Autonomous Path Tracking Using Recorded Orientation and Steering Commands

Thomas Hellström, Ola Ringdahl
Department of Computer Science
Umeå University
901 87 Umeå, Sweden
{thomash, ringdahl}@cs.umu.se

Abstract

This paper describes a novel algorithm, Follow-the-Past, for autonomous path-tracking vehicles. Common algorithms, like Pure Pursuit and Follow the Carrot, compute steering commands that make a vehicle follow approximately a predefined path. One problem with these algorithms is that they tend to cut corners, since they do not explicitly take into account the actual curvature of the path. The method presented in this paper uses recorded orientation and steering commands to overcome this problem. The algorithm is constructed within the reactive paradigm, common in modern robotics, and is divided into three separate behaviors, each responsible for one aspect of the path-tracking task. We present results from both a simulator for autonomous forest machines and experiments with a physical robot. The results are compared with the Pure-Pursuit and the Follow-the-Carrot algorithms, and show a significant improvement in performance.

1. Introduction

The work presented in this paper is part of the project Autonomous Navigation for Forest Machines (Georgsson et al., 2005) 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 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 developed path-tracking algorithm steers like the driver, plus an additional angle, based on the distance to the path and the error in orientation. 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 additional information from the human driver. There are many variations on the basic Pure-Pursuit algorithm. The Adaptive Pure-Pursuit algorithm (Hebert et al., 1997) addresses computational issues and stability at high speeds. The Feedforward Pure-Pursuit algorithm (Hebert et al., 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 our suggested algorithm, 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 new algorithm, denoted Follow the Past, presents simulated runs as well as tests with a physical robot, *Pioneer 2-AT8* (ActivMedia, 2005), and compares these to traditional path-tracking algorithms. All simulations are run on a dedicated forest machine simulator, described in (Ringdahl, 2003). More detailed tests of the Follow-the-Past algorithm can be found in (Hellström and Ringdahl, 2004).

2. Description of the algorithm

The algorithm has been developed to RUN on a forest vehicle that uses articulated steering. The definitions of heading, orientation, and steering angle may not be obvious to the reader and are therefore illustrated in Figure 1:

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. $\theta = \eta - \phi/2$.

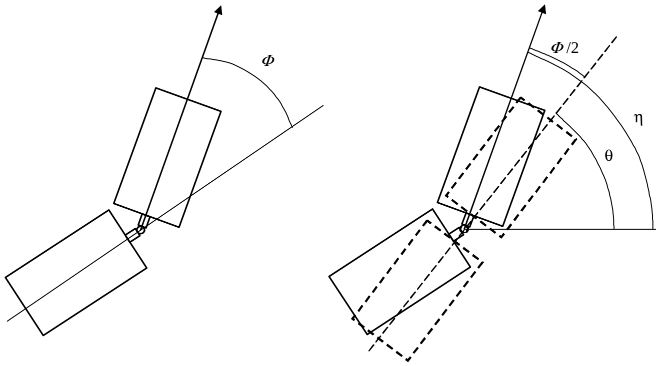


Figure 1: Definitions of steering angle ϕ, heading η, and orientation θ.

The orientation θ and the heading η are expressed in a global system of coordinates.

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 behaviors:

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

Each behavior 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 positions (x', y') and actual position (x, y) as inputs. ϕ_β uses recorded orientation θ' and actual orientation θ as inputs. ϕ_γ uses the recorded steering angle ϕ' as input. The three behaviors are fused into one action, the commanded steering angle ϕ_t, as shown in Figure 2. The three independent behaviors ϕ_α, ϕ_β, and ϕ_γ operate in the following fashion:

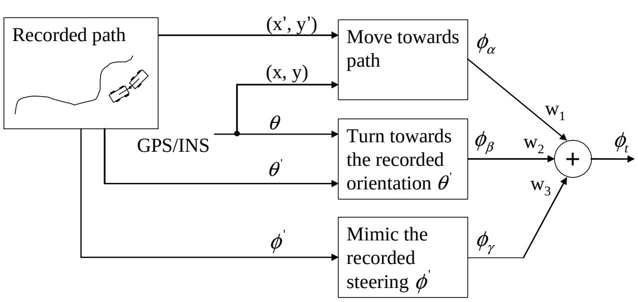


Figure 2: Path tracking with reactive control of steering angle ϕ_t.

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 behavior simply returns the recorded steering angle ϕ' at 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 to methods like Pure Pursuit (Coulter, 1992) and Follow the Carrot (Barton, 2001).

ϕ_α: Move towards the path This behavior 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 behavior, 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 One: directly proportional to the distance from the path
The simplest way is to calculate a value, directly proportional to the distance between the vehicle and the path point. It is done by multiplying the distance d by a constant k $[deg/m]$, i.e.:

$$\phi_\alpha = kd \tag{3}$$

with the constraint

$$|\phi_\alpha| \leq 90°.$$

d is the signed distance between the vehicle and the path point, and is negative or positive, depending on the side of the path at which the vehicle is. 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 behaviors 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 Two: use a Look-Ahead Point
In Method One, the calculation of the behavior ϕ_α is linear. This may result in oscillations about the path, when the vehicle is close to it. Reducing the value of k results in a slow response, even for a large d, meaning that the vehicle requires more time to reach the path. Method Two overcomes this problem by using a Look-Ahead Point, and calculating the angle between the vehicle and the Look-Ahead Point. See Figure 3. The algorithm for computing ϕ_α in Method Two 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 θ' 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.: $\phi_\alpha = \psi - \delta$.

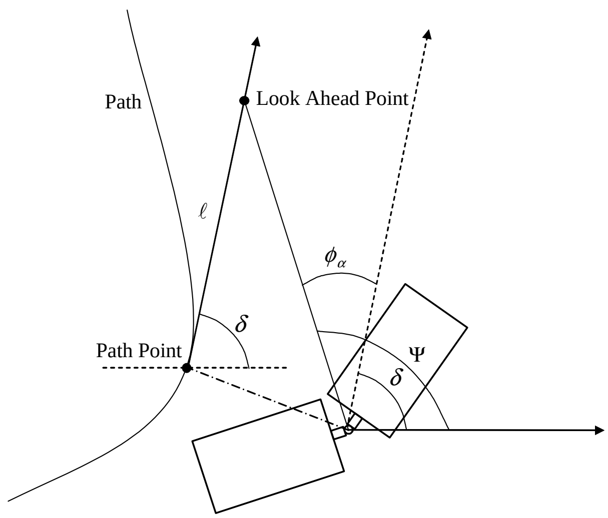


Figure 3: Calculation of ϕ_α in Method Two.

2.1 Command fusion

The three behaviors ϕ_α, ϕ_β, and ϕ_γ all return a suggested steering angle, aiming at fulfilling the goals of the respective behaviors. These three values are fused into one value ϕ_t by a weighted addition, as shown in Figure 2:

$$\phi_t = w_1\phi_\alpha + w_2\phi_\beta + w_3\phi_\gamma. \tag{4}$$

In our tests, all weights have been set to 1. I.e.:

$$\phi_t = \phi_\beta + \phi_\gamma + \phi_\alpha. \tag{5}$$

For Method One above, this expression is:

$$\phi_t = \theta' - \theta + \phi' + kd. \tag{6}$$

For Method Two above, the expression for the fused ϕ_t is:

$$\begin{aligned} \phi_t &= \psi - \delta + \phi_\beta + \phi_\gamma \\ &= \psi - (\phi' + \theta') + (\theta' - \theta) + \phi' \\ &= \psi - \theta. \end{aligned} \tag{7}$$

Other values for the weights in equation 4 can be used to balance the effect of the three behaviors in certain situations. For example, lowering the value of w_2 eliminates the effect of a faulty or unreliable orientation signal θ. This is sometimes necessary when using a magnetic compass as orientation sensor. The compass most often works very well but sometimes produces a completely erroneous orientation signal. Reducing w_2 to 0 in these situations results in a path tracking guided only by the distance to the path and the recorded steering signals. w_2 may be reset to its original value when the compass has recovered and produces reliable orientation estimates.

3. Tests

The developed algorithm has been tested on both a simulator for forest machines (Ringdahl, 2003) and a Pioneer robot (ActivMedia, 2005), 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 Two are shown here. In the simulator, a Look-Ahead Distance ℓ of 12 meters is used, and in the robot test cases ℓ = 1.2 meter. 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 difficulties with sharp turns. This gives some indication as to how the algorithm will perform on a real forest machine. In the examples below 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[1] for positioning and a magnetic compass combined with a gyro to measure the heading of the vehicle. The optimal accuracy of the GPS position is ±2 cm.

Figure 6(a) shows 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, Figures 6(b) and 6(c) show how the vehicle behaves when using the Follow-the-

[1]Real-Time Kinematics - Differential GPS

Carrot and Pure-Pursuit methods respectively, under the same conditions. This path has some sharp turns, which makes it difficult for both Follow-the-Carrot and Pure-Pursuit, as they tend to "cut corners" instead of following a highly curved path. This is especially important in a forest environment as the short cut may be impassable or at least impractical to drive. As the path is recorded by a human driver, you can assume that he had a good reason to drive as he did. Under ideal conditions, the problem with cutting corners is avoided by the Follow-the-Past algorithm. As we can see in Figure 7, the Follow-the-Past algorithm still has superior performance over the other two algorithms when applied to the physical Pioneer robot. The behaviors of the algorithms are very similar to the simulator tests.

It is illuminating to study how the three behaviors 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 meters away from the recorded path. Figure 8 shows the behaviors produced by Follow-the-Past, when tracking the path in Figure 4. The resulting steering angle is composed of all three behaviors as long as the vehicle is far away from the path, and has an incorrect heading. About nine seconds from the start, both ϕ_α and ϕ_β get reduced to almost zero, and the total behavior 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 greater than the maximum steering angle of the vehicle (20°), and therefore the actual steering angle differs from the set value in this situation.

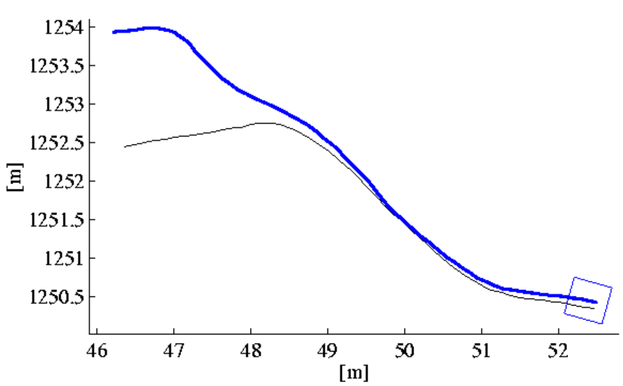


Figure 4: The robot starts 1.5 meters from the recorded path and tracks it with the Follow-the-Past algorithm. The three behaviors that define the set steering angle are shown in Figure 8.

3.1 Obstacle avoidance

To function in the forest-machine application described in Section 1., the path tracker must be able to work together with routines for obstacle avoidance. Figure 5 shows the vehicle avoiding obstacles with the VFH+ method (Ulrich and Borenstein, 1998), while tracking the path in the Follow-the-Past method. Follow-the-Past works well in combination with obstacle-avoidance, and the deviation from the path is caused by the obstacle avoiding system, when obstacles are detected near the vehicle. The example presented in Figure 5 is done in the simulator environment, since it provides a better illustration of the behavior. Tests that have been carried out on the Pioneer robot show similar results.

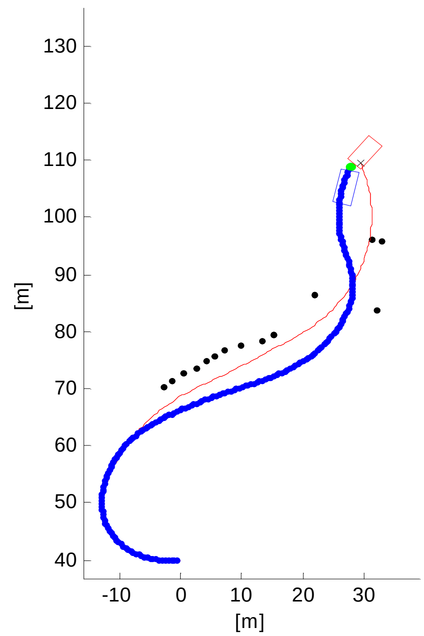


Figure 5: Avoiding obstacles with the VFH+ method in combination with Follow The Past works well. The dots represents obstacles, the thin line is the recorded path and the thicker line shows how the vehicle moved. The deviation from the path is caused by the obstacle avoidance.

4. Summary

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. Another advantage in using recorded information is that even if a sensor emits false information (for example a large offset on the heading), the information can still be used as long as the difference is the same at the time of recording as when driving autonomously. The algorithm is currently being implemented on a full-size forest machine.

5. 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 (2005). http://www.activmedia.com/.

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

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.

Georgsson, F., Hellström, T., Johansson, T., Prorok, K., Ringdahl, O., and Sandström, U. (2005). Development of an autonomous path tracking forest machine — a status report —. Technical Report UMINF 05.08, Department of Computing Science, University of Umeå.

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

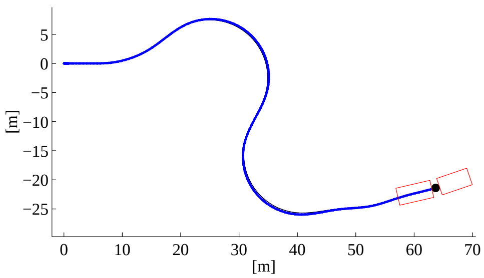
Hellström, T. and Ringdahl, O. (2004). Follow the past — a path tracking algorithm for autonomous forest vehicles. Technical Report UMINF 04.11, Department of Computing Science, University of Umeå.

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

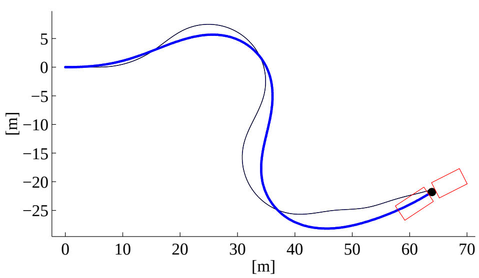
Ollero, A., Ferruz, J., Snchez, O., and Heredia, G. (2001). Mobile robot path tracking and visual target tracking using fuzzy logic. In Driankov, D. and Saffiotti, A., (Eds.), *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*, pages 51–72. Physica-Verlag Heidelberg.

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

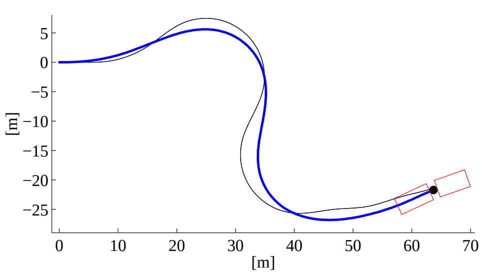
Ulrich, I. and Borenstein, J. (1998). VFH+: Reliable obstacle avoidance for fast mobile robots. *IEEE Int. Conf. on Robotics and Automation*, pages 1572–1577.



(a) Follow the Past

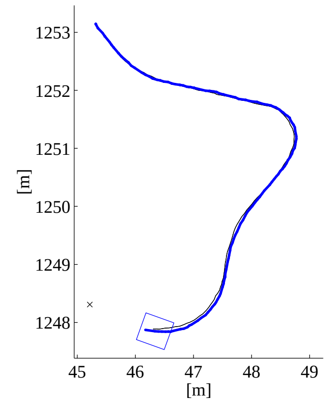


(b) Follow the Carrot

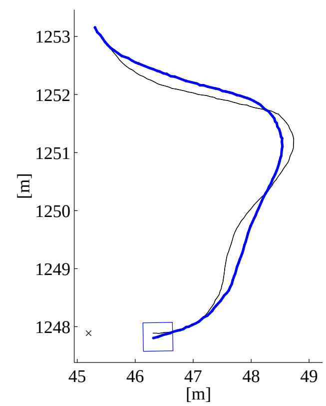


(c) Pure Pursuit

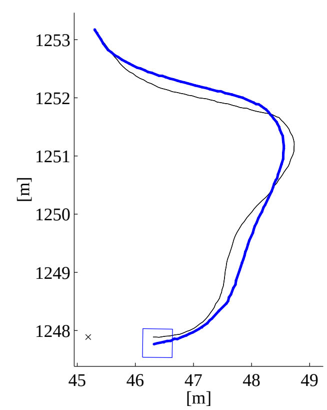
Figure 6: The Follow-the-Past algorithm does not cut the corners like the other algorithms do. The shown examples are from the forest machine simulator.



(a) Follow the Past



(b) Follow the Carrot



(c) Pure Pursuit

Figure 7: The Follow-the-Past algorithm still performs better than the other two algorithms when applied to a physical vehicle. The examples above are from tests with the Pioneer robot.

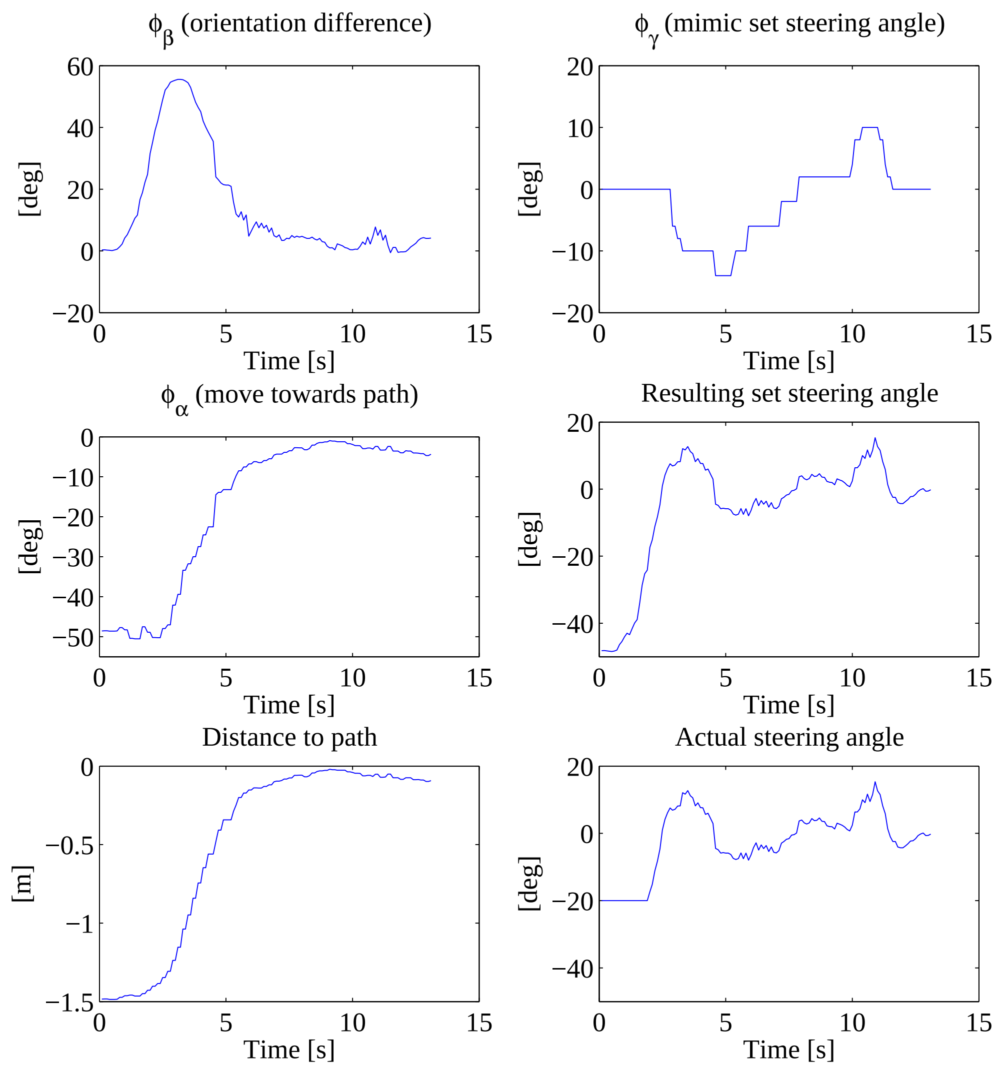


Figure 8: Illustration of the three different behaviors in Follow the Past, when following the path in Figure 4. The Pioneer robot was initially placed 1.5 meters away from the recorded path. The resulting set steering angle is the sum of ϕ_α, ϕ_β and ϕ_γ. When the distance to the path has been reduced close to zero, the set steering angle is essentially ϕ_γ, i.e. the robot tries to mimic the steering commands of the human driver.