

Biologically Inspired Multi-Robot System based on Wolf Hunting Behavior

Zachary Hinnen
Computer Science and Eng
University of South Florida
Tampa, FL, USA
zhinnen@usf.edu

Chance J. Hamilton
Computer Science and Eng
University of South Florida
Tampa, FL, USA
chamilton4@usf.edu

Alfredo Weitzenfeld
Computer Science and Eng
University of South Florida
Tampa, FL, USA
aweitzenfeld@usf.edu

Abstract— Studies involving the group predator behavior of wolves have inspired multiple robotic architectures to mimic these biological behaviors in their designs and research. In this work, we aim to use robotic systems to mimic wolf packs' single and group behavior. This work aims to extend the original research by Weitzenfeld et al [7] and evaluate under a new multi-robot robot system architecture. The multiple robot architecture includes a 'Prey' pursued by a wolf pack consisting of an 'Alpha' and 'Beta' robotic group. The Alpha Wolf will be the group leader, searching and tracking the 'Prey.' At the same time, the multiple Beta 'Wolves' will follow behind the Alpha, tracking and maintaining a set distance in the formation. The robotic systems used are multiple raspberry pi-robots designed in the USF bio-robotics lab that use a combination of color cameras and distance sensors to assist the Beta 'Wolves' in keeping a set distance between the Alpha "Wolf" and themselves. Several experiments were performed in simulation, using Webots, and with physical robots. An analysis was done comparing the performance of the physical robot in the real world to the virtual robot in the simulated environment.

Index Terms—Wolf Packs, Computer Vision, Webots

I. INTRODUCTION

Many animals dwell in groups developing cooperative behaviors that can improve their survival [1]. There exists extensive research on animal collaboration and cooperation. For example, Ranjbar et al. [8] explore the coordination of social insects like honey bees and ants to mimic foraging and pheromone signaling in robotics. A similar work by Novitzky et al. [9] explores the behaviors of honey bees and the ability of the bee to communicate through dances and actions.

Animal collaboration, such as wolf pack hunting, can result in more effective preying as a group than individually [1]. Animal collaboration can take many forms, from multiple animals working together having similar individual behaviors to those showing complementary behaviors, performing distinct individual roles [2]. Several hunters working together through collaboration and coordination gain the ability to take down larger prey where individuals would fail. This paper analyzes wolf pack hunting behavior, where individualistic behavior demonstrates linear and simplistic movements. In contrast, small packs show geometric size formations, such as polygons, with more behavioral differentiation between members, resulting in more complex group dynamics [3].

Our work uses a combination of lab-designed robots and an open-source robot simulator, Webots, to develop and test robot behaviors and formations. This work is inspired by Weitzenfeld et al. [7].

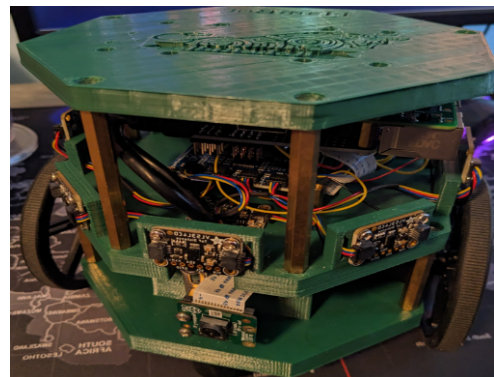


Figure 1: USF Biorobotics Lab "Hambot" that was used as a basis for physical robot development and tests. The alpha wolf robots are in red while the beta wolf robots are in green. Robot scale is 1ft height with 1ft diameter.

II. RELATED WORK

Multi-robot systems have been developed for a variety of environments, including air, land and underwater [4]. Multi-robot systems extend the idea of swarm intelligence, where, for example, Liu and Liu [6], define specifications for small-sized robots using the high-energy realization of an energy-efficient hardware architecture that supports complex formations in a distributed robotic system. The system's neural network learns the behaviors of other agents within the process and maps the state of the system, providing pose information on all agents with respect to the center agent within the formation [6]. In Pierson et al. [10] authors describe work on controlling a group of dog-inspired robots to direct a herd of uncooperative sheep-like robots to a target goal. The objects in the experiment can be biological or robotic; however, the object must be capable of responding to dog-like robots. The work by DeCastro et al [5], describes a robotic controller team that can perform high-level assignments for capturing actuation and locomotion to match the behavior needed for reactive and sensor events. The overarching goal for the work is to ensure that motion planning is collision-free between the multiple robots and dynamic obstacles.

III. MODEL

The wolf pack hunting model described in the paper extends prior work by Weitzenfeld et al [7]. The multi-robot system architecture consists of four robots representing one “alpha” wolf (the leader) and three “beta” wolves (the followers). The alpha wolf leads the hunt for prey, while the beta wolves follow the alpha while maintaining a hunting formation. The model is based on a small pack organization, having the benefit of providing greater precision and control of the hunting formation, when compared to collective hunting packs of greater size that can result in more individualistic behavior and less cooperation and cohesion [1].



Figure 2: Hunting in smaller packs allows for greater cohesion and collaboration.

The collective hunting method proposed in this work is based on the following basic assumptions:

1. Wolf hunting groups are composed of a group leader (alpha wolf robot) and at least one follower (beta wolf robot).
2. Beta wolf robots form around the alpha, trying to keep a constant angle and distance between themselves and the alpha wolf robot.
3. Every wolf robot uses only visual information to evaluate its environment, determining positions, angles, velocities, and behaviors.
4. Movement speeds are established as a baseline and fluctuate based on the alpha's or target's position in relation to the beta wolf robot.
5. The tests are completed in a controlled environment with a minimal number of obstacles, designed to facilitate low noise conditions.

A. Physical Robot Systems

The physical robot implementations are based on Raspberry Pis [11], as shown in Figure 1. The robots include actuators and sensors, such as distance sensors, IMU's, wheel motors, and cameras, comparable to those included in the Webots implementation. The robot includes visual perception libraries with functionality for perceiving information relative to other robots or objects.

B. Webots Simulation

Webots [12] is an open-source robot simulator that allows the design of complex environments on custom robots. Webots includes built-in object and color recognition. Once recognized, object's size and position can be easily obtained, as

illustrated in Figure 3.



Figure 3: Webots has built-in object recognition that provides relative position, relative orientation, size, position on the image, size on the image, and color.

Webots allows for the incorporation of multiple robots and objects of different shapes and sizes, and different layouts that can quickly be adjusted, as illustrated in Figure 4.

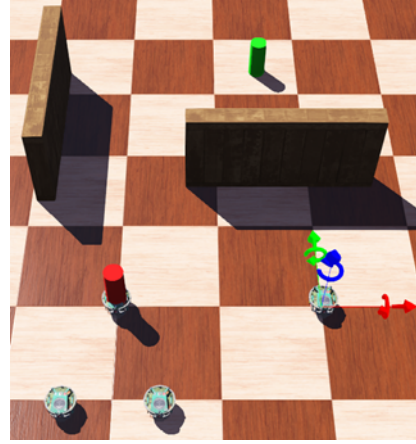


Figure 4: This is an example of an environment with adjustable obstacles and robot positions.

C. Image Processing

The robot image processing system performs object and color recognition to identify the target "Prey", “Alpha” and “Beta” wolf robots, as described in Figure 5. The image processing system receives the object contours to compute distances between the robot and the object. Using these shapes, the system returns X and Y coordinate values along with the bounding box H (height) and W (width) to provide an approximate object size. The angular values keep the beta wolf robots in formation behind the alpha wolf robot.

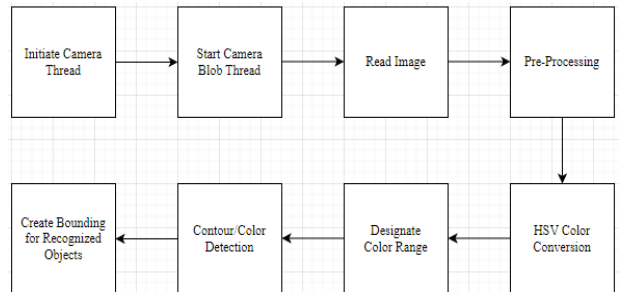


Figure 5: The block diagram illustrates camera image processing. The blob

thread reads an image and converts it from RGB to HSV. Objects are recognized by color and contour. Small colored objects or other items too far away are removed by the perception system.

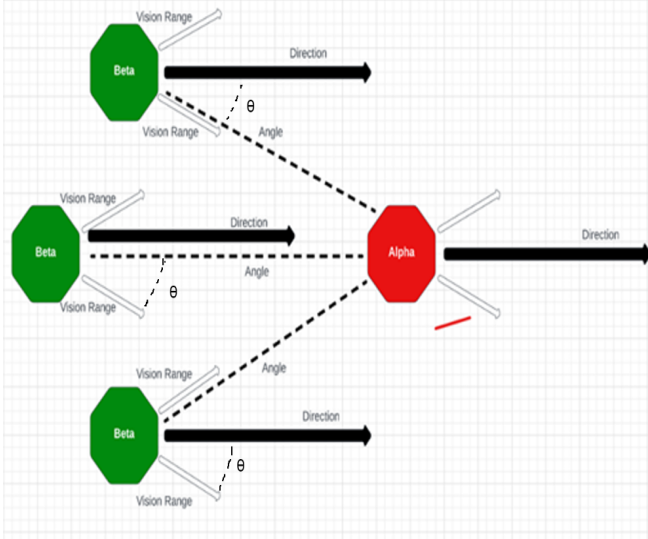


Figure 6: The beta wolf robot attempts to keep a constant distance between themselves and the alpha robot using the X angle values by adjusting their velocities and orientations. The illustration is based on the original work by Weitzenfeld et al [7].

Beta wolves try to maintain a constant distance from the alpha wolf, while forming a pack centered around the alpha wolf, as shown in Figure 6. Movement direction is represented by black arrows, while arrows in white, labeled vision range, represent the visual range with a horizontal FOV (Field of View) of 62.2 degrees and vertical FOV of 48.8 degrees, as shown in Figure 7.

Beta wolves follow the alpha's movement and adjust their velocity in response to the change in direction, creating a fixed target position angle (the vision-range median). Eq. 1 and 2 describe the relative positioning of beta wolf in relation to the alpha wolf. The alpha robot's position, represented by (r), is derived from the current alpha angle (θ) subtracted from the initial alpha angle (θ_0). The relative area size (α) of the alpha robot (see Equation 2) is calculated based on the perceived height (h) multiplied by the perceived width (w).

$$r = \theta - \theta_0 \quad (1)$$

$$\alpha = h * w \quad (2)$$

An initial velocity of 0.25 inch per sec was given to the physical robots, with the velocity constantly adjusted (see equation 3). An initial velocity is given so we have a base robot speed; 0.25 was chosen so the robots can perform controlled tests at a slower speed. An adjustment value (*adj*) is set to 0, and the new position (r), calculated in equation 1, is converted into an abs value and divided by 1000 (see equation 4).

$$v_0 = 0.25, v = v_0 \pm adj \quad (3)$$

$$adj = (adj - abs(r))/1000 \quad (4)$$

A division of 1000 is applied for a gradual velocity adjustment to avoid jerky transitions.

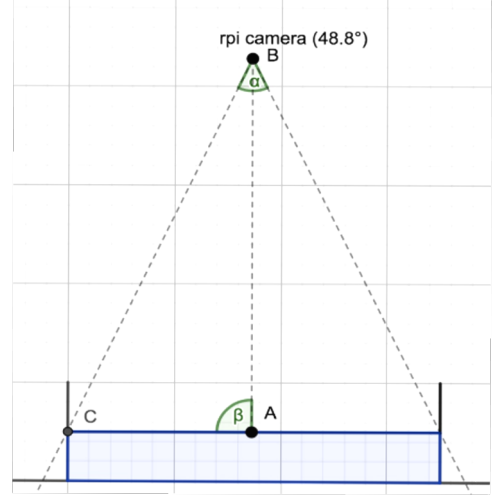


Figure 7: The figure illustrates the physical robot's FOV.

D. Behaviors

Alpha Wolf Robot. The alpha wolf behavior is determined by three states: Wander, Pursuit, Catch, as illustrated in Figure 8. Corresponding pseudo code is shown in Figure 9.

- Wander:** This state is initiated if the alpha robot has no view of the prey. The alpha robot will maneuver around the environment performing simple obstacle avoidance until the prey comes back into view. Once that occurs, the alpha robot will transition into a pursuit behavior.
- Pursuit:** In this state, the alpha robot makes a pursuit towards the prey, continually adjusting its movement relative to the prey's current position. During the pursuit mode, if the alpha robot loses sight of the prey, then the alpha robot will enter wandering.
- Catch:** This state results from the alpha robot being within a set range from the prey. Once this happens, the alpha will complete its behavior, unless the prey starts moving.

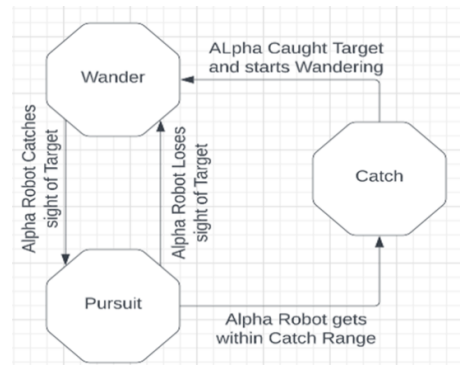


Figure 8: The alpha wolf performs state transitions based on the stimulus from its environments meeting the conditions for a state change. The state changes vary between Wander, Pursuit, and Catch.


```

Alpha ← CallAlphaClass;
while True do
    Update Blob Key-points;
    if TargetAngle is 0 then
        Alpha.Wander;
    Else
        Alpha.Pursuit;
        if TargetSize > MaxTarget
            Alpha.Catch;
        end
    end
end
end

```

Figure 9: The pseudo code describes the alpha robot state transitions in response to different visual perceptions.

Beta Wolf Robot: The beta robot has four states: Wander, Pursuit, Formation, and Catch, as illustrated in Figure 10. Corresponding pseudo code is shown in Figure 11.

- Wander:** This state is initiated if the beta robot does not have a visual on the alpha robot or the prey. This will lead to wandering behavior. If the prey is found first, the beta robot will transition into pursuit mode. The beta will transition into formation mode if the alpha robot is found first.
- Pursuit:** In this state, the beta moves toward the prey, continually adjusting its movement based on its current position. During pursuit mode, if the beta robot loses sight of the prey, the beta robot will enter the Wander behavior. If the beta robot perceives the alpha robot, it will continue pursuing the Alpha robot until it is within a set range and transitioning into formation mode.
- Formation:** This state is responsible for making the beta robot keep a constant distance and angle from the alpha robot and a safe distance from the other beta robots. The angle is determined by the initial starting position of the beta robot in proximity to the alpha. Using the initial angle (X coordinate), the beta robot stays at that angular position independent of directional movement. If a beta robot were to start in wandering mode, the initial angle would be determined when the beta robot reached a set distance from the alpha robot. If the beta robots were to lose sight of the alpha robot, they would move into Wander mode. If the beta robot perceives the prey, the beta robot would only initiate catch mode if the prey is within a set distance.
- Catch:** This state results from the beta robot being within a set range of the prey. Once this happens, the beta robot will maintain its current position until the prey begins to move again.

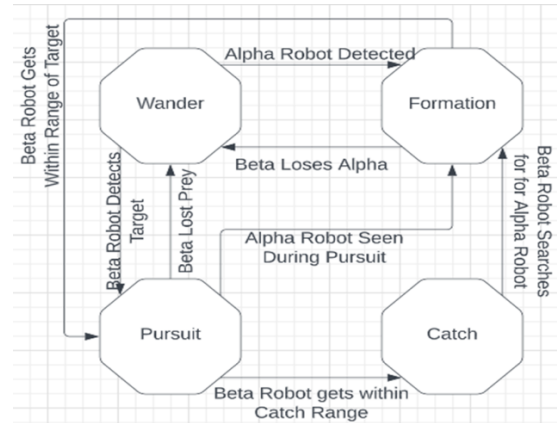


Figure 10: The Beta wolf performs state transitions based on the stimulus from its environments meeting the conditions for a state change. The state changes vary between Wander, Pursuit, Catch, and Formation.

```

beta ← CallBetaClass;
status ← false;
while True do
    Update Blob Key-points;
    if AlphaAngle is 0 then
        if PreyAngle is greater than 0 then
            beta.Stalk;
        else
            beta.Wander;
        end
    else
        if status is False then
            Update Blob Key-points;
            beta.BaseValues;
            status ← True;
        end
        beta.Formation;
    end
end
end

```

Figure 11: The pseudo code describes the behavioral algorithm for the beta robot and for transitioning between states.

IV. EXPERIMENTS AND RESULTS

The robot experiments had success in the ability to transition between states and perform the assigned behavioral task. The tests were conducted in three major events:

- One beta robot followed behind the alpha robot pursuing the target. The beta robot and alpha robot are given initial positions, as shown in Figure 12 (left).
- Three beta robots followed behind one alpha robot, pursuing the target. The three beta robots had an initial position behind the alpha robot, as shown in Figure 12 (right).
- One beta robot followed behind the alpha robot, pursuing the prey (in blue). In contrast to the previous two test cases, two beta robots have a random starting position, not directly behind the alpha robot.

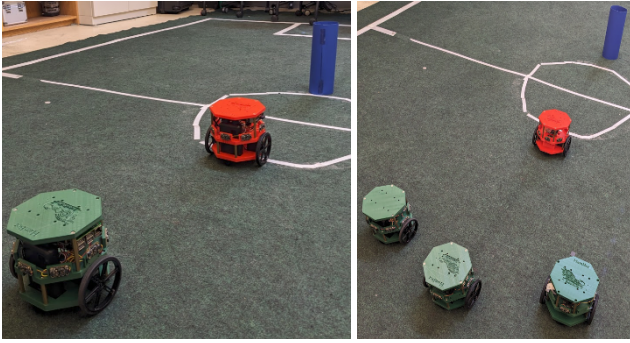


Figure 12: (Left) The figure shows the initial setup for one alpha robot and one beta robot. (Right) The figure shows the initial setup with three beta robots and one alpha robot.

The overall task goal is for the beta robots to remain in formation while following the alpha robot to the prey, while minimizing formation deviation. If there is deviation in the formation, it should only be when the beta robot perceives the prey and is within a range of three feet.

A. Single Alpha and Single Beta Wolf (Preset Starting Formation)

The single alpha and single beta robots test were consistent in its ability for the alpha and beta robots to transition between states and achieve the expected conditions. The beta robot maintained a constant angle and distance from the alpha robot regardless of motion. The beta robot can stay in place and rotate if the alpha robot is moving horizontally, only moving until the alpha robot gets outside of the accepted size range. Once the set size range is exceeded, the robot applies the proper velocity computations to get back within range of the alpha robot. During implementation, color object recognition caused multiple blobs to be detected for the red alpha robot instead of a singular blob. To fix this issue, a large pink cylindrical object was placed on the robot to allow for more precise blob detection. An image of this experiment is shown in Figure 13.

B. Single Alpha and Multiple Beta (Preset Starting Formation)

The single alpha and multiple beta robot test performed similarly to the single alpha and single beta robot test. The beta robots could stay in formation and make proper transitions between pursuing the prey, following the alpha robot, and wandering around the environment, attempting to find the alpha robot or the prey. The color recognition issue of detecting multiple-colored objects belonging to the same robot was fixed to detect only the largest object within a cluster. Since there was a system-imposed range on how large an object must be to get detected, this solved the issue of too many objects being detected.

There were additional problems with very sudden state transitions. There would be times when the beta robot would detect the alpha robot and the prey simultaneously. Depending on the size of the prey object, the beta robot would sometimes transition to pursuit if the size of the alpha robot was too small.

There were also issues with obstacle avoidance and returning to the initial formation from a pursuit stage. This issue resulted when beta robots would aimlessly wander. Overall, it was a minor concern, and the robots stayed in formation, and were able to avoid collisions with each other. Images of this experiment are shown in Figure 14 and 15.



Figure 13: (Left) The figure shows the red cylinder object on top of the alpha robot in its initial pose. (Right) The figure shows the same setup but with the target and alpha robot two feet away from the initial position.

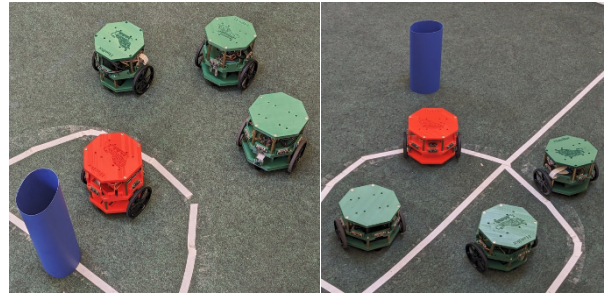


Figure 14: (Left) The figure shows the alpha in its initial pose with the three beta robots. (Right) The figure shows the robots remaining in formation.

C. Single Alpha and Multiple Beta (Random Starting Formation)

The single alpha and multiple beta robot test were successful in finding the prey, as well as having the beta robot find and follow behind the alpha robot. A major issue resulted from the beta robot fixating onto the prey despite perceiving the alpha robot. This happened whenever the beta robot obtained image values of size greater than the alpha robot. There were also issues with the beta robots getting lost and taking more than 1 minute (4 wandering cycles) to encounter either the alpha robot or the prey. More adjustments need to be made to the wandering behavior to better account for object finding.

D. Webots Implementation vs Real World

The Webots implementation of the previous tasks performed similarly with physical robots in relation to task completion. In general, the physical robots were slower to run due to hardware limitations including image processing for color recognition. Figure 16 summarizes the results for the single alpha and beta physical robot experiments.

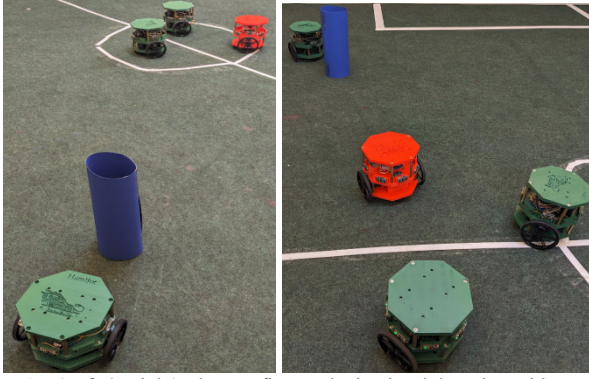


Figure 15: (Left & Right) The two figures depict the alpha robot with two beta robots attempting to find the prey. The alpha robot finds the prey, and the two beta robots follow behind it. One of the beta robots managed to find the prey before finding the alpha robot and stayed in that catch position.

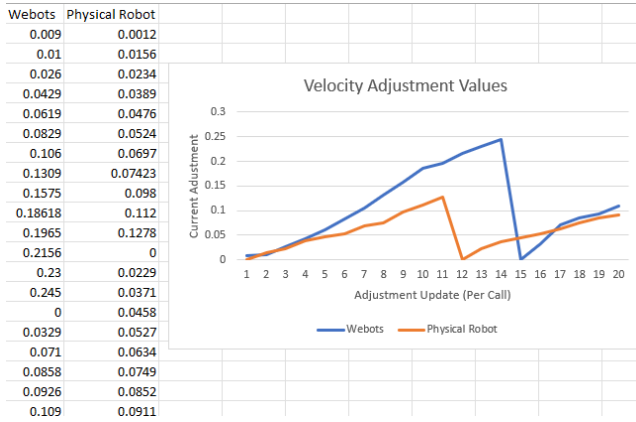


Figure 16: The table shows the adjustment values for Webots and the physical robots. Webots can run at a much faster speed with smoother velocity adjustments. The physical robot moves slower, however the performance in terms of adjusting velocity is comparable between the two environments.

V. DISCUSSION

The current work serves as a baseline model maintaining robot formation, by adjusting beta robot velocities while tracking a single alpha robot. We tested the model with Webots and with real-world physical robots. Implementing the code within Webots demonstrated smoother velocity adjustments and better object detection with no noise disruption during the search for the alpha robot and the prey. While not as smooth as the Webots implementation, the physical robots could still perform the experimental tasks and overcome environmental noise. Future work will include updating the computer vision system from a color recognition model to an object recognition model. Color recognition, while quick, detects only major colors corresponding to either the alpha robot (in red) or the prey (in blue). The current work can inspire future work on robot swarms and mimicry of other animals, including birds, fish, or insects.

VI. ACKNOWLEDGEMENTS

This work was funded in part by NSF IIS Robust Intelligence research collaboration Grant No. #1703225 at the

University of South Florida and Grant No. #1703440 at the University of Arizona, entitled "Experimental and Robotics Investigations of Multi-Scale Spatial Memory Consolidation in Complex Environments."

VII. REFERENCES

- [1] Nowak MA. Five rules for the evolution of cooperation. *Science*. 2006 Dec 8;314(5805):1560-3. doi: 10.1126/science.1133755. PMID: 17158317; PMCID: PMC3279745.
- [2] Escobedo R, Muro C, Spector L, Coppinger RP. "Group size, individual role differentiation and effectiveness of cooperation in a homogeneous group of hunters." *J R Soc Interface*. 11(95), Apr 2, 2014.
- [3] J. D. Madden, R. C. Arkin, and D. R. MacNulty, "Multi-robot system based on model of wolf hunting behavior to emulate wolf and elk interactions," 2010 IEEE International Conference on Robotics and Biomimetics, Tianjin, China, 2010.
- [4] Z. Zhou, J. Liu, and J. Yu, "A Survey of Underwater Multi-Robot Systems," in *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 1-18, January 2022, doi: 10.1109/JAS.2021.1004269.
- [5] DeCastro, J.A., Alonso-Mora, J., Raman, V., Rus, D., Kress-Gazit, H. (2018). Collision-Free Reactive Mission and Motion Planning for Multi-robot Systems. In: Bicchi, A., Burgard, W. (eds) *Robotics Research*. Springer Proceedings in Advanced Robotics, vol 2. Springer, Cham.
- [6] Liu J.-C. and Liu, T.-T., "Multi-Robot Formation Control using Collective Behavior Model and Reinforcement Learning," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 2261-2265, doi: 10.1109/ISCAS48785.2022.9937572.
- [7] Weitzenfeld, A., Vallesa, A., Flores, H. "A Biologically Inspired Wolf Pack Multiple Robot Hunting Model," in *Latin American Robotics Symposium and Intelligent Robotic Meeting (LARS 2006)*, Santiago, Chile, 26-27 Oct. 2006.
- [8] Ranjbar-Sahraei B., Tuyls, K., Caliskanelli, I., Broeker, B., Claes, D., Alers, S., Weitss, G., "Bio-inspired multi-robot systems," *Biomimetic Technologies*, 31-Jul-2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780081002490000136>. [Accessed: 29-Mar-2023].
- [9] M. Novitzky, C. Pippin, T. R. Collins, T. R. Balch and M. E. West, "Bio-inspired multi-robot communication through behavior recognition," 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 2012, pp. 771-776, doi: 10.1109/ROBIO.2012.6491061.
- [10] A. Pierson and M. Schwager, "Bio-inspired non-cooperative multi-robot herding," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015, pp. 1843-1849, doi: 10.1109/ICRA.2015.7139438.
- [11] The Raspberry Pi Guide. [Online]. Available: <https://raspberrypi-guide.github.io/electronics/camera-positioning>. [Accessed: 30-Mar-2023].
- [12] Webots Guide. [Online]. Available: <https://www.cyberbotics.com/>. [30-Mar-2023].