Trajectory Based Autonomous Vehicle Following using a Robotic Driver

Article · January 2009

CITATIONS
10

READS
315

4 authors, including:

Martin Spencer
Freie Universität Berlin
5 PUBLICATIONS **112** CITATIONS
SEE PROFILE

Karl Alexander Stol
University of Auckland
110 PUBLICATIONS **2,523** CITATIONS
SEE PROFILE

Trajectory Based Autonomous Vehicle Following using a Robotic Driver

Martin Spencer
University of Auckland,
New Zealand
martinspencer1@gmail.com

Daniel Jones
University of Auckland,
New Zealand
danielrjones1@gmail.com

Martin Kraehling
Hamburg University of Technology,
Germany
martin.kraehling@tuhh.de

Dr. Karl Stol
Department of Mechanical Engineering,
University of Auckland
k.stol@auckland.ac.nz

Abstract

An autonomous vehicle following system including control approaches is presented in this paper. An existing robotic driver is used to control a standard passenger vehicle such that no modifications to the car are necessary. Only information about the relative position of the lead vehicle and the motion of the following vehicle is required, and methods are presented to construct a reference trajectory to enable accurate following. A laser scanner is used to detect the lead vehicle and the following vehicle's ego-motion is sensed using an IMU and wheel encoder. An algorithm was developed and tested to locate the lead vehicle with RMS position and orientation errors of $65mm$ and 5.8° respectively. Several trajectory-based lateral controllers were tested in simulation and then experimentally, with the best controller having an RMS lateral deviation of $37cm$ from the path of the lead vehicle. A new trajectory-based spacing controller was tested in simulation which allows the following vehicle to reproduce the velocity of the lead vehicle.

1 Introduction

Autonomous vehicles offer many benefits by performing monotonous and time consuming tasks in the place of humans, reducing costs and reducing the risk to human life in dangerous environments. They have many applications such as in automated highways, vehicle following cruise control systems, military convoys and agriculture.

This paper outlines a project to improve an existing robotic driver to allow it to follow another vehicle. The robotic driver was originally constructed at the University of Auckland to enter the 2005 DARPA Grand Challenge and has since been used in dynamometer research [Namik *et al.*, 2006]. Recently, Wong *et al.* [2008] added sensors to allow detection of the lead vehicle and motion of the following vehicle, and simulated various vehicle identification and lateral control strategies. The robotic driver was designed as a retrofit solution and is able to fit in most standard passenger cars (Figure 1), with no modifications to the vehicle required. The steering wheel and pedals are actuated by the robot in the same way as a human driver. As a retrofit solution, the robotic driver presents an inexpensive and rapid method to automate most vehicles. The current robot design requires a vehicle with automatic transmission and power steering.

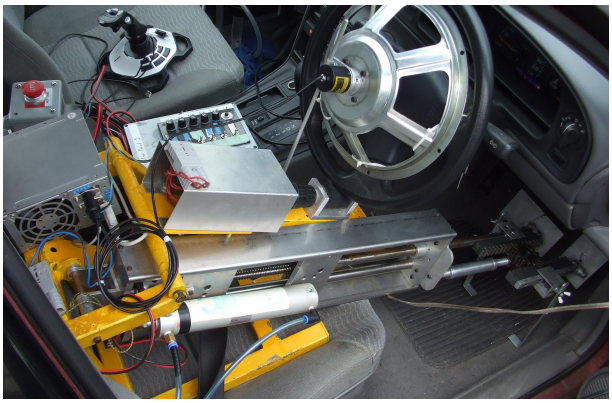

Figure 1: The robotic driver fitted in a vehicle

The aim of this project is to allow the robotic driver to safely and autonomously follow the trajectory of a lead vehicle. It is assumed that the task can be decoupled into lateral control, to enable the following vehicle to adhere to the path taken by the lead vehicle, and longitudinal control, to reproduce the velocity of the lead vehicle with a constant time headway. This approach has been used by many previous authors, such as Sotelo [2003]. No information about the absolute position of the vehicle, such as that provided by GPS, landmarks or external communication is used: vehicle following is approached in a purely relative sense, by making use only of the relative position of the lead car and the motion of the following car to construct a series of waypoints representing the path of the lead vehicle over time, in a similar way to the approach in Fujioka and Omae [1998] and Gehrig and Stein [1998]. Both of these papers demonstrated that following the trajectory of the lead vehicle rather than the vehicle itself allows more accurate tracking.

Wong *et al.* [2008] developed and tested three methods for determining the relative position of the lead vehicle, which use reflections from the rear and side of the car with no modification necessary. A new algorithm is presented in this paper and compared with these three methods.

In this paper, several algorithms for following the trajectory of the lead vehicle are tested in simulation. The bicycle approximation is used to model the steering of the vehicle, which has been used by many previous authors with good results, for example Sotelo [2003] and Linderoth *et al.* [2008]. A great deal of literature has been published on methods for following a trajectory. Fujioka and Omae [1998] and Gehrig and Stein [1998] discuss algorithms for path tracking which involve selecting a lookahead point on the path, which are relevant to this work. Other lateral controllers include those developed for the DARPA Grand Challenge such as those by Thrun and others [2007] and Linderoth *et al.* [2008].

Much research into longitudinal control for vehicle following has been carried out in the context of vehicle platooning and automated highways. These often aim at string stability of multiple vehicles following one lead car in a row or optimising traffic flow. Different constant and variable time headway approaches were successfully used by Chien *et al.* [1994], Swaroop *et al.* [2001] and Zhang and Ioannou [2005]. A PID controller was used by Chien, Ioannou and Lai [1994] to maintain the desired spacing. A nonlinear controller is presented by Alain Girault [2003] which does not take a spacing error, but a ratio of actual and desired distance. The present work considers only one lead vehicle.

Object avoidance is not considered in this project and it is assumed that the lead vehicle is following a safe path. Real-world tests are conducted only on flat terrain and in clear weather. The speed of both vehicles is limited to $30km/h$ and the target headway time is $3.0s$, chosen to allow a safe margin of error during preliminary testing. Due to the limited angular range of the laser scanner, the radius of curvature of the lead vehicle is constrained to be not less than $18m$. The lead vehicle is a standard passenger car.

The remainder of this paper is organised as follows: Section 2 outlines the robotic driver system including hardware and software, Section 3 describes the methods used to determine the relative position of the lead vehicle and to create a trajectory, Section 4 describes the lateral controllers that were implemented and tested in simulation, and in Section 5 longitudinal control of the vehicle is explained. Concluding remarks are given in Section 6.

2 Robotic Driver System

Hardware The hardware of the system has been explained by Wong *et al.* [2008]; a brief summary is presented here.

All electrical power to the robotic driver is provided by the car battery of the vehicle. The robotic driver has two feet to actuate the brake and accelerator pedals. They are connected via a rack and pinion mechanism, enabling a single DC motor to actuate both pedals. The response time of the pedal actuator has been shown to be similar to that of a human driver [Wong *et al.*, 2008].

The vehicle steering wheel is turned using a friction wheel connected to a DC motor. The response of the steering actuator is significantly slower than that of a human driver; during testing it was found to take up to 9s to turn from full lock to full lock. This has implications for the lateral controller.

Motion Mind motor controller boards from Solutions Cubed LLC are used to control the pedal and steering actuators. On-board PID controllers regulate the motor positions using feedback from rotary encoders. The desired setpoints are received via serial connection with the control computer.

An emergency stop mechanism is implemented using pneumatics, enabling the entire pedal drive mechanism to be forced downwards on to the brake.

A joystick is used to control the robot manually during initial testing; this allows the vehicle to be driven by a human with the robot in the driver's seat for collecting performance data and checking the system.

Sensors A Sick LD MRS laser scanner (a prototype version of the Ibeo LUX) provides information about the location of the lead vehicle. The laser scanner has a working azimuth range of 94° and up to $200m$, which is more than adequate for the conditions in this project.

Additionally, the Sick LD MRS scans four horizontal layers independently, making the system robust to small changes in the relative height of the lead vehicle, and registers multiple laser beam echoes for each target, enabling the laser scanner to function from behind the vehicle windscreen.

An Xsens Mti-G Inertial Measurement Unit (IMU) is used to measure the yaw rate of the following vehicle. This IMU includes an integrated GPS with a Kalman filter. For this project the GPS was not used because it is only accurate to $5m$ which was not considered sufficient for vehicle following, and the aim of the project was to achieve vehicle following in a purely relative sense.

A rotary encoder measures the velocity of the following vehicle, providing greater accuracy than the IMU.

Software and Control Figure 2 gives an overview of the control structure of the system. The open source Player/Stage software [Vaughan *et al.*, 2009] was used to interface with the hardware and for performing simulations. Player handles all low-level communication with the hardware and provides a standard set of interfaces for client programs to connect to. In previous work [Wong *et al.*, 2008], several Player drivers were developed to communicate with each of the hardware components of the robotic driver.

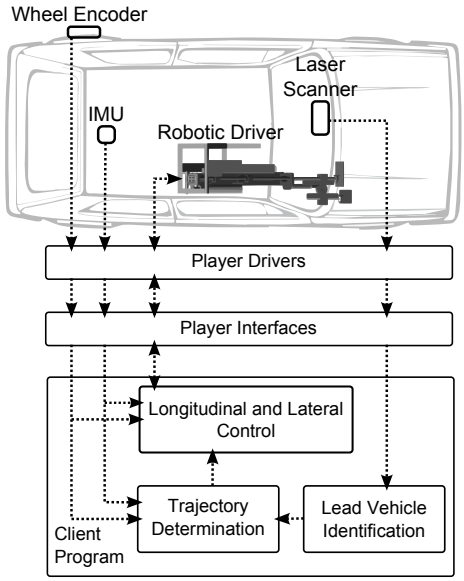

Figure 2: Control overview

The client program for high-level control of the robotic driver was written in C++ using the client libraries included with Player. There are three primary independent modules to the program: the main control loop, the data logger and the graphical user interface (GUI). The main control loop reads all sensor data from the Player server, performs control calculations and sends commands to the actuators; it was written as a finite state machine, where each state is a particular mode of operation. This structure allows for well-defined conditions for changing between any two states. The loop operates at a control frequency of $10Hz$ which is slightly slower than the rate at which the laser scanner, the slowest sensor, updates ($12.5Hz$). The data logger writes all important information to a log file, and the GUI allows for easy interaction with the program.

3 Lead Vehicle Tracking

Tracking of the lead vehicle is divided into two tasks: determining the position and orientation of the lead vehicle relative to the following vehicle, and the integration of this information with the following vehicle's velocity and yaw rate to calculate the trajectory followed by the lead vehicle.

Determination of Lead Vehicle's Relative Position The lead vehicle's relative position and orientation are calculated from the laser scanner data. Generally the lead vehicle's point cloud can be simplified to a single line segment, or two adjacent line segments at right angles, as shown in Figure 3. For this task, we define the lead vehicle's relative position to be the centre of the lead vehicle's rear axle, relative to the laser scanner.

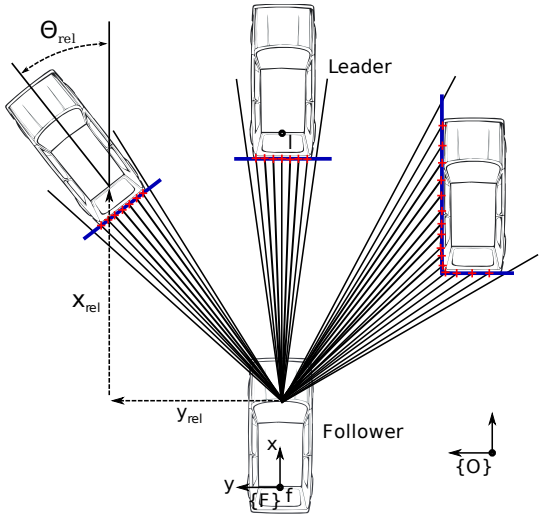

Figure 3: Lead vehicle identification schematic

Wong *et al.* [2008] investigated three algorithms to determine lead vehicle position: Box Classification (using least-mean-squares to fit a line segment to the data points representing the rear side of the lead vehicle), Hough Transform, and Hough Classification (a combination of both techniques). However, it was not concluded which was most accurate.

There are several disadvantages with these techniques. All three are dependent on the previous lead vehicle's position, which makes them susceptible to temporary laser data dropouts and require careful consideration of initial values. These algorithms do not detect when inaccurate results are produced, so the system does not recognise when it is following an incorrect trajectory. Finally, these algorithms have a tendency to produce large orientation errors.

A new algorithm was tested which eliminates these disadvantages, and reduces the overall position and orientation error. First, the point cloud data is passed into an image kernel filter, locating possible lead vehicle positions. The lead vehicle will be surrounded by empty space, therefore a 7x7 Laplacian image kernel with negative outer coefficients was selected to accentuate regions with high concentrations of data points surrounded by empty spaces. Image kernel filtering was implemented as an alternative method of excluding irrelevant laser data, eliminating the dependency on the previous position of the lead vehicle.

For each region output by the image kernel filter, an angular histogram of the nearby laser points is created. The angular histogram will generally have two peaks 90° apart, representing the back and sides of the vehicle. These two peaks are combined to find a peak angle ranging from 0° to 90°. The true vehicle orientation will be either at this angle, or shifted by 90°.

Finally, a rectangle the size of the lead vehicle is placed to enclose the laser points representing the vehicle, determining the correct lead vehicle's orientation and position. In addition, there are physical constraints on the variation of vehicle position and orientation between laser scanner frames. These constraints will force impossible vehicle positions to be ignored, unless they are persistent across multiple frames, in which case we assume the lead vehicle's previous position was incorrect.

A real-world driving test was conducted to quantify the accuracy of all four algorithms. A sample view and laser scan are shown in Figure 4. For this test the two vehicles were driven manually in oval and figure-eight patterns without sharp turns, ensuring the lead vehicle was within range of the laser scanner at all times. The path was chosen so that other vehicles and obstacles such as trees were too far away to interfere with vehicle identification. Some authors [White and Tomizuka, 2002; Fujioka and Omae, 1998] have used reflective markers on the rear of the lead car to enable easy detection, but for this test only a flat white board was attached to the rear of the lead car to ensure sufficient reflections.

A ground-truth analysis was conducted using 65 of the laser scanner frames collected from the real-world driving test outlined above. These frames were intentionally selected to ensure the first three algorithms would not select entirely incorrect vehicle positions. For each frame, the ground-truth vehicle position was determined by hand by looking at the laser range data, and compared against the output of each algorithm. The errors in position and orientation are summarised in Table 1 below.

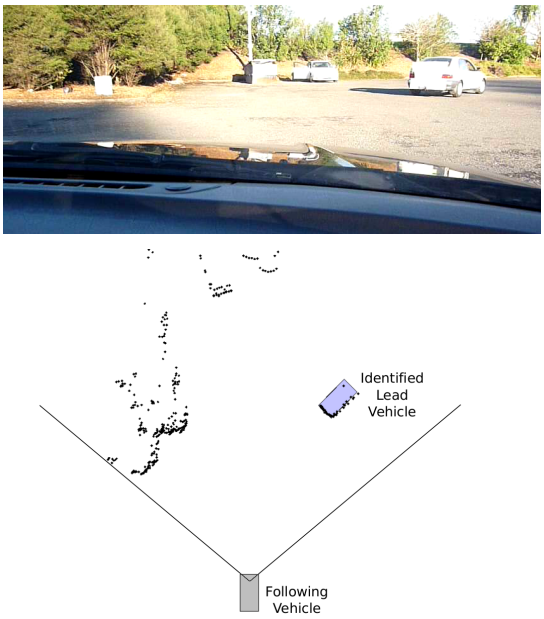

Figure 4: View from following vehicle (top) and corresponding laser data visualisation (bottom) from a real-world test

	RMS Error	
Algorithm	**Position (mm)**	**Orientation** (°)
Box Class.	71	20.5
Hough Trans.	77	7.9
Hough Class.	94	13.5
New Alg.	65	5.8

Table 1: RMS error in measured position and orientation of lead vehicle

The new algorithm has increased accuracy compared to the previous three. Note that these results only quantify accuracy in frames which produce reasonable results in all four algorithms; this test does not quantify the robustness of the new algorithm to laser data dropouts, incorrect initial values etc. The robustness and accuracy of the new algorithm makes it a viable method for determining lead vehicle position.

There are some situations in which all four techniques would fail. It is assumed that the lead vehicle orientation will never exceed the range 90° to −90°. Additionally, it

is assumed that there will be a single vehicle in the laser view range, so if another vehicle appears the results will be unpredictable. Possible improvements to these techniques include detection of multiple vehicles and other objects such as obstacles and pedestrians.

Determination of Lead Vehicle Trajectory The calculated relative lead vehicle positions are used to create a trajectory over time. This is achieved by transforming each position into an arbitrary ground-based reference frame, where the position of the following vehicle in this reference frame is updated each iteration by the measured velocity and yaw rate, as in the approach taken by Gehrig and Stein [1998]. In this way a series of waypoints are created which can then be followed.

We write the position of the following car as (x_f, y_f, θ_f), and the lead car as (x_l, y_l, θ_l). In the relative frame the x axis is forwards, the y axis is to the left and θ (orientation) is positive anticlockwise. Points in the following frame are denoted with a superscript F, while points in the ground-based frame have a superscript O. These reference frames are shown in Figure 3. For each iteration k, the estimated position of the following vehicle is updated as follows:

$$\bar{\omega}_f = \frac{\dot{\theta}_{f,k} + \dot{\theta}_{f,k-1}}{2}, \quad \bar{v}_f = \frac{v_{f,k} + v_{f,k-1}}{2},$$

$$\begin{bmatrix} {}^O x_{f,k} \\ {}^O y_{f,k} \\ {}^O \theta_{f,k} \end{bmatrix} = \begin{bmatrix} {}^O x_{f,k-1} \\ {}^O y_{f,k-1} \\ {}^O \theta_{f,k-1} \end{bmatrix} + T \begin{bmatrix} \cos({}^O\theta_{f,k} + \bar{\omega}_f \frac{T}{2}) & 0 \\ \sin({}^O\theta_{f,k} + \bar{\omega}_f \frac{T}{2}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{v}_f \\ \bar{\omega}_f \end{bmatrix} \tag{1}$$

where $\dot{\theta_f}$ is the yaw rate and v_f is the forward velocity of the following vehicle, and T is the sampling time, $0.1s$. Next, the lead car's current position is transformed to the ground-based reference frame and added to the list of waypoints:

$$\begin{bmatrix} {}^O x_{l,k} \\ {}^O y_{l,k} \\ {}^O \theta_{l,k} \end{bmatrix} = \begin{bmatrix} {}^O x_{f,k} \\ {}^O y_{f,k} \\ {}^O \theta_{f,k} \end{bmatrix} + M \begin{bmatrix} {}^F x_{l,k} \\ {}^F y_{l,k} \\ {}^F \theta_{l,k} \end{bmatrix}, \tag{2}$$

where

$$M = \begin{bmatrix} \cos({}^O\theta_{f,k}) & -\sin({}^O\theta_{f,k}) & 0 \\ \sin({}^O\theta_{f,k}) & \cos({}^O\theta_{f,k}) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Any waypoint in the list can be transformed back into the following frame by reversing equation (2).

The resulting list of waypoints, ignoring the orientations,

$$\mathbf{X} = \begin{bmatrix} {}^F x_{l,1} & {}^F y_{l,1} \\ \vdots & \vdots \\ {}^F x_{l,n} & {}^F y_{l,n} \end{bmatrix} \tag{4}$$

is then smoothed by fitting a two dimensional third order parametric polynomial vector function to all n waypoints, using a standard least-squares approach. The path of the lead vehicle is then given by

$$\mathbf{s}(t) = \begin{bmatrix} {}^F x(t) \\ {}^F y(t) \end{bmatrix} = \begin{bmatrix} \alpha_x^T \\ \alpha_y^T \end{bmatrix} \cdot \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix} \tag{5}$$

where $\alpha_x, \alpha_y \in \mathbb{R}^4$ are the vectors of path coefficients. For $t \in [t_1, t_n]$ the polynomial returns an interpolation of the true path of the lead vehicle, and times outside this interval are not used. At this stage, the closest point on the polynomial to the origin, $P_0 = \mathbf{s}(t_0)$, is found by iterating over values in the range $[t_1, t_n]$. All waypoints for which $t_0 - t_i > 4, i > 0$ are then removed from the list. This means that waypoints are kept for $4s$ after the following vehicle has driven past them.

The velocity and acceleration at a point on the lead vehicle's trajectory are calculated as follows:

$$\mathbf{v}(t) = \begin{bmatrix} {}^F \dot{x}(t) \\ {}^F \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \alpha_x^T \\ \alpha_y^T \end{bmatrix} \cdot \begin{pmatrix} 3t^2 \\ 2t \\ 1 \\ 0 \end{pmatrix} \tag{6}$$

$$\mathbf{a}(t) = \begin{bmatrix} {}^F \ddot{x}(t) \\ {}^F \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} \alpha_x^T \\ \alpha_y^T \end{bmatrix} \cdot \begin{pmatrix} 6t \\ 2 \\ 0 \\ 0 \end{pmatrix} \tag{7}$$

The velocity vector is always tangent to the path, whereas the acceleration vector may have a component in the longitudinal as well as lateral directions:

$$\mathbf{a}_{long}(t) = \frac{\mathbf{v} \cdot \mathbf{v}^T}{\mathbf{v}^T \cdot \mathbf{v}} \cdot \mathbf{a}(t) \tag{8}$$

$$\mathbf{a}_{lat}(t) = \mathbf{a}(t) - \mathbf{a}_{long}(t) \tag{9}$$

The curvature of the path (the reciprocal of the turn radius, R) is a scalar, given by the normalized inner product

$$\kappa(t) = \frac{1}{R(t)} = \frac{\mathbf{a}^T(t) \cdot \mathbf{a}_{lat}(t)}{||\mathbf{v}(t)|| \cdot ||\mathbf{a}(t)|| \cdot ||\mathbf{a}_{lat}(t)||}. \tag{10}$$

This method of smoothing the waypoints offers several advantages over simply using the raw waypoints. A continuous time solution for the path that the lead vehicle follows is obtained, allowing position, velocity and acceleration to be easily calculated at any time, and allowing handling of the situation where the laser scanner is unable to see the lead vehicle for short periods. The method also reduces noise in the waypoints. The main disadvantage is a loss of accuracy in some cases due to the smoothing, however we have observed that this is

negligible for the turns permitted by the scope of this paper. The polynomial fit also ignores the estimated orientation of the lead car; including this information may provide an advantage and could be the focus of future work. An example from a real-world test is shown in Figure 5.

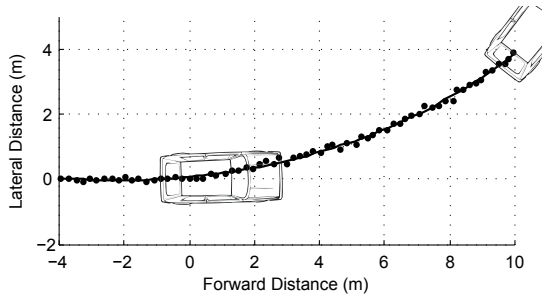

Figure 5: Sample frame, showing waypoints and the estimated path

A test was carried out to estimate how accurately the waypoints represent the true path travelled by the lead vehicle. Two cars were driven by human drivers at a low speed of approximately $10km/h$ along a straight line, at a distance of $20 \pm 0.5m$ (Figure 6). This was repeated with the following car directly behind the lead car, and with the following car driving with a $\pm 6m$ lateral offset from the lead car. The positions of the waypoints as the following car drove past them were then compared with the true path. Over 9 tests it was found that the absolute difference was up to $1.2m$. This is more than half the width of a car, so this level of accuracy was not considered acceptable. Methods for achieving a target accuracy of $\pm 0.5m$ are being investigated, such as developing a mounting system to allow the laser scanner to be more accurately aligned with the vehicle.

Figure 6: Straight line test of trajectory estimation

4 Lateral Control

The method of creating waypoints transforms the problem from vehicle following into that of following a predefined trajectory. Several algorithms were investigated and tested in simulation for this. The algorithms all produce a desired curvature, which is the input to the steering controller. The notation used to describe points on the path is shown in Figure 7. P_0 is the position vector to the closest point on the trajectory, and $d_e = ||P_{lat}||$. P_{lat} is the lookahead point, and is the input to the lateral controllers. Wit *et al.* [2004] describe the issues surrounding the choice of this point: increasing the lookahead distance tends to increase stability but also cause the lateral controller to cut corners. In this work, the lookahead point is calculated as $\mathbf{s}(t_0 + t_h)$, where t_h is the lookahead time.

The bicycle steering model is used. The centre of the rear axle of the vehicle is used as the control point and side slip at the angles is ignored.

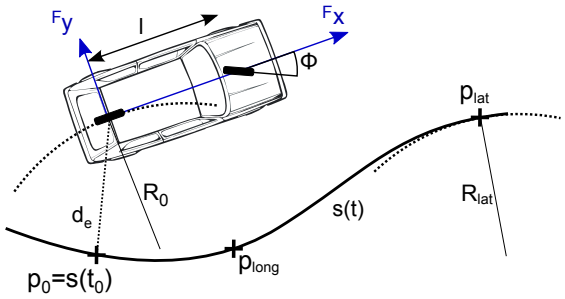

Figure 7: Coordinate system and bicycle steering

Five lateral controllers were tested in simulation.

1. Pure Pursuit [Coulter, 1992], a widely used method in which a circle is fitted passing through the vehicle and P_{lat}:

$$\kappa_{ctrl} = \frac{2 \cdot P_{lat_y}}{P_{lat_x}^2 + P_{lat_y}^2} \tag{11}$$

2. A simplified version of the third algorithm described by Fujioka and Omae [1998] was tested. We neglect both side slip and acceleration of the following vehicle, so the algorithm reduces to Pure Pursuit with a term proportional to the current tracking error added to the control output:

$$\kappa_{ctrl} = \frac{2 \cdot P_{lat_y}}{P_{lat_x}^2 + P_{lat_y}^2} + K_1 \cdot d_e \tag{12}$$

3. Vector Pursuit [Wit *et al.*, 2004]

4. An algorithm which calculates the curvature of the trajectory at the lookahead point, $\kappa_{P_{lat}}$ and adds a term proportional to the current tracking error to this to obtain a desired curvature:

$$\kappa_{ctrl} = \kappa_{P_{lat}} + K_3 \cdot d_e \tag{13}$$

5. A new algorithm similar to the third method, except that a predictor is used to compensate for the time

delay in the steering actuator response. The predictor is an estimated model of the vehicle which includes the steering dynamics with no time delay, so that the current position of the model is an estimate of the true vehicle position $0.3s$ in the future. The desired curvature is then calculated in the same manner as (13), using the predicted vehicle position. It was found that adding a derivative term to the control output improved the performance for all four controllers. For the first three algorithms equation (14) was used, and for the other two equation (15) was used. The value of 0.8 for the gain scheduling in equation (15) was determined by tuning the controllers in simulation at different speeds, with the other gains constant, and fitting a curve to the optimal K_d values.

$$\kappa_{des} = \kappa_{ctrl} + K_d \cdot \frac{P_{laty,t} - P_{laty,t-1}}{T} \quad (14)$$

$$\kappa_{des} = \kappa_{ctrl} + \frac{K_d}{V^{0.8}} \cdot \frac{P_{laty,t} - P_{laty,t-1}}{T} \quad (15)$$

Once a desired curvature has been obtained, commands must be sent to the steering actuator to achieve this curvature. A simple feedback loop, similar to the approach taken by [Fujioka and Omae, 1998], is used for this, whereby a desired change in steering wheel position (ST) is calculated from the current error in curvature:

$$\Delta ST = \frac{K_{st}}{V}(\kappa_{des} - \frac{\dot{\theta}}{V}) \quad (16)$$

The choice of gain K_{st} will affect the speed of response. When tuning controllers in simulation at different speeds, it was found that the optimal K_{st} tended to decrease in inverse proportion to the speed, so K_{st} is gain scheduled. The inverse relationship means an upper limit must be imposed on ΔST at low speeds. The result of this method is that the relationship between steering wheel angle and front wheel angle does not need to be known exactly, so the controller can be easily adapted to different vehicles.

Simulation Results The simulations carried out to test the lateral controllers included realistic parameters for the steering dynamics. During drive-by-wire tests with the robot in a real vehicle, it was found that the steering actuator response is well characterised by a first order model with a saturation on the rate of change, in series with a pure time delay. The estimated parameters were a time constant of $0.55s$, a delay of $0.3s$, and a saturation equivalent to $7.3s$ to turn the steering wheel from full lock to full lock. It was also found that the relationship between steering wheel position (measured in degrees) and curvature is approximately linear across most of the range, with a gain of $3.44 \times 10^{-4} m^{-1}$. These characteristics were used for the simulation.

Using the bicycle model of the vehicle a simple simulation environment was written as a plugin driver for Player, updating at a rate of $50Hz$. Speed and steering angle commands are sent to the driver and the position of the vehicle over time is calculated. Stage (the 2D robot simulation environment for Player) was not used for this because the overhead associated with running a full Stage world meant that it was difficult to accurately control the update rate and to obtain repeatable results.

The model of the robot and vehicle was written as another plugin driver for Player, which receives actuator commands from the client program and calculates the vehicle speed and steering angle.

Simulations were carried out over a variety of speeds from 5 to $30km/h$ and a headway time of $3.0s$, with the lead vehicle following the path shown in Figure 8. In the first test, the velocity of the lead car was varied and the system parameters were unchanged. In the second test, the gain between vehicle curvature and steering actuator position was varied from half to twice the true value. In the third test the steering actuator time delay was varied from 0 to $0.6s$. For the last two tests the lead car's velocity was constant at $20km/h$.

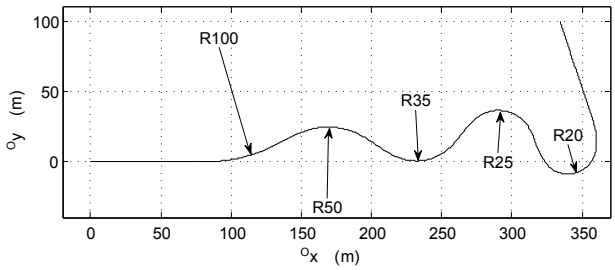

Figure 8: Path followed by lead vehicle in simulation

The controller gains were tuned manually by finding optimal values over several velocities. For the first three algorithms $t_h = 1.5s$ was found to give good results. For the curvature methods $t_h = 0.3s$ was used.

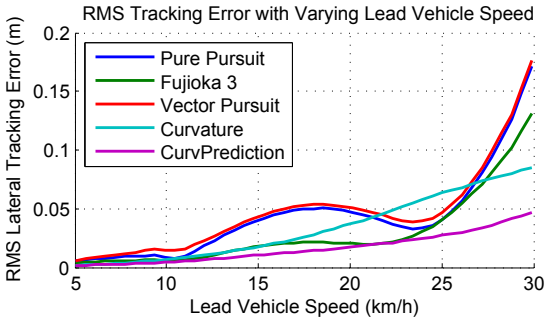

Figure 9: Simulation results with varying velocity

Figure 9 shows that all the controllers are stable across the range of velocities considered in this paper, with

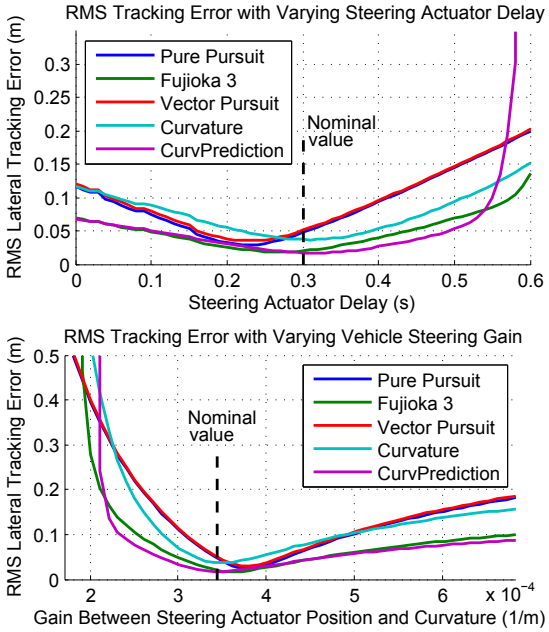

Figure 10: Simulation results, showing robustness to parameter uncertainty

the curvature prediction method giving the best performance. As can be seen in Figure 10, the controllers show similar robustness to parameter uncertainty, except for actuator delays greater than $0.5s$. However this is not expected to occur.

Experimental Results Real-world tests were conducted to quantify the performance of the overall system. A figure eight path (Figure 11), with two circles of radius 20m and 25m, was marked out with 25 reflectors. The lateral error from the marked path of both the lead car and the following car were determined by placing video cameras above the towbar of each car, facing downwards, and continuously recording. The lateral deviation of the following vehicle's path from the lead vehicle's path was then calculated as the difference between both vehicle's lateral error at each marker.

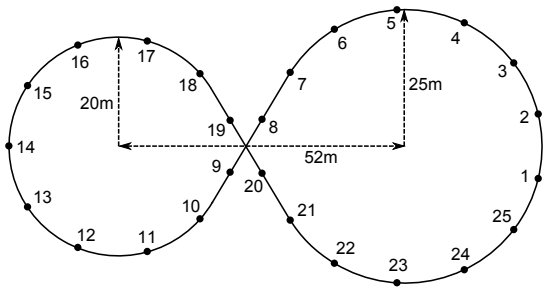

Figure 11: Path used to test the lateral controller

A total of three tests were conducted. Each test consisted of two full laps around the figure eight starting and finishing at marker #1. The lead vehicle was driven at approximately 15km/h and the following vehicle's speed was controlled via joystick, with the aim of keeping a following distance of roughly 20m. The inherent inaccuracy in this method may have affected the results. The three controllers tested were Pure Pursuit, Pure Pursuit with a lateral offset term (Fujioka 3), and the new Curvature Prediction method.

The results of the tests are summarised in Table 2. The Curvature Prediction method gave the best results out of the three, in terms of both maximum and RMS tracking error, which is consistent with the simulation. However, the tracking errors are significantly larger than any seen in simulation and the controllers showed some tendency to oscillate about the path, which did not occur in simulation. This could be due to unmodelled effects such as sensor noise and wheel slip, the fact that the controller gains have not been tuned for the real system, and the fact that the steering rate of the lead vehicle was not controlled so it may have been too fast for the steering actuator to follow.

	Lateral Deviation (m)	
Controller	**RMS**	**Max**
Pure Pursuit	0.47	1.28
Fujioka 3	0.44	1.46
CurvPrediction	0.37	0.77

Table 2: Experimental lateral deviation results

5 Longitudinal Control

The Control Approach The objective of the longitudinal control is to maintain a safe inter-vehicle spacing. A minimum bound in spacing is required to avoid collision between the two cars whereas a maximum bound is required to ensure robust operation of the laser scanner. Also the control approach has to be robust, considering nonlinearities and varying system parameters as well as delays and discontinuities partly introduced by the self-imposed condition of portability. Because of its robust nature the control approach is therefore chosen to be a cascaded PID control as shown in Figure 12. The outer control loop takes a reference spacing and a reference speed to calculate a desired speed. The inner control loop takes this desired speed and calculates an appropriate pedal position. Due to the decoupled longitudinal and lateral control a special emphasis is put on reproducing the speed trajectory of the lead vehicle.

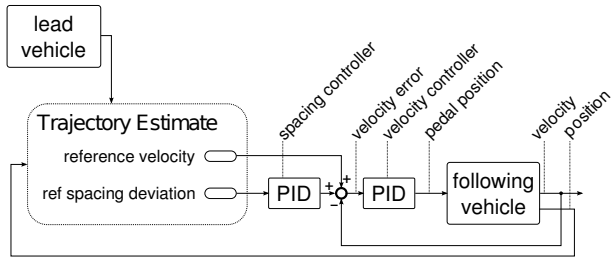

Figure 12: Longitudinal Control Structure

Modelling A vehicle model was designed to allow for controller design and simulation based on the suggested model structure by Cho and Hedrick [1989]. The model takes the desired pedal position as its input and returns the vehicle speed as its output. As the pedals are controlled through only one linear motion actuator they can not be activated separately. This makes the vehicle model a SISO system.

The vehicle model was fitted to different sets of reference data. The reference data was collected through drive-by-wire tests. The simulated speed was compared with the reference data through the following equation:

$$f = 100 \cdot \left(1 - \frac{||v - v_{sim}||}{||v - \bar{v}||}\right) \qquad [\%] \qquad (17)$$

where v is the speed of the real car, $\bar{v}$ is the mean value of that speed and v_{sim} is the simulated speed returned by the model. The simulated speed matches a reference data with at least $f = 63.8\%$ for different random drive cycles. Due to changing parameters of the real vehicle it is hard to find a more precise model.

Control Performance A PID controller was tuned for speed control representing the inner loop of the cascaded controller. Experimental results show a rise time of $t_r = 1.4s$ for a step in desired velocity from $v = 0m/s$ to $v = 5m/s$. The speed keeps oscillating around the reference value with an amplitude of about $0.8m/s$. This might be caused by discontinuities and delays in the system as well as the uneven ground of the test area.

The most widely used approach for controlling inter-vehicle spacing is the constant time headway (CTH) approach [Chien *et al.*, 1994; Girault, 2003]. This approach calculates a desired velocity

$$\dot{x}_{des} = \frac{d - d_0}{\Delta t} \qquad (18)$$

using the actual spacing between the two cars d and the constant lookahead time Δt. d_0 in this case is a safety distance that the vehicles will maintain at low speeds. However, this approach leads to an instant acceleration if the lead vehicle starts accelerating.

The CTH method was adapted for this application so that both vehicles pass a certain point on the path with the same speed. This results in a new interpretation of the constant time headway, not based on the actual vehicle speed, but on the time delay with which the two cars pass a point on the trajectory. The advantage of this is that the following vehicle is able to reproduce the velocity of the lead vehicle with a certain headway time delay. This way the assumption of decoupled longitudinal and lateral control can be justified by assuming the appropriate speed to curvature ratio and safe driving of the lead vehicle.

The reference speed, v_{ref}, is the speed of the lead vehicle a certain time headway, t_{long}, ago. This is calculated by setting t to be the current time minus t_{long} in equation (6). A second control input takes the distance error δ_p along the path from the closest point to the following vehicle, P_0, to the point P_{long} on the path where the lead vehicle was at time t_{long} ago. The distance is calculated by integrating along the polynomial. The control equation is then given by

$$v_{des} = v_{ref} + k_p \cdot \delta_p + k_i \int_0^t \delta_p dt + k_d \cdot \dot{\delta}_p. \qquad (19)$$

Figure 13 shows the advantage of the new spacing control approach in simulation. The lead vehicle's velocity profile along the path is reproduced more accurately by the new approach, allowing the following vehicle to keep a safe speed if the lead vehicle slows down to go around a corner, such as at the section marked by the grey stripe. The constant time headway approach leads to a velocity which is 40% to 50% faster than the lead vehicle in this region whereas the new approach tracks the desired velocity to within 15%. Experimental tests for the spacing control are ongoing.

6 Conclusions

A robust algorithm has been developed for tracking the lead vehicle using a laser range finder. The algorithm has been tested in real-world experiments, with RMS position error of $65mm$ and orientation error of $5.8°$.

A technique to convert relative lead vehicle positions into a trajectory over time has been developed and tested, with a lateral error of up to $1.2m$ in a real-world test.

Various lateral control schemes have been implemented and tested in simulation and then experimentally at speeds of approximately $15km/h$. The best controller performed with an RMS lateral deviation of $37cm$ and a maximum deviation of $77cm$.

A new trajectory-based spacing control approach was developed and successfully tested in simulation, allowing lateral and longitudinal control to be decoupled.

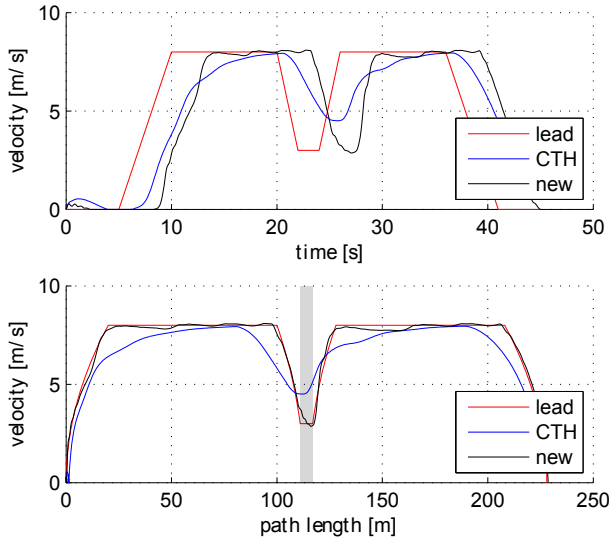

Figure 13: Velocity over time and over distance along the path

Future tests will combine lateral and longitudinal control to achieve full vehicle following. The On-Board Diagnostics (OBD) interface could also be explored to provide ego-motion.

Acknowledgments

The authors would like to thank the Energy and Fuels Research Unit at the University of Auckland for access to the dynamometer lab and the vehicle, as well as the Dynamics and Control lab for technical support.

References

[Chien *et al.*, 1994] C. C. Chien, P. Ioannou, and M. C. Lai. Entrainment and vehicle following controllers design for autonomous intelligent vehicles. *Proceedings of the American Control Conference*, pages 6–10, 1994.

[Cho and Hedrick, 1989] D. Cho and J. Karl Hedrick. Automotive powertrain modeling for control. *Journal of Dynamic Systems, Measurement, and Control*, 111:p. 568–576, 1989.

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

[Fujioka and Omae, 1998] T. Fujioka and M. Omae. Vehicle following control in lateral direction for platooning. *Vehicle System Dynamics*, 29:p. 422–437, 1998.

[Gehrig and Stein, 1998] S. K. Gehrig and F. J. Stein. A trajectory-based approach for the lateral control of car following systems. *IEEE International Conference on Systems, Man, and Cybernetics*, 4:p. 3596–3601, 1998.

[Girault, 2003] Alain Girault. A hybrid controller for autonomous vehicles driving on automated highways. 2003.

[Linderoth *et al.*, 2008] M. Linderoth, K. Soltesz, and R. M. Murray. Nonlinear lateral control strategy for nonholonomic vehicles. *Proceedings of the American Control Conference*, 2008.

[Namik *et al.*, 2006] H. Namik, T. Inamura, and K. Stol. Development of a robotic driver for vehicle dynamometer testing. ACRA, Auckland, New Zealand, 2006.

[Sotelo, 2003] M. A. Sotelo. Lateral control strategy for autonomous steering of ackermann-like vehicles. *Robotics and Autonomous Systems*, 45(3):p. 223–233, 2003.

[Swaroop *et al.*, 2001] Darbha Swaroop, Karl Hedrick, and S.B. Choi. Direct adaptive longitudinal control of vehicle platoons. *IEEE Transactions on Vehicular Technology*, 50(1):150–161, 2001.

[Thrun and others, 2007] S. Thrun et al. *Stanley: the robot that won the DARPA Grand Challenge*, pages 1–43. Springer Tracts in Advanced Robotics. Springer, 2007.

[Vaughan *et al.*, 2009] Richard T. Vaughan, Andrew Howard, and Brian P. Gerkey. The player project. http://playerstage.sourceforge.net/, 2009.

[White and Tomizuka, 2002] R. White and M. Tomizuka. Estimating relative position and yaw with laser scanning radar using probabilistic data association. *Proceedings of the American Control Conference*, 2002.

[Wit *et al.*, 2004] J. Wit, C.D. Crane, and D. Armstrong. Autonomous ground vehicle path tracking. *Journal of Robotic Systems*, 21(8):439–449, 2004.

[Wong *et al.*, 2008] N. Wong, C. Chambers, K. Stol, and R. Halkyard. Autonomous vehicle following using a robotic driver. *15th International Conference on Mechatronics and Machine Vision in Practice*, pages 115–120, 2008.

[Zhang and Ioannou, 2005] Jianlong Zhang and Petros Ioannou. Adaptive vehicle following control system with variable time headways. *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 3880–3885, 2005.