Abstract—In this paper we present an experimental implementation of the flocking algorithm presented in [1] for the double integrator model, in the case of wheeled mobile robots (WMR). We use the look ahead concept to feedback linearize the WMR model and obtain its double integrator form in which we apply the control signals in the form of voltages in wheel motors. A virtual leader is used to steer the flock in the desired direction. Comparisons of the experimental results with simulation verify the robustness of this algorithm to model uncertainties and communication delays.

I. INTRODUCTION

A landmark paper by Reynolds [9] in 1987 attracted a significant interest in studying the flocking behavior of animals in the context of coordination and formation of multiple agents. Inspired by this work, numerous research groups have studied the problem of coordinating the motion of multiple autonomous agents for the possible applications in broad areas like automated highways, satellite formation, coordinated movement of a cluster of vehicles for the search and rescue operation [5], and formation flying of unmanned aerial vehicles (UAVs) for the purposes like surveillance and distribution of sensor networks [7].

The primary objective of the approaches presented in [1], [6], [5] and [7] is to design a local decentralized controller for individual agents so that a global objective for the flock can be achieved. In most of the work done in this area of formation control, the basic objectives of controller design can be summarized as:

- Avoid inter-agent collisions and maintain group cohesion by keeping a desired separation.
- Align to the average group heading.
- Match the velocity with neighbors.

In [1] and [2], authors take double integrator models and design decentralized control laws based on potential functions for both, fixed and dynamic topology of flocks. The control laws have been extended for nonholonomic agents in [10]. In addition to the potential function, virtual leaders (beacon in [6] and γ-agents in [5]) are used in [6] and [5] to steer the flock in the desired direction.

A control Lyapunov function approach is proposed in [8]. Instead of designing a controller for some standard model, they have addressed the coordination problem for a class of robots for which control Lyapunov function can be found. Although there has been a lot of research in an effort to understand this problem, practical implementations of such control algorithms have rarely been reported. The work in [7] is motivated by a practical implementation of meter-scale UAV formation and many practical issues are considered in their control laws, yet, a successful implementation of flocking algorithms on UAVs has not been reported.

In this paper we present an implementation of flocking control laws on wheeled mobile robots. In [1], the authors present a class of control laws for fixed topology for a group of double integrator agents such that their velocities converge to the average group velocity, and collisions between the agents are avoided by maintaining the desired separation.
between all the agents. We introduce the concept of look-ahead to feedback linearize the WMR model in order to obtain it in the double integrator form, and then implement the control law proposed in [1]. We have successfully implemented the coordination algorithm mentioned in [1], in two WMRs and one virtual leader.

The remainder of this paper is organized as follows. In section II, we describe and model the WMR that we are using in the implementation. Feedback linearization and control is presented in section III. Simulation and experimental results are discussed in section IV and conclusions are drawn in section V.

II. WHEELED MOBILE ROBOT MODELING

Advancement in computation and communication technology has opened up a whole new domain of applications for the networked, distributed and decentralized control. In the course of preparation of a test bed for the practical implementation of proposed control strategies in the areas of coordination of multiple agents and networked control systems, we have built two wheeled mobile robots (3 & 4) in the Networked Control Systems Laboratory at the Electrical & Computer Engineering department of the University of New Mexico. These WMRs are actuated by DC motors attached to each of the wheels in a differential steering New Mexico. These WMRs are actuated by DC motors & Computer Engineering department of the University of New Mexico. These WMRs are actuated by DC motors attached to the center of mass of the WMR. (x, y) denote the position of the frame B with respect to the global coordinate frame U. The orientation of the frame B is given by the angle θ of X-axis with respect to the X-axis, and ω is the angular speed. The dynamic equations for the motion of the WMR are given by,

\[ m \ddot{x} = -\eta_1 \dot{x} + (F_R + F_L) \cos \theta \]
\[ J \ddot{\theta} = -\phi_1 \dot{\theta} + (F_R - F_L)l \cos \theta \]

where \( m \) is the mass of the WMR, \( J \) is its moment of inertia, \( F_L \) and \( F_R \) are the forces generated by the left and right wheels respectively, \( \phi_1 \) and \( \eta_1 \) are the coefficients of rotational and viscous frictions respectively and \( l \) denotes the moment arm of the forces where the center of geometry and center of mass in the WMR are assumed to coincide. Differentiating (1) and (2), yields

\[ \dot{x} = \dot{u} \cos \theta - u \sin \theta \dot{\theta} \]
\[ \dot{y} = \dot{u} \sin \theta + u \cos \theta \dot{\theta} \]

Substituting \( \dot{u} \) from (4) and \( \omega \) from (5) gives a complete model of the vehicle in the global coordinate frame.

\[ m \ddot{x} = -\eta_1 \dot{x} - \dot{y} \dot{\theta} + (F_R + F_L) \cos \theta \]
\[ m \ddot{y} = -\eta_1 \dot{y} - \dot{x} \dot{\theta} + (F_R + F_L) \sin \theta \]
\[ J \ddot{\theta} = -\phi_1 \dot{\theta} + (F_R - F_L)l \cos \theta \]

Since we do not have access to the force inputs, we would like to express the model so that the voltage applied to the left and right wheels appears as control inputs. We know that the force, \( F \) generated by a DC motor connected to a wheel with radius \( r \) when supplied with a voltage of \( V \) is

\[ F = KV - \eta_2 \nu \]

where, \( K = \frac{K_b \nu}{r} \) and \( \eta_2 = \frac{K_b \nu}{r} \), \( K_b \) is motor torque constant, \( K_b \) is the back emf constant, \( R \) is the armature resistance and \( \nu \) is the linear speed of the wheel. Now, the forces generated by left and right wheels when supplied with voltages \( V_L \) and \( V_L \) are,

\[ F_L = KV_L - \eta_2 \nu_L \]
\[ F_R = KV_R - \eta_2 \nu_R \]

where \( \nu_L \) and \( \nu_R \) are the linear speed of the left and right wheel respectively. Combining (8), (11), we obtain,

\[ m \ddot{x} = -\eta_1 \dot{x} - \dot{y} \dot{\theta} - \eta_2 (\nu_R + \nu_L) \cos \theta + K(V_R + V_L) \cos \theta \]

Define \( \eta_1 + 2 \eta_2 = \eta \), and since, \( \frac{(\nu_L + \nu_R)}{2} \cos \theta = \dot{z} \)

\[ m \ddot{x} = -\eta \dot{x} - \dot{y} \dot{\theta} + K(V_R + V_L) \cos \theta \]

Figure 1, are given by,

\[ \dot{x} = u \cos \theta \]
\[ \dot{y} = u \sin \theta \]
\[ \dot{\theta} = \omega \]
Combining (10), (11) and (12) we obtain, (17), (18) while the leader has simple dynamics of the form, 

\[ u_i = k_1 \alpha_i + k_2 a_i \quad i = 1, 2, \ldots, n - 1 \]  

(21)

where, \( k_1 \) and \( k_2 \) are the gains introduced to avoid saturating the control voltages. \( \alpha_i \) in (21) causes the velocity vectors of all agents to align in the same direction with the same magnitude. \( a_i \), which is a vector in the negative gradient of an artificial potential function \( U_i \), contributes to collision avoidance and cohesion in the group. The acceleration inputs for agent \( i \) are defined as:

\[
\begin{bmatrix}
    u_{ix} \\
    u_{iy}
\end{bmatrix} = 
\begin{bmatrix}
    -k_1 \sum_{j \in N_i} (v_{ix} - v_{jx}) - k_2 \sum_{j \in N_i} \nabla_{r_{ij}} U_{ij} \\
    -k_1 \sum_{j \in N_i} (v_{iy} - v_{jy}) - k_2 \sum_{j \in N_i} \nabla_{r_{ij}} U_{ij}
\end{bmatrix}
\]

where \( U_{ij} \) is an artificial potential function and is a function of the distance \( ||r_{ij}|| \) between agent \( i \) and its neighbor \( j \). \( N_i \) is a set which contains all the neighbors of the agent \( i \). Neighbors of an agent \( i \) are those agents with whom it can communicate. The potential \( U_{ij} \) is a differentiable, nonnegative, radially unbounded function of the distance \( r_{ij} \) between agents \( i \) and \( j \), such that [1]

1) \( U_{ij}(r_{ij}) \rightarrow \infty \) as \( ||r_{ij}|| \rightarrow 0 \)

2) \( U_{ij} \) attains its unique minimum when agents \( i \) and \( j \) have the desired separation.

A typical potential function is shown in Figure 2. We choose the following potential function candidate,

\[
U_{ij} = \frac{1}{(||r_{ij}||/d)^2} \log \left( \frac{||r_{ij}||}{d} \right)^2
\]

(22)

where, \( ||r_{ij}|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \). Differentiating \( U_{ij} \) with respect to \( x_i \) and \( y_i \), we obtain

\[
\nabla_{r_{ix}} U_{ij} = \frac{\partial U_{ij}}{\partial x_i} = \frac{-2(x_i - x_j)d^2}{((x_i - x_j)^2 + (y_i - y_j)^2)^2} \]

\[
\nabla_{r_{iy}} U_{ij} = \frac{\partial U_{ij}}{\partial y_i} = \frac{-2(y_i - y_j)d^2}{((x_i - x_j)^2 + (y_i - y_j)^2)^2}
\]

Next, we feedback-linearize the vehicle model (16)-(18) to transfer it to the double integrator form so that we can substitute the acceleration inputs. Let us choose the following two outputs,

\[
y_1 = x + L \cos \theta \quad \text{(23)}
\]

\[
y_2 = y + L \sin \theta \quad \text{(24)}
\]
Differentiating again and substituting for $\ddot{x}$, $\ddot{y}$ and $\ddot{\theta}$, we obtain

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} = A \begin{bmatrix} V_R \\ V_L \end{bmatrix} + B$$

where,

$$A = \begin{bmatrix} \frac{K}{J} \cos \theta - \frac{L}{J} K r \sin \theta & \frac{K}{J} \cos \theta + \frac{L}{J} K r \sin \theta \\ \frac{K}{J} \sin \theta + \frac{L}{J} K r \cos \theta & \frac{K}{J} \sin \theta - \frac{L}{J} K r \cos \theta \end{bmatrix}$$

$$B = \begin{bmatrix} -\frac{u}{m} \dddot{x} - \dddot{y} \dot{\theta} - L \cos \theta \dddot{\theta}^2 + \frac{L \eta}{m} \sin \theta \dot{\theta} \\ -\frac{u}{m} \dddot{y} + \dddot{x} \dot{\theta} - L \sin \theta \dddot{\theta}^2 - \frac{L \eta}{m} \cos \theta \dot{\theta} \end{bmatrix}.$$ 

Note that $A$ is invertible. Hence, we can design

$$\begin{bmatrix} V_R \\ V_L \end{bmatrix} = A^{-1} \left( -B + \begin{bmatrix} u_x \\ u_y \end{bmatrix} \right)$$

### IV. Simulation and Experiment

We implemented the coordination scheme for two WMRs, Lobot (Figure 3) and Pino (Figure 4), and one virtual leader. The physical parameters of the WMRs are listed in Table I, where $m$ is in $Kg$, $J$ is in $Kg m^2$, $K$ is in $N m sec^{-1}$, $\eta$ is in $Kg m^2$, $\phi$ is in $N m sec^{-1}$, and $r$ is in $m$. We define the control laws in continuous-time, however, we implemented it in a discrete-time fashion. We implemented the controller in LabView™ and because of the computation delay we could manage to sample only as fast as 200msec. So, at the interval of 200msec, Pino and Lobot update their position, and velocity using odometry and send it to the respective neighbor. The virtual leader is simulated in LabView™ and it also sends its position and velocity to Pino and Lobot at the same time interval. Communication between the agents is done using UDP across the WLAN. In each agent, a separate thread of receiver is running continuously and it updates the received data every time new data arrives. Control calculation is done using the latest information in receiver.

The initial condition for Pino is: surge speed 0.05 m/sec, angular speed 0.0rad/sec and orientation 35°. The initial condition for Lobot is: surge speed 0.11 m/sec, angular speed 0.0 rad/sec, and orientation 55°. The virtual leader is simulated to have surge speed 0.113m/sec, angular velocity 0.0 rad/sec and orientation 45°. At the start of the experiment, the separation between Pino and Lobot ($d_{L,P}$) is 1.59m, between Pino and the Leader ($d_{L,P}$) is 1.125, and between Lobot and the Leader ($d_{L,L}$) is 1.125. The desired separation between each of them is 1m. After implementing the coordination algorithm, the surge velocities of Pino and Lobot converge to 0.113 m/sec, orientation of them converge to 45°, and separation between all of them converges to the desired separation of 1m as shown in Figures (5, 7, 9, 11).

Simulation results for the same initial condition and sampling time are presented along with the experimental results (Figures 6, 8, 10, 12). Comparing the simulation and the experimental results, we can observe that the transient behavior of the surge velocity (Figure 5) in experiment is different than that of the simulation (Figure 6). This is because of the uncertainties involved in the modelling of the WMR and inaccurate system parameters. The varying nature of transmission and computation delay may also have contributed in the differences between the experimental and the simulation behavior. In the transient of the experiment, velocities of both, Pino and Lobot, decreases to almost zero. In this situation, if we had not used the virtual leader, both of them would have stopped and it would have looked

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
& m & J & K & \eta & \phi & r \\
\hline
Pino & 25 & 1.1094 & 3.1054 & 178.37 & 7.5817 & 0.203 \\
Lobot & 27 & 1.4415 & 9.5001 & 225 & 12.04 & 0.23 \\
\hline
\end{array}
\]

**Table I**

**Physical Parameters of WMRs**
like they converged to a zero group velocity. The virtual leader provided a reference group objective and even in the presence of uncertainties, directed the group to flock.

V. CONCLUSIONS

In this paper we presented an experimental implementation of flocking in wheeled mobile robots. We implemented control laws presented in [1] for the double integrator model in WMRs. We employed feedback linearization with look ahead to represent the WMR model in normal form and computed the control inputs for coordination. We introduced a virtual leader to steer the flock in the desired direction. Simulation and experimental results were presented. We observed a slight discrepancy between the simulation and experimental results, which we attribute to the uncertainties in the modelling of the robots and inconsistent computation.
time and network delays.

REFERENCES


