Of_The_Wild( ) : A Robotic Wolf Pack

William Campbell Martin

Adviser: Amy Briggs

Senior Thesis in Computer Science

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Arts Department of Computer Science Middlebury College Middlebury, Vermont

May 2011
ABSTRACT

Wolves are natural hunters. Their ability to hunt in a pack presents an effective model for controlling the motion of multiple robots. In particular, this is highly useful in the field of swarm robotics, where numerous agents act together to produce a collective behavior. This thesis investigates and implements a physical wolf pack in order to create emergent swarm hunting behaviors. These are especially useful for pursuit and evasion tasks in the field of robotics.
You, the people, have the power, the power to create machines, the power to create happiness! You, the people, have the power to make this life free and beautiful, to make this life a wonderful adventure. Then in the name of democracy, let us use that power. Let us all unite. Let us fight for a new world, a decent world that will give men a chance to work, that will give youth a future and old age a security.

Charlie Chaplin
ACKNOWLEDGEMENTS

I would like to thank my advisor Amy for being extremely helpful and kind. I must also thank my family for their support in my studies and the Senior Research Project Supplement fund, which has helped this project financially. Lastly, I am grateful for my friends, who have been nice enough to humor me while I rattled on about building robotic wolves. The wolf pack lives on!
TABLE OF CONTENTS

1 Introduction 1

2 Swarm Intelligence 4
   2.1 Description ................................................. 4
   2.2 Decentralized Decisions .................................... 5
      2.2.1 Foraging .................................................... 5
      2.2.2 Exploring .................................................. 7
      2.2.3 Choosing a Nest ........................................... 8
   2.3 Swarm Movement ............................................. 9
      2.3.1 Grouping .................................................... 9
      2.3.2 Direction ................................................... 11

3 Wolf Packs 12
   3.1 Behavior ..................................................... 12
      3.1.1 Hierarchical ............................................... 12
      3.1.2 Structureless .............................................. 13
   3.2 Modeling ..................................................... 16

4 Robotic Platform 20
   4.1 Goals ......................................................... 20
   4.2 Communication ............................................... 20
   4.3 Motors ......................................................... 22
   4.4 Circuit Design ............................................... 23
      4.4.1 Power Regulation .......................................... 23
      4.4.2 H-bridge ................................................... 24
      4.4.3 Microcontroller ............................................ 25
      4.4.4 Sensing With Photodiodes ................................. 27
      4.4.5 Use of LEDs ............................................... 27
      4.4.6 Loading Programs .......................................... 29

5 Conclusion 31

Bibliography 33
LIST OF FIGURES

1.1 Comparison of wolf body size to that of prey [17]................. 2
2.1 Percentage of ants passing on two equal bridges over time [11]... 7
2.2 Mormon cricket with radio transmitter [18]......................... 10
3.1 Pack formation with alpha and beta wolves [29].................... 13
3.2 Wolf behavioral transition probabilities [20]....................... 15
3.3 Finite state acceptor for wolf behavior [20]....................... 15
3.4 Set of $n$ wolves and prey $P$.................................. 16
3.5 Wolf rules to move towards prey (a) and repel from neighbors (b). 17
4.1 A completed wolf bot............................................. 21
4.2 Rubber wheel (left) and pager motor (right)....................... 23
4.3 Power regulation circuitry......................................... 24
4.4 (a) H-bridge circuit [10] and (b) L293D IC........................ 25
4.5 Pinout of ATmega168 microcontroller [1].......................... 26
4.6 Circuit for communication and sensing ambient light............. 28
4.7 LEDs used for communication (a) and state indication (b)........ 29
4.8 Arduino used as an ISP flashing a bot............................ 30
CHAPTER 1
INTRODUCTION

Wolves are considered to be some of the fiercest predators on earth. It is no coincidence that they exist in the majority of northern ecosystems; a generalized hunting behavior allows them to feed on an extremely diverse selection of prey. They are able to accomplish this feat through pack hunting and basic behavioral heuristics that can be adapted to a variety of scenarios. Perhaps their most impressive trait is a reliance on animals significantly larger than themselves (figure 1.1) as a primary food source. Prey such as moose or bison not only typically weigh five to eight times that of wolves, but are covered in thick hide and often run faster [17]. This peculiar dependence gives a possible evolutionary explanation for why wolves developed pack-based hunting tactics. For these reasons wolf hunting behavior demonstrates a highly successful model for the motion of multiple actors working towards a common goal.

One area where multiple body systems is highly relevant is the field of swarm robotics. In particular it is concerned with the coordination and overall behavior of large numbers of robots. Although swarm research has traditionally drawn inspiration from the social behavior of insects [25], any biological system that relies on the actions of many similar creatures can be used. Natural systems have the advantage of evolutionary refinements that can help coordinate complex interactions between robots and their environment. The ultimate goal is to derive synchronized system-level behavior from numerous actors that individually cannot handle most tasks. Furthermore, the robots are designed to be extremely simple, have limited ability to interact, and lack any central coordination [26].

Wolf packs present an interesting base for swarm robotics research because of their unique ability to capture targets that are both larger and faster than
an individual wolf. This appeal has inspired past work by Weitzenfeld [29] and Madden [20], whom both investigate the use of wolf pack hunting behavior as a model for multi-robot systems. These projects present two thorough approaches to creating a behavioral model for pack hunting. Despite differences in theory, both research groups primarily implement these models using simulation software, such as JavaBots or MissionLab. Additionally, Weitzenfeld attempts a small scale (three wolf robots) hardware implementation using Sony AIBO robotic dog platforms. These models are able to successfully control virtual wolf packs and mimic situations observed in the wild.

Aside from the novelty of using wolves as inspiration, there are a number of other reasons why new multi-robot swarm models are useful to investigate. One of the most desirable goals is to find methods and techniques that can reduce the necessary amount of communication and planning for robots. Since swarm robotics focuses on animals with either very few or very simple communication processes, it can provide both insight and innovation for other robotic applications that depend
heavily on communication or planning. In addition, a project based on something like a wolf pack presents problems and solutions that are geared towards pursuit and evasion tasks. Applying new models to such a problem uncovers both the limitations of different approaches and improvements that can be made to solve the task at hand.

This thesis provides an investigation into swarm robotics and describes how a model inspired by wolf pack hunting can be implemented. The focus of this is on building a physical hardware platform from discrete components and integrated circuits. In addition, firmware is designed to demonstrate wolf pack behavior and hunting tactics. In all cases, the design decisions aim to be as cost-effective and simple as possible in order to make the robots easy to build and program. This implementation will provide an analysis of how wolves can be used to solve pursuit and evasion scenarios for multi-robot systems.
CHAPTER 2

SWARM INTELLIGENCE

2.1 Description

Although it may not always be obvious, swarms permeate both our media and everyday lives. It seems impossible to be outside for any extended period of time without encountering a noisy flock of birds or a dedicated ant colony. If we turn towards the ocean we can find countless schools of fish navigating around blooms of jellyfish. Given how common animal groupings are in our natural world, it is easy to see why Hollywood commonly uses them in films to display destruction and horror. From Alfred Hitchcock’s 1963 movie *The Birds* to the animated Pixar feature *Finding Nemo*, swarms are presented as massive calculating armies of physically identical beasts. Similarly, the Biblical Plagues of Egypt are frequently used in plots to create hordes of insects wielding unimaginable destructive power. These rather dramatic portrayals of swarms tend to focus on showing a chaotic nature arising from large numbers of creatures. While this notion may serve well to induce fear, it does not give justice to the order that arises from a collective behavior. The ability of a decentralized, self-organized system of actors to tackle significant tasks is what defines the intelligence of a swarm.

“Swarm intelligence” is a term introduced by Beni, Hackwood, and Wang [5, 6, 7, 8] to describe cellular robotic systems in one- or two-dimensional environments. These systems utilize agents which have the ability to self-organize and develop patterns due to interactions with near neighbors. However, this definition of the term is somewhat limiting because it does not allow for more advanced swarm actions involving complex actors. Bonabeau, Dorigo, and Theraulaz [9] extend it to include “any attempt to design algorithms or distributed problem-solving de-
vices inspired by the collective behavior of social insect colonies and other animal societies.” Using this definition, swarm intelligence is able to encompass advanced collective behaviors, such as picking an optimal nesting area or consistently directing thousands of bees in flight. The next few sections will focus on how swarm intelligence arises in nature and present a biological foundation for its seemingly impossible coordination and control.

2.2 Decentralized Decisions

Honey bees in a hive can number in the thousands while ant colonies can easily support millions of individuals. These extreme sizes of many swarms make it seem impossible that a consensus could ever be reached on any important decision. Yet like clockwork these masses of animals make daily decisions that are not only quickly executed but consistently near optimal. Surely there must be some conductor insect instructing these countless drones in the correct choices. Or perhaps they act “blindly using the highest mathematics by divine guidance and command” [2]. However, there is a much more satisfying answer to this conundrum that can be explained mathematically. In order to present it, we will look at a series of examples of intelligent swarm behavior and show how collective decisions arise.

2.2.1 Foraging

A classic case of swarm decision making involves determining the optimal food source for a colony of ants. When weighing different sources, an ant must consider both the quality of the food and its distance from the nest. Despite lacking prior knowledge of an area or its sources of sustenance, foraging ants are able to
continually find the best possible paths to nearby food. The key to this uncanny ability lies in the recruitment techniques that ants use to organize foraging. These techniques are found throughout many species of social insects and are defined as any behavior that is responsible for increasing the number of members at a particular location [12]. This can be incredibly useful for concentrating foragers along shortest paths in expansive areas with patchy food sources.

Deneubourg et al. [11] studied the recruitment abilities of foraging Argentine ants by experimenting with different possible path choices. This methodology is commonly referred to as the double bridge experiment because it presents the ants with the option of just two different paths to a food source. The ratio between the lengths of these two bridges is then varied in order to see how it affects the ants’ decisions. As the ants traverse their environment they lay down a pheromone, which acts as a recruiting mechanism. This chemical attracts other ants to its trail, making them more likely to follow a particular path. Thus, a positive feedback loop is created as more ants follow a distinct path and deposit their own pheromone. In addition, the pheromone evaporates over time so weak trails fade away.

In the first test, bridge lengths were kept equal and the ants were allowed to freely walk from the nest to the food. Although initially the ants chose to take a random bridge, as time elapsed all of the ants converged to a single bridge despite the identical lengths (figure 2.1). This can be explained by random fluctuations where occasionally more ants will follow one bridge than another. Eventually the positive feedback will cause this bridge to be more favored due to increased pheromone trails. When the ants were exposed to bridges of different lengths the researchers witnessed similar phenomena. However, rather than favoring an arbitrary bridge, the ants consistently chose the shorter of the two. Initial bridge choice was random but the ants that chose the shorter path were the first to reach
the food source and thus the first to turn back. This resulted in higher pheromone density on the shorter path, so it was more probable that future ants would follow it. This process explains how ants are able to develop efficient paths from their nest to nearby food sources.

2.2.2 Exploring

There is a careful balance that must be maintained for an ant colony between exploration and exploitation. Under stable conditions, the foragers should ideally always follow the shortest path to the best food. If this source can support an infinite number of foragers then this is exactly what is observed [24], but in practice there is more to consider. Chiefly, a colony of foragers must also deal with steadily dwindling food sources. Their pheromone recruitment mechanism must somehow allow for new sources to be discovered and depleted ones forgotten. Luckily, there is never a 100 percent chance that an ant will pick up a trail, even if it is highly saturated and marked as the optimal foraging path. This leads to a so-called "strategy of errors" [13] because there is always some chance that a forager ant

Figure 2.1: Percentage of ants passing on two equal bridges over time [11].
CHAPTER 2. SWARM INTELLIGENCE

will go off the trail, making it an impromptu explorer. The explorer has the chance of discovering new and perhaps better food sources for when the current food source is depleted. The number of active explorers at any time is then related to the current strength of trails. If no nearby food source has been discovered (and thus there are no strong trails) then there will be a large number of scouts. On the other hand, with several strong pheromone paths there will be few scouts because there is a low probability of losing a trail.

2.2.3 Choosing a Nest

An interesting decision problem that swarms must frequently handle is picking a new nest to inhabit when the current nest becomes too small or is destroyed. The ant *Temnothorax albipennis* has an interesting way of tackling this problem that only needs to ask about half of the ants which nest site is optimal. Although half of the ants do not directly compare potential nests, they nevertheless contribute to the colony’s decision. Mallon et al. [21, 23] detail how they are able to make this collective decision without the use of pheromone trails.

*T. albipennis* utilizes a technique called tandem running in which one ant can bring another to a specific location by acting as a guide. In addition, the species practices social carrying, which results in one ant simply picking up a fellow ant and taking it to a new nest. The nest picking process typically begins with just scouts comparing the quality of potential locations. Tandem running is useful here because it allows a high quality nest to quickly be investigated by several scout ants. Also, tandem running occurs somewhat slowly, so it allows for higher quality sites to still be found. The greater the quality of the nest site, the more likely a scout will attempt to initiate tandem running. Overtime if more scouts agree with the first scout’s decision, the number of tandem runnings will increase dramatically.
and consequently so will the number of ants at the high quality site. Once a certain population threshold, referred to as a quorum, is reached at the new nest, tandem runnings will cease and be replaced with social carrying. At this point, the new nest is filled very rapidly along with any eggs or larvae from the old nest.

2.3 Swarm Movement

A flock of birds or swarm of bees presents an aesthetically pleasing marvel to the human eye. The motion is both chaotic and ordered, simple yet visually complex. If one were to only view a small section of the entire group, it would be hard to spot any patterns or coordination. Yet as a whole, a swarm manages to maintain a consistent velocity and direction without any sudden stops or hiccups. This is the mystery behind swarm movement. Whether we are considering a handful of seagulls or 10,000 locusts, the individual agents manage to aggregate and flow to the same place. In order to learn more about how this motion occurs, we will look at two biological examples.

2.3.1 Grouping

The Mormon cricket, *Anabrus simplex*, is a mass-migrating insect that forms large migratory bands similar to that of the locust. Beekman et al. [4] discuss the Mormon cricket’s unique group movement and how it arises. These migratory bands can span areas of over ten kilometers in length and manage to travel about two kilometers per day. They are extremely damaging to crops and as a result typically require immediate pesticides to combat. Since the insect is rather large (about 7 centimeters in length), scientists are able to tag them with tiny radio transmitters (Figure 2.2) in order to study their movement.
Sword et al. [28] utilized radiotelemetry in order to study the movement of Mormon crickets both inside and outside of a migratory band. The study originally aimed to solely observe their collective movement. Unexpectedly, it also yielded information about what benefits and pressures lead to massive migratory bands. When the crickets were inside a band, they were significantly less likely to be killed by predators. In contrast, subjects that were taken out of a band suffered high mortality rates within just two days. In this same period of time, none of the banded crickets were killed. Thus, the researchers concluded that these bands form as a technique to avoid predation. Despite this protection, there is one major disadvantage to being a member of the band. Mormon crickets are cannibalistic, so members that stop or fall behind have a high probability of being eaten by their swarm-mates. The rapid movement of the entire band over large distances can be described as a forced march driven by cannibalism [27]. Since the bands still manage to form a cohesive group, the risk of outside predation must exceed that of being eaten by a neighbor.
2.3.2 Direction

Once a honeybee swarm has picked a new nest location, the hive must somehow navigate its thousands of members in unison to this destination. This is made even more difficult by the fact that only about five percent of the total population actually knows where the new nest is located [4]. Martin Lindauer [16] was first to observe that some bees in a moving swarm fly at high speeds through the group in the direction of travel. He posited that these fast moving bees were in fact the informed scout bees, directing the rest of the swarm to the new nest. In order to confirm this hypothesis, Beekman, Fathke, and Seeley [3] studied swarms of bees by taking several photographs against a light sky. This resulted in small streaks on each bee that could be measured to determine their individual velocities. They were able to verify that not only did a portion of the swarm fly faster than the rest, but that these bees were concentrated in the upper half of the swarm.
CHAPTER 3
WOLF PACKS

3.1 Behavior

It is important to understand the behavior of a wolf pack in order to investigate how it can be adapted to swarm robotics. Although wolves have numerous behaviors, the ones we are particularly interested in for swarm robotics are those involved with hunting prey. This includes not only the collective motion of the pack but how it transitions from one behavioral state to another. We will look at how these behaviors have been interpreted by other research groups concerned with robotic wolf packs and with what success these have been implemented.

3.1.1 Hierarchical

A classical interpretation of wolf pack structure is the hierarchy of alpha and beta wolves. Researchers who subscribe to this train of thought see the alpha wolves as the most experienced pack members, leading a group of beta wolves during a hunt [30]. This social structure allows a group of wolves to coordinate and control a hunt. Weitzenfeld, Vallessa, and Flores [29] studied how this model could be to applied robotic applications. Their pack model used the following assumptions:

1. The pack consists of a leader (alpha) and one or more followers (betas).
2. Beta wolves aggregate around the alpha wolf but keep a certain distance.
3. Wolves are only allowed visual information. No communication is used.
4. A single field of vision is used. Anything outside this is unknown to a wolf.
5. Each wolf’s head is pointed in the direction of motion.
6. Walking speeds are constant for all wolves.
CHAPTER 3. WOLF PACKS

Figure 3.1: Pack formation with alpha and beta wolves [29].

Using these six rules, the researchers created a wolf pack model consisting of alpha and beta wolves (figure 3.1). Each wolf was allowed to transition between behavioral states of eating, attacking, stalking, and wandering. Simulations of this model were first performed using the JavaBot simulator and using the NSL/ASL schema language. From here, the team moved on to test a physical implementation using Sony AIBO ERS-210 four-legged dog robots. Although the pack only consisted of three robots, they were able to successfully follow their model of alpha and beta wolf structure. However, it is difficult to say whether or not this implementation acts according to the behavior of an actual wolf pack.

3.1.2 Structureless

Another interpretation of wolf pack behavior is an unstructured model that lacks hierarchy or social structure between the wolves. Although the roles of alpha and beta wolves may exist, they change frequently during a hunt and are not based on a preexisting dominance hierarchy. Instead, each wolf simply follows basic heuristics
and rules of thumb during a hunt. The collective behavior of many wolves following these rules in close proximity results in a highly successful hunt. This behavior is similar to many examples of swarm intelligence, so it is easy enough to use it for swarm robotics. Madden, Arkin, and MacNulty [20] investigated how well this type of a model would fare for robotic applications. They made two major changes to Weitzenfeld’s model:

1. No tight structure is maintained between wolves.

2. No specific roles, such as alpha or beta, are assigned to the wolves.

The unstructured robotic model is based on five different general states that the agents transition between. These states are labeled search, approach, attack group, attack individual, and capture. A typical hunt breaks down as follows. After navigating around an environment in the search state, the pack eventually encounters a herd of prey. The wolves will then approach the herd at a moderate speed, causing the prey to flee. This will initiate the attack group phase where the wolves will chase the group until weak or unfit prey fall behind. These are chased individually by the wolves until they are either captured or manage to outrun the predator.

Transitions between behavioral states were handled using probabilities observed in the wild (figure 3.2). Although in nature the decision to move between these states is determined based on the current environment, it was controlled by a weighted random function in this model. Valid transitions were encoded in a finite state acceptor (figure 3.3) that only allowed certain state progressions. The group simulated this model using MissionLab software for several scenarios. They were able to successfully mimic wolf behavior that is observed in the wild.
### Figure 3.2: Wolf behavioral transition probabilities [20].

<table>
<thead>
<tr>
<th>Preceding State</th>
<th>Search</th>
<th>Approach</th>
<th>Watch</th>
<th>Attack Group</th>
<th>Attack Individual</th>
<th>Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>.00</td>
<td>.68</td>
<td>.00</td>
<td>.31</td>
<td>.01</td>
<td>.00</td>
</tr>
<tr>
<td>Approach</td>
<td>.09</td>
<td>.00</td>
<td>.12</td>
<td>.69</td>
<td>.09</td>
<td>.01</td>
</tr>
<tr>
<td>Watch</td>
<td>.32</td>
<td>.35</td>
<td>.00</td>
<td>.27</td>
<td>.06</td>
<td>.00</td>
</tr>
<tr>
<td>Attack Group</td>
<td>.24</td>
<td>.09</td>
<td>.03</td>
<td>.13</td>
<td>.51</td>
<td>.00</td>
</tr>
<tr>
<td>Attack Individual</td>
<td>.16</td>
<td>.06</td>
<td>.02</td>
<td>.16</td>
<td>.08</td>
<td>.52</td>
</tr>
</tbody>
</table>

### Figure 3.3: Finite state acceptor for wolf behavior [20].
3.2 Modeling

One aspect of the wolf hunt that is particularly appealing to swarm robotics is the circular pattern formation that occurs when wolves are encircling their prey. Muro, Escobedo, and Spector [22] present a simulated mathematical model that offers one possible solution for emergent circular patterns in a wolf pack. Starting with the MacNulty ethogram [19] used in the unstructured model, Muro et al. distinguish between pursuit and encircling behaviors. Pursuit behavioral states include approach, watch, attack group, and attack individual. Encircling behavioral states are attack individual and harass. Pursuit and encircling can then be considered emergent behaviors because they arise from the combination of simple rules followed by each of the individual wolves. Furthermore, they occur completely decentralized without any communication or coordination between the wolves.

We can describe a robotic wolf model with a prey $P$ and set of $n$ wolves $W = \{w_1, w_2, \ldots, w_n\}$ as shown in figure 3.4. Similar to the unstructured pack model, we assume homogeneous agents have identical roles. In addition, there will be no direct wolf-to-wolf communication. All information supplied will be from a wolf’s local information of the environment, the prey, and near neighbors. We then introduce two rules that each wolf in the pack will follow (figure 3.5).
Figure 3.5: Wolf rules to move towards prey (a) and repel from neighbors (b).

1. Move towards the prey $P$ until you are at a critical distance, $d_c$.

2. If distance to $P$ is less than or equal to $d_c$, repel from nearby wolves.

In order to create a mathematical formulation, we begin with the coordinates of an individual actor $i$.

$$\vec{p}(t) = (x_p(t), y_p(t))$$
$$\vec{u}_i(t) = (x_i(t), y_i(t))$$

$$i = 1, \ldots, n$$

We can then describe the forces between all of the different entities as

$$\vec{F}_{i,j} = \begin{cases} 
\text{force of } P \text{ on } w_i : & i = j \\
\text{force of } w_j \text{ on } w_i : & i \neq j 
\end{cases}$$

$$\vec{F}_{i,j} = f_{i,j}^x \vec{e}_1 + f_{i,j}^y \vec{e}_2$$

$(\vec{e}_1, \vec{e}_2)$ is an orthonormal basis

The individual force components can be further written as

$$f_{i,j}^x = \sigma_{i,j} s_{i,j} \text{sign}(x_i - x_j) \cos \alpha_{i,j}$$
$$f_{i,j}^y = \sigma_{i,j} s_{i,j} \text{sign}(y_i - y_j) \sin \alpha_{i,j}$$
If \( i \neq j \) then

\[
\vec{F}_{i,j} = \begin{cases} \\
\left| \tan^{-1} \frac{y_i - y_j}{x_i - x_j} \right| : & x_i \neq x_j \\
\pi/2 : & x_i = x_j 
\end{cases}
\]  

(3.5)

If \( i = j \) then

\[
\vec{F}_{i,j} = \begin{cases} \\
\left| \tan^{-1} \frac{y_i - y_p}{x_i - x_p} \right| : & x_i \neq x_p \\
\pi/2 : & x_i = x_p 
\end{cases}
\]  

(3.6)

We use \( \sigma_{i,j} \) to denote the modulus of the force, \( ||\vec{F}_{i,j}|| \), and \( S_{i,j} \) to describe if it is an attractive or repulsive force. Finally, we can sum all of the forces on any one actor to figure out the resulting force that acts upon it.

\[
\vec{F}_i = \vec{F}_{i,1} + \vec{F}_{i,2} + \ldots + \vec{F}_{i,i} + \ldots + \vec{F}_{i,n} = \sum_{j=1}^{n} \vec{F}_{i,j}
\]  

(3.7)

If we would like to figure out how this will evolve over time we can pick some appropriate initial conditions and solve the following system of ordinary differential equations.

\[
\frac{d\vec{u}_i(t)}{dt} = \beta_w \sum_{j=1}^{n} \vec{F}_{i,j}
\]  

\[
\frac{d\vec{p}_i(t)}{dt} = \beta_p \sum_{i=1}^{n} \vec{F}_{i,i}
\]  

(3.8)

where \( b_w \) and \( b_p \) are bounds used to control the prey and wolf velocities. They can be expressed mathematically as

\[
\beta_w = \min \left( 1, v_w \left\| \sum_{j=1}^{n} \vec{F}_{i,j} \right\| \right) \sum_{j=1}^{n} \vec{F}_{i,j}
\]

\[
\beta_p = \min \left( 1, v_p \left\| \sum_{i=1}^{n} \vec{F}_{i,i} \right\| \right) \sum_{j=1}^{n} \vec{F}_{i,i}
\]  

(3.9)
Using this mathematical formulation of the encircling model, we can consider several different cases of both stationary and moving prey with an arbitrary number of wolves. In all of these cases, the result will be an evenly spaced circle formed by the final positions of the wolves around the prey. The amount of time required for the system to settle into this stable state depends on the velocity of the prey and the number of wolves used to encircle.
4.1 Goals

The robotic platform I settled on for this project is strongly based on the Formica swarm robot project [14]. Throughout this chapter you will find inspiration from this and other swarm robotics projects that have created successful robots. The primary focus for this particular project was on small and inexpensive parts that would be easy to assemble by hand. This wolf bot platform (shown in figure 4.1) aimed to incorporate the minimum number of sensors and actuators needed to model wolf pack behavior. Not only does this reduce the cost of the entire platform, but it minimizes complexity and decreases build time.

4.2 Communication

Communication between the wolves is used as a form of vision for measuring distances and relaying general near neighbor information between the pack. Physically, it is implemented using infrared (IR) communication with photodiodes and emitters. Although this type of communication has a limited range, it is inexpensive and allows for the photodiodes to also be used as ambient light sensors. It does introduce possible issues if there is a lot of reflection with the surrounding environment, so dark materials must surround the test chamber to minimize this noise.

The IR communication protocol used was adapted from the Formica project and is based on multiple frequency-shift keying (MFSK). Frequency shift keying transfers digital information through discrete frequency changes in a wave [15]. Utilizing multiple frequencies, MFSK switches between symbols when a change in
CHAPTER 4. ROBOTIC PLATFORM

Figure 4.1: A completed wolf bot.
frequency is detected. This is used in a rotating fashion to move between symbols; these transitions are then interpreted as binary data. Each packet transmitted includes a delimiter, length, data block, and checksum. Despite this organization, it is still difficult to prevent all packet loss but when it is fully operational, data can be transferred at a few hundred bytes per second.

4.3 Motors

Motors can be quite expensive depending on their size, speed, and torque. Manufacturing tiny gear boxes is rather difficult so using one will significantly increase costs further. In order to meet the small size requirement, the wolf bots are equipped with extremely small pager motors (figure 4.2). Although these offer very little torque, they are extremely cheap (less than a dollar) and have low current requirements. This makes them ideal for tiny robots and relatively painless to replace if one breaks. The motors are secured to the base of the wolves using an adhesive so that they can be attached in a variety of ways.

Finding a small enough wheel for a pager motor was a bit of a challenge. Although some swarm robotic projects have resorted to cutting the wheels out of rubber sheets, I chose to cut my wheels from a 12mm diameter rubber rod. Long lengths of this can be bought at a low price and the resulting wheels are nice circles. In order to make straight cuts on this thick piece of rubber, I utilized a tubing cutter intended for cutting lengths of hose in a computer cooling system. A hole could then be drilled into the center of the resulting disc with a Dremel tool.
4.4 Circuit Design

The circuit was designed and laid out using the EAGLE Layout Editor by CadSoft. This free software offered a lot of power and flexibility when adding components to the circuit. In addition, there are readily available component libraries for many parts due to its wide use. The physical circuit was constructed by hand with a regular soldering iron on small prototyping boards. All wiring was done with either thin enamel-coated magnet wire or high-gauge hookup wire.

4.4.1 Power Regulation

The system utilizes a LM7805 linear voltage regulator to maintain a constant 5 volt supply for the microcontroller, motors, and LEDs. The current requirement for the entire robot was less than 500mA so a regulator was chosen that could supply up to 1A. Large (4.7µF) tantalum capacitors were used to smooth input voltage from the battery. Non-polarized, smaller (0.1µF) ceramic disc capacitors
were placed on the output line in order to filter out any high frequency noise that might occur. This was of particular concern because the same power source was used for both motors and the microcontroller. Since motors are inductive devices that often produce back emf and electrical noise on their power lines, there is some risk of causing instability for other circuits sharing power. Fortunately, the motors used for these robots were extremely small pager motors that only drew around 200mA when stalled. The power for the entire device was toggled by a SPDT mechanical switch. Although these switches were easy to work with, they were somewhat easy to break with an accidental short circuit.

4.4.2 H-bridge

Although some swarm robotics projects, such as Formica, have chosen to forgo using an h-bridge IC, I decided to include one due to its power and simplicity. Building an h-bridge from scratch is simple enough and may have brought down the overall cost, but it would have required soldering quite a few individual diodes by hand. Since this platform is intended to be soldered by hand without the use of a reflow station, it was important to integrate ICs when possible to simplify
design. The package chosen for this platform was the L293D four channel driver (figure 4.4b), which can control two motors separately with currents up to 1.2A.

An h-bridge allows for bi-directional, variable speed control over a DC motor. The theory behind h-bridge control is straight forward; as shown in figure 4.4a, current is allowed to flow into the motor depending on which of the four transistors are switched on. By controlling these transistors with different logic values, the motor can spin in either direction and even brake with a shunt. The clamp diodes are also essential to prevent back emf from destroying the transistors. The motor speed can then be controlled via a pulse-width modulation (PWM) signal fed in from a microcontroller. PWM presents a highly efficient way to vary the voltage across a motor by rapidly switching a power source on and off. The average amount of time that the power is on is directly proportional to the voltage that the motor sees.

4.4.3 Microcontroller

The microcontroller selected for this design was an Atmel AVR ATmega168 (figure 4.5). This 8-bit device contains 16kB of flash memory for storing programs and 1kB of SRAM. It is a promising candidate for swarm robotics platforms because it
offers 23 programmable I/O lines in a 28-pin plastic dual in-line (PDIP) package. Although it is possible to buy this chip in a thin quad flat pack (TQFP) package for reflow soldering, the PDIP version was selected so that it could be attached to the circuit with an ordinary soldering iron. The chip operated at the clock speed of 1MHz (instead of the typical 16MHz with a crystal oscillator) to maintain a low power consumption and simplify the circuitry. Since the ATmega168 natively supports pulse-width modulation, it was very easy to provide input to the h-bridge and control the motors. In addition, the package provides six channels of 10-bit analog-to-digital conversion, which is useful for measuring input from the photodiodes.
4.4.4 Sensing With Photodiodes

Both distance sensing and communication is handled by the use of photodiodes and two different bias resistors (figure 4.6). The bias resistors are controlled by two microcontroller connections, BIAS1 and BIAS2. These are used to switch between a 10kΩ resistor for low-sensitivity ambient light detection and a 100kΩ resistor for high-sensitivity communication.

The 100kΩ resistor acts with a capacitor as a high-pass filter, which removes any DC photocurrent generated by ambient light. The resulting high frequency communication signal is then superimposed on half of the supply voltage, generated via a low-current voltage divider. The signal at RX must then be fed into a operational amplifier to compare it with half of the supply voltage. A microcontroller timer measures rising edges from the comparator output, so the frequency of an incoming signal can be measured.

Detecting ambient light levels is somewhat easier because we can directly compare the voltages at PD1, PD2, and PD3 using the 10-bit ADC channels available on the ATmega168. Since these photodiodes are only sensitive to the infrared range of light waves, this navigation circuitry will only operate near good infrared sources, such as the 100 watt incandescent light bulb used in this project. The distance from a constant infrared source can then be measured to provide both location and direction information.

4.4.5 Use of LEDs

Transmission of signals between wolves was handled with high power infrared emitters (SFH4244) that provided 940nm light. These were used because they cover a wide angle of 120 degrees, so only three are needed to cover all directions around a robot. Switching time between on and off states is just 11 ns, an excellent value
for producing very fast signals. However, there are a few complications with using this particular LED on the robot. First, they consume a decent amount of current, 35mA each for this project, so a field-effect transistor (FET) was used to toggle each one (figure 4.7a). This was done to take strain off the current requirements for the microprocessor’s I/O lines. In addition, these LEDs pose a mechanical problem because they are only offered as surface mount devices (SMDs). This makes them somewhat difficult to solder wires to since they are intended for reflow soldering. Nevertheless, this was managed using 26-gauge wire and a steady hand.

There were four additional LEDs used on each robot as status lights (figure 4.7b). These consisted of cheap 3mm red and green LEDs that could be run at less than 20mA. Since this current requirement is quite small, it could be handled easily by the microcontroller, so they were directly connected to output pins. These four LEDs were used to relay information about which hunting state the wolves were in and which actions they were attempting.
4.4.6 Loading Programs

An Arduino prototyping platform was used as an in-system programmer (ISP) to flash each of the wolf bots with the appropriate firmware (figure 4.8). This was an effective method to program the microcontrollers, but it was somewhat time consuming to connect the several required wires to each robot. Further work should experiment with faster flashing methods, such as with IR communication or other wireless protocol. In particular, it would be beneficial if swarm members could program each other with the most current firmware.

Figure 4.7: LEDs used for communication (a) and state indication (b).
Figure 4.8: Arduino used as an ISP flashing a bot.
CHAPTER 5

CONCLUSION

Over the course of this project we have explored how a wolf pack can be implemented with swarm robotics. As we have seen with many of the currently proposed models, it is possible to describe a system of wolves in several different ways depending on what assumptions are made about wolf hunting behavior. The most successful models appear to be those based on an unstructured wolf pack without any sort of hierarchy. For this reason, I chose to implement my version of the wolf pack based on work done by the groups led by Madden and Muro. Furthermore, these models allowed me to make intelligent decisions as to what sensors and electrical components to include in the physical implementation.

Our physical implementation of a robotic wolf pack introduces not only a possible approach to wolf pack robotics, but yet another swarm robotics platform that can be built. While it is clearly based on previous and existing robotic platforms, it offers an extremely cheap option that can even be built by hand. Although all of the wolves were built one at a time with a soldering iron, I would not recommend this for future work with this robot. It is quite time consuming and becomes quite complex as more and more wires are added. Ideally, I would have the circuit board fabricated from my EAGLE designs so that it would be much quicker to build and more organized. This would also allow the use of surface mount devices, which are much smaller than their through-hole counterparts used in this implementation. I did not choose to go this route for this project because of the lengthy turn around time on inexpensive fabrication options.

The firmware for each robot is not too complex and there are several great examples freely available on the web, such as the Formica project. However, actually flashing the robots with new code takes some time as mentioned earlier, so I rec-
ommend investigating quicker methods to program the robots. The ATmega168 is an excellent and cost-effective chip for swarm robots, but it does lack an internal operational amplifier. To cut down on the number of IC packages, it may be useful to migrate future wolf bots to a different line of microcontrollers with this feature.

The particular sensors and actuators used on this robotic platform were appropriate for the tasks being considered, but more advanced swarm behavior could be achieved with a larger suite of features. Higher accuracy positional and directional information could be gathered with advanced sensors, such as ultrasonic transducers or magnetometers. Obviously, this would significantly increase the cost of a single robot, but it may allow for faster and more effective simulations.

In conclusion, the wolf pack robot presented here is a practical option for investigating wolf swarm behavior and emergent patterns. At first glance wolves may not seem like members of a swarm, but their collective behavior indicates that they do in fact share many things in common with an intelligent swarm. Future work will look at how the platform implemented during this project can be used for different robotic applications, swarm or otherwise. Unique multi-robot control schemes can help provide us with both advanced motion planning and methods for cutting down on the required amount of communication between robots.
BIBLIOGRAPHY


