

SAN097-2352C  
SAND-97-2352C

# Decentralized Fuzzy Control of Multiple Nonholonomic Vehicles

CONF-980623--

Brian J. Driessen  
John T. Feddema  
Kwan S. Kwok

(Sandia National Laboratories, Albuquerque, NM 87185-0439)

RECEIVED

OCT 10 1997

OSTI

## Abstract

This work considers the problem of controlling multiple nonholonomic vehicles so that they converge to a scent source without colliding with each other. Since the control is to be implemented on simple 8-bit microcontrollers, fuzzy control rules are used to simplify a linear quadratic regulator control design. The inputs to the fuzzy controllers for each vehicle are the (noisy) direction to the source, the distance to the closest neighbor vehicle, and the direction to the closest vehicle. These directions are discretized into four values: Forward, Behind, Left, and Right, and the distance into three values: Near, Far, Gone. The values of the control at these discrete values are obtained based on the collision-avoidance repulsive forces and the change of variables that reduces the motion control problem of each nonholonomic vehicle to a nonsingular one with two degrees of freedom, instead of three. A fuzzy inference system is used to obtain control values for inputs between the small number of discrete input values. Simulation results are provided which demonstrate that the fuzzy control law performs well compared to the exact controller. In fact, the fuzzy controller demonstrates improved robustness to noise.

## 1. Introduction

The distributed control of multiple robotic vehicles has been a subject of significant recent interest. When hundreds or thousands of such vehicles are involved, clearly to be cost effective each vehicle must be cheap. Thus, sensors are likely to be cheap, and therefore noisy, and the amount of compute power and memory on board each vehicle is likely to be small. This brings about the need to be able to have robustness to noisy sensor measurements while simultaneously using a simple controller. Kalman filters can be relatively expensive to implement, in terms of required computation power and memory. Fuzzy control provides an alternative approach. In this section, we will review previous related research.

Miyata et al used a steepest descent method to tune fuzzy rules. The cost function in the steepest descent search was the sum of the squares of the deviations from a set of training data, and the search variables were the parameters that defined the piece wise linear membership functions of the inputs to the fuzzy inference system. They demonstrated that general piece wise linear membership functions took

significantly fewer steepest descent iterations than conventional triangular membership functions and gaussian membership functions, especially for the approximation of general complicated input/output relationships.

Nomura et al also considered the tuning of fuzzy rules to match closely a set of input-output data. They also used steepest descent. They included the crisp output values as part of the optimization variable set. They showed that the number of iterations required was four orders of magnitude smaller than for a neural network using back propagation. Kim et al also applied a steepest descent approach in the tuning of fuzzy rules, in a robotic visual servoing application.

Fukuda and Shimojima used a combination of genetic algorithms and steepest descent to tune fuzzy rules. Here the genetic algorithm approach allowed them to actually adjust the *number* of membership functions in addition to their shapes.

There has been much work in multiple mobile robotic vehicle problems, much of which has not considered the use of fuzzy logic. Some of these will now be discussed. Reynolds [8] considered the formation of flocks, herds, and

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

19980423 094

THIS QUALITY INFORMATION

## DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

schools in simulations in which multiple autonomous agents were driven away from each other and other obstacles by inverse square law repulsive forces. Part of the motivation behind Reynold's work was the impression of centralized control exhibited by actual bird flocks, animal herds, and fish schools, despite the fact that each agent (bird, animal, or fish) is responding only to its limited-range local perception of the world, in Reynold's opinion. Reynolds stated "Natural flocks seem to consist of two balanced, opposing behaviors: a desire to stay close to the flock and a desire to avoid collisions within the flock." He also stated "A bird might be aware of three categories: itself, its two or three nearest neighbors, and the rest of the flock." The desire to be close to the flock while avoiding collisions was modeled by force fields that attracted a bird to its two to three closest neighbors but prevented it from getting too close to these neighbors.

Arkin [1] studied an approach to "cooperation without communication," for multiple mobile robots. He states that "by enabling many simple robots to cooperate together on a large task, it becomes possible to solve a problem that would be infeasible using a large-scale machine." Arkin also points out that centralized master/slave or hierarchy-based approaches have drawbacks of potential communication bottlenecks and less robustness than completely decentralized approaches in which each agent is autonomous and has the same control law. Arkin demonstrated such an approach for robots that are supposed to forage and retrieve objects (in a hostile environment). Anticipated applications mentioned were "housekeeping on the exterior of a space station or undersea base" and "mining operations at remote locations." Kube and Zhang [6] also considered decentralized robots performing tasks "without explicit communication." Much of their study examined comparisons of behaviors of different social insects such as ants and bees. They considered a box-pushing task and utilized a subsumption approach as well as ALN (Adaptive Logic Networks). Asama [2] intelligently points out that "an autonomous and decentralized system has two essentially contradictory characteristics, autonomy and cooperativeness, and the biggest problem in the study of distributed autonomous robotic systems is how to reconcile these two features."

Noreils [7] conducted work toward achieving cooperation between mobile robots for

indoor environment applications. He dealt with robots that were not necessarily homogeneous. That is, one subset of the robots may have capabilities that another subset does not have. His architecture consisted of three levels: functional level, control level, and planner level. The planner level was the high-level decision maker. He developed a special purpose language for his application.

Chen and Luh [5] examined decentralized control laws that drove a set of mobile robots into a specific formation. In particular, a circle formation was considered. Similarly, Yamaguchi and Arai [9] studied line-formations, and so did Yoshida et al, [10].

The work herein considers the problem of controlling multiple nonholonomic robotic vehicles which converge to a scent source without collisions between vehicles, in the presence of noisy measurements and a small amount of compute-power and memory on board each vehicle.

## 2. Problem Statement

In this section, we will present a description of the problem. We have  $N$  nonholonomic robotic vehicles, whose detailed description will be presented shortly. Each vehicle has the following (noisy) measurements available to it.

1. Distance to the closest neighbor vehicle ( $r$ , meters).
2. Direction to the scent source in vehicle coordinates ( $\phi$ , radians).
3. Direction to the closest neighbor vehicle in vehicle coordinates ( $\alpha$ , radians).

The objective is to control these vehicles to converge to the scent source, without colliding with each other. We will now present a detailed description of the geometry of each robotic vehicle.

Figure 1 below shows a schematic of the vehicle.

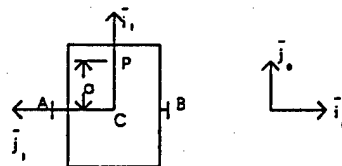


Figure 1. Schematic of Vehicle, Showing Reference Frames

Frame 0 (whose axes are  $\bar{i}_0$  and  $\bar{j}_0$ ) is a fixed frame. Frame 1 is fixed to the vehicle. Point C is at the midpoint of the independent driving wheels at A and B. Point P is fixed to the vehicle and offset forward by distance  $a$ . Let  $u_1$  and  $u_2$  denote the velocities of points A and B, respectively (in the  $\bar{i}_1$  direction). Also, let  $R$  denote the distance between A and B.

### 3. Preliminary, Non Fuzzy, Control Law

This section presents a preliminary, non fuzzy, control law, which will be fuzzified to produce the fuzzy controller in the next section of this paper. Clearly we want a vehicle to move toward the goal but avoid collisions with other vehicles. We want an attractive force that pulls the vehicle in the direction  $\phi$  of the scent source. This attractive force will be given by

$$k \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} \quad (3.1)$$

where the constant  $k$  is a scaling coefficient and where the vector in (3.1) is expressed in frame 1 coordinates.

The collision avoidance is obtained by using a  $1/r^3$  repulsive force exerted along the line from the closest neighbor vehicle to the vehicle whose control we are considering. The direction of this repulsive force, in frame 1 (vehicle coordinate frame), is obtained through measurement of  $\alpha$  (angular direction to the closest vehicle, in vehicle coordinates). In particular, this repulsive force, expressed in frame 1 coordinates is

$$-\frac{kc_2}{r^3} \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \quad (3.2)$$

where the value of  $c_2$  is chosen large enough that any pair of vehicles will maintain a safe distance apart.

Because of the no-slip condition at the two wheels (at A and B in Figure 1), the velocity of point C (Figure 1) cannot be arbitrarily specified. It can have no component in the  $\bar{j}_1$  direction (Figure 1). Thus, we cannot specify the motion of point C. However, as will be shown shortly, we can specify the motion of point P (in Figure 1). Therefore the approach will be to let the repulsive and attractive forces determine the velocity of point P. The absolute velocity of point P, expressed in frame 1 coordinates, can be shown to be

$${}^1\bar{v}_{P/0} = \begin{bmatrix} 1/2 & 1/2 \\ -a/R & a/R \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (3.3)$$

The determinant of the matrix in (3.2) is  $a/R$  and therefore this matrix is always nonsingular. So, if  ${}^1\bar{v}_{P/0}$  is specified, we can always solve for the required inputs  $u_1$  and  $u_2$  that produce this velocity. In particular, we have

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{bmatrix} 1 & -R/2/a \\ 1 & R/2/a \end{bmatrix} {}^1\bar{v}_{P/0} \quad (3.4)$$

So, we will let

$${}^1\bar{v}_{P/0} = k \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} - \frac{kc_2}{r^3} \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \quad (3.5)$$

and then the control  $(u_1, u_2)^T$  is determined by (3.4).

The direction of motion of point P in (3.5) can be shown to be the LQR solution for a single obstacle, where the cost function  $J$  is given by

$$J \equiv 1/2((x_p - x_s)^2 + (y_p - y_s)^2) + c_2 \frac{1}{(x_p - \bar{x})^2 + (y_p - \bar{y})^2} \quad (3.6)$$

where  $(\bar{x}, \bar{y})$  is the position of the obstacle and  $(x_s, y_s)$  is the position of the goal. Differentiating (3.6) with respect to  $(x_p, y_p)^T$ , we obtain

$$\nabla J = \begin{pmatrix} x_p - x_s \\ y_p - y_s \end{pmatrix} - c_2 \frac{1}{[(x_p - \bar{x})^2 + (y_p - \bar{y})^2]^2} \begin{pmatrix} x_p - \bar{x} \\ y_p - \bar{y} \end{pmatrix} \quad (3.7)$$

If we note that  $r = \sqrt{(x_p - \bar{x})^2 + (y_p - \bar{y})^2}$  and let  $\hat{\phi}$  and  $\hat{\alpha}$  be the angular direction from P to the goal and the obstacle to P, respectively, in frame 1 coordinates, and also let  $s \equiv \sqrt{(x_p - x_s)^2 + (y_p - y_s)^2}$ , then (3.7) becomes

$$J = -s \begin{pmatrix} \cos \hat{\phi} \\ \sin \hat{\phi} \end{pmatrix} + c_2 / r^3 \begin{pmatrix} \cos \hat{\alpha} \\ \sin \hat{\alpha} \end{pmatrix} \quad (3.8)$$

We will now see that the first term of  $(-\nabla J)$  is in the same direction as our attractive force in (3.1) and the second term is in the same direction as our repulsive force in (3.2). Let  $d_A$  be the direction vector from P to  $(x_s, y_s)$  and let  $d_R$  be the direction vector from point  $(\bar{x}, \bar{y})$  to point P. Then,  $d_A$ , expressed in frame 0 coordinates, is

$${}^0d_A = \begin{pmatrix} \cos \hat{\phi} \\ \sin \hat{\phi} \end{pmatrix} \quad (3.9)$$

and  $d_R$ , expressed in frame 0 coordinates, is

$${}^0d_R = -\begin{pmatrix} \cos \hat{\alpha} \\ \sin \hat{\alpha} \end{pmatrix} \quad (4.0)$$

Let  $R$  be the 2 by 2 rotation matrix between frame 0 and frame 1. Then,

$${}^0d_A = R^T {}^1d_A = R^T \begin{pmatrix} \cos \hat{\phi} \\ \sin \hat{\phi} \end{pmatrix} = \begin{pmatrix} \cos \hat{\phi} \\ \sin \hat{\phi} \end{pmatrix} \quad (4.1)$$

$${}^0d_B = R^T {}^1d_B = -R^T \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} = -\begin{pmatrix} \cos \hat{\alpha} \\ \sin \hat{\alpha} \end{pmatrix} \quad (4.2)$$

We also have

$${}^0v_{P/0} = R^T {}^1v_{P/0} \quad (4.3)$$

or, considering the first term of  ${}^1v_{P/0}$  in (3.5),

$${}^0v_{P/0} \leftarrow kR^T \begin{pmatrix} \cos \hat{\phi} \\ \sin \hat{\phi} \end{pmatrix} = k \begin{pmatrix} \cos \hat{\phi} \\ \sin \hat{\phi} \end{pmatrix} = k {}^0d_A \quad (4.4)$$

and the second term:

$${}^0v_{P/0} \leftarrow -kc_2 / r^3 R^T \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} = -kc_2 / r^3 \begin{pmatrix} \cos \hat{\alpha} \\ \sin \hat{\alpha} \end{pmatrix} = -\frac{kc_2}{r^3} {}^0d_B \quad (4.5)$$

Thus the velocity  ${}^0v_{P/0}$  is in fact composed of two terms, each of which is in the direction of the negative gradient of the corresponding term of  $J$ .

#### 4. Fuzzification of the Control Law of Section 3

We discretized the inputs to the controller into the following possible values:

Direction to Goal ( $r$ ): Forward1, Left, Behind, Right, or Forward2

Direction to Obstacle ( $\phi$ ): Forward1, Left, Behind, Right, or Forward2

Distance to Obstacle ( $\alpha$ ): Small, Medium, Gone

Thus, we have  $5*5*3 = 75$  possible "fuzzy" inputs to the controller. The value of "Small" was chosen to be 1.5 meters, that of "Medium" to be 3 meters, and that of "Gone" to be 10 meters. A Sugeno Fuzzy Inference System was used to interpolate between the exact values of the control as given in equations (3.4) and (3.5) at these 75 data points. Triangular membership functions were used for each of the discrete values of the three inputs to

the controller (Direction to Goal, Direction to Obstacle, and Distance to Obstacle). The resulting fuzzy controller has the same input-output relationship as the exact controller at these discrete 75 points, but has different values, in general, at any points in between. The membership functions are shown in Figures 2 and 3 below.

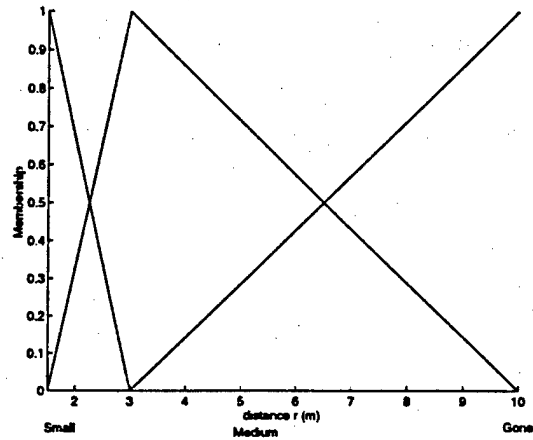


Figure 2: Membership Functions for Distance to Obstacle

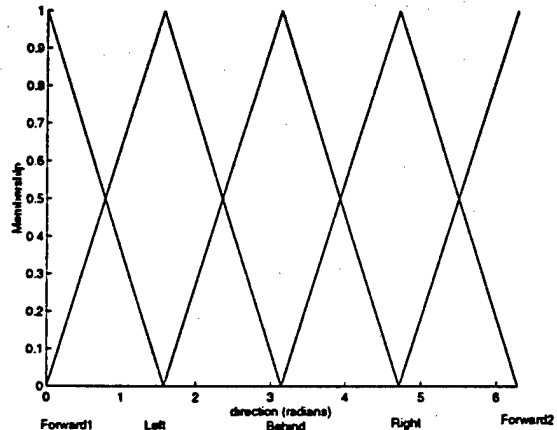


Figure 3: Membership Functions for Direction to Goal and Direction to Obstacle

The reason that we can use rather coarse fuzzy logic membership functions is that convergence is achieved as long as the estimate of the angle to the goal is within  $\pm 90$  degrees of the actual angle. This can be shown by considering a linear perturbation of nonlinear dynamics of the vehicle.

$$\dot{x} = f(x, u) = f(x_o, u_o) + \frac{\partial f}{\partial x} \Big|_{x_o, u_o} (x - x_o) + \frac{\partial f}{\partial u} \Big|_{x_o, u_o} (u - u_o)$$

(4.1)

where  $\mathbf{x}$  is the  $(x,y)$  position of the point "p" on the vehicle and orientation  $\theta$ ,  $\mathbf{u}$  are the commanded right and left linear wheel velocities,  $f(\mathbf{x}, \mathbf{u})$  are the first order vehicle dynamics, and  $\mathbf{x}_o$  and  $\mathbf{u}_o$  are linearized operating points. This can be rewritten as

$$\Delta \dot{\mathbf{x}} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_o, \mathbf{u}_o} \Delta \mathbf{x} + \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{x}_o, \mathbf{u}_o} \Delta \mathbf{u} \quad (4.2)$$

where

$$\begin{aligned} \Delta \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) - f(\mathbf{x}_o, \mathbf{u}_o) \\ \Delta \mathbf{x} &= \mathbf{x} - \mathbf{x}_o \end{aligned}$$

The first order model of a skid-driven vehicle  $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$  is

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\theta} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \cos\theta - \frac{a}{R} \sin\theta & \cos\theta + \frac{a}{R} \sin\theta \\ \sin\theta + \frac{a}{R} \cos\theta & \sin\theta - \frac{a}{R} \cos\theta \\ \frac{1}{R} & -\frac{1}{R} \end{bmatrix} \begin{bmatrix} u_r \\ u_l \end{bmatrix} \quad (4.3)$$

or

$$\dot{\mathbf{x}} = \mathbf{B}(\mathbf{x})\mathbf{u}$$

where  $R$  is one-half the wheel base,  $a$  is the distance between the vehicle center C and point p, and  $\theta$  is the orientation of the vehicle.

$$\text{If } \mathbf{u}_o = \mathbf{0}, \text{ then } \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_o, \mathbf{u}_o} = \mathbf{0} \text{ and } \Delta \mathbf{u} = \mathbf{u}.$$

$$\text{Since } \left. \frac{\partial f}{\partial \mathbf{u}} \right|_{\mathbf{x}_o, \mathbf{u}_o} = \mathbf{B}(\mathbf{x}_o), \text{ then}$$

$$\Delta \dot{\mathbf{x}} = \mathbf{B}(\mathbf{x}_o) \Delta \mathbf{u} \quad (4.4)$$

Suppose we choose the control to be a weighted inverse Jacobian which is a function of the estimated state  $\hat{\mathbf{x}}$ . Then,

$$\Delta \mathbf{u} = [\mathbf{WB}(\hat{\mathbf{x}})]^{-1} \Delta \mathbf{x} \quad (4.5)$$

where

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The matrix  $\mathbf{W}$  is chosen to drive  $x_p, y_p$  to the desired reference position yet leave  $\theta$  unconstrained.

Considering only the position of the vehicle, let

$$\mathbf{p} = \begin{bmatrix} x_p \\ y_p \end{bmatrix}$$

then

$$\Delta \dot{\mathbf{p}} = \mathbf{WB}(\mathbf{x}_o) [\mathbf{WB}(\hat{\mathbf{x}})]^{-1} \Delta \mathbf{p} \quad (4.6)$$

or

$$[s\mathbf{I} - \mathbf{WB}(\mathbf{x}_o) [\mathbf{WB}(\hat{\mathbf{x}})]^{-1}] \Delta \mathbf{p} = \mathbf{0} \quad (4.7)$$

For  $\Delta \mathbf{p} \rightarrow \mathbf{0}$ ,  $\mathbf{WB}(\mathbf{x}_o) [\mathbf{WB}(\hat{\mathbf{x}})]^{-1}$  must be positive definite [15]. It can be shown that for the skid driven dynamics in Equation (4.3), this occurs if and only if  $-90^\circ < \hat{\theta} - \theta < 90^\circ$ .

## 5. Application of the Exact and Fuzzy Control Laws to the Multi-Vehicle Problem

In this section, we consider the case of many vehicles trying to reach the goal point. The obstacle for each vehicle  $i$  is taken to be the vehicle that is nearest to vehicle  $i$ . Then, the same control law (from sections 3 and 4) is applied to this multi-vehicle problem, in which all of the vehicles are trying to converge to the goal position. An additional constraint, not considered in the previous sections, was imposed. Namely, a maximum wheel velocity (velocity of points A and B in Figure 1) of  $v_{\max} = 0.45 \text{ m/s}$  was imposed. For the generation of the fuzzy controller data points of section 2, we scaled the speed-coefficient  $k$  (equation (3.5)) down to  $k = 0.1136$ . This automatically guaranteed that the fuzzy controller would never generate a wheel velocity greater than  $v_{\max}$ . The exact controller also uses this value of  $k$  and additionally truncates  $\bar{v}_{p/0}$  to a small enough value so that neither wheel velocity exceeds  $v_{\max}$ . The parameter values used in all

simulations herein will be  $a=R=0.58$  meters, and  $c_2 = 10$ .

### 5. Simulations of the Multi-Vehicle Problem in the Presence of Noise in Measurements and a One-Second ZOH

We wanted to assess the effect of having a 1-second update rate on  $r$ ,  $\phi$ , and  $\alpha$  and noise in  $\phi$ ,  $r$ , and  $\alpha$ . Figures 4 and 5 below correspond to simulations with the exact and fuzzy controllers, respectively, in which there was not only  $\pm 90$  degrees of noise in the direction-to-goal,  $\pm 9$  degrees of noise in the direction-to-obstacle and  $\pm 0.4$  meters of noise in the distance to the obstacle (closest vehicle), but also a slow 1-second update rate on the inputs to the controller. The controls were held constant over each such interval. The noise on the direction to the obstacle was taken to be smaller than the noise on the direction to the goal, since the former will come from communication signals between bugs, while the latter might come from cheap chemical sensors. The total integration time for both Figure 4 and Figure 5 was 780 seconds. Also, Figure 6 shows a blown up view of the paths for Figure 4 and Figure 5.

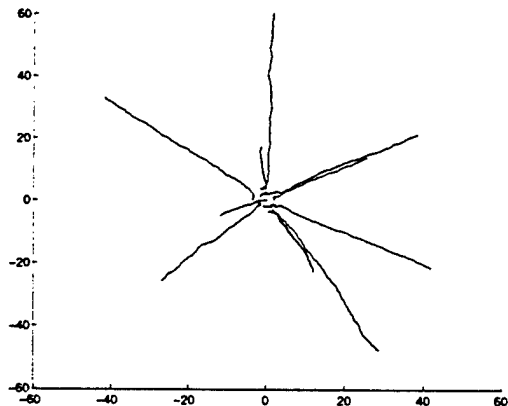


Figure 4. Simulation Results for Exact Controllers With  $\pm 90$  Degrees of Noise on Direction to Goal,  $\pm 9$  Degrees of Noise on Direction to Closest Vehicle, and  $\pm 0.4$  Meters of Noise on Distance to Closest Vehicle, and 1-second ZOH

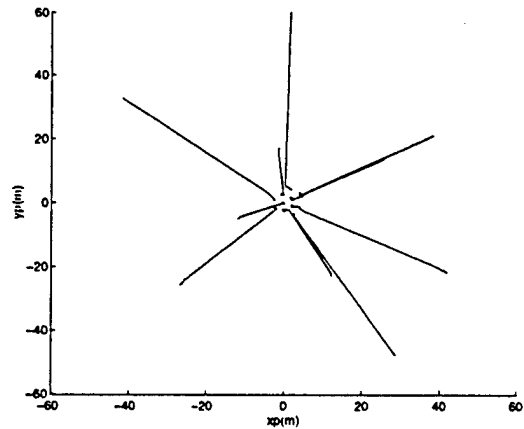


Figure 5. Simulation Results for Fuzzy Controllers With  $\pm 90$  Degrees of Noise on Direction to Goal,  $\pm 9$  Degrees of Noise on Direction to Closest Vehicle, and  $\pm 0.4$  Meters of Noise on Distance to Closest Vehicle, and 1-second ZOH

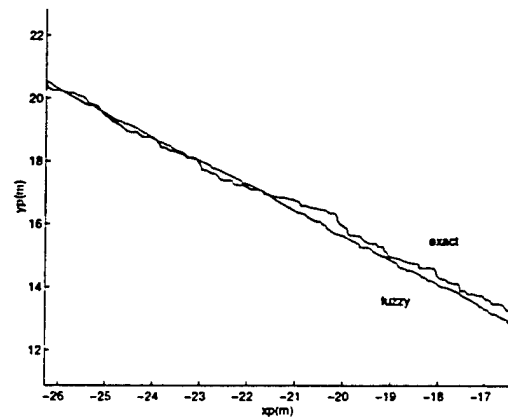


Figure 6. Blown Up View of Simulation Results of Figures 4 and 5

### 7. Comparison of Fuzzy and Exact Controllers With Kalman Estimation

It is interesting to compare Figures 4 and 5. The fuzzy controller exhibits a smoother approach to the goal. A more expensive smoothing controller would be one that uses a Kalman estimator. A comparison of such a controller will be made in this section. A summary of an extended Kalman filter implementation is as follows. Suppose we have a nonlinear system:

$$\dot{x} = f(x, u) \quad (7.1)$$

$$y = h(x) \quad (7.2)$$

with  $x \in R^n$ ,  $u \in R^r$ ,  $y \in R^m$ , and where (7.2) is the output equation and we have a new noisy

measurement  $\bar{y}$  every  $T$  seconds. Also, suppose that we have either an approximate or exact state-transition function  $\hat{f}$  which takes us from time  $t$  to time  $t+T$ , where the input  $u$  is constant over the  $T$ -second interval:

$$x(t+T) = \hat{f}(x(t), u(t)) \quad (7.3)$$

Let  $\Lambda \in R^{m \times m}$  be an estimate of the output covariance matrix and  $Q \in R^{r \times r}$  an estimate of the input covariance matrix. Let  $P_0$  be an initial estimate for  $P$  (to be defined). Let  $F \equiv \partial f(\hat{x}, u) / \partial x$  and  $G \equiv \partial f(\hat{x}, u) / \partial u$ . One extended Kalman estimator, whose purpose is usually to provide an estimate  $\hat{x}$  of the state  $x$ , is as follows, where an initial estimate of  $\hat{x}$  would be provided. At each time step (of length  $T$ ), calculate:

$$H = \partial h(\hat{x}) / \partial x \quad (7.4)$$

$$K = PH^T (\Lambda + HPH^T)^{-1} \quad (7.5)$$

$$P = (I - KH)P \quad (7.6)$$

$$\hat{x} = \hat{x} + K(\bar{y} - h(\hat{x})) \quad (7.7)$$

Calculate the control  $u = u(\hat{x})$ . (In our case this controller is the one in section 3.) Then further update  $\hat{x}$  and  $P$  for the next time step by

$$P = FPF^T + GQG^T \quad (7.8)$$

$$\hat{x} = \hat{f}(\hat{x}, u(\hat{x})) \quad (7.9)$$

In our problem the state equation (7.1) is:

$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \frac{1}{2} \cos \theta & \frac{1}{2} \cos \theta \\ \frac{1}{2} \sin \theta & \frac{1}{2} \sin \theta \\ -1/R & 1/R \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (7.10)$$

where  $\theta$  is the angle from  $\bar{i}_0$  to  $\bar{i}_1$ , measured counterclockwise, in Figure 1 and where  $R$  is the distance between points A and B in Figure 1 and our output is  $\phi$ , which can be easily calculated as a function of  $x$  as can the Jacobean  $H = \partial \phi / \partial x$ . Initially, however, to avoid  $0-2\pi$  wrap around issues, two outputs were used:  $\cos \phi$  and  $\sin \phi$ . An exact state transition function  $\hat{f}$  can be derived and is shown below

$$\theta(t+T) = \theta(t) + T(-u_1 + u_2) / R \quad (7.11)$$

$$x_c(t+T) = x_c(t) + R/2 \frac{u_1 + u_2}{u_2 - u_1} [\sin(\theta(t+T)) - \sin(\theta(t))], u_1 \neq u_2 \quad (7.12)$$

$$x_c(t+T) = x_c(t) + Tu_1 \sin(\theta(t)), u_1 = u_2 \quad (7.13)$$

$$y_c(t+T) = y_c(t) - R/2 \frac{(u_1 + u_2)}{(u_2 - u_1)} [\cos(\theta(t+T)) - \cos(\theta(t))], u_1 \neq u_2 \quad (7.14)$$

$$y_c(t+T) = y_c(t) + Tu_1 \sin(\theta(t)), u_1 = u_2 \quad (7.15)$$

Figures 7 and 8 show single-vehicle simulations, for ease in comparison with such a controller. The total integration time is 1000 seconds. The values of  $\Lambda$ ,  $Q$ , and  $P_0$  used were 10I, 0.1I, and 10I, respectively. In this Kalman estimator implementation, measurement of  $\phi$  and  $u$  alone are not sufficient to achieve estimation of the full state  $x$ . More measurements, such as the distance to the goal, would be needed to obtain full state estimation. However, in scent seeking problems, distance to the goal may not be measurable since the strength of the scent source may not be known.

We see that the more computationally intensive Kalman filter also exhibits more smoothing than the unfiltered exact controller, but the controller overshoots the target a little before coming back to it, whereas the fuzzy controller does not.

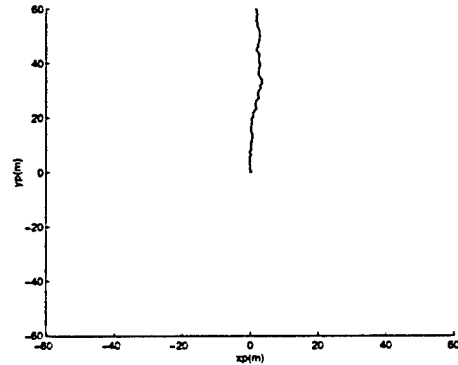


Figure 7. Exact Controller, Single Vehicle, +/-90 Degrees Noise on Direction To Goal and a 1-Sec ZOH

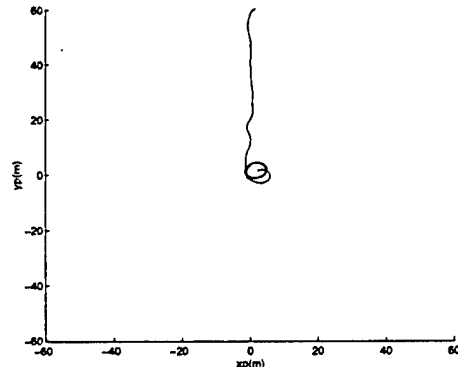


Figure 8. Kalman Controller, Single Vehicle, +/-90 Degrees of Noise on Direction To Goal and a 1-Sec ZOH

## 8. Conclusion

This work investigated decentralized fuzzy control of multiple nonholonomic vehicles. A preliminary non fuzzy controller was designed using a change of variables that reduced the control problem from one with three degrees of freedom to a non-singular one with only two degrees of freedom. Fuzzy control was investigated because the control is to be implemented on simple 8-bit microcontrollers in the future. The fuzzy controller exhibited a smoother response than the preliminary non fuzzy controller in the presence of noisy measurements and its smoothness and performance compared favorably to both a controller that used a Kalman estimator and the preliminary non fuzzy controller, respectively.

#### ACKNOWLEDGMENTS

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

#### References

- [1] Arkin, Ronald C., "Cooperation Without Communication: Multiagent Schema-Based Robot Navigation," *Journal of Robotic Systems* 9(3), 1992, pp. 351-364.
- [2] Asama, "Distributed Autonomous Robotic System Configured With Multiple Agents and Its Cooperative Behaviors," *Journal of Robotics and Mechatronics*, Vol. 4, No. 3, 1992, pp. 199-204.
- [3] Brooks, Rodney A. and Flynn, Anita M., "Fast, Cheap and Out of Control: A Robot Invasion of the Solar System," *Journal of the British Interplanetary Society*, Vol. 42, 1989, pp. 478-485.
- [4] Brooks, Rodney A., "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, March 1986, pp. 14-23.
- [5] Chen, Qin and Luh, J.Y.S., "Coordination and Control of a Group of Small Mobile Robots," *IEEE International Conference on Robotics and Automation*, Vol. 3, 1994, pp. 2315-2320.
- [6] Kube, Ronald C and Zhang, Hong, "Collective Robotics: From Social Insects to Robots," *Adaptive Behavior*, Vol. 2, No.2, pp. 189-218.
- [7] Noreils, Fabrice R., "Toward a Robot Architecture Integrating Cooperation Between Mobile Robots: Application to Indoor Environment," *The International Journal of Robotics Research*, Vol. 12, No. 1, February 1993, pp. 79-98.
- [8] Reynolds, Craig W., "Flocks, Herds, and Schools," *Computer Graphics*, Volume 21, No. 4, July 1987, , pp. 25-34.
- [9] Yamaguchi, Hiroaki, and Arai, Tamio, "Distributed and Autonomous Control Method for Generating Shape of Multiple Mobile Robot Group," *IEEE International Conference on Intelligent Robots and Systems*, Vol. 2, 1994, pp.800-807.
- [10] Yoshida, E.; Arai, Tamio, Ota, Jun; Miki, Tomoyoshi, "Effect of Grouping in Local Communication System of Multiple Mobile Robots," *IEEE International Conference on Intelligent Robots and Systems*, Vol. 2, 1994, pp. 808-815.
- [11] Miyata, Hitoshi; Ohki, Makoto; and Ohkita, Masaaki, "Self-Tuning of Fuzzy Reasoning by the Steepest Descent Method and Its Application to a Parallel Parking," *IEICE Trans. Inf. & Syst.*, Vol. E79-D, No. 5, May 1996, pp. 561-569.
- [12] Nomura, Hiroyoshi; Hayashi, Isao; and Wakami, Noboru, "A Learning Method of Fuzzy Inference Rules by Descent Method," *IEEE International Conference on Fuzzy Systems*, March, 1992, pp. 203-210.
- [13] Fukuda, Roshio and Shimojima, Koji, "Fusion of Fuzzy, NN, GA to the Intelligent Robotics," *1995 IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 3, October, 1995, pp. 2892-2897.
- [14] Kim, Joo Gon; Cha, Dong Hyuk; Cho, Hyung Suck; Kim, Seung Ho, "An Auto-Tuning Fuzzy Rule-Based Visual Servoing Algorithm for a Slave Arm," *IEEE International Symposium on Intelligent Control*, August, 1995, pp. 177-182, pp. 177-182.
- [15] Samson, C., Le Borgne, M., and Espiau, B., *Robot Control: The Task Function Approach*, pp. 218-264, 1991.

M98000242



Report Number (14) SAND--97-2352C  
CONF-980623--

Publ. Date (11) 199709

Sponsor Code (18) DOE/MA, XF

UC Category (19) UC-905, DOE/ER

DOE