry. When the accuracy of the GPS pose drops, we fuse the information from the secondary sensor(s) to get a reliable pose estimate. Because the GPS antennas are mounted on top of the forest machine, about three meters above ground, the GPS position varies due to the pitching and rolling of the vehicle while moving on uneven ground. To compensate for this, we use the roll and pitch readings from the gyro to calculate the correct pose for the vehicle.

2.1 Satellite Navigation

Our main position sensor is Javad's Real-Time Kinematics Differential GPS (RTK DGPS). RTK means that the receiver uses the carrier wave phase in addition to measuring on the code phase as in ordinary GPS applications. DGPS means that a stationary GPS receiver is connected by a radio link to a mobile GPS receiver. Correction signals for timing, ionospheric and tropospheric errors are transmitted by radio from the stationary to the mobile GPS, resulting in a centimeter accuracy under ideal conditions. The Javad receiver is capable of receiving signals from both American GPS system and Russian GLONASS system. While providing a lower accuracy than the GPS, GLONASS provides important backup, especially at high latitudes (64 degrees north), at which the work has been conducted, thanks to the inclination angle of 64.8 degrees. In addition to position, satellite navigation can give the speed and heading of the vehicle. To calculate this, the difference between two consecutive positions are used. This method gives quite low accuracy, especially when standing still. To increase the accuracy of the heading, we use two GPS antennas (and receivers), which give an accuracy about $0.3°$ if the antennas are placed one meter apart from each other. The position accuracy is in the order of a few centimeters. When the carrier wave phase measurements fail we get a *fix loss* and the accuracy of the position drops to 0.5 - 1 meter, and the heading to several degrees.

2.2 Laser-based localization

The general idea for most laser-based localization techniques is to compare two or more laser scans taken from different viewpoints, but covering at least partly, the same objects in the environment. By comparing the scans, the change in robot pose (position and heading) can be estimated. A number of algorithms for this have been proposed. Bailey and Nebot [1] developed a method based on matching identified landmarks such as cylinders and corners. Selkäinaho [11] proposed an efficient pixel-based matching method that works in unstructured outdoor environments. Similar techniques have been used in indoor environments [9] [1]. In our work we have adopted and modified the algorithms described in [11].

The techniques for estimated pose changes can be used in two major ways. For relative localization, the scans are generated with a very short time difference, and represent two vehicle positions very close to each other. The change in vehicle pose is estimated by finding the optimal transformation (translation and rotation) that makes the two scans match each other. The technique is often called laser odometry and, just like ordinary wheel odometry, it suffers from accumulation of errors. For global localization, the vehicle records a database with reference scans during one or many passages along a fixed route. For localization, the current laser scan is compared to the database, to find the best matching one(s). In this way the vehicle's position is estimated relative to the positions at which the reference scans were recorded. This corresponds to a global localization. There is no accumulation of localization errors in this method since the recorded reference scans are used in each step of the localization.

In the presented work, we implement relative localization or laser odometry. The estimated pose changes are used in situations where the GPS loses its fix solution (or the GPS signal disappears completely). The accuracy of the laser odometry decreases over time as illustrated in Figure 3 and 2. The (a)-figures compare heading and position respectively for GPS and laser odometry. The (b)-figures show how the difference (error) increases over time. The values plotted at time T, err_T, are averages of all drifts for periods of length T, i.e the drift we can expect after a time T after the GPS failed, refer to Equations 1 to 4. To calculate the heading error, the same equations hold, with the difference that heading is not a vector as position is.

$$\Delta GPS_t = GPS_{t+T} - GPS_t \tag{1}$$

$$\Delta Odo_t = Odo_{t+T} - Odo_t \tag{2}$$

$$drift_t = ||\Delta GPS_t - \Delta Odo_t|| \tag{3}$$

$$err_T = \frac{\sum_{t=1}^{N-T} drift_t}{N-T} \tag{4}$$

where

GPS_t : GPS-position at time t (x, y)

Odo_t : Odometry-position at time t (x, y)

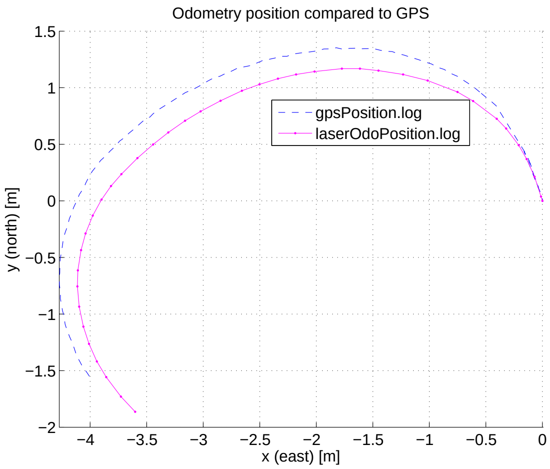
err_T : Average of all drifts for periods of length T.

3 A new path-tracking algorithm - Follow the Past

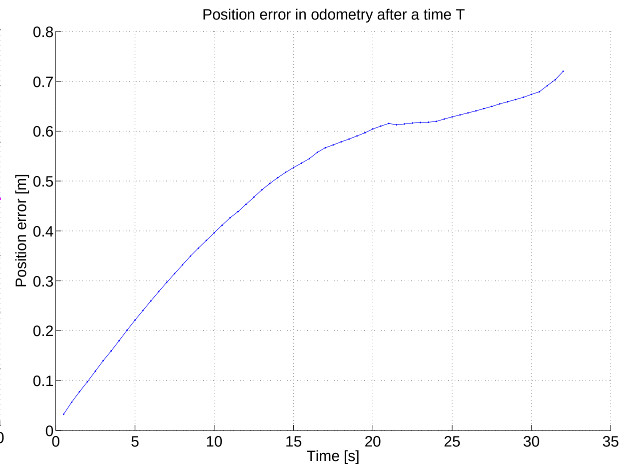
We developed a new path-tracking algorithm, called Follow the Past that makes use of the fact that an operator drives the path once and records data on that run. The idea behind the algorithm is to drive exactly as the driver did as long as the vehicle is on the learnt path. If the vehicle deviates from the learnt path, the algorithm adds a term to get back to the path again. This could happen for example when avoiding obstacles, because of inaccurate sensor readings or wheel slippage. The algorithm uses three reactive behaviors to achieve this:

- Turn towards the recorded vehicle orientation
- Mimic the recorded steering angle
- Move towards the path if the vehicle is too far away from it

These three behaviors are then fused into one steering command to the vehicle. More details can be found in [7],[8].

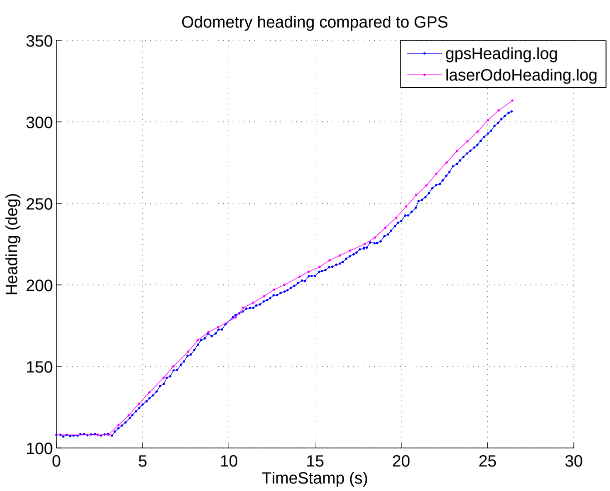


(a) Position from GPS and laser odometry

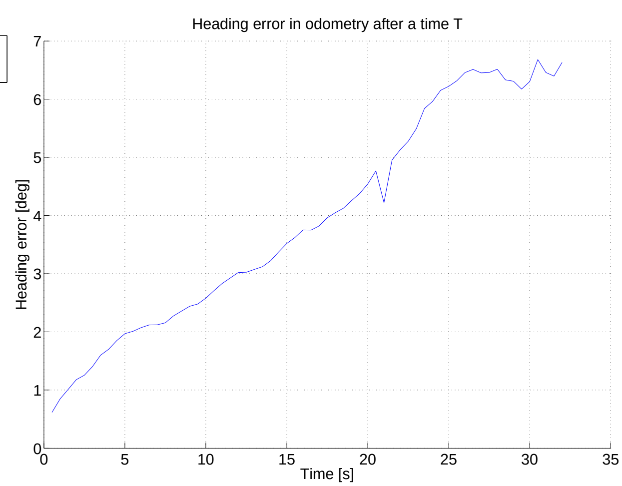


(b) Mean difference between the position from laser odometry and GPS after a time T.

Figure 2: Laser odometry position compared to position from the GPS when driving along an arc. The difference between the two sensors is less than one meter and normally increases with time. However, by pure chance the estimated path may sometimes converge to the actual path, depending on its shape. This may result in a locally decreasing error as in the end of the graph above.



(a) Heading from laser odometry and GPS.



(b) Mean difference between the heading from laser odometry and GPS.

Figure 3: Laser odometry heading compared to heading from the GPS for the same run as in Figure 2. The difference between the two sensors is less than seven degrees after 30 seconds.

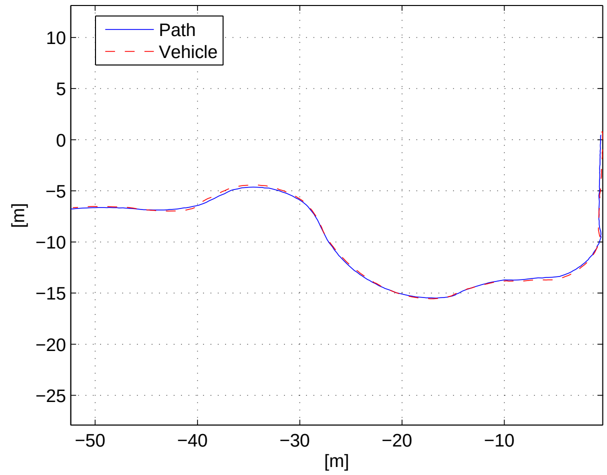


Figure 4: By using Follow the Past the vehicle is able to follow the path with minimal errors. The vehicle diverges at most 0.24 meters from the path. The average distance error is 0.13 meters.

3.1 Results

The developed Follow the Past algorithm is tested in a forest environment on a forwarder equipped with a GPS for position and heading, and a gyro for sensor fusion and pose transformation. In the following section the algorithm is compared to implementations of the Follow the Carrot and Pure Pursuit path-following algorithms. The vehicle's path-tracking abilities are presented graphically and with numerical error measurements in the form of max and mean deviation from the path. As shown in Figure 4, the vehicle is able to follow the path with minimal errors by using Follow the Past. Figures 5 and 6 show the vehicle tracking the same path with two common path-tracking algorithms; Follow the Carrot [2] and Pure Pursuit [3] methods. We can see that Follow the Past is able to avoid the problem of "cutting corners" that both Follow the Carrot and Pure Pursuit have. The *look-ahead distance* parameter in the algorithms is set to six meters in this example, as this gave the best results for Follow the Carrot and Pure Pursuit. To minimize the effect of uneven ground, these tests were performed on a reasonably flat surface.

References

[1] T. Bailey and E. M. Nebot. Localisation in large-scale environments. In *Robotics and Autonomous Systems 37*, pages 261–281, 2001.

[2] Matthew J. Barton. *Controller Development and Implementation for Path Planning and Following in an Autonomous Urban Vehicle*. Undergraduate thesis, University of Sydney, November 2001.

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

[4] H Durrant-Whyte. A critical review of the state-of-the-art in autonomous land vehicle systems and technology. Technical report, Sandia National Laboratories Albuquerque, 2001.

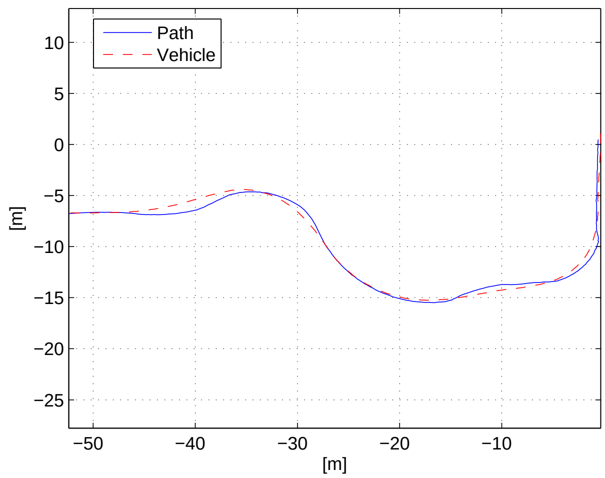


Figure 5: By using the Follow the Carrot the vehicle diverges at most 1 meter from the path. The average distance error is 0.29 meters.

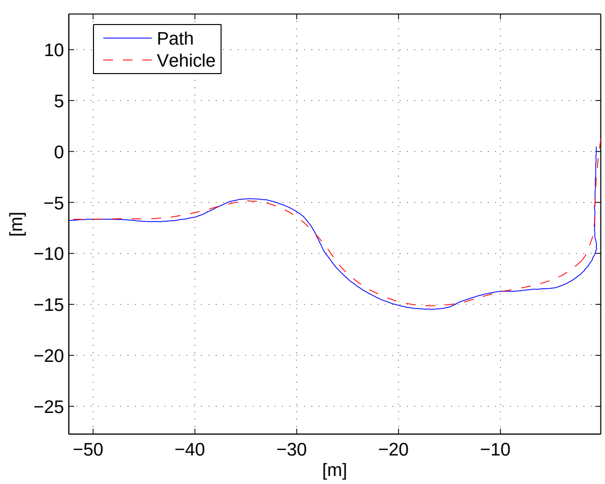


Figure 6: By using the Pure Pursuit the vehicle diverges at most 0.92 meters from the path. The average distance error is 0.33 meters.

[5] Ulf Hallonborg. Förarlösa skogsmaskiner kan bli lönsamma [autonomous forest machines can be profitable]. *Skogforsk RESULTAT*, (9), 2003.

[6] Thomas Hellström, Thomas Johansson, Ola Ringdahl, Fredrik Georgsson, Kalle Prorok, and Urban Sandström. *Development of an Autonomous Forest Machine for Path Tracking*, volume 25 of *Springer Tracts in Advanced Robotics*, pages 603 – 614. Springer, field and service robotics: results of the 5th international conference edition, 2006.

[7] Thomas Hellström and Ola Ringdahl. Follow the past — a path tracking algorithm for autonomous forest vehicles. Technical Report UMINF 04.11, Department of Computing Science, University of Umeå, 2004.

[8] Thomas Hellström and Ola Ringdahl. Follow the past — a path tracking algorithm for autonomous vehicles. *To be published in International Journal of Vehicle Autonomous Systems*, 2006.

[9] K. Konolige and K. Chou. Markov localisation using correlation. In *Proceedings of Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 1154–1159, 1999.

[10] Ola Ringdahl. Path tracking and obstacle avoidance for forest machines. Master's Thesis UMNAD 454/03, Department of Computing Science, University of Umeå, april 2003.

[11] Jorma Selkäinaho. *Adaptive Autonomous Navigation of Mobile Robots in Unknown Environments*. PhD thesis, Helsinki University of Technology, Finland, 2002.