Dancing Swarm of Robots

Team sdmay21-40

Abdalla Abdelrahman, Daniel Nikolic, Benjamin Schneider, Noah Thompson, Mason Walls, Cole Weitzel
Faculty Advisers: Dr. Akhilesh Tyagi and Dr. Diane Rover
Client: Dr. Akhilesh Tyagi

Problem Statement

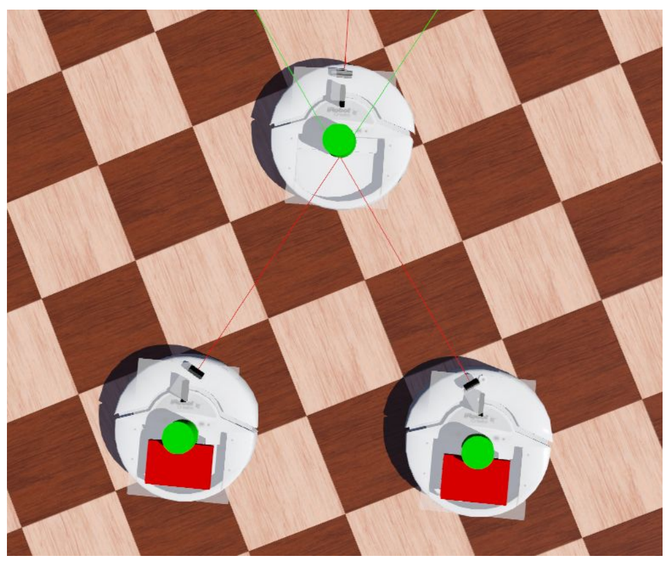
- Bird Swarms
  - Bird swarms are thought to move with a single leader that determines the direction of the flock. All other birds in the swarm maintain a local position relative to the bird in front of them, and the flock seamlessly maintains its shape as the birds fly.
- Robotics
  - Swarms like this have a multitude of applications in robotics due to their scalability and robustness.
- Project
  - This project seeks to model this localized swarm design in a 2-D robot swarm of ground-based robots designed around the principle of maintaining local position.

Design Requirements

- Functional Requirements
  - Robot swarm should model a 2-D triangular bird flock
  - Swarm consists of two followers and a leader
  - Only the leader may receive movement instructions, followers must determine moves from sensors alone
  - Followers must maintain 60 cm of separation and a 30° relative angle from the leader
  - 10% maximum follower position error margin
- Non-functional Requirements
  - Swarm system should be robust enough to hold its shape during complicated maneuvers
  - Leader shouldn't have unreasonable restrictions on how it may move
- Operating Environment
  - 40 m x 40 m virtual arena
  - No noise, signal interference, or obstacles
- Engineering Constraints
  - No lab access due to COVID restrictions
  - All project components must be simulated
  - $500 total project budget
- Standards
  - Project design process followed Agile-like structure
  - Version control through Github
  - IEEE 14764-2006 (Software Life Cycle Processes) Standard
  - IEEE 29119-2-2013 (Software Testing) Standard

Intended Users and Uses

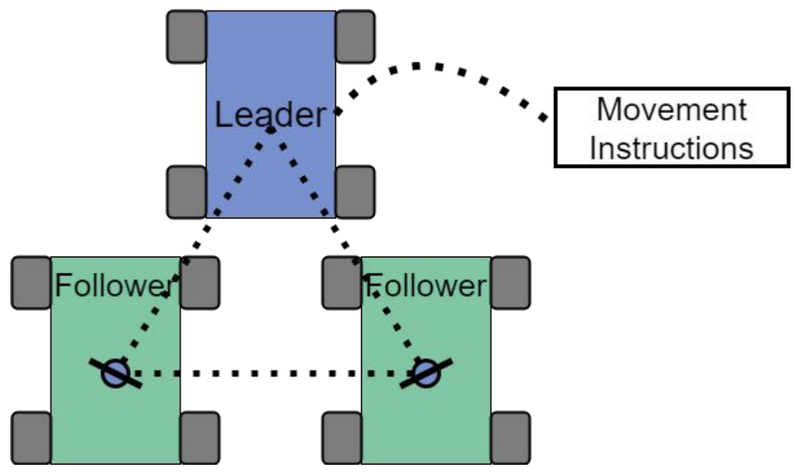
This project is intended to be used by ISU's CprE 288 professors as a possible addition to the Embedded Systems course. The results of this project may be incorporated into the course's lab in some capacity, either as a demonstration or student lab project.



WeBots Simulation Running

Design Approach

- System
  - Virtually simulated swarm using WeBots software suite due to COVID lab access restrictions
  - Robot models based off of CprE 288 CyBot platform with a cylindrical reflector in the center of the robot
  - All modelling, experimentation, testing, and final product to be done within WeBots virtual world model
- Swarm
  - A single lead robot will receive movement instructions and follow them to direct the swarm
  - Both followers use their distance sensor mounted on a servo motor to scan their surroundings for the leader's reflector and use this data to determine their movements



Conceptual Design Sketch

Technical Details

- Follower Movement Algorithm
  - Each follower has a 120° sensor sweep arc centered on 30°
  - Samples are made in 1° increments with movement decisions made every sample based on distance readings
- System
  - The swarm consists of three robot models with independent controllers written in C in WeBots
  - All robot geometric parts are modelled to CyBot spec
  - Distance sensors and servos are configured to CyBot spec
  - Each robot has a green cylindrical reflector added on
- Allowed Leader Moves
  - The leader is limited to movements similar to a bird in flight
  - Turning in place and reversing is not allowed

Testing

- Unit Testing Components
  - Distance sensor, servo, and speed control formula were fed sample data sets and their behaviors were verified
- System Integration Testing
  - Straight Line - Followers would lock on to the leader using the distance sensor and servo, the leader would move forward, and followers would keep pace using the speed control formula
  - Turning - The Straight Line Test was performed again, but with the leader turning as it moved while the followers would use the movement algorithm to steer in formation
  - Combined Movement - The Turning Test was performed for up to 30 minutes and the followers' positions were checked
- Results
  - All unit tests passed successfully
  - Followers maintain an error margin under 6% for normal operation and 8% when using illegal moves

WeBots Robot Model Class Diagram