Marco-Polo MPBots: Group 4 Final Report

Max Ernst
University of Michigan
maxernst@umich.edu

Andrew Greer
University of Michigan
alogreer@umich.edu

Dhillon Patel
University of Michigan
dhipatel@umich.edu

John Pye
University of Michigan
jepye@umich.edu

Abstract*—In this paper we extend the capabilities of two MBots [1] to enable environment-agnostic demonstration of pursuit and evasion behavior. The pursuer MPBot (Marco-Polo Bot), dubbed "Marco", has been outfitted with an advanced directional microphone array a high-resolution, wide-angle camera, and a speaker. The evader MPBot, dubbed "Polo", has been equipped with a four-direction April Tag [2] cube for target acquisition, a speaker, and a simple omnidirectional condenser microphone. In audio mode, these MPBots will thus possess the ability to track the origin of certain sound signals, and the ability to emit those signals. In visual mode, the April Tag visual fiducials on Polo will instead provide a pursuit target able to be acquired by the camera mounted on Marco. Using these new functionalities we build on the existing MBot autonomy architecture to develop autonomous pursuit-evasion behaviors, and implement these behaviors on the MPBots. We finally test the efficacy of our hardware and software upgrades through a series of competitive trials, demonstrating both audio-based and vision-based modes of pursuit. We believe this development project will lend itself both to applications in human-robot interaction as well as to search-and-rescue scenarios, in which reliable target positions may be unknown, and alternate methods of task acquisition may be required.

***Keywords*—** human-robot interaction, pursuit-evasion, search-and-rescue, audio navigation, visual navigation, directional microphone array, source finding, man-and-lion, autonomy

I. Introduction

The Pursuit-Evasion (PE) Problem has long attracted the interest of the robotics and autonomy field [3] [4] [5]. Such algorithms typically, but not always, assume perfect information - namely, that both predator and prey know the others location, and the map environment. More robust solutions have addressed sensing limitations present in the real world, whether it be a limited sensor range [6] or knowledge of only a bearing [7]. It is this variant of the classic PE problem which we propose to explore and implement.

The bearing-only PE problem is characterized by two main challenges. First; that neither pursuer nor evader are aware of the other's position. Second; that guaranteed target convergence is reliant on a continuous bearing. We propose to introduce two additional constraints: first, that impeding obstacles lie on the straight-line bearing from pursuer to evader; and second, that the bearing of the target may be discontinuous (caused by, for example, an occluding obstacle). These additional constraints will necessitate either the implementation of new pursuit logic, or of new control logic. We plan to pursue the latter strategy, and determine whether the guarantees of the family of pursuit logic presented in [7] hold under these non-ideal control circumstances. We then plan to explore more contemporary solutions which take obstacles in the arena into account in the behavioral planning stage.

Our project is essentially broken into three components, which will be pursued in parallel.

1) Obscured bearing pursuit behavior.
2) Auditory bearing determination, and visual contingency preparation.
3) High-speed path planning and control.

These efforts have produces two MPBots able to conduct our modified pursuit-evasion game effectively. The MPBots are able to, alternately, detect sound or emit sound, and give chase or attempt evasion. We have expanded the functionalities of the MBot platform, and have also upgraded their underlying control abilities [8]. Furthermore; we have developed a visual navigation contingency, using a MPBot-mounted wide-angle camera and April Tags [2] as an alternate method of calculating and supplying the relative bearing from pursuer to evader to Marco. This has eliminated a major source of risk stemming from our reliance on an experimental method of bearing generation, which is susceptible to ambient noise levels.

Our project concludes with a series of tests and demonstrations of the two robots in various arenas. These arenas will be enclosed (to prevent robot escape or human intrusion), and will be littered with obstacles to provide meaningful navigational challenges to Marco and Polo. Utilizing these obstacles, Marco and Polo will begin in line-of-sight of each other, to provide an initial bearing when in visual pursuit mode, or in any position and heading while in audio pursuit mode.

II. Related Work

Study of the PE problem in general began with [5] and further expanded upon in [9]. These texts outline the fundamentals of the problem; that is, a pursuit agent, an evasion agent, and the conflicting objectives to alternately converge to the target, or prevent the same. This problem was extended in [10] where bounded strategies for both "lion" (pursuer) and "man" (evader) were evolved. Since then, strategies for the PE problem and variants therein have been rapidly developed and iterated upon, with a survey of state-of-the-art techniques provided in [4]. [6] and [7] are examples of work done to solve the PE problem under less ideal circumstances, especially sensor degradation. It is here that we will turn for early prototypes of pursuit-evasion logic, to be refined (or changed) later as we are able to test effectiveness in a physical arena. Modern research [11] [12] in applying the PE problem to ocean surface confrontation has yielded both closed-form strategies based around potential fields, as well as machine learning-based strategies which are able to navigate even more complex environments. This work provides inspiration for our initial prototype PE behavior, as well as a possible avenue for extension work incorporation machine learning into our autonomy stack to enable for potentially superior performance in complex environments.

However, further literature review of existing standard pursuit-evasion strategies underlined their general incompatibility with our concept of operations. The cornerstone assumption of these strategies is a continuous bearing measurement - however, both in visual mode and audio mode, such a continuous measurement was impractical. In the visual case, simple occlusion or inopportune headings occasionally led the camera to lose track of the April Tag fiducial targets. In audio mode, the beam-forming microphone array would malfunction when presented with a constant signal, and fail to adjust to the changing heading of that signal. Therefore, while we drew upon the established theoretical methods of pursuit and evasion for

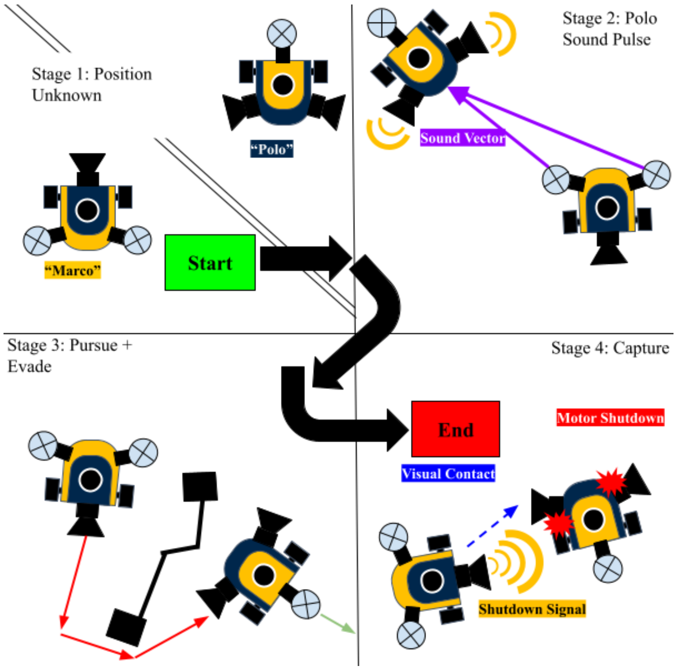

Fig. 1: Pursuit-Evasion Game Concept of Operations

guidance and inspiration, we ultimately reduced the complexity of our pursuit logic and instead focused on the development of a more robust state machine able to adapt to changing data streams and re-acquire the target in the case of a loss of target.

III. Methodology

Figure 1 displays the basic project concept of operations. At first, the positions of the MPBots are unknown to one another. After a sound pulse, or in the visual case, target acquisition, the pursuit-evasion game begins. The respective autonomy modules send commands to Marco and Polo, which are interpreted by the motion planner to ensure no environmental collisions, and to plan the intermediate states using the A* search algorithm. Finally, once Marco comes into range of Polo, a shutdown signal is broadcast, and both MPBots cease operation.

A. Physical Design and Assembly

The physical design of the MPBot differs substantially from the standard MBot model. An elevated platform has been added just above the LiDAR module on each robot to carry sound equipment: on Marco, the UMA-8 directional microphone, and on Polo, the speaker which broadcasts the locator signal. The raised location of these items helps to reduce echo, and minimizes occlusion of the LiDAR module. Above this sits the wide-angle camera on Marco, and the April Tag box on Polo. The height of this top platform was consciously chosen to sit above the lab-standard plastic wall sections, to eliminate any possibility of obstacle occlusion. This choice was made so that the visual mode of pursuit differed as little as possible from the audio mode of pursuit, which does not suffer from occlusion at all. Testing of each MPBot in this configuration demonstrated their stability up to their maximum speed (0.8 m/s).

To allow for remote shutdown, Polo is additionally equipped with an omnidirectional condenser microphone mounted flush to the Raspberry Pi computer, and Marco is similarly mounted with a speaker. When Marco comes into shutdown range of polo, as measured by the camera to the April Tags, it will broadcast a shutdown signal at 4 kHz. Polo, performing a thresholded Fast Fourier Transform on all signals received through its omnidirectional microphone, recognizes

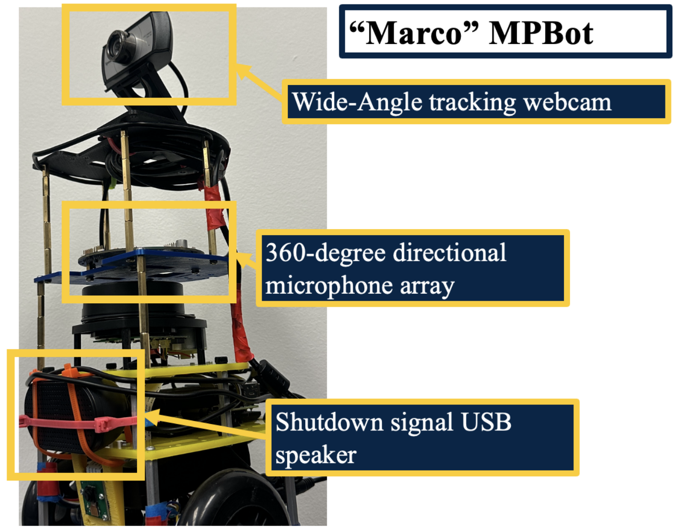

Fig. 2: Assembled Marco MPBot

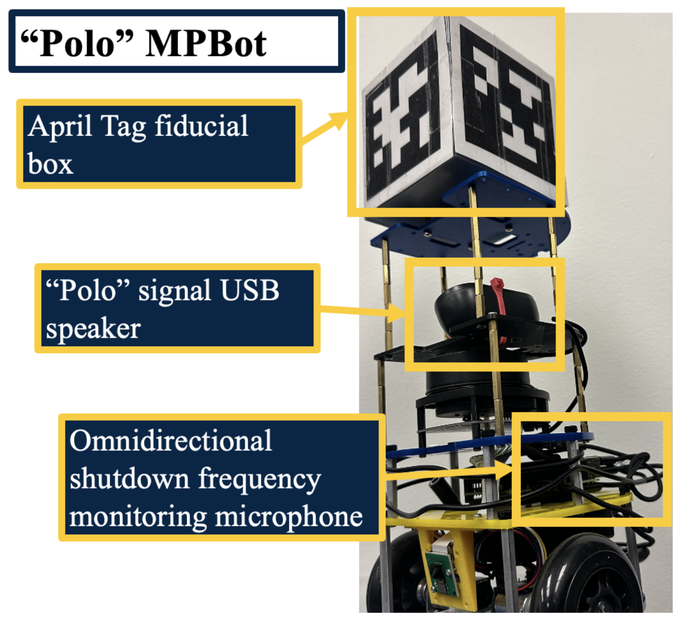

Fig. 3: Assembled Polo MPBot

this signal and then shuts down by disabling its motion planner. Likewise, Marco also disables its own motion planner upon sending this signal.

B. Pursuit-Evasion Simulation

To aid in the development of the MPBot, a pursuit-evasion simulation was developed to test different methodologies. This allowed for tuning of pursuit-evasion strategies before MPBot assembly and integration was complete.

In simulation, we studied three evasion patterns against and three main pursuit strategies.

1) Evasion Patterns: We studied three evasion patters: random walk, clockwise motion around the arena, and counterclockwise motion around the arena. We chose these strategies to illustrate the principal sources of pursuit failure present in pursuit strategies, those being a failure to converge and a failure to intercept.

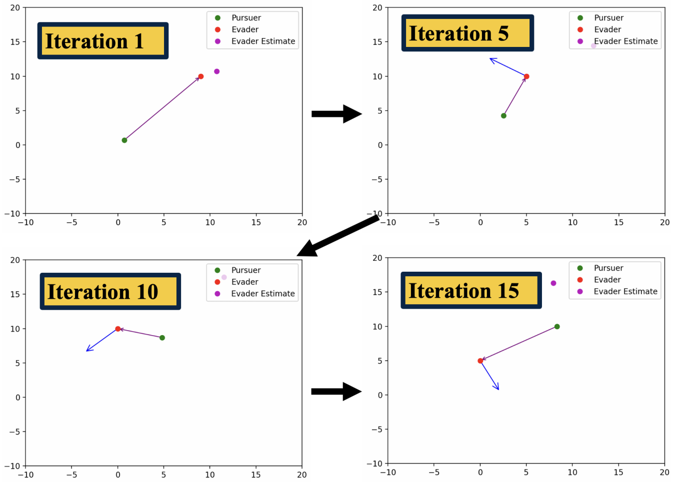

Fig. 4: Example Simulation Iterations; Predictive Chase Behavior

Convergence failure occurs when, rather than closing the distance between itself and its target, the pursuer agent attempts to merely match the movements of the evader, leading to an essential stalemate. This failure mode is revealed through a random walk evader, whose moves are by definition unpredictable and thus, requires strong convergence instead of prediction.

Interception failure occurs when the pursuer fails to accurately predict the motion of the evading agent, allowing for actual interception, and instead greedily converges to the current position of the evader at all times. This greedy convergence leads to an inability to make close contact with the evader, as the pursuer instead simply falls in behind the evader's motion and chases it. This failure mode is revealed through the counterclockwise and clockwise evasion patterns, which require interception to predict the motion along the walls of the evader.

2) Pursuit: Direct Chase: The direct chase strategy is the simplest pursuer strategy, and was primarily developed as an integration test for the MPBots as well as a simple baseline of comparison. In this strategy, the pursuing MPBot simply chases the evading MPBot at all times, setting waypoints to go to the current evader position. This results in a guaranteed interception failure, and a guaranteed convergence success.

3) Pursuit: Smooth Chase: The smooth chase strategy was an evolution of the direct chase strategy, and aimed to explore the efficacy of velocity prediction of the evader MPBot. In this strategy, the pursuing MPBot predicts the velocity of the evader over a fixed time horizon by successive bearing measurements, and then attempts to move in the same direction. This results in a near-guaranteed interception success, and likewise a near-guaranteed convergence failure, due to the lack of a direct pursuit component.

4) Pursuit: Predictive Chase: The predictive chase strategy was the synthesis of the direct chase strategy and the smooth chase strategy, and aimed to solve both the convergence and prediction problems by estimating the velocity of the evader over a set time frame, projecting that velocity forward in time, and directly converging to that point instead. This strategy thus has potential to successfully converge and intercept.

All three of these pursuit and evasion strategies were implemented in simulation, as seen in Figure 4. This simplistic simulation nevertheless allowed for early refinement of our pursuit-evasion algorithms.

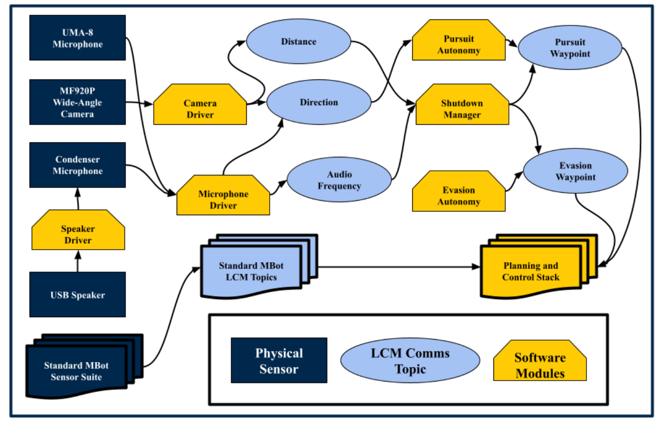

Fig. 5: MPBot Information Flow

C. MPBot Integration

The integration of new sensors, new guidance algorithms, and new objectives, required a significant expansion of the current robot informational interchange over the Lightweight Communication and Marshalling (LCM) protocol. Figure 5 details the new nodes which were created in order to realize the full capabilities of the MPBot. The implementation of a new state machine, built on the model of the MBot's exploration module but focused instead on, for Marco, the acquisition and constant tracking of Polo and the pursuit of Polo. For Polo, this state machine instead focused on the successful execution of Polo's circuit routes (counterclockwise and clockwise around the arena) and the collision-free planning of its random routes.

Joined to this main state machine, the pursuit/evasion command module interfaced directly with the appropriate pursuit/evasion algorithm, subscribed to sensor data to generate new waypoints, and published those waypoints back to the MPBot's state machine. This allowed for a more modular design approach, as new state machine logic had no direct reliance on the inner-workings of the waypoint generation command module, and vice-versa.

Feeding information to both the MPBot state machine and the waypoint generation command module, the sensor drivers of the MPBot allowed for immediate data processing and refinement, adjusting for physical offsets and filtering incoming data to allow for smooth working of the rest of the new pursuit/evasion autonomy stack. Microphone drivers filtered out noise frequencies and listened for specific signals, such as the shutdown or tracking signal. The camera driver, on the other hand, monitored for the April Tag and evaluate whether shutdown distance had been reached; sending the shutdown signal to both MPBots if it had been. This structure allowed for easily switchable pursuit and evasion modes, which was of great benefit as different environments warranted different modes (for example, the atrium of the busy Ford Robotics Building was too noisy for audio-only pursuit, but perfectly lit for visual pursuit).

Upon completion of all individual software modules, Figure 6 shows the initial integration Pursuit-Evasion test performed in the EECS 467 Laboratory, with an extremely simple layout. In this test, the evading robot was initially kept stationary to test the basic functionalities of the pursuing robot. Then, the pursuing robot was kept still while the evading robot moved freely, to test its motion. Finally, both robots were allowed to move, and a successful pursuit was demonstrated.

IV. Results

The Marco-Polo MPBots were tested in two different types of arena configuration, with obstacles scattered semi-randomly within.

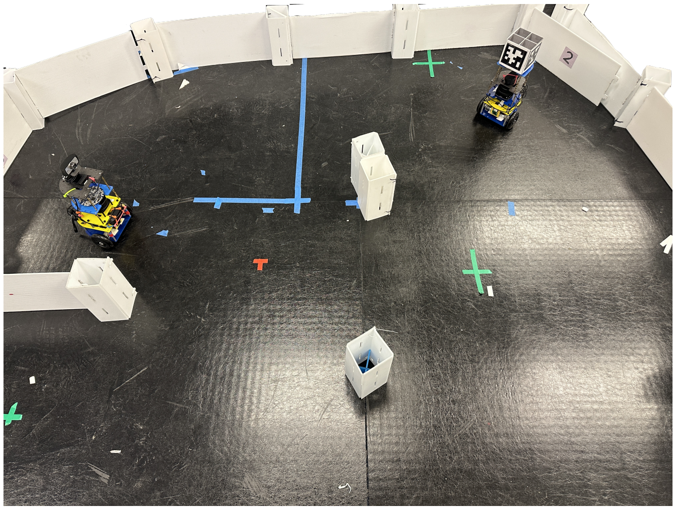

Fig. 6: MPBot Integration Test

Fig. 7: Easy Layout - In Ford Robotics Building Atrium

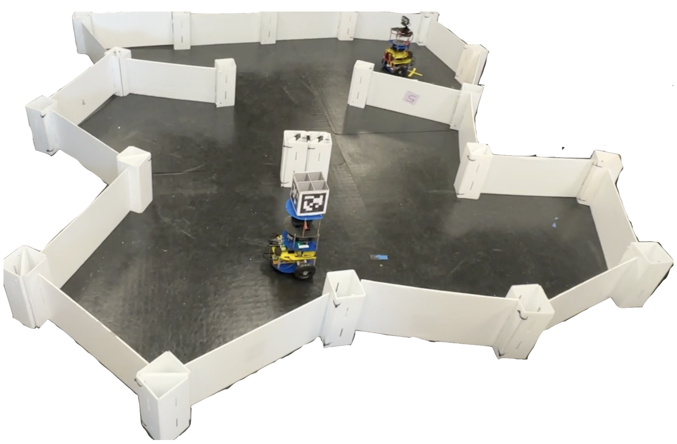

Fig. 8: Difficult Layout - In Lab

One set of trials took place in the atrium of the Ford Robotics Building, while the other took place in the EECS 467 Laboratory. This differentiation allowed for different lighting conditions, floor textures, and echoing, thus returning a wider array of results.

A. Audio Pursuit

Audio pursuit, due to its ambient noise limitations, was only conducted in the environment shown in Figure 8, which was in a controlled and quiet laboratory space. We achieved the desired results, whilst conducting trials in a quiet space, as Marco was able to catch Polo during a pursuit-evasion game. As shown in our demo video, as soon as Marco hears Polo emit a sound, in this case the word "Hello," Marco sets a waypoint in the general direction of the audio and begins its pursuit of Polo. The initial concept of only having an audio-based pursuit, where Marco would chase Polo based on the direction Polo emits sound, was met with some difficulty as the beamforming filtering output the direction of closest voice. We conducted multiple trials of testing, in which we had crowd noise close to and far from the MPBots. In each of these tests the UMA-8 microphone would pick up the sound emitted from Polo, as well as the crowd noise causing Marco to set waypoints in the wrong direction and away from Polo. We attempted to combat this was by changing the thresholds of the microphone and adding keywords, so that the microphone would only pick up certain words, which Polo would be programmed to emit; however, end-of-life drivers hampered this effort, and so we were forced to rely on the microphone's inherent speech recognition logic.

Despite these setbacks, audio guidance remains a promising solution to our pursuit-evasion game. Due to the omnidirectionality of the UMA-8, the MBot had almost no need to stop and turn, to reacquire a visual lock on Polo. This trait was especially useful due to the underlying unicylce dynamics of the MBot, which limits motion only along the heading of the robot; while the camera would often be pointed away from Polo to effect some maneuver, the microphone never was.

B. Vision Pursuit

Vision pursuit was conducted in the environments shown in Figures 6, 7, and 8. It was initially conceived as a fallback method of pursuit, but after uncovering the difficulties and potential inaccuracies of the UMA-8 solution, it became a primary method of pursuit - especially in environments with other people. However, even this method carried difficulties with it. More computationally expensive than the UMA-8 due to the lack of any offboard processing, we were forced to reduce our planned three cameras to only one - this, in turn, forced us to develop new state machine stages, to account for loss-of-signal as well as iteration-to-iteration inaccuracies due to camera distortions. This lack of reliable omnidirectional bearing discernment led to the abandonment of more guaranteeably optimal pursuit algorithms, which relied on constant bearing measurements to formulate control signals. Furthermore, we encountered difficulties due to interference from other April Tags in the environment, forcing us to exert a degree of control over the environment to ensure no false April Tag readings were recorded.

Once these setbacks were overcome, vision-based pursuit proved to be an effective solution. The tag-camera sensor system was always a necessity, due to the need for proximity shutdown to avoid damaging the MBots. Simply tapping into this feed to additionally record bearing angles proved efficient, and so this method was used for much of the development of the system. Its primary failing is in the processing rate of the computer itself - although when used in isolation, the camera exhibits negligible motion blur, when forced to rely on the onboard Raspberry Pi computer, the processed images do blur at times, causing a loss-of-lock unrelated to the bearing angle. We overcame this difficulty by adding a pause state into our loss-of-lock movement, which could be interrupted once lock was reestablished. This pause allowed for the image to unblur enough for the April Tag processing software to reacquire its target.

V. Discussion and Future work

Although the current MPBot configuration has demonstrated its basic ability to perform a pursuit-evasion game, every aspect of the MPBot platform - hardware, software, and fundamental algorithms - can and should be studied and improved for future demonstration of higher autonomy.

Computing: The Raspberry Pi 4, although a capable enough computer for the standard tasks allocated to the MPBot (SLAM, A* navigation, basic PID control), is not a computer capable of large-scale video processing. Initial project concepts called for the use of three, not one, wide-angle cameras. However, the RPi simply could not intake that much data simultaneously, let alone process each stream to search for April Tags. This reduced performance of the pursuer agent, and disallowed use of more complex algorithms which carried with them the basic assumption of a constant bearing measurement on the evading target. Future iterations of this project should look towards a more capable computer with integrated graphics processing, such as the NVIDIA Jetson family of single-board computers.

Signal Processing: The UMA-8 microphone, although successful in our limited trials, displays some troubling behavior which limits the further development of this project. Namely, the beam-forming logic of the pre-packaged microphone drivers are optimized for simple noise reduction, rather than specifically signal source detection. This means that, once a beam is formed, it does not re-form unless there is a period of complete silence. Therefore, while on the move, the direction reading of the UMA-8 microphone may drift as it only slowly adjusts its bearing measurement. One solution to this issue would be to instead use the "raw" microphone drivers, which allows access of the raw microphone readings with essentially no onboard processing. This would allow for the development of a custom direction-finding application to be run on the computer, which could be tuned to prioritize direction-finding over any other functionality. This approach should be followed if an increase in agent speed was found to be desirable, since it is at high speeds where the current microphone directional drift is most troublesome. However, this approach should be joined with the acquisition of a superior computer, to allow for the greatly increased processing load.

Sensor Fusion: Both the UMA-8 microphone and the wide-angle camera employed in the Marco MPBot have unique strengths and weaknesses. The camera, with its superior angular resolution and noise rejection, is an accurate sensor - but limited in its field of view. The UMA-8 microphone, meanwhile, has an unlimited field of view, but poor range and accuracy. These sensors appear to be ripe for some kind of sensor fusion, to combine the heading data streams of both the camera and the microphone to create a single, accurate, and always-active heading data stream at the disposal of Marco. Such a data stream would further enable optimal algorithms to be employed, and would additionally allow for the simplification of our state machine logic, to eliminate costly "wait" states which are required when using the camera by itself for guidance.

Platform Alterations: The basic MBot platform, with its unicycle dynamics, is perhaps the worst possible platform to use when conducting a pursuit-evasion game. Motion perpendicular to the current heading of the MPBot is almost guaranteed to result in a loss of target acquisition due to our one-camera limitation. Furthermore, the cumbersomeness of the MPBot gives a significant advantage to the evader, as the pursuer struggles to navigate tight corridors and obstacles. In this way, a high-performance pursuit-evasion game using the current MPBot design would essentially devolve into a test of the robots' controllers, rather than a test of the actual pursuit-evasion behavior. Should such high-performance testing become desirable, the MPBots should be retrofitted with a heading-agnostic system of movement [13] better able to respond to pursuit or evasion commands and thus to develop and demonstrate their behavior.

Control Logic: The current approach to robot control is inextricably linked to the robot motion planner. This, in itself, is not an issue, as waypoint-based control represents the dominant form of control required by almost any autonomy algorithm. However, its current implementation fundamentally eliminates the possibility of direct motion control of the robot. This, in turn, limits the ability of the pursuit-evasion algorithms to make fine heading adjustments to regain visual lock or to make quick movements to either apprehend or evade. Therefore, to increase physical robot performance, the underlying control architecture should be altered to allow for direct and overriding translational and rotational velocity commands to be sent to the robot controller and applied to the motors.

Evader Upgrades: Currently, Polo is not an advanced MPBot. It is not able to sense Marco's position, and thus it lacks the ability to adaptively evade, relying instead on pre-programmed heuristics to guide its movement. Although sufficient for a demonstration of basic pursuit functionality, the addition of a Polo wide-angle camera and of April Tag fiducials on Marco would allow for Polo to provide a more challenging demonstration of evasion. Furthermore, it would allow for the development of theoretically sound evasion algorithms, able to give some kind of potential guarantee of performance which the current heuristics lack.

Fundamental Algorithms: Once all or even a majority of the above improvements are made to the MPBot platform, the implementation of more advanced pursuit and evasion logic should be pursued. With improvements made to the robot drive system, controller, processing, and sensors, the guarantee of a perpetual heading lock could be satisfied, and thus a more advanced family of guaranteeable successful pursuit algorithms could be implemented.

VI. Statement of Contributions

John Pye did substantially write the final report and create the demonstration video, developed the simulation environment and attendant pursuit and evasion behaviors, developed the algorithm-autonomy interfaces necessary to control the robot, implemented the pursuit and evasion algorithms in the autonomy stack, created the state machine necessary to drive the pursuit-evasion behaviors of the MPBots, tested both MPBots, assembled both MPBots, and helped to tune the microphone drivers of the MPBots.

Dhillon Patel did the initial set up of the code, find and implement all of the speaker, microphone, camera, and AprilTag drivers, calibrate the cameras, configure the Raspberry Pi boards for multiple USB devices, implement and test all of the drivers, helped with integration and the state machine of the pursuer, and a substantial amount of testing and tuning of the MPBots.

Andrew Greer assisted in various aspects of the project wherever needed. Contributed to initial testing of beam-forming microphone, troubleshooting issues, testing, among others.

Max Ernst helped with testing and integration of cameras, microphones, and speakers. Additionally, wrote the code for and served as a reference for LCM messaging and, and created the framework for pursuer and evader command. Aided in testing, tuning, and bug fixing where needed.

VII. Demo Video Link

Group 4 Demo Video

VIII. Code Repo Link

Group 4 Botlab Repo

IX. Conclusion

This project set out to extend the sensing, control, and autonomy functionalities of the MPBot platform to enable the performance of a pursuit-evasion game. Despite technical difficulties with the UMA-8 directional microphone, processing limitations which restricted the MPBot to the use of a single tracking camera, and general platform nonlinearities (unicylce dynamics are one of the worst dynamical models for both pursuit and evasion, due to its reliance on heading for movement), we have largely succeeded. We first demonstrated the ability to track a moving April Tag target, plan around obstacles, re-acquire the target after a loss of lock, and successfully intercept the evader. We then demonstrated the ability to replicate this behavior using audio-only tracking with the UMA-8 directional microphone array, proving the original concept. The upgraded MPBot firmware, designed with modularity in mind, will be able to be used in future projects, attempting more ambitious autonomous behaviors.

REFERENCES

[1] EECS 467 Course Staff. *EECS 467 WN24 Slides*. Courier Corporation, 2024. URL: https://drive.google.com/drive/folders/1k2LsNlQYenkD3fTVSE6PTB4Cy1tz1eGF?usp=drive_link.

[2] John Wang and Edwin Olson. "AprilTag 2: Efficient and robust fiducial detection". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2016.

[3] Dave W. Oyler, Pierre T. Kabamba, and Anouck R. Girard. "Pursuit–evasion games in the presence of obstacles". In: *Automatica* 65 (2016), pp. 1–11. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2015.11.018. URL: https://www.sciencedirect.com/science/article/pii/S0005109815004823.

[4] Timothy Chung, Geoffrey Hollinger, and Volkan Isler. "Search and Pursuit-evasion in Mobile Robotics". In: *Auton. Robots* 31 (Nov. 2011). DOI: 10.1007/s10514-011-9241-4.

[5] Rufus Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[6] Shaunak D. Bopardikar, Francesco Bullo, and JoÃo P. Hespanha. "On Discrete-Time Pursuit-Evasion Games With Sensing Limitations". In: *IEEE Transactions on Robotics* 24.6 (2008), pp. 1429–1439. DOI: 10.1109/TRO.2008.2006721.

[7] Nikhil Karnad and Volkan Isler. "Bearing-only pursuit". In: *2008 IEEE International Conference on Robotics and Automation*. 2008, pp. 2665–2670. DOI: 10.1109/ROBOT.2008.4543614.

[8] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2026.

[9] Tamer Basar and G Jan Olsder. "Dynamic noncooperative game theory, ser". In: *Classics in Applied Mathematics. SIAM* (1999).

[10] Jiří Sgall. "Solution of David Gale's lion and man problem". In: *Theoretical Computer Science* 259.1 (2001), pp. 663–670. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(00)00411-4. URL: https://www.sciencedirect.com/science/article/pii/S0304397500004114.

[11] Xiuqing Qu et al. "Pursuit-evasion game strategy of USV based on deep reinforcement learning in complex multi-obstacle environment". In: *Ocean Engineering* 273 (2023), p. 114016. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2023.114016. URL: https://www.sciencedirect.com/science/article/pii/S0029801823004006.

[12] Yueying Wang et al. "Threat potential field based Pursuit–Evasion Games for underactuated Unmanned Surface Vehicles". In: *Ocean Engineering* 285 (2023), p. 115381. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2023.115381. URL: https://www.sciencedirect.com/science/article/pii/S0029801823017651.

[13] Mariane Dourado Correia, André Gustavo, and Scolari Conceição. "Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models". In: *IFAC Proceedings Volumes* 45.22 (2012). 10th IFAC Symposium on Robot Control, pp. 7–12. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20120905-3-HR-2030.00002. URL: https://www.sciencedirect.com/science/article/pii/S1474667016335807.