Designing Stable Finite State Machine Behaviors using Phase Plane Analysis and Variable Structure Control

John T. Feddema, Rush D. Robinett, Brian J. Driessen*
Sandia National Laboratories
P.O. Box 5800, MS 1003
Albuquerque, NM 87185

Abstract

This paper discusses how phase plane analysis can be used to describe the overall behavior of single and multiple autonomous robotic vehicles with finite state machine rules. The importance of this result is that we can begin to design provably asymptotically stable group behaviors from a set of simple control laws and appropriate switching points with decentralized variable structure control. The ability to prove asymptotically stable group behavior is especially important for applications such as locating military targets or land mines.

1. Introduction

In recent years, there has been considerable interest in the control of cooperative multiple robotic vehicles. The vision being that multiple robotic vehicles can perform tasks faster and more efficiently than an individual.

In the area of distributed mobile robot systems, much work has been performed and is summarized in [1]. The strategies of cooperation encompass theories from such diverse disciplines as artificial intelligence, game theory/economics, theoretical biology, distributed computing/control, animal ethology, and artificial life.

For instance, Reynolds [2] simulated the formation of flocks, herds, and schools 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 Reynolds' 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.

Arkin [3] studied an approach to "cooperation without communication" for multiple mobile robots that are supposed to forage and retrieve objects in a hostile environment. Kube and Zhang [4] 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 [5, 6] as well as ALN (Adaptive Logic Networks). Noreils [7] dealt with robots that were not necessarily homogeneous. His architecture consisted of three levels: functional level, control level, and planner level. The planner level was the high-level decision maker. Chen and Luh [8] examined decentralized control laws that drove a set of mobile robots into a circle formation. Similarly, Yamaguchi and Arai [9] studied line-formations, and so did Yoshida et al, [10].

Most of these works do not include a formal development of the system controls from a stability point of view. Many of the schemes such as the Subsumption approach rely on stable controls at a lower level while providing coordination at a higher level.

In this paper, we address the stable control of single and multiple vehicles using phase plane analysis and variable structure control techniques. The objective is to design individual finite states (in the computer science sense of the word "state") which may or may not be convergent. However, when the transitions between states are correctly designed, a convergent stable behavior results. In the phase plane, these states are represented by sets of trajectories. The initial condition of each state trajectory is the final condition of the preceding state trajectory.

This interpretation of finite state machine behavior lends itself nicely to Variable Structure Control (VSC) design tools. A VSC system changes the dynamics of a system by abruptly switching at defined states (in the controls sense of the word) to an alternative function of a set of possible continuous functions of the state [11-13]. This switching may be realized with simple relays. Sliding mode control is a special case of VSC and relies on feedback linearization to create switching surfaces which the system dynamics are guided onto. Lyapunov's direct

* 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.
method is often used to design control surfaces which
guide the system to a given goal.

The next section develops a dynamic model of a skid-
driven vehicle. Section 3 discusses a finite state machine
implementation used to control a single vehicle tracking a
line, and Section 4 describes how phase plane analysis has
been used to analyze this behavior. Section 5 expands this
analysis to multiple vehicles which converge to the origin
while not running into each other. Sliding mode control is
used to design the convergent behaviors. The last section
discusses the conclusions of this work and future research
directions.

2. Vehicle Dynamics

The following is a brief summary of the derivation of the
equations of motion of a skid-driven vehicle. In
particular, these dynamics represent the 16 cm$^2$ (1 in$^2$)
Miniature Autonomous Robot Vehicle (MARV) shown in
Figure 1. Although the vehicle has four wheels, the front
wheels slide and the back wheels are independently driven
much like a skid-driven vehicle.

![Figure 1. Picture of MARV.](image)

The following derivation is similar to [14] except that the
parameters of the right and left sides are assumed equal
and the motor's armature effects are added. Referring to
Figure 2, the vehicle's equations of motion are

\[ M \frac{dv}{dt} = f_r + f_l \]  
\[ J \frac{d\omega}{dt} = B \left( f_r - f_l \right) \]

where $B$ is the wheel base, $M$ is the mass, $J$ is the
rotational moment of inertia, and $f_r$ and $f_l$ are forces
generated by the right and left wheels. Assuming no
slippage, the linear and angular velocities of the vehicle are

\[ v = \frac{1}{2} \left( R\omega_r + R\omega_l \right) \]  
\[ \omega = \frac{1}{B} \left( R\omega_r - R\omega_l \right) \]

where $R$ is the wheel radius, and $\omega_r$ and $\omega_l$ are the right
and left wheel angular velocities.

![Figure 2. Skid driven vehicle notation.](image)

The force generated by each wheel is related to the motor
torque, which in turn is related to the applied voltage of
the motor by the following equations.

\[ \tau_r = \gamma J_m \omega_r + \frac{1}{\gamma} \left( J_m \omega_r + D \omega_r + R f_r \right) \]  
\[ \tau_l = \gamma J_m \omega_l + \frac{1}{\gamma} \left( J_m \omega_l + D \omega_l + R f_l \right) \]

\[ \tau_r = \frac{K_I V_r - \gamma K_b K_I \omega_r}{\Omega} \]  
\[ \tau_l = \frac{K_I V_l - \gamma K_b K_I \omega_l}{\Omega} \]

where $\tau_r$ and $\tau_l$ are the right and left motor torque values, $\gamma$
is the ratio of the motor gearbox, $J_m$ is the moment of
inertia about the motor axis, $J_w$ is the moment of inertia
about the wheel, $D$ is the friction constant of the wheel, $\Omega$
is the motor armature resistance, $K_b$ is the motor's back
EMF, $K_I$ is the torque constant, and $V_r$ and $V_l$ are the right
and left motor voltages.
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.
The velocity and acceleration of the vehicle in the x and y directions are given by

\[
\begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = \begin{bmatrix}
\cos \theta \\
\sin \theta
\end{bmatrix} v 
\]

(9)

\[
\begin{bmatrix}
\ddot{x} \\
\ddot{y}
\end{bmatrix} = \begin{bmatrix}
\cos \theta \\
\sin \theta
\end{bmatrix} v + \begin{bmatrix}
-\sin \theta \\
\cos \theta
\end{bmatrix} \ddot{\theta}
\]

(10)

Note that \( \omega = \dot{\theta} \) and \( v = \sqrt{\dot{x}^2 + \dot{y}^2} \). Combining the above equations, the resulting equations of motion are

\[
\begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = \begin{bmatrix}
\cos \theta \\
\sin \theta
\end{bmatrix} \left( \frac{\gamma K_I}{R \Omega M_{eff}} (V_r + V_l) - \frac{2}{R^2 M_{eff}} \left( D + \frac{\gamma^2 K_b K_I}{\Omega} \right) \sqrt{\dot{x}^2 + \dot{y}^2} \right) 
+ \begin{bmatrix}
-\sin \theta \\
\cos \theta
\end{bmatrix} \dot{\theta} \sqrt{\dot{x}^2 + \dot{y}^2}
\]

(11)

where the effective mass and moment of inertia are given by

\[
M_{eff} = M + \frac{2}{R^2} \left( \gamma^2 J_m + J_w \right) 
\]

(13)

\[
J_{eff} = J + \frac{\beta^2}{2R^2} \left( \gamma^2 J_m + J_w \right)
\]

(14)

Two capacitance sensors are located in front of the vehicle's skid center (see Figure 2) and their positions are given by

\[
(x_{sr}, y_{sr}) = (x + a \cos \theta + a \sin \theta, y + a \cos \theta - a \sin \theta)
\]

(15)

\[
(x_{sl}, y_{sl}) = (x + a \cos \theta - a \sin \theta, y + a \cos \theta + a \sin \theta)
\]

(16)

The output voltage of the sensors \( s_r \) and \( s_l \) is inversely proportional to the distance from the wire. If we assume that the vehicle is tracking a straight wire along the x axis, then the sensor measurements are given by

\[
s_r = \frac{1}{\sqrt{(y + a \cos \theta - a \sin \theta)^2 + h^2}}
\]

(17)

\[
s_l = \frac{1}{\sqrt{(y + a \cos \theta + a \sin \theta)^2 + h^2}}
\]

(18)

where \( h \) is the height of the sensor from the wire.

### 3. Control Algorithm

MARV was designed to track a single conducting wire carrying a 96kHz signal. Particular attention was paid to the design of the control system to search out the wire, track it, and recover if the wire was lost. In Section 4, this example will be used to illustrate a simple phase plane analysis and to develop provably stable finite state machine behavior.

Approximately 250 lines of assembly code were written to implement the MARV's control in an embedded microcontroller. A set of if/then statements switches the control routine between four finite states: SEARCH, ROTATE, TRACK, and BACKUP. In the SEARCH state, the vehicle simply moves straight ahead until a change is detected in the two capacitance sensors on the bottom of the vehicle. If both sensors detect that the wire is right underneath the vehicle, the controller switches to the ROTATE state where one wheel drives forward while the other drives backward. Once the vehicle is straddling the wire, the control is switched to the TRACK state where right and left wheel velocities are regulated so that the difference between the right and left capacitance sensor is zero. Using this control, the vehicle will follow a curved wire. If the vehicle loses the wire and the capacitance measurements drop below a given threshold, then the vehicle backs up until it detects the wire again. If it does not detect the wire after a specified time out period, then it switches to the SEARCH state.

![Figure 3. State Transition diagram of MARV.](image-url)
is the state time, and $T_{out}$ is a constant time-out period. Based on the state decision, the program jumps to different routines which determine the duration of the pulse width modulated (PWM) signals that control the velocities of the two motors. During these routines, the state and time in that state are updated, and each routine ends by reading the sensor inputs. This organization resembles an augmented finite state machine. Similar to the work by Brooks [5], there is a time out associated with each state; however, we do not compute the results of each state in parallel and then decide which state to apply.

Both simulations and actual experiments were performed to test the performance of MARV's control. The vehicle tracked the wire as desired, but an interesting phenomenon occurred when approaching the wire at a more perpendicular angle. The vehicle would overshoot when in the TRACK state and then switch into the BACKUP state. By switching between these two states the vehicle was eventually able to track the wire as shown in Figure 4.

![Figure 4. Path of MARV when approached at a 52 degree angle.](image)

This behavior might be what some would call emergent. Emergent behavior is generally thought of as a complex global behavior arising from the interaction of simple local rules [2]. The simple rules in this case are the control laws in the TRACK and BACKUP states. The complex behavior is the ability to track the wire at larger angles than the system was initially designed to accommodate.

While emergence is usually viewed as having beneficial properties, it is very difficult to design for in general since it is presently not well understood. Some form of mathematical modeling is needed to explain the phenomenon, and in this paper, phase plane analysis is suggested as a means of explaining emergent behavior resulting from finite state machine programming.

4. Goal Searching With A Single Vehicle

In this section, we analyze the phase plane transition between the TRACK and BACKUP states. By switching between these two states when the sensor values reach predefined thresholds, the motion of the vehicle follows the trajectory in the phase plane plot in Figure 5.

![Figure 5. Phase plane trajectory using full nonlinear dynamics and simple MARV control laws. Transitions occur between TRACK, BACKUP, and TRACK states.](image)

In the SEARCH state, the vehicle approaches the line at a fixed orientation $\theta$. During the TRACK state, the vehicle will asymptotically orient itself over the line and $\theta$ will go to zero if $\theta$ is small. If $\theta$ is too large, the vehicle will overshoot the line and the controller will switch to the BACKUP state. Once the vehicle has backed up over the line, the controller will again switch into the TRACK state. This is repeated until $\theta$ is small enough to converge to the origin.

During the TRACK state, the proportional control law with gain $G$ is:

$$\text{If } s_r \geq s_l, \text{ then } V_r = V_o \text{ and } V_l = V_o + G(s_r - s_l).$$

$$\text{If } s_r < s_l, \text{ then } V_r = V_o + G(s_l - s_r) \text{ and } V_l = V_o. \quad (19)$$

The constant voltage $V_o$ keeps the vehicle moving forward. During the BACKUP state, both the left and the right voltages are $-V_o$. 

During the TRACK state, the proportional control law with gain $G$ is:

$$\text{If } s_r \geq s_l, \text{ then } V_r = V_o \text{ and } V_l = V_o + G(s_r - s_l).$$

$$\text{If } s_r < s_l, \text{ then } V_r = V_o + G(s_l - s_r) \text{ and } V_l = V_o. \quad (20)$$

The constant voltage $V_o$ keeps the vehicle moving forward. During the BACKUP state, both the left and the right voltages are $-V_o$. 

4
The phase plot in Figure 5 can be more easily interpreted if we simplify the sensor model and substitute into Equation (12). Assuming that the sensor is straddling the wire and that $y = 0$ and $h = 0$, the control law may be reduced to

$$V_r - V_l = -\frac{2G \sin \theta}{a \cos 2\theta}.$$  

If $s_r \geq s_l$, then $V_r - V_l = -\frac{2G \sin \theta}{a \cos 2\theta}$.  

If $s_r < s_l$, then $V_r - V_l = \frac{2G \sin \theta}{a \cos 2\theta}$.  

(21)  

(22)

For small angles of $\theta$, the control law can be approximated as:

$$V_r - V_l = -\frac{2G}{a} \theta.$$  

(23)

Substituting into Equation (12), the resulting equation of motion during the TRACK state is the familiar second order underdamped system

$$\ddot{\theta}(t) + 2\xi \omega_0 \dot{\theta}(t) + \omega_0^2 \theta(t) = 0.$$  

(24)

During the SEARCH and BACKUP states, the control is open loop (i.e., the capacitance sensors are not used). Since $V_r = V_l$, the resulting equation of motion is

$$\dot{\theta}(t) + \frac{1}{\tau} \dot{\theta}(t) = 0.$$  

(25)

Therefore, during the TRACK state, the vehicle moves along a trajectory approximately equal to a stable second order response. If this trajectory leads the vehicle outside the range of the sensors, a threshold value is reached and the controller goes into the BACKUP state. The BACKUP state moves the vehicle in the phase plane plot of $\theta$ to a zero angular velocity. Another threshold is reached when the vehicle straddles the wire, and the controller goes back into the TRACK state. This process is repeated until the vehicle is on a trajectory which is within the range of the sensors.

This simple example shows how phase plane analysis can be used to analyze and design asymptotically convergent finite state machine control laws. Each finite state has a trajectory on the phase plane whose initial conditions are determined from the final conditions of the previous finite state. It is important to note that individual finite state control laws do not have to be convergent for the overall control (which consists of many different finite states) to be convergent. For example, the BACKUP state is not convergent, yet when combined with the convergent TRACK state, the vehicle achieves its goal. As seen in the above example, this can be a powerful tool which can be used to overcome many sensing and actuation limitations.

This interpretation of finite state machine behavior also lends itself nicely to VSC design tools. In the next section, we will use VSC techniques to design the control of multiple vehicles performing a coordinated motion.

### 5. Goal Searching With Multiple Vehicles

Consider $N$ skid-driven vehicles randomly distributed on a planar surface. The goal of these vehicles is to go towards the origin without running into each other. To begin with, let's consider two finite states: goal seeking when no other vehicle is present, and goal seeking while avoiding other vehicles. Two sliding mode surfaces which seamlessly switch between these two finite states will be designed to guide the vehicles to the origin. At the end of this section, an additional finite state will be added to account for limited sensor range.

The $N$ subsystems can be described by the following nonlinear dynamics:

$$\dot{x}_i(t) = f(x_i(t), \dot{x}_i(t)) + B(x_i(t))u_i(t) \quad i = 1, \ldots, N$$

where

$$x_i(t) = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} \quad \text{and} \quad u_i(t) = \begin{bmatrix} V_{ri} \\ V_{li} \end{bmatrix}$$

and $f(x_i(t), \dot{x}_i(t)) \in \mathbb{R}^{3 \times 1}$ and $B(x_i(t)) \in \mathbb{R}^{3 \times 2}$ are given in Equations (11) and (12). When the vehicles are not close to one another, we can define a sliding surface which is directed to the origin as

$$s_i(t) = W_i x_i + \bar{W}_i \dot{x}_i = 0$$

where

$$W_i = \begin{bmatrix} w_{1i} & 0 & 0 \\ 0 & w_{2i} & 0 \end{bmatrix} \quad \text{and} \quad \bar{W}_i = \begin{bmatrix} \bar{w}_{1i} & 0 & 0 \\ 0 & \bar{w}_{2i} & 0 \end{bmatrix}.$$  

(27)

The nonzero diagonal terms can be used to prescribe the transient response while on the sliding surface. The equivalent control is determined by setting the time derivative of the sliding surface to zero and solving for the control.

$$\dot{s}_i(t) = W_i \dot{x}_i + \bar{W}_i \ddot{x}_i = W_i \dot{x}_i + \bar{W}_i f(x_i, \dot{x}_i) + \bar{W}_i B(x_i)u_i = 0$$

Therefore, the equivalent control on the sliding surface is given by

$$u_i = \left[ \bar{W}_i B(x_i) \right]^{-1} \left[ -W_i \dot{x}_i - \bar{W}_i f(x_i, \dot{x}_i) \right]$$

(28)

(29)

When not on the sliding surface (in the reaching mode), the term $-A_i \text{sgn}(s_i)$, where $A_i$ is positive definite, may be added to drive the system to the sliding surface.
We can prove asymptotic stability of the reaching control with the Lyapunov candidate function
\[ V_i = \frac{1}{2} s_i^T s_i \geq 0 \]
(31)

For an asymptotically stable solution, the time derivative of the Lyapunov function must be less than zero.
\[ \dot{V}_i = s_i^T \dot{s}_i = -s_i^T A_i \text{sgn}(s_i) < 0 \]
(32)

As seen in the equations above, the control is valid only if the inverse of \( \mathbf{W}_i \mathbf{B}(\mathbf{x}_i) \) exists. Unfortunately, \( \mathbf{W}_i \mathbf{B}(\mathbf{x}_i) \) is singular for Equation (11). This difficulty occurs because we are trying to control three free variables, \( \mathbf{x}_i \), with only two control parameters, \( \mathbf{u}_i \). If the state vector is modified to drive a point \( p \) in front of the vehicle to the origin, the number of free variables is reduced to two. If \( a \) is the distance to \( p \) in the moving body frame, then
\[ \begin{align*}
  x_p &= x + a \cos \theta \\
y_p &= y + a \sin \theta \\
  \dot{x}_p &= \dot{x} - \hat{\theta} a \cos \theta \\
  \dot{y}_p &= \dot{y} - \hat{\theta} a \sin \theta \\
  \end{align*} \]

The state equations can be rewritten as
\[ \ddot{p}_i(t) = \ddot{\mathbf{F}}(p_i(t), \dot{p}_i(t)) + \mathbf{B}(p_i(t))u_i(t) \]
(33)

where
\[ \begin{align*}
  p_i(t) &= \begin{bmatrix} x_{pi} \\ y_{pi} \\ \theta_i \end{bmatrix} \\
  \ddot{\mathbf{F}}(p_i, \dot{p}_i) &= \begin{bmatrix} -\left( K_1 v_i + a \hat{\theta}_i^2 \right) \cos \theta_i + (aK_2 - v_i) \hat{\theta}_i \sin \theta_i \\ \left( K_1 v_i + a \hat{\theta}_i^2 \right) \sin \theta_i + (-aK_2 + v_i) \hat{\theta}_i \cos \theta_i \\ -K_2 \hat{\theta}_i \end{bmatrix} \\
  \mathbf{B}(p_i) &= \begin{bmatrix} K_3 \cos \theta_i - aK_4 \sin \theta_i & K_3 \cos \theta_i + aK_4 \sin \theta_i \\ K_3 \sin \theta_i + aK_4 \cos \theta_i & K_3 \sin \theta_i - aK_4 \cos \theta_i \\ K_4 & -K_4 \end{bmatrix} \\
  v_i &= \sqrt{x_{pi}^2 + y_{pi}^2 + a^2 \hat{\theta}_i^2 + 2a \hat{\theta}_i \left( x_{pi} \sin \theta_i - y_{pi} \cos \theta_i \right)} \\
  \end{align*} \]

Since \( \mathbf{W}_i \mathbf{B}(\mathbf{p}_i) \) is not singular, the sliding mode control in Equation (30) with \( \mathbf{p}_i, \dot{\mathbf{p}}_i, \ddot{\mathbf{p}}_i, \mathbf{W}_i \mathbf{B}(\mathbf{p}_i) \) replacing \( \mathbf{x}_i, \dot{\mathbf{x}}_i, \ddot{\mathbf{x}}_i, \) and \( \mathbf{W}_i \mathbf{B}(\mathbf{x}_i) \) can be used to guide the vehicles to the origin. However, multiple vehicles will run into each other since there is no feedback between them which would drive them apart.

Fortunately, recent work on decentralized VSC (or DVSC) of interconnect systems [15-17] provides some tools for designing a controller which provide vehicle avoidance and at the same time guide them towards the origin. Let the dynamics be defined as:
\[ \ddot{p}_i(t) = \ddot{\mathbf{F}}(p_i(t), \dot{p}_i(t)) + \mathbf{B}(p_i(t)) \left[ \mathbf{u}_i(t) + \mathbf{g}(p_i(t), p_j(t)) \right] \]
(35)

where the term \( \mathbf{g}(p_i(t), p_j(t)) \) is the interaction between vehicles \( i \) and \( j \). A repulsive force, which is proportional to the inverse of the squared distance between the vehicles and directed away from vehicle \( j \), is created when
\[ \mathbf{g}(p_i(t), p_j(t)) = K \left( \frac{1}{d^2} - \frac{1}{d_0^2} \right) \left[ \mathbf{W}_i \mathbf{B}(p_i(t)) \right]^T \left[ \frac{(r_{pi} - r_{pj})}{d} \right] \left[ \frac{(r_{pi} - r_{pj})}{d} \right] \]
for \( d < d_0 \)
(36)

where \( K \) is user specified gain, \( d \) is the distance between vehicle \( i \) and \( j \), and \( d_0 \) is a user specified distance within which the repulsive force takes affect. A complementary sliding surface is
\[ s_i(t) = \mathbf{W}_i \dot{x}_i + \mathbf{W}_i \dot{x}_j - \mathbf{W}_i \int_0^t \mathbf{B}(p_i) \mathbf{g}(p_i(q), p_j(q)) dq \]
(37)

The equivalent control is the same as before (Equation 30), and it can be shown that the control is asymptotically stable (i.e. \( \dot{V}_i = s_i^T \dot{s}_i < 0 \)).

The response of three vehicles with and without repulsive forces is shown in Figures 6 and 7. In Figure 6, the vehicles move along a diagonal line to the x axis because the sliding surface parameters \( w_{1i}, w_{2i}, \) and \( \dot{w}_{2i} \) were set equal to one. Notice the vehicles run over each other while moving to the origin. In Figure 7, they
move along the same sliding surface defined by Equation (29) until they come within $d_0$ of each other. At that point, they switch to the controller with the repulsive force term $g(p_i(t), p_j(t))$ and are attracted to the sliding surface defined by Equation (37). The result is that the vehicles swarm about the origin and do not run into each other.

Although the trajectories overlap, the vehicles do not run into each other because of the repulsive force given in Equation (36). Figure 9 shows the final position of these vehicles. The cross shaped pattern is caused by the selection of the sliding surface weights.

Other behaviors can be added by changing the sliding surface. For example, if the vehicles have only a limited sensing range to the target, the attractive force could be changed to attract vehicles which cannot sense the target to those that can. Figure 8 shows the trajectories of 25 vehicles, some of which can not initially sense the origin. However, the trajectories overlap, the vehicles do not run into each other because of the repulsive force given in Equation (36). Figure 9 shows the final position of these vehicles. The cross shaped pattern is caused by the selection of the sliding surface weights.

**Figure 6.** Simulation of 3 vehicles using a sliding mode controller without repulsive force to move to the origin.

**Figure 7.** Simulation of 3 vehicles using a sliding mode controller with repulsive forces to avoid each other while moving to the origin.

**Figure 8.** Simulation of 25 randomly distributed vehicles with a limited sensing range of 100. If they do not sense the goal (the origin), they follow a vehicle which can.

**Figure 9.** Final position of 25 vehicles with a limited sensing range of 100. Two vehicles did not make it to the goal because they did not detect another vehicle which detected the goal.
6. Conclusion

Both the single and multiple vehicle examples show how phase plane analysis and variable structure control can be used to analyze and design convergent control behaviors. The single vehicle example showed that switching between finite states can be viewed as the switching of trajectories in the phase plane. For a convergent behavior, it is important to switch onto a trajectory which asymptotically leads to the goal. The multiple vehicle example showed how cooperative motion can be designed using sliding modes. Complex nonlinear problems can be solved by designing several sliding surfaces which perform different tasks and then defining when to switch to the appropriate sliding surface. Future research is needed to define sliding surfaces for other tasks such as searching for targets and maintaining geometric formations.

Acknowledgments

The authors greatly appreciate the help of Tom Weber and Barry Spletzer in designing and building MARV.

References

Report Number (14) SAND--98-0622C
CONF--980537--

Publ. Date (11) 19980310
Sponsor Code (18) DOE/ER, XF
UC Category (19) UC-900, DOE/ER