Follow the Past: a path-tracking algorithm for autonomous vehicles

Thomas Hellström* and Ola Ringdahl

Department of Computing Science,
Umeå University,
S-901 87 Umeå, Sweden
E-mail: thomash@cs.umu.se E-mail: ringdahl@cs.umu.se
*Corresponding author.

Abstract: A number of algorithms for path tracking are described in the robotics literature. Traditional algorithms, like Pure Pursuit and Follow the Carrot, use position information to compute steering commands that make a vehicle follow a pre-defined path approximately. These algorithms are well known to cut corners, since they do not explicitly take into account the actual curvature of the path. In this paper we present a novel algorithm that uses recorded steering commands to overcome this problem. The algorithm is constructed within the behavioural paradigm common in intelligent robotics and is divided into three separate behaviours, each responsible for one aspect of the path-tracking task. The algorithm is implemented both on a simulator for autonomous forest machines and a physical small-scale robot. The results are compared with the Pure Pursuit and the Follow the Carrot algorithms and show a significant improvement in performance.

Keywords: behavioural robotics; outdoor navigation; path tracking.

Reference to this paper should be made as follows: Hellström, T. and Ringdahl, O. (2006) 'Follow the Past: a path-tracking algorithm for autonomous vehicles', *Int. J. Vehicle Autonomous Systems*, Vol. 4, Nos. 2-4, pp.216–224.

Biographical notes: Thomas Hellström is currently working as an Associate Professor at the Department of Computing Science at Umeå University and has received his PhD from the same department in 2001. His research interests are in learning robots, field robotics and machine learning.

Ola Ringdahl works as a Research Engineer at the Department of Computing Science at Umeå University. He received his MSc from the same department in 2003. His research interests are in path tracking, obstacle avoidance and system integration.

1 Introduction

The Follow the Past algorithm has been developed as a part of the project Autonomous Navigation for Forest Machines (Hellström, 2002) at Umeå University, Sweden. The goal of this project is to develop a path-tracking forest vehicle. A human driver drives the path once, while a computer is continuously recording the position, velocity, orientation and steering angle. This information is then used to control the vehicle each time it autonomously travels along the path. If the vehicle gets off course, for example, as a result of avoiding an obstacle or because of noise in the positioning sensors, the Follow the Past algorithm steers like the driver, plus an additional angle, based on the distance to the path. Traditional algorithms, like Follow the Carrot (Barton, 2001) and Pure Pursuit (Coulter, 1992), use position information only and sometimes run into problems that can be avoided, by taking into account the additional information from the human driver.

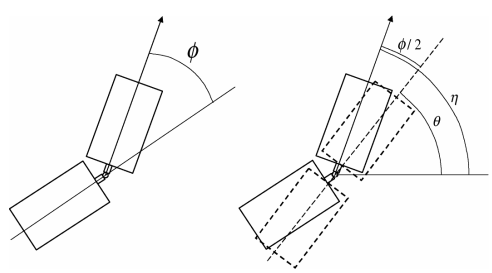
There are many variations on the basic Pure Pursuit algorithm. The Adaptive Pure Pursuit algorithm (Hebert, Thorpe and Stentz, 1997) addresses computational issues and stability at high speeds. The Feedforward Pure Pursuit algorithms (Hebert, Thorpe and Stentz, 1997) simulates the outcome of various control commands and selects the most appropriate one. In Coulter (1992), the average curvature of the path is used instead of the recorded steering angle in a proportional path-following algorithm. Other researchers, e.g. Ollero et al. (2001), have approached the problem with a fuzzy-logic controller that uses the same additional information as Follow the Past, but has a more complex design. A brief survey of more path-tracking control algorithms can be found in (Mäkelä, 2001).

This report describes the derivation of the Follow the Past algorithm, presents simulated runs as well as tests with a physical robot, Pioneer 2-AT8 (ActivMedia_Robotics, 2004) and compares these with traditional path-tracking algorithms. A full description of the simulator environment with mathematical derivations can be found in (Ringdahl, 2003). Initial tests and results can be found in our technical report (Hellström and Ringdahl, 2004).

2 Follow the Past algorithm

Since the forest vehicle uses articulated steering, the definitions of heading, orientation and steering angle may not be obvious to the reader. In this paper we use the following definitions, also illustrated in Figure 1.

Figure 1 Definitions of steering angle ϕ, heading η and orientation θ



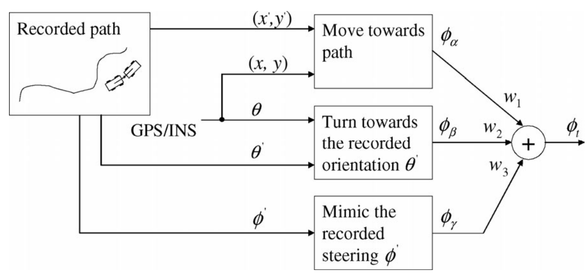
The steering angle ϕ is defined as the angle between the front and rear sections of the vehicle. The heading η is defined as the direction of the front part of the vehicle. The orientation θ is defined as the direction in which the vehicle would travel if the steering angle was zero and the baseline was maintained, as illustrated by the dashed vehicle in Figure 1. θ can be computed as the heading of the vehicle minus half the steering angle, i.e. $\theta = \eta - (\phi/2)$. The orientation θ and the heading η are expressed in a global coordinate system.

During the manual driving along the path, the orientation and steering angle are recorded together with the position at every moment. The recorded orientation θ' and the recorded steering angle ϕ' are used by the Follow the Past method, which is composed of three independent behaviours:

- ϕ_β: Turn towards the recorded orientation θ'.
- ϕ_γ: Mimic the recorded steering angle ϕ'.
- ϕ_α: Move towards the path.

Each behaviour suggests a steering angle and is reactive, i.e. operates on the current input values; orientation, steering angle and estimated shortest distance to the path. ϕ_α uses recorded closest position (x', y') and actual position (x, y) as inputs. ϕ_β uses recorded orientation θ ' and actual orientation µ as inputs. ϕ_γ uses the recorded steering angle ϕ ' as input. The three behaviours are fused into one action, the commanded steering angle ϕ_t, as shown in Figure 2. The three independent behaviours ϕ_α, ϕ_β and ϕ_γ operate in the following fashion:

Figure 2 Path tracking with reactive control of steering angle ϕ_t



ϕ_β: *Turn towards the recorded orientation*. The angle θ' is defined as the recorded orientation at the closest point on the recorded path. This point is called the path point. ϕ_β is computed as the difference between the current orientation θ and the recorded orientation θ':

$$\phi_\beta = \theta' - \theta \tag{1}$$

ϕ_γ: *Mimic the recorded steering angle*. This behaviour simply returns the recorded steering angle ϕ' at the the path point:

$$\phi_\gamma = \phi' \tag{2}$$

By using the recorded steering angle, the curvature of the path is automatically included in the final steering command. This is a great advantage compared with methods like Pure Pursuit (Coulter, 1992) and Follow the Carrot (Barton, 2001).

ϕ_α: *Move towards the path.* This behaviour is responsible for bringing the vehicle back to the path if the vehicle deviates from the path for some reason. Such a deviation can be caused by noise in the position signal or by the obstacle-avoidance system, which is a separate behaviour, not described in any detail in this report. ϕ_α can be implemented in many ways. We have implemented two different methods as described below.

Method 1: Directly proportional to the distance from the path. The simplest way is to calculate a value, which is directly proportional to the distance between the vehicle and the path point. This is done by multiplying the distance d by a constant k (degree m^{-1}), i.e.

$$\phi_\alpha = kd \qquad (3)$$

with the constraint

$$|\phi_\alpha| \leq 90^\circ$$

d is the signed distance between the vehicle and the path point. It is negative or positive, depending on the side of the path of the vehicle. A typical value for k in the tested applications is 0.07. To ensure that ϕ_α does not become too large when the distance to the path is substantial, a constraint is required. The two behaviours, ϕ_β and ϕ_γ, see to it that the vehicle remains parallel to the path and, therefore, the angle ϕ_α must be within [–90°, 90°], i.e. $|\phi_\alpha| \leq 90°$.

Method 2: Use a look-ahead point. With Method 1, the calculation of the control angle ϕ_α is linear. This may result in oscillations about the path, when the vehicle is close to it. Reducing the value for k results in a slow response, even for a large d, meaning that the vehicle requires more time to reach the path. Method 2 overcomes this problem by using a Look-Ahead Point and calculating the angle between the vehicle and the Look-Ahead Point (Figure 3). The algorithm for computing ϕ_α with Method 2 is:

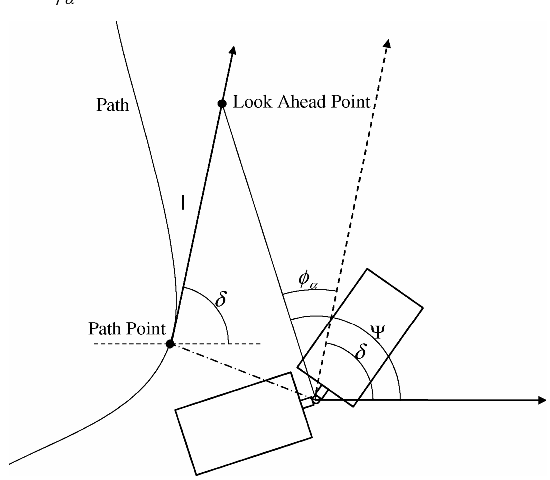
1 Determine the closest point on the recorded path (i.e. the path point).

2 Compute a Look-Ahead Point at a Look-Ahead Distance l from the path point, in a direction δ, defined as the sum of the recorded orientation ϕ' and the recorded steering angle ϕ' at the path point, i.e.: $\delta = \phi' + \theta'$.

3 Calculate a Look-Ahead Angle ψ, defined as the polar angular coordinate for the vector between the vehicle's current coordinates and the Look-Ahead Point.

4 Compute ϕ_α as the difference between ψ and the angle δ., i.e. $\phi_\alpha = \psi - \delta$.

Figure 3 Calculation of ϕ_α in Method 2



2.1 Command fusion

The three behaviours ϕ_α, ϕ_β and ϕ_γ, return a suggested steering angle, aiming at fulfilling the goals of the respective behaviours. These three values are fused into one value ϕ_t by a weighted addition, as shown in Figure 2. In our tests, all weights have been set to 1, i.e.

$$\phi_t = \phi_\beta + \phi_\gamma + \phi_\alpha. \tag{4}$$

For Method 1, this expression is

$$\phi_t = \theta' - \theta + \phi + kd. \tag{5}$$

For Method 2, the expression for the fused ϕ_t is

$$\begin{aligned} \phi_t &= \psi - \delta + \phi_\beta + \phi_\gamma \\ &= \psi - (\phi' + \theta') + (\theta' - \theta) + \phi' \\ &= \psi - \theta. \end{aligned} \tag{6}$$

3 Testing and results

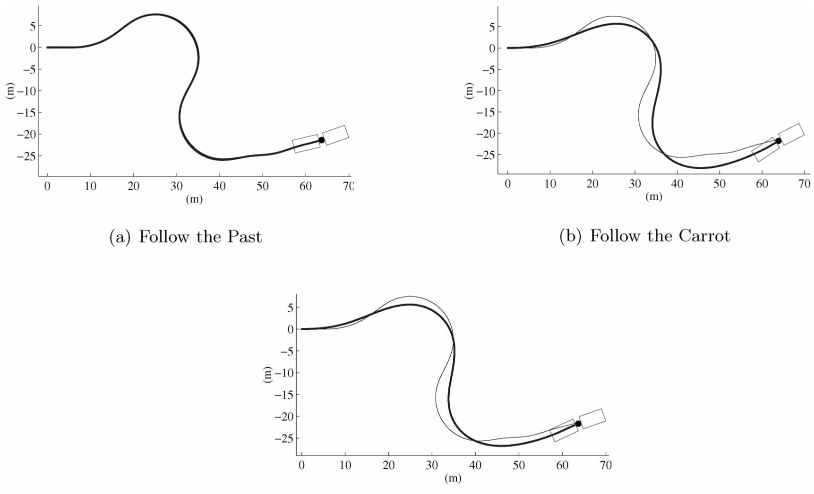
The developed algorithm has been tested both in a simulator for forest machines (Ringdahl, 2003) and on a Pioneer robot (ActivMedia_Robotics, 2004) and is compared to the already existing implementations of the Follow the Carrot (Barton, 2001) and Pure Pursuit (Coulter, 1992) methods. No significant difference between the two methods for calculating ϕ_α can be seen in the presented examples; therefore, only the tests done with Method 2 are shown here. In the simulator, a Look-Ahead Distance l of 12 m is used and in the robot test cases $l = 1.2$ m. These values are determined experimentally and affect the vehicle's ability to quickly return to the path; for example, when avoiding obstacles, as well as the sensitiveness to noise in the position sensors. Extensive tests with different values for k and l at different noise levels in the simulator can be found in Hellström and Ringdahl (2004).

The reason for using a simulator in addition to a real robot is that the physical layout of the Pioneer robot differs significantly from the eventual target machine: a real forest machine, with articulated steering. The simulator simulates a large forest vehicle that takes time to turn and has more diffculties with sharp turns. This gives some indication as to how the algorithm will perform on a real forest machine. In the examples given later, it is assumed that there is no noise in the simulated position sensors. This illustrates how the algorithm performs with ideal sensors. The Pioneer robot is equipped with a RTK-DGPS for positioning and a magnetic compass combined with a gyro to measure the heading of the vehicle. The maximal accuracy of the GPS position is ±2 cm.

Figure 4a shows the results for path tracking in the forest machine simulator. The vehicle (thick line) is capable of perfectly following a recorded path (thin line) by the Follow the Past algorithm, with practically no deviation from the path. As a reference, Figure 4b and c show how the vehicle behaves when using the Follow the Carrot and Pure Pursuit methods, respectively, under the same conditions. This path has some sharp turns, which makes it diffcult for both Follow the Carrot and Pure Pursuit, as they tend to 'cut corners' instead of following a highly curved path. Under ideal conditions, this problem is avoided by the Follow the Past algorithm. As we can see in Figure 5, the

Follow the Past algorithm still has superior performance over the other two algorithms when applied to the physical Pioneer robot. The behaviours of the algorithms are very similar to the simulator tests.

Figure 4 The follow the Past algorithm does not cut the corners like the other algorithms do. The shown examples are from the forest machine simulator

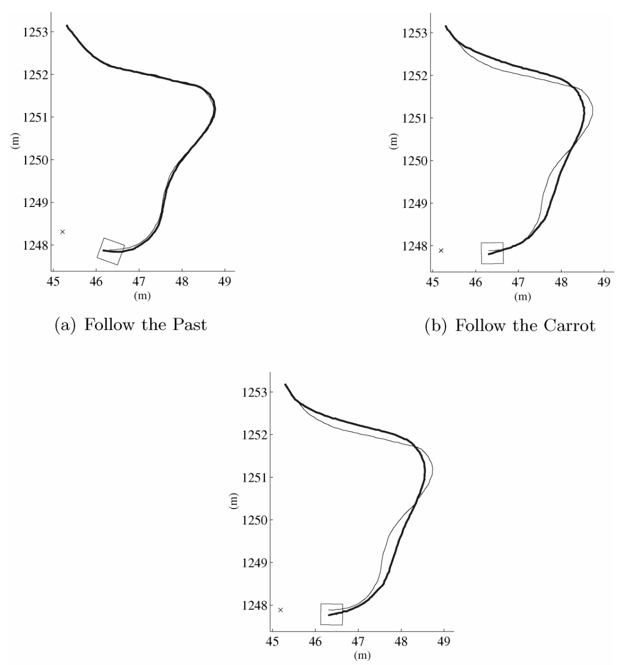


(a) Follow the Past

(b) Follow the Carrot

(c) Pure Pursuit

Figure 5 The follow the Past algorithm still performs better than the other two algorithms when applied to a physical vehicle. These examples are from tests with the Pioneer robot



(a) Follow the Past

(b) Follow the Carrot

(c) Pure Pursuit

It is illuminating to study how the three behaviours ϕ_β, ϕ_γ and ϕ_α contribute to the path-following in different situations. For the purpose of illustration, the robot was placed 1.5 m away from the recorded path. Figure 6 shows the behaviours produced by Follow the Past, when tracking the path in Figure 7. The resulting steering angle is composed of all three behaviours as long as the vehicle is far away from the path and has an incorrect heading. About 9 sec from the start, both ϕ_α and ϕ_β get reduced to almost zero and the total behaviour consists almost entirely of mimicking the steering angle, i.e. $\phi_t \approx \phi_\gamma$. For the first two seconds, the resulting set steering angle is less than the minimum steering angle of the vehicle (–20°) and therefore the actual steering angle differs from the set value in this situation.

Figure 6 Illustration of the three different behaviours in Follow the Past, when following the path in Figure 7. The Pioneer robot was initially placed 1.5 m away from the recorded path. The resulting set steering angle is the sum of ϕ_α, ϕ_β and ϕ_γ. When the distance to the path has been reduced close to zero, the set steering angle is essentially ϕ_γ, i.e. the robot tries to mimic the human driver's steering commands

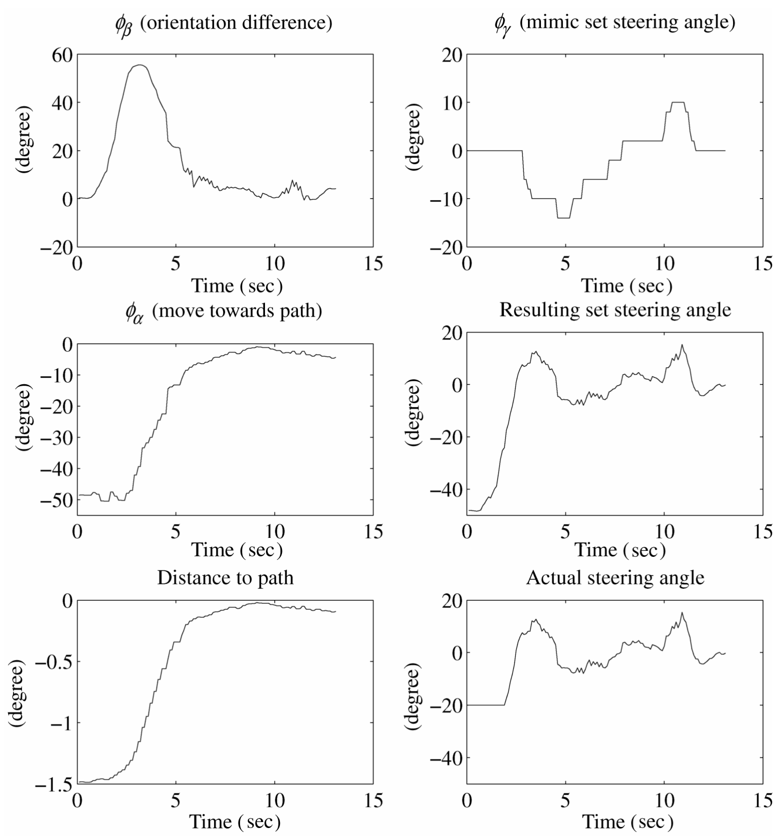
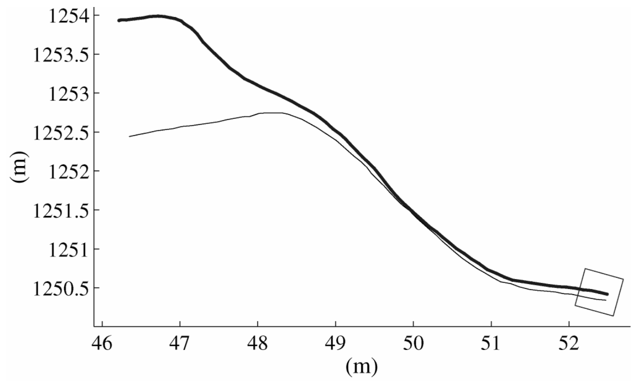


Figure 7 The robot starts 1.5 m from the recorded path and tracks it with the follow the Past algorithm. The three behaviours that define the set steering angle are shown in Figure 6



4 Conclusions

We have implemented and evaluated a new path-tracking algorithm that uses the operator's steering commands and the recorded heading values as inputs, in addition to actual position and heading information. The algorithm has proven to be a much more appropriate choice than the traditional Pure Pursuit and Follow the Carrot algorithms for our application. The reason for this is primarily the inclusion of extra information into the algorithm. Taking into account the human driver's steering commands makes it possible to avoid the well-known drawbacks of the traditional algorithms. The two described methods for computing ϕ_α, the behaviour responsible for moving towards the path, have comparable performances. The algorithm is currently being implemented on a full-size forest machine.

Acknowledgements

The authors would like to acknowledge Thomas Johansson, who has designed and implemented the control and sensor software for the Pioneer robot used in the experiments. This work was funded by The Kempe Foundation and VINNOVA under Award No. 2003-00777.

References

ActivMedia_Robotics (2004) Robots, agv's and robotic sensing. Available at: http://www.activmedia.com; 17 December 2004/.

Barton, M.J. (2001) *Controller Development and Implementation for Path Planning and Following in an Autonomous Urban Vehicle*. Undergraduate thesis, University of Sydney, November.

Coulter, R.C. (1992) 'Implementation of the pure pursuit path tracking algorithm', Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January.

Hebert, M., Thorpe, C. and Stentz, A.T. (1997) *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. London: Kluwer Academic Publishers.

Hellström, T. (2002) 'Autonomous navigation for forest machines', Technical Report UMINF 02.13, Department of Computing Science, University of Umeå aug 2002.

Hellström, T. and Ringdahl, O. (2004) 'Follow the past - a path tracking algorithm for autonomous forest vehicles', Technical Report UMINF 04.11, Department of Computing Science, University of Umeå.

Mäkelä, H. (2001) *Outdoor Navigation of Mobile Robots*. PhD thesis, Helsinki University of Technology, 2001.

Ollero, A., Ferruz, J., Snchez, O. and Heredia, G. (2001) 'Mobile robot path tracking and visual target tracking using fuzzy logic', In D. Driankov and A. Saf- fiotti (Eds), *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Physica-Verlag Heidelberg.

Ringdahl, O. (2003) 'Path tracking and obstacle avoidance for forest machines', Master's Thesis UMNAD 454/03, Department of Computing Science, University of Umeå, April.