# MULTI-AGENT AUTONOMOUS PILOT FOR SINGLE-TRACK VEHICLES

Dana Vrajitoru
Intelligent Systems Laboratory
Computer and Information Sciences
Indiana University South Bend
South Bend, IN 46634, USA
email: danav@cs.iusb.edu

Russel Mehler
Intelligent Systems Laboratory
Computer and Information Sciences
Indiana University South Bend
South Bend, IN 46634, USA
email: rmehler@cs.iusb.edu

**ABSTRACT**
In this paper we're introducing a physically-based model of a motorized single-track vehicle (a motorcycle) that we have implemented in an interactive application. The vehicle can be driven by a multi-agent autonomous pilot using a set of perceptual information to make decisions on the control units of the vehicle.

**KEY WORDS**
Physically-based modeling, agents, autonomous vehicle

## 1 Introduction

Single track vehicles (STV) present somewhat different challenges than double-track vehicles like cars because their balance is not automatically achieved by the disposition of the wheels, but is a dynamic result of their motion. The input from the rider also has an additional component using the centrifuge force to change direction.

The autonomous pilots are an important aspect of developing the vehicles of the future and they represent an interesting challenge for intelligent control applications as well as for traffic control [8, 13]. This project starts from a simulation of a vehicle with a multi-agent autonomous pilot using perceptual information. The application aims to control the vehicle in a non-deterministic way inspired from the behavior of a human driver and using the same kind of perceptual information to make decisions.

In this paper we introduce a simulation of a vehicle with a multi-agent autonomous pilot using perceptual information. The goal of this application is not to develop a pilot capable of driving the vehicle in a stable and deterministic way, but to simulate the behavior of a human driver under various circumstances on the road.

The intelligent agents represents a modern approach in artificial intelligence and they have been extensively utilized for many applications [16, 17]. Several approaches have applied multi-agent models to the simulation of autonomous drivers [10, 15] and our application follows similar ideas. A related research direction focuses on traffic flow simulation [14, 8] or trajectory planning [2].

Most of the research on autonomous pilots is directed toward piloting aircrafts [9, 11, 1, 4], and cars [12]. Our approach targets motorcycles which have not yet been studied as extensively as the other types of vehicles and which represent a more challenging modeling problem.

The application we are presenting in this paper is developed using the ideas and concepts that can be observed in game engines. It is implemented using the OpenGL library and provides real-time interaction for a human player. The visual interface of the application allows the human user to adjust the point of view and to drive the vehicle, which in our case is a motorcycle. The application includes an autonomous pilot that can be toggled on and off as well as a test circuit that the human or automatic driver must attempt to complete.

The automatic pilot is a multi-agent probabilistic application with a separate configuration interface where each agent is an independent process acting on one of the control units of the vehicle, as for example, the gas, the brakes, the handlebars, or the steering wheel. The agents use some information about the current status of the vehicle to make a decision about an action to be taken on their respective control units. This information includes both status data, like the current speed, and perceptual data, as the visible distance on the road in the direction of movement, the lateral distance to the border of the road, and the current slope. The performance of the automatic pilot is compared with the performance of a trained human.

The paper is structured the following way. Section 2 describes the physical model and the equations that we have used in our simulation. Section 3 introduces our multi-agent automatic pilot. Section **??** presents the GUI and other implementation details for our application. Section 4 presents some results of our simulation and compares them with the performance of a human player. We finish the paper with some conclusions.

## 2 Physical Model of the Single-Track Vehicle

In this section we introduce the physical model of a single-track motorized vehicle, as for example a motorcycle, and the motion equations we have used to implement the interactive application.

## 2.1 The Vehicle Control and Degrees of Freedom

A motorized STV, as for example, a motorcycle, can be model as a system made of several units with various degrees of freedom that can be driven through several control units. Figure 1 shows these parameters of the physical model for a motorcycle.

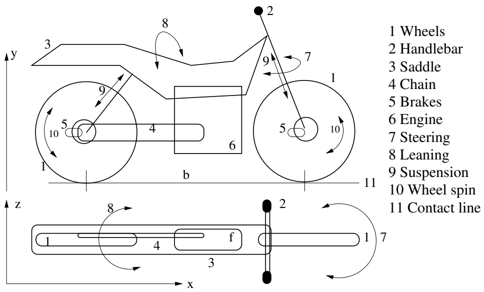


Figure 1. A motorcycle with control units and degrees of freedom

The STV is a non-holonomic dynamic system with six degrees of freedom: the rotation of the wheels around the axis parallel to $Oz$, the rotation of the handlebar and the front wheel around the fork axis (steering), the front and back translation along the suspension axis, and the rotation of the whole vehicle around the $Ox$ axis in a system of coordinates relative to the motorcycle where the origin is in the center of the vehicle, on the ground level.

The driver can control the vehicle through five inputs: the handlebar steering, leaning the vehicle laterally, the throttle and the two brakes, front and back.

The state of the STV is described at any moment by the current position of the vehicle's center on the road, the current direction of movement which can be described either as a vector or as an angle in the $(x, z)$ plane plus a slope, in general determined by the road. The model must also include the state of each unit with a degree of freedom and the current action of the driver on each of the control units. These two components are in general defined relative to the STV's internal system of reference.

Various aspects of the physical model of the bicycle have been studied before. The closest model to our purposes is [5] which considers a bicycle as a nonlinear, nonholonomic, non-minimum phase system. The stability and control of a bicycle are also of interest [7], [6], as well as the study of its aerodynamics [3].

## 2.2 STV Motion and Control

The STV is modeled as a reduced state system of continuous variables. The generalized coordinates of the vehicle at a particular moment are by

$$q = (s, \alpha, \theta)^T \qquad (1)$$

where $s(t) = (x(t), z(t))$ represents the *spatial position* of the STV, $\theta$ is the *orientation angle* determining the *direction of movement* $d = (\cos\theta, \sin\theta)$, $\alpha$ the *leaning angle*. Let $\phi$ be the *steering angle*. Since our STV is based on a motorcycle and not a bicycle, we have imposed the constraint that $-\pi/3 \leq \phi \leq \pi/3$. Figure 2 shows these angles and coordinates.

The vertical component of both $s$ and $d$ is determined by the road altitude and slope at the given spatial position and considering the vehicle orientation. In this paper we consider the road to be close enough to the surface of the Earth such that the gravitational acceleration is the constant $g$ and the altitude of the vehicle does not really influence its motion. Let $\sigma(s, d)$ be the angle made by contact line of the vehicle with the $(x, z)$ plane, depending on its position and orientation.

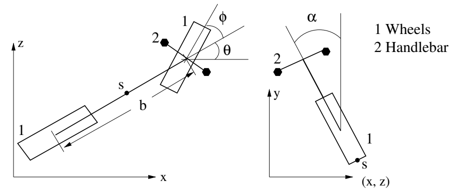


Figure 2. STV coordinate system

The driver's input into the system consists in $u = (\tau, \beta_f, \beta_r, \phi, \alpha)$ where $\tau$ is an acceleration component along the moving direction $d$ and $\beta_f, \beta_r$ represent the front and rear brakes respectively.

The nonholonomic constraints can be expressed by the following, where $b$ is the distance between the two contact points of the wheels on the ground:

$$-x' \sin\theta + z' \cos\theta = 0 \qquad (2)$$

$$b \cos\phi\, \theta' - \sin(\phi + \theta) x' + \cos(\phi + \theta) z' = 0 \qquad (3)$$

Equation 2 expresses the fact that the STV moves in the direction of the vector $d$. Equation 3 allows us to compute the change in orientation due to steering. Both equations are adapted from [5].

In particular, if $-\pi/2 < \phi < \pi/2$, we can compute the change in the orientation angle due to steering as

$$\Delta\theta = \frac{\sin(\phi + \theta)\Delta x - \cos(\phi + \theta)\Delta z}{b \cos\phi} \qquad (4)$$

Let $v = s'$ be the momentary speed or velocity in the direction of movement, and $a = v' = s''$ the momentary acceleration in the direction of movement. We consider the motion of the vehicle defined by Newtonian mechanics. The position and velocity of the vehicle at $t + \Delta t$ are defined by $s(t + \Delta t) = s(t) + \Delta s$, $v(t + \Delta t) = v(t) + \Delta v$ where:

$$\Delta s = d\left(v \cdot \Delta t + a \frac{\Delta t^2}{2}\right), \quad \Delta v = a \cdot \Delta t \qquad (5)$$

In our case, the acceleration is defined by the gravity, the friction, the drag, and the throttle. The brakes do not act as a simple negative acceleration, but rather have the effect of adding to the drag coefficient which otherwise depends on the air and is rather small. We also take into consideration the engine brake which is also a drag force and which prevents the speed of the vehicle from increasing indefinitely.

$$a = \tau + g\ \sin\sigma - k\ g\ \cos\sigma - Dv^2 \tag{6}$$

In this equation, $g = 9.8\ m/s^2$ is the gravitational acceleration on the surface of the Earth, $k$ is the coefficient of friction, and $D$ is the coefficient of drag, defined as a sum of the air resistance, of the brakes input $B_f$ and $B_r$, and the engine brake. These forces cause the speed of the vehicle to become constant after a while for any given throttle input $\tau$, and also cause the vehicle to eventually stop when no throttle input is given anymore. Together with the friction force, they will prevent a resting motorcycle from going downhill if the slope $\sigma$ is not null, and prevent the speed from increasing indefinitely due to a gravitation in the direction of movement when the vehicle is going downhill.

The latest model of the motorcycle also implements an additional change in direction of movement due to leaning that is already available for the human player, together with failsafe conditions that can cause the vehicle to crash if they are not met, as for example, leaning too far depending on the velocity. We are also developing a model where the gravity center of the vehicle depends on the acceleration being applied to the vehicle and where the acceleration and speed of the front and rear wheels are dissociated.

Figure 3 shows the main window of the application displaying the motorcycle and the road with some of the perceptual cues and the outline of the road triangulation.

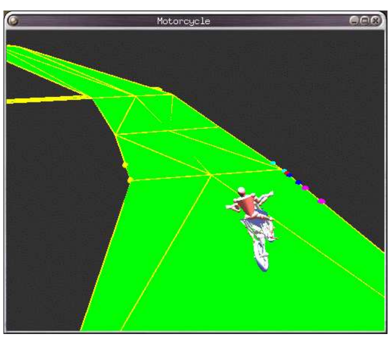


Figure 3. The main application window displaying the vehicle

## 3 The Autonomous Pilot

In this section we present the ideas that we have used to implement the autonomous pilot for our motorcycle.

### 3.1 Perceptual Information

The autonomous pilot is using perceptual information to make decisions about the vehicle driving. This information is inspired from the perceptual cues that a human driver would also be paying attention to while driving a vehicle.

In our application, the pilot is aware of the following measures:

The *visible front distance*, denoted by $front$, defined as the distance to the border of the road from the current position in the direction of movement, scaled by the length of the vehicle. This distance is a measure of how much of the road is visible ahead and also of how straight the road is in front of the vehicle. We will also mention this as the horizon.

The *front probes*, denoted by $frontl$ and $frontr$, are defined as the distances to the border of the road from the current position of the vehicle in directions rotated left and right by a small angle from the direction of movement. They give the pilot an indication as to which way from the direction of movement the front distance would become larger.

The *lateral distances*, denoted by $leftd$ and $rightd$, are measure of the lateral distance from the vehicle to the border of the road, at a short distance in front, simulating what the pilot might be aware of without turning their head to look. In our computations we use $lat_n = |\frac{leftd-rightd}{\max(leftd,rightd)}|$, the normalized difference between the lateral distances. A high value of this measure indicates a turn in the road, or the vehicle being close to the border. A value close to 0 indicates that the vehicle is in the middle of the road.

The *slope*, denoted by $slope$, is a perceptual version of $\sigma$ which is discretized to simulate the intuitive notion of road inclination that a human driver would have, like almost flat, slightly inclined up or down, or highly inclined up or down. This simulates the fact that a human pilot is not aware of the precise value of $\sigma$.

Figure 4 shows an example of the geometrical definition of these measures.

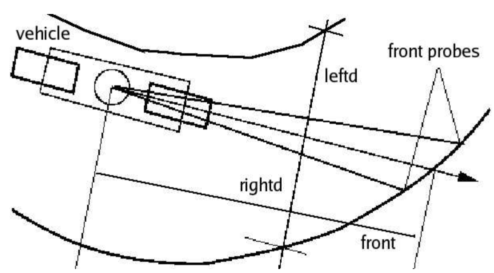


Figure 4. Perceptual information used by the autonomous pilot

Beside the perceptual information, the autonomous pilot is using the current status of the motorcycle to make decisions about the action to be taken on each of the control units of the vehicle. The status includes measures like the $v$, $\tau$, $B_f$, $B_r$, $\phi$.

## 3.2 Control Units

The motorcycle is driven by several control units (CUs). Each of them is controlled by an independent agent with a probabilistic behavior. The agents are not active during the computation of each new frame simulating the evolution of the vehicle on the road, but only once in a while in a non-deterministic manner. This simulates the behavior or a human driver that may not be able to instantly adapt and take action based on the road situation and requires a certain reaction time.

The current control units focus on the gas (throttle), the brakes, and the handlebar. Each of these control units is independently adjusted by an agent whose behavior is intended to drive the motorcycle safely in the middle of the road at a speed close to a given limit. In our case, the agents controlling the throttle and the handlebar are in general more active than the agent controlling the brakes.

The next paragraphs introduce the equations used by each of our agents to make a decision and to perform an action. The equations comprise a fair number of coefficients and thresholds. The configuration of each agent uses independent values for the coefficients.

**The Throttle.** This CU and its corresponding agent controls the amount of gas that is supplied to the engine and determines the acceleration that the vehicle is submitted to.

The input for this agent is represented by $(v, front, leftd, rightd, slope)$. The agent uses a minimal speed threshold $v_{low}$, a maximal speed threshold over which the speed is considered unsafe, and the given speed limit $v_{limit}$. The agent aims to keep the vehicle speed above $v_{low}$ and below the maximal one, and also close below the $v_{limit}$.

The agent detects a turn in the road by testing $lat_n$ and eventually decreases $\tau$ to allow a safe turn. A similar rule is applied to the visible distance in front of the driver: a low value for $front$ indicates an unsafe road situation requiring a reduced $\tau$. In any other situation it attempts to keep the speed close to $v_{limit}$.

Equation 7 represents the conditions under which $\tau$ should be decreased, causing the vehicle to slow down. If the condition bellow i not fulfilled, $\tau$ is given an appropriate value to keep the speed close to the limit.

$$v > v_{low} \wedge$$
$$(v > v_{limit} \ \vee \ tr_{lat} > lat_n \ \vee \ tr_{fr} > front) \qquad (7)$$

Equation 8 illustrates the change in throttle performed by the agent, where $c_{incv}$, $c_{decv}$, and $c_{sl}$ are configurable coefficients. The actual amount of the change is a probabilistic quantity equally distributed in a small neighborhood around the computed value.

$$\Delta\tau = c_{incv}(front - thr_{fr})(v - v_{low}) +$$
$$c_{decv}\left((v - v_{limit}) + tr_{lat} + tr_{fr}\right) + c_{sl} \cdot slope \qquad (8)$$

**The Brakes.** The agent controlling the brakes has a similar behavior to the one controlling the throttle because we have assumed that the rules deciding when the speed should decrease are of general purpose. The equations of this agent though are simpler because the brakes can only decrease the speed and not increase it. The speed is only decreased when a more drastic change is necessary than we can assume will be achieved by decreasing $\tau$ only.

Equation 7 is also used to determine when to apply a force on the brakes, but the coefficients $c_{incv}$, $c_{decv}$, and $c_{sl}$ can have different values for this agent. The change in either $B_r$ or $B_f$ is the following equation.

$$\Delta B_{r,f} = c_{decv}\left((v - v_{limit}) + tr_{lat} + tr_{fr}\right) - c_{sl} \cdot slope \qquad (9)$$

**The Handlebar.**

The agent controlling the handlebar of the motorcycle is the one with the most complex behavior. This agent is also using the lateral distances to the border of the road, as well as the front probes $frontl$ and $frontr$, to make decisions about turning the handlebar left or right. The agent turns the handlebar in the direction of the longer distance between the left and right, getting away from the closest border.

The agents first considers the immediate distance to either side, given by the lateral distances. Thus, if the vehicle is not situated within a given percentage (like 20%) of the center of the road, then the agent moves the handlebar to direct the vehicle towards the center.

If the first measure (lateral distances) doesn't provide a condition for the vehicle to turn, the agent estimates the distance forward to the horizon. Based on the front probes and the front distance, the agent moves towards the center of the horizon (given by the front probes). The amount of handlebar turning depends on the distance to the horizon, a bigger change being required if the horizon is closer.

The agent starts by making a decision whether to use the lateral distances as reference or the front probes. Let us denote by $probe_n$ the normalized of the difference between the front left and right probes as shown in Equation 10 and by $probe_{abs} = |probe_n|$ the absolute value of this quantity.

$$probe_n = \frac{frontl - frontd}{\max(frontld, frontd)} \qquad (10)$$

Let us denote by $lat_{diff}$ the quantity used by the agent to decide if it must turn and in which direction, computed according to the following equation.

$$lat_{diff} = \begin{cases} lat_n & \text{if} \quad lat_n > thr_1 \text{ and} \\ & \qquad frontd > thr_2 \\ \frac{lat_n + probe_n}{2} & \text{if} \quad lat_n > thr_3 \text{ and} \\ & \qquad frontd > thr_4 \\ probe_n & \qquad otherwise \end{cases} \qquad (11)$$

where $thr_i, i = 1, 4$ are configurable coefficients.

The amount of the change depends on how different the left and right lateral distances are either right next to the vehicle or at the intersection with the road in front of it, based on the measure $lat_{diff}$, and on the speed. Thus, if the speed of the motorcycle is lower, the handlebar has to be turned more to achieve a given change in direction. If the vehicle moves at a higher speed, smaller changes in the orientation of the handlebar are necessary to obtain the same change in direction.

The handlebar agent will update the handlebar position if the condition expressed in Equation 12 is fulfilled. This means that a change is necessary either if the lateral difference measure is greater than the threshold $thr_{lat}$, or if the distance in the direction of movement to the border of the road is smaller than another threshold, $thr_{front}$.

$$|lat_{diff}| > thr_{lat} \text{ or } front < thr_{fr} \tag{12}$$

If we denote $\Delta\phi = \phi(t+\Delta t) - \phi(t)$, then the general rule for modifying the orientation of the handlebar is shown in Equation 13, but the actual amount of the change is a probabilistic quantity equally distributed in a small neighborhood around the computed value.

$$\Delta\phi = c_{hbar}\left(lat_{diff} + \frac{thr_{fr} - front}{thr_{fr}}\right) \tag{13}$$

**Alerting Agent.** Beside all the agents that are in direct control of the motorcycle, the pilot comprises a fourth agent that does not perform any action on the vehicle. While the other agents are active only occasionally, this agent is probing the vehicle and road condition for every new frame and is capable of activating one of the other agents if the situation case requires special attention. Thus, if the speed of the vehicle is either too high or too low, or if the visible front distance is too short, or if the difference between the left and right lateral distances is too high, this agent considers the situation to be exceptional, meaning unsafe, and generates an alert event that will randomly activate one of the agents that can take action on the motorcycle and correct the issue.

Equation 14 describes the condition that must be true for the alerting agent to consider that the vehicle's status is not safe and one of the agents coordinating the vehicle must be triggered to take some action and correct the situation. The alerting agent only generates an alert message and does not decide which other agent will perform the necessary action.

$$\begin{aligned} v < \ c_{vlow} v_{limit} \ \vee \ v > \ c_{vhigh} v_{limit} \ \vee \\ lat_{abs} < \ thr_{lat} \ \vee \ front < \ thr_{fr} \end{aligned} \tag{14}$$

## 4 Experiments

We have performed our experiments with a circuit consisting of 3 loops such that a portion of it being elevated with respect to the rest of the road. The circuit was designed with the intention to test the ability of the pilot to drive correctly in situations where the road is turning both to the left and to the right, and also where the slope of the road is ascending and descending.

The autonomous pilot was capable of completing the circuit with an average speed comparable to the speed at which a human player is capable of handling the vehicle correctly along the entire length of the circuit.

Beside being successful at completing the circuit, the autonomous pilot has also shown interesting behavior as compared to the human player. In the case of the human, the entire set of control keys is rarely used and once the player achieves a speed that is perceived as comfortable, the rest of the circuit is covered by controlling only the lateral movement of the motorcycle. In the case of the automatic pilot, we can observe a higher variation of the speed, which makes the simulation closer to a real-life situation.

Considering the general direction of movement, we have also observed that the autonomous pilot is much more sensitive to the differences between the left and right distance to the border of the road than the human player and the changes in direction happen a lot more often. The pilot also seems to be capable of remedying dangerous situations better than the human player but the general impression of the ride is that it is less smooth.

To evaluate the autonomous pilot, we have computed a number of statistics based on 5 completed circuits by the autonomous pilot as well as by two human subjects. Table 1 shows these results in which the rows have to following meaning: average time to complete the circuit (time), average speed over the entire circuit (v), maximal speed that the player has achieved at any time(max v), the average value of $lat_{abs}$, the total distance covered to complete the circuit, the total number of left and right turns, the number of times that the player has exited the road. The last row shows the number of times that the pilot has exited the road with no immediate recovery, in which case the experiment was restarted.

From this table we can note that the average timing of the human player and of the autonomous pilot are comparable, even though the human can still handle higher speeds. Experiences with higher speeds for the pilot resulted in the vehicle leaving the road without managing to recover.

## 5 Conclusions

In this paper we have presented an application simulating a motorcycle that can be driven by both a human player and an autonomous pilot. The application is implemented based on the physical equations describing the vehicle's attributes, motion, and road behavior. The physical model of the vehicle has been described in Section 2.

The main focus of the paper has been the description of the automatic pilot. This part of the application is implemented using a multi-agent model in which each control unit of the vehicle, like the throttle, the handlebar, and the brakes, is controlled by an independent agent with a

Table 1. Comparison of the performance of the autonomous pilot and the human players

| | Human 1 | Human 2 | Pilot |
|---|---|---|---|
| Total time | 97.4 | 79.2 | 357.9 |
| Average speed | 6.19 | 8.94 | 2.2 |
| Maximum speed | 8.75 | 12.26 | 6.34 |
| Total distance | 2312.05 | 2316.83 | 2465.39 |
| Left turns | 121.4 | 119.2 | 277.4 |
| Right turns | 51.4 | 47 | 185.3 |
| Lateral balance | 0.29 | 0.36 | 0.66 |
| Exit the road | 0 | 0.4 | 3.9 |
| Recovery time | 0 | 11.2 | 9.36 |
| Completed circuits | 100% | 100% | |
| Perfect circuits | 100% | 60% | 20% |

probabilistic behavior. Section 3 described in details the equations used by each of the agents to drive the vehicle.

The experiments described in Section 4 have shown that the autonomous pilot is capable of successfully driving the motorcycle over the entire length of a test circuit in conditions that are comparable to the performance of a human driver.

As a limitation of our system in its current state, the coefficients that determine the behavior of the pilot have to be chosen by the user and the task of finding good values for them is a tedious one. As a direction for future research we intend to explore some methods that would allow the agents to find the appropriate behavior through adaptation and learning.

## References


[1] T. F. Abdelzaher, E. M. Atkins, and K. G. Shin. QoS negotiation in real-time systems and its application to automated flight control. *IEEE Transactions on Computers*, 49(11):1170–1183, 2000. Best of RTAS '97 Special Issue.

[2] E. M. Atkins, R. H. Miller, T. VanPelt, K. D. Shaw, W. B. Ribbens, P. D. Washabaugh, and D. S. Bernstein. Solus: An autonomous aircraft for flight control and trajectory planning research. In *Proceedings of the American Control Conference (ACC)*, volume 2, pages 689–693, 1998.

[3] A. Fuchs. Trim of aerodynamically faired single-track vehicle in crosswinds. In *Proceedings of the 3rd European Seminar on Velomobiles*, Roskilde, Denmark, 1998.

[4] V. Gavrilets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron. Aggressive maneuvering of small helicopters: a human centered approach. *International Journal on Robotics Research*, 2001.

[5] N. Getz. Control of balance for a nonlinear nonholonomic no-minimum phase model of a bicycle. In *American Control Conference*, Baltimore, June 1994.

[6] N. Getz and J. Marsden. Control for an autonomous bicycle. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 21-27 1995.

[7] D. Jones. The stability of the bicycle. *Physics Today*, 23(4):34–40, 1970.

[8] T. Kelly. Driver strategy and traffic system performance. *Physica A*, 235:407–417, 1997.

[9] I. Martinos, T. Schouwenaars, J. De Mot, and E. Feron. Hierarchical cooperative multi-agent navigation using mathematical programming. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2003.

[10] T. Al-Shihabi and R.R. Mourant. Toward more realistic behavior models for autonomous vehicles in driving simulators. *Transportation Research Record*, (1843):41–49, 2003.

[11] J. De Mot and E. Feron. Spatial distribution of two-agent clusters for efficient navigation. In *IEEE Conference on Decision and Control*, Maui, HI, 2003.

[12] R.R. Mourant and S. Marangos. A virtual environments editor for driving scenes. In *Proceedings of the International Conference on Computer, Communication and Control Technologies*, volume IV, pages 379–384, Orlando, Florida, 2003.

[13] R.R. Mourant and D. Refsland. Developing a 3d sound environment for a driving simulator. In *Proceedings of the Ninth International Conference on Virtual Systems and Multimedia*, pages 711–719, Montreal, Canada, 2003.

[14] K. Nagel. Particle hopping models and traffic flow theory. *Physical Review E*, 53:4655, 1996.

[15] Rahul Sukthankar, Shumeet Baluja, and John Hancock. Multiple adaptive agents for tactical driving. *Applied Intelligence*, 9(1):7–23, 1998.

[16] G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 2000.

[17] M. Wooldridge. *Introduction to MultiAgent Systems*. John Wiley & Sons, 2002.