Measurable Augmented Reality for Prototyping Cyberphysical Systems: A Robotics Platform to Aid the Hardware Prototyping and Performance Testing of Algorithms

Measurable Augmented Reality for Prototyping Cyber-Physical Systems

Shayegan Omidshafiei, Ali-akbar Agha-mohammadi,

Yu Fan Chen, N. Kemal Ure, Shih-Yuan Liu, Brett T. Lopez,

Rajeev Surati, Jonathan P. How, and John Vian

Planning, control, perception, and learning are current research challenges in multi-robot systems. The transition dynamics of the robots may be unknown or stochastic, making it difficult to select the best action each robot must take at a given time. The observation model, a function of the robots' sensor systems, may be noisy or partial, meaning that deterministic knowledge of the team's state is often impossible to attain. Moreover, the actions each robot can take may have an associated success rate and a probabilistic completion time. Regardless of the control scheme, planning, or learning algorithms used for a specific problem, robots designed for real-world applications require careful consideration of such sources of uncertainty. Understanding the underlying mechanisms of planning algorithms can be challenging due to the latent variables they often operate on. When performance-testing such algorithms on hardware, simultaneous use of the debugging and visualization tools available on a workstation can be difficult. This transition becomes especially challenging when experiments need to replicate some feature of the software toolset in hardware, such as simulation of visually-complex environments. This article

details a robotics prototyping platform, called Measurable Augmented Reality for Prototyping Cyber-Physical Systems (MAR-CPS), which directly addresses this problem, allowing real-time visualization of latent state information to aid hardware prototyping and performance-testing of algorithms.

Hardware-in-the-loop experiments are a key step for transitioning the implementation of planning and learning algorithms from simulations to real-world systems. They are not only important for verification of an algorithm's performance in a real-world setting, but also for conveying behavioral characteristics of algorithms to researchers and other observers. They allow determination of an algorithm's robustness to uncontrollable factors such as environmental uncertainty and sensor noise, which need to be verified before deployment onto consumer-level platforms. Operation of experimental hardware in an outdoor environment may be disfavored, in some instances, due to safety, cost, or regulatory concerns. For instance, although Federal Aviation Administration (FAA) regulations have allowed limited testing of Unmanned Aircraft Systems (UAS) in outdoor spaces, several regulatory constraints are in place which affect rapid prototyping of such systems, including required registration of UAS and operator certification [1].

During the execution of planning and learning algorithms, numerous latent variables such as probability distributions over the system state, predicted agent trajectories, and transition probabilities are manipulated in software, but are difficult to convey on hardware testbeds. In frameworks designed for planning under uncertainty, which in a general form can be described as Partially Observable Markov Decision Processes (POMDPs) [2], agents operate in belief space

(the space of probability distributions over states) rather than state space. Though hardware experiments enable spectators to observe the performance of planning algorithms in the real world, it is difficult to simultaneously convey latent information such as an agent's belief state alongside physical platforms. The perceived performance of algorithms can, therefore, suffer due to this complexity not being appropriately conveyed to spectators.

MAR-CPS (illustrated in Fig. 1) is an experimental architecture that enables controlled testing of planning and learning algorithms in an indoor setting which closely emulates outdoor conditions. The presented architecture leverages motion capture technology with edge-blended multi-projection displays to improve state-of-the-art indoor testing facilities by augmenting them with interactive, dynamic, partially unknown simulated environments.

This article discusses related CPS prototyping environments and presents the system architecture for MAR-CPS. Technical features of MAR-CPS are then outlined, providing details of high-level system capabilities. An 'Application Focus' section presents a tutorial-style discussion of the improvements MAR-CPS provides for two example experiment domains (high-speed quadrotor maneuvering and multi-agent intruder monitoring) from both researcher and spectator perspectives. Step-by-step descriptions of each mission, visualization elements, and associated benefits provided to researchers and spectators are provided. Finally, a sequence of higher-level case studies demonstrating remaining capabilities are presented, with a focus on planning, perception, and learning algorithms for autonomous single-robot and multi-robot systems which actively sense and interact with the augmented laboratory space.

Framework for Prototyping Autonomous Vehicles

The ancestor of MAR-CPS, referred to as Real-time indoor Autonomous Vehicle test ENvironment (RAVEN), was designed to facilitate rapid prototyping of autonomous vehicle systems through modular mission, task, and vehicle components [3]. The inherent flexibility in the architecture enables system managers to easily change mission specifications such as high-level goals or the number/type of vehicles involved in tasks. With the combination of both air and ground vehicles it is possible to create scenarios that stress test a variety of planning and learning algorithms, although it is typically difficult to generate hardware environments that are complex and time-varying enough to performance-test the vehicles' onboard perception systems. RAVEN can be outfitted with physical obstacles and target vehicles to test algorithms for a target-tracking scenario, but such a domain is fixed for the duration of the experiment. Motivated by the goals of increasing domain complexity and enabling the possibility that it be time-varying (possibly the output of a dynamic simulation) this article presents an extension of RAVEN to include augmented reality. The approach taken maintains the modularity of the original design in that the software and hardware required for the augmented reality architecture are decoupled from the rest of the mission, allowing experiments to be conducted even when the projection system is offline.

The primary contribution of this work is an augmented reality system architecture that can be implemented in research laboratories to transform barren indoor flight spaces into dynamic interactive environments that are more representative of the outside world.

Related Work

Various prototyping environments for CPS have been developed in the past [3]–[9], but the addition of augmented visualization capabilities to indoor platforms is of ongoing interest. For example, display of dynamically changing events using projectors for hardware experiments involving quadrotors has been investigated [10]. Specifically, reward and damage information for quadrotors involved in an aerial surveillance mission was displayed in real time, although simulation of complex mission scenarios, measurement of the augmented environment using onboard sensors, and display of state transition and observation probability distributions were not demonstrated. Also, [8] utilized physical props (small blocks) as surrogates of building components enabling the construction of small-scale structures.

An investigation of augmented reality for multi-robot mission scenarios has also been conducted [11], including applications in pedestrian perception and tracking of swarm robotics. However, their applications are limited to display of this information in software only, and integration of the data into a physical laboratory space was not conducted.

Onboard projection systems have also been investigated for human-robot interaction situations [12], with applications in robot training demonstrated, though the projection footprints are limited. Due to increasing affordability of virtual reality headsets, such as the Oculus VR [13] and SteamVR [14], their usage in a CPS-prototyping setting may be possible. Usage of virtual reality head-mounted displays to superimpose mission data over a live camera feed has previously been investigated [15], with applications to intruder monitoring in swarm robotics.

For demonstrations involving many spectators, virtual reality headsets are typically infeasible due to the large amount of hardware and supporting infrastructure required. Information displayed in a virtual reality headset is also not measurable using onboard sensors on a vehicle, whereas projected images are physically visible in a lab and can be directly measured.

MAR-CPS leverages the emergence of motion capture technology as well as multi-projection systems to change the state-of-the-art for CPS prototyping. MAR-CPS not only allows display of latent state information, but also enables hardware-level interaction of vehicles' sensor systems with augmented, customizable mission domains of arbitrary complexity with little hardware overhead. Previous work on virtual and augmented reality interfaces is additionally extended by demonstrating that measurement of projected environments using noisy hardware sensors is a useful validation tool in situations where outdoor testing is infeasible.

System Architecture

Fig. 2 illustrates the system architecture. The system has several components: i) a high-level mission manager, ii) autonomous vehicles (each with access to a planning, control, and perception CPU) equipped with onboard sensors, iii) motion capture system, and iv) projection system.

The central mission planner coordinates high-level tasks for the vehicles, given the mission objective. For instance, in a multi-robot package delivery scenario, the central planner assigns packages and delivery destinations to individual vehicles. In a partially-observable domain, where the vehicle state is not known deterministically, this planner could be assigning

tasks based on each vehicle's belief of its state. This architecture can be extended to decentralized systems where each vehicle chooses its own tasks based on local observations.

Each vehicle communicates with a designated CPU for planning, perception, and low-level control. Given a task, the planning CPU defines a valid trajectory for the vehicle. The trajectory is relayed to a control CPU, which defines low-level control inputs to the vehicle using feedback from the motion capture system. Note that the planning CPU also has knowledge of the controllability of the vehicle in question, and its role can be combined with the control CPU if desired.

The vehicle can simultaneously perceive or measure the projected virtual environment and convert these observations to useful features using its perception CPU. The perception CPU can process still images from a camera sensor to find objects of interest, which can then be tracked using the planning and control CPUs. The perception CPU has the additional task of performing state estimation using motion capture data, which is done by Kalman Filtering the raw position and orientation data provided by the motion capture system for each vehicle.

The capability to perceive the projected augmented reality environment allows replication of outdoor test environments in a controlled, indoor space. Sensor systems used in outdoor environments can be used within MAR-CPS to obtain noisy measurements, allowing tests of complex domains with noisy observation models. The modular architecture of MAR-CPS allows tests in a variety of simulated environments to be conducted with low overhead.

Hardware

Fig. 3 illustrates a hardware overview for MAR-CPS. The visualization system is implemented in MIT Aerospace Controls Laboratory's RAVEN [3] indoor flight testbed. This system uses 18 Vicon T-Series motion capture cameras allowing tracking of heterogeneous teams of autonomous vehicles [16]. A unique pattern of reflective motion capture markers is affixed to each vehicle, allowing the motion capture system to determine the position and orientation of the vehicles, as seen in Fig. 3.

State and latent information for the team is published to the laboratory network using Robot Operating System (ROS) [17], allowing feedback control of the vehicles, as well as a computer dedicated to visualization-rendering to package this information in an intuitive format for researchers. The visualization is then projected onto the experiment area using 6 ceiling-mounted Sony VPL-FHZ55 ground projectors, with latent data animations and physical systems being run synchronously. This augmented reality interface allows designers and spectators to observe hardware while simultaneously gaining an intuitive understanding of decisions made by the planning and learning algorithms.

A primary challenge in implementing this system was to ensure that the footprint of the experiment testbed would not be downsized. The 1200-plus square foot RAVEN laboratory space is used for experiments involving a variety of ground and air vehicles, therefore restricting its size to be equal to the footprint of a single projector is infeasible. Two solutions are presented for this. First, the projected area can be treated as a window into the underlying belief space.

Though the hardware itself may run in a larger physical space, visualization can be presented only for specific sub-regions of this space, providing observers an understanding of the decision-making scheme used by the algorithm, allowing them to extend its behavior to regions with no visualization. Second, the projected area is not necessarily constrained to the size of a single projector. Instead, multiple projectors are combined to increase the overall visualization footprint.

Additional hardware can be appended to MAR-CPS to simulate outdoor environments more realistically. Fig. 4 shows usage of an industrial fan for simulating turbulent wind conditions while testing a UAS in a forest firefighting experiment.

Multi-Projection System Calibration

MAR-CPS uses a multi-projector system for data visualization in the physical lab space. Misalignment of the projectors' mountings requires careful treatment, as it may cause affine warping and distortions in these visualizations. It is infeasible to permanently align the edges of the projected images in hardware due to ground vibrations moving the projectors over time, resulting in overlapping or gaps in the images.

To counter this, a software calibration scheme was implemented in collaboration with Scalable Display Technologies, a company specializing in multi-projector displays [18]. Driver-level customizations on a computer running NVIDIA Mosaic-capable K5000 graphics cards allow edge-blending of projector displays, as well as de-warping of images in the lab environment. The result is a seamlessly blended projection region with the majority of affine distortions removed. Note that during the calibration phase, the placement of the camera is important for reducing

warping near the edges of the projections. Specifically, the calibration camera orientation must be such that the angle between the projection plane normal vector (for all projection surfaces) and the camera image plane normal vector is not highly oblique. Otherwise, warping of the projected image occurs in certain regions.

Additional calibration is required for the mapping between the pixel space and the physical (Vicon) space so that visual elements such as markers and trajectories align with the physical counterparts which they represent. Since the canvas in the pixel space is rectangular but the projection footprints might not be due the placement of projectors and the edge blending, the calibration is non-trivial. Piecewise-linear mappings on Delaunay triangulation are used to transform coordinates between the pixel space and the physical space. The calibration procedure for the mappings is described as follows. Markers are generated on a regular grid in the pixel space and projected onto the floor. The markers' positions in the physical space are measured by the Vicon system as shown in Fig. 5. To map the pixel space to the Vicon space, a Delaunay triangulation is constructed in the pixel space using the markers. Two piecewise-linear surfaces are constructed on top of the Delaunay triangulation with the height of the surface at each marker set to the x and y coordinates of the marker in the physical space respectively. A point in the pixel space can then be mapped to the Vicon space efficiently by querying the two piecewise-linear surfaces. Mapping from the Vicon space to the pixel space can be handled with the same approach.

Technical Features

MAR-CPS includes several features that extend prototyping of autonomous systems in traditional laboratory spaces, as well as debugging and demonstration of planning and learning algorithms in real-time. These features are outlined in the following sections.

Rapid Prototyping in Simulated Environments

The modular architecture of MAR-CPS is focused on minimization of logistics for autonomous vehicle research labs, allowing new vehicles and software capabilities to be added on-the-fly. MAR-CPS provides visual awareness of both high and low-level information from software and hardware platforms, making it an efficient testbed for rapid prototyping of autonomous vehicles. If desired, MAR-CPS can be designed to be disjoint from experiments, allowing it to be "turned off" without affecting vehicle behavior or performance of the experiments conducted.

Traditionally, testing an algorithm's performance in simulation and on physical systems has been disjointed. A typical framework for development of algorithms for autonomous vehicles involves initial experimentation in simulation and a subsequent transfer to physical platforms (either in an indoor or outdoor environment). This transfer may cause problems such as discrepancies between models used in simulation and real-world models (of sensors, actuators, and environment), including miscalibrations. In particular these discrepancies in complex physical systems consisting of several interacting vehicles can make it difficult to understand the behavior

of algorithms or root causes of performance problems.

Traditional debugging schemes for such scenarios are iterative, requiring update of software, verification in simulation, and testing in hardware [19]. Specifically, software debugging requires extracting and understanding low-level system information while experiments are conducted. MAR-CPS alleviates this time-exhaustive process by allowing display of such information in real time. Information regarding the location and velocity of vehicles can be visually shown next to them in the physical space, allowing immediate identification of discrepancies between hardware sensors and software variables.

MAR-CPS is a platform designed to transform indoor laboratories into controlled simulations of outdoor environments in which experiments involving autonomous vehicles can be conducted prior to real-world deployment. This technology provides researchers an inexpensive pathway to field testing, which is distinct from traditional testing in simulations or indoor environments. Specifically, visual presence of obstacles, environmental conditions such as varying terrain type or wind velocity vector fields, and presence of restricted regions in a mission scenario can be implemented and visualized using MAR-CPS. Fig. 22 indicates trajectories as well as detected obstacles in a self-driving vehicle scenario.

Vehicles used in MAR-CPS are subject to lower levels of wear-and-tear and unforeseen environmental factors that may damage them during a field test, leading to increased lifespans and less effort spent by researchers on re-calibration or troubleshooting of hardware. Debugging of software issues in the simulated MAR-CPS environment is also eased, as full access to laboratory resources (which may be too costly or difficult to transport outdoors) is maintained.

Finally, in situations where physical vehicles are expensive, use of real-time projection in MAR-CPS allows integration of virtual vehicles in experiments. Physical and virtual vehicle interactions can be modeled in software, allowing inexpensive testing of complex, multi-agent mission scenarios. Fig. 23 illustrates a path-planning with obstacle avoidance demonstration involving 2 physical and 2 virtual quadrotors, a scenario in which vehicle collisions may be expected during early phases of testing and debugging. Using MAR-CPS, virtual vehicles can be incrementally replaced with physical counterparts as the collision avoidance algorithm is made more robust. Thus, physical-virtual vehicle interactions can be leveraged to reduce the risk associated with the hardware-in-the-loop debugging process.

Window into Belief Space

Application of planning and learning algorithms in real-world settings often requires handling of stochasticity in the environment. In such scenarios, each vehicle constructs a "belief" or probability distribution over the state space, using it to decide its next best action. Decisions made by planning algorithms may be counter-intuitive to human observers if a thorough representation of a vehicle's perception of the environment is not conveyed.

In the planning under uncertainty problem for an autonomous agent, the agent operates in a partially observable domain and takes a series of actions (with stochastic outcome) in order to achieve a set of tasks. This problem can be formulated as a Partially Observable Markov Decision Process (POMDP) [20].

In partially-observable domains, since the state is not explicitly known, decision-making

is performed in *belief space*. Belief $b(s)$ is a probability distribution over state s of the agent [20], where $b(s) = P(s)$. Given current belief $b(s)$, action a, and observation o, the agent transitions to a new belief $b'(s')$. POMDPs and their multi-agent counterpart, Decentralized POMDPs, have been applied to networking, multi-robot exploration, and surveillance problems [21], [22]. Though agents in the POMDP framework operate in belief space, no prototyping or debugging hardware architecture exists for visualizing the agent belief $b(s)$ as it propagates over time. While visualizations of state transition probabilities $P(s'|s, a)$ and observation probabilities $P(o|s', a)$ in a hardware prototyping setting would be useful aids in understanding the sources of uncertainty in the problem domain, no means of doing so exist. Note that these needs are not specific to the POMDP problem, and that a similar gap in debugging hardware exists for other planning and learning frameworks which operate on latent variables.

In certain scenarios, once experimental data is gathered, latent or belief information can be visualized through simulation software. But it can be difficult for observers to synchronously monitor behavior in the simulator and on the physical platform, especially in the case of real-time algorithms. It is beneficial to augment the experiment area with real-time visualization of this data, allowing direct perception of the progress of the planning and learning algorithm.

Visualization of such information, specifically the level of uncertainty in a vehicle's perceived state, aids researchers' understanding of the behavior of the algorithms. In the past, conveying this information in real time has been difficult. MAR-CPS uses ceiling-mounted projectors to show latent information about an autonomous agent, such as its perception of its own location, nearby obstacles, future trajectories, failure states, battery levels, and communication

links with other teammates.

Fig. 6 shows an example of belief space visualization for robot localization. In such domains, robots observe features of their surroundings in order to determine their location. However, observations are typically made using noisy sensors, leading to a posterior distribution on the robot's location. MAR-CPS allows visualization of this distribution. In Fig. 6, regions with high posterior probability for the robot's location are highlighted in red. The robot in Fig. 6 uses a Gaussian mixture model to represent belief on its location, and MAR-CPS prominently indicates the underlying multi-modal belief state. Using MAR-CPS, spectators can visualize and intuitively understand the concept of 'belief space'. They can also see the transformation of the robot's belief state in real-time (and with the correct scale in the laboratory) as it performs actions and gathers observations.

In Fig. 7a, a second example of belief visualization in MAR-CPS is shown. In this domain, a quadrotor maintains a belief over its health state, which is partitioned into low, medium, and high health. As the quadrotor executes its mission, it uses noisy observations from a health sensor to update its health belief, which is projected underneath it as a donut chart. The visualization follows the quadrotor and is dynamically updated as the mission progresses. Such a visualization is important, in this context, since the actions of the quadrotor are associated with its belief over health. When the quadrotor believes it is in low health, it temporarily halts its mission and performs a 'repair' action. As the quadrotor increases and decreases altitude, the donut chart expands and shrinks, indicating the field of view of the downward-facing camera onboard the vehicle. This dynamic resizing of the health belief visual also ensures that it is never

observed by the quadrotor's onboard camera, and does not corrupt the vehicle's observations of its environment. Fig. 7b illustrates the full mission domain with 4 quadrotors, each with an associated health belief. Also indicated are the planned paths for each vehicle, as well as labeled locations where each vehicle can perform specific tasks. Although the monitoring of belief and path-planning information for a single quadrotor may be possible using conventional methods, MAR-CPS has been found to be indispensable for real-time understanding of complex scenes involving many dynamic elements.

Measurable Augmented Reality: Perception and Interaction in Simulated Environments

The quality of outdoor environment simulations is heightened by enabling perception of the projected imagery, closing the loop on the simulation architecture. More specifically, the combination of a projected simulated environment and sensors observing it creates a lab environment which is essentially a replacement for the outdoor world. Using MAR-CPS, noisy observations of the state space can be obtained and state observation probabilities $P(o|s', a)$ can be modeled directly in the lab space, as seen in Fig. 8. Since MAR-CPS uses real-time projections in the laboratory space, simultaneous perception of physical as well as virtual agents and environmental features is possible.

Communication and Teaching Tool for Spectators

Conveying valuable information about autonomous vehicle algorithms is not only useful for researchers, but for spectators outside the research field as well. In some scenarios, even

high-level descriptions of algorithms may prove difficult for spectators to understand. Though display of visual information or explanatory animations on a computer monitor may be effective, it can also detract from the experience of spectators as they must divide their attention between computer displays and physical experiments.

MAR-CPS is capable of showing latent or meta-information during demonstrations to spectators, which can be especially useful for transferring an intuitive understanding of the specific topic of research or specific mission scenarios. Information regarding a given vehicle's overall objective or messages declaring each vehicle's current task can be projected in MAR-CPS.

Vehicle Safety

MAR-CPS allows implementation of useful safety features for testing autonomous vehicles, and can aid compliance to regulatory restrictions placed on research institutions.

In scenarios where interaction of humans and autonomous hardware is dangerous (such as flight of high-speed quadrotors in an enclosed environment), physical barriers provide a means of protection for operators and spectators. However, minimization of vehicle crashes and collisions due to software or hardware failures is also desirable. Though planned trajectories and health states [23] of vehicles can be visualized on a computer display, it may be difficult for researchers to monitor such information while simultaneously observing the vehicles themselves. Using MAR-CPS, the above information can be projected directly on the vehicle testbed, allowing researchers to observe and even predict dangerous behavior and react accordingly with fast response time. Fig. 9 illustrates a scenario where a vehicle undergoes actuator damage and must

leave the mission premise until it is repaired. Using MAR-CPS, observers gain an understanding of such events without need for additional explanation.

The combination of motion capture technology and the visualization system also allows safe interaction of humans with robots in the augmented reality workspace. Fig. 10 shows a demonstration of multi-vehicle path-planning with humans present in the domain. In this demonstration, a team of quadrotors plan paths to randomly-generated goal destinations while avoiding collisions with each other and with a human wearing a motion-capture helmet. Using this setup, the perceived position of humans in the experiment domain can be directly projected underneath them, allowing safe interaction with the quadrotors.

Regulations

In some scenarios, experiments involving autonomous vehicles cannot be conducted in a public or outdoor setting, due to regulatory restrictions. For instance, although FAA regulations allow limited testing of UAS in outdoor settings, imposed requirements such as operator UAS certification may be over-constraining in some cases [1]. Using MAR-CPS, institutions can instead leverage existing private, indoor laboratory spaces to conduct simulated outdoor experiments without violating regulations.

Application Focus

This section details two robotics domains where MAR-CPS increases understanding of complex planning and control algorithms. The first domain involves a high-speed quadrotor performing difficult maneuvers, and presents a researcher perspective on the algorithm prototyping advantages offered by MAR-CPS. The second domain is a multi-agent decision-making system situating teams of competing robots in a surveillance environment. This example focuses on challenges in demonstrating complex robotics algorithms to spectators with limited background in this research field.

Researcher Focus: Algorithm Insights using Augmented Visualization

Sustained high-speed flight of quadrotors has not been extensively studied because of outdoor flight regulations and indoor space limitations. A thorough understanding of how aerodynamic effects and large attitudes impact the flight controller's performance is important since many of the projected applications of these vehicles place them within the high-speed flight regime. In MAR-CPS, some of the challenges of aggressive turning at high-speeds are addressed by studying the vehicle's performance during a $1\ \mathrm{m}$ radius circular trajectory at speeds greater than $4\ \mathrm{ms}^{-1}$. There were two main contributions of this work which benefited from the MAR-CPS visualization architecture. First, an improved nonlinear cascade-feedback control architecture that generated desired angular rates that were not dependent on small angle approximation [24] or hand-tuned gains [25] was shown to greatly enhance flight performance. Second, the vehicle's response was greatly enhanced by using a drag model and the geometric properties of a desired

trajectory for the feedforward command.

A major challenge in conducting controls experiments is concisely displaying data and error in real-time. For instance, Fig. 11 shows the vehicle banking at 60° during an aggressive turn. For high-speed maneuvers, it is difficult for researchers to extract information about position and altitude error in real-time, while simultaneously observing vehicles.

Traditionally, additional plots showing position, velocity, and attitude tracking would supplement the previous figure. However, MAR-CPS can display real-time flight data and thus provide all the necessary information to evaluate the vehicle's performance. Fig. 12a defines the visual elements used in the subsequent subplots. The blue circle is the desired position, the red circle is the actual position, the red arrow protruding from the red circle is desired heading, and relevant flight data is shown in the center. Fig. 12b shows the vehicle's poor performance when pure feedback control is used. The vehicle has a notable pitch of 30.2° because the lag between its actual and desired position is significant, shown by the offset between the blue and red circle. Further, the circular contours beneath the quadrotor indicate the history of its trajectory. Without MAR-CPS, it would be difficult to convey this error in real-time on the experiment platform.

When the feedforward model is added to the controller, the previous position lag is non-existent and the required pitch angle is reduced, as shown in Fig. 13a. A decrease in throttle is also observed, leading to an increase in control authority. Throttle is lowered since the vehicle no longer needs a large pitch. Note that for both experiments a steady-state roll angle greater than 50° is achieved. Lastly, Fig. 13b shows the vehicle can still track a desired trajectory near its maximum throttle setting.

The ability to display real-time data during control experiments with MAR-CPS is a valuable tool. It provides a gateway into what the vehicle is doing without having to analyze numerous plots. Researchers can use this as a debugging tool and also easily compare the results of different experiments. Display of such visuals also reduces the post-processing time following an experiment.

Spectator Focus: Aiding Understanding of Complex Missions

Due to its modular nature, MAR-CPS can be used in large-scale multi-agent planning problems. An example application is intruder monitoring using a team of quadrotors [26]. In this problem, a team of autonomous ground vehicles attempts to reach a goal location in a discretized world while a competing team of quadrotors attempts to push or "herd" them away.

In this problem domain, the quadrotors solve a planning problem where they first locate ground vehicles using a simulated radar system, use the stochastic state transition model $P(s'|s, a)$ to predict each ground vehicle's most likely next state, and use this information to choose which ground vehicles to focus their efforts on. Joint actions involving 2 quadrotors herding ground agents are more effective, but consume resources. Decisions related to health management (such as refueling or requests for repair) are also made autonomously by the quadrotors.

MAR-CPS is used as a visualization platform in this application. Fig. 14a illustrates the domain with no projected visualizations, with a team involving 1 physical and 1 virtual quadrotor. Without use of MAR-CPS, only the physical quadrotor is visible to spectators, making it difficult

to understand the underlying decision-making sequence. In this setting, the target destination for the ground vehicles is not visible (although could be indicated by a physical marker in the scene). Finally, there is no indication of the task each vehicle is conducting, nor any indication of whether the ground vehicles have been detected by the quadrotors. The experiment domain makes no indication of whether the planning algorithm is running in a continuous or discrete state space. This view provides spectators with a limited, high-level understanding of the experiment. For deeper understanding, underlying domain mechanisms must be explicitly conveyed to each spectator, as well as step-by-step descriptions of each vehicle's actions over the course of the experiment.

Next, the domain is incrementally augmented with elements enhancing both visual complexity as well as improving spectator understanding. Since the state of ground vehicles is initially unknown to the quadrotors, satellite imagery is projected to convey a sense of 'camouflaged ground intruders' to spectators (Fig. 14b). This imagery could be modified to highlight pathways of interest or varying terrain types as well.

In many experiments, the vehicle must localize itself within an unknown environment. Thus, the physical state of the vehicle does not always match its 'perceived' state. In MAR-CPS, the perceived state of the vehicle (or mean of the belief state) can be projected directly in the hardware domain to indicate discrepancies with actual vehicle location. This is done using projected crosses underneath vehicles in Fig. 15a. An additional benefit of this visual element is that virtual vehicles can also be indicated, as shown in the center of Fig. 15a. Now, the domain which spectators initially perceived to include only 1 quadrotor has transformed to include an

additional virtual agent. Actions involving cooperation between quadrotors make intuitive sense to spectators as a result. Another advantage of MAR-CPS is that existing visual elements can be leveraged to efficiently convey additional latent information in the experiment. The crosses in Fig. 15a not only indicate position, but also the health state of each quadrotor. One of the arms of the virtual quadrotor is red, indicating low health. The color of the crosses also indicate the current task for each quadrotor (yellow indicating surveillance, blue indicating a "herding" task). Therefore, by looking at a given quadrotor's cross, spectators can quickly infer the vehicle's perceived location, health state, and current task.

The sensor footprint of a given vehicle can be visualized as seen in Fig. 15b, which adds the radar footprint of the physical quadrotor. This visual also adds the underlying discretized state space of the domain using grid cells. Grid cells which have been visited by a surveying quadrotor are removed, with the remaining represented as foggy regions. A yellow grid square is also added to indicate the goal destination of the ground vehicles. Using these visuals, spectators gain an understanding of the competing decisions made by the aerial and ground teams.

Fig. 16a adds a visual element indicating tracking of ground vehicles. A ground vehicle which has been previously detected by a surveillance quadrotor is given a highlighted color. This creates an easily-discernible contrast between detected and undetected ground vehicles.

The effectiveness of the quadrotors' planning algorithm is also indicated in real-time using MAR-CPS. As Fig. 16b illustrates, green arrows protruding from each ground vehicle indicate their most likely next state, while black arrows indicate their most likely next-next state had the quadrotors not been there. Likewise, white arrows indicate their next-next most likely state

given the quadrotors' current positions (Fig. 16b). These serve as real-time visualizations for the transitions of vehicle state, $P(s'|s, a)$, giving spectators an immediate and intuitive understanding of the impact of the quadrotors' actions on the overall objective of keeping the ground vehicles away from their target destination.

Figures 17a-17c indicate various stages of the full experiment as it would appear to spectators. In Fig. 17a, the experiment begins with 2 physical quadrotors and 1 virtual quadrotor taking off and initiating surveillance. In Fig. 17b, the quadrotors have surveyed most of the domain and detected the top-left ground vehicle (in purple). Finally, in Fig. 17c, the experiment nears conclusion with all quadrotors performing "herding" of the ground vehicles away from their target destination (yellow square).

Using MAR-CPS, visual elements representing various aspects of the mission domain can be incrementally added to the laboratory space. The result is an easy-to-understand representation of the complex underlying domain for spectators.

Additional Case Studies

This section presents a selection of additional case studies which demonstrate benefits of MAR-CPS in research and communication settings. MAR-CPS was designed to be generalizable enough to be applicable to a variety of research topics and experiments, such that it can be a standardized testing and prototyping environment for CPS. Numerous experiments running autonomous teams of ground and air vehicles have already been conducted in MAR-CPS. The system has been tested in scenarios such as forest fire management (Fig. 18) and multi-

agent intruder monitoring (Fig. 17). In each case, the projector visualization is used to improve understanding of underlying behaviors using information such as vehicle position, health state, and viability of future actions.

Visualization and Perception of a Dynamic Forest Fire

MAR-CPS allows researchers to construct dynamic mission environments, visualize them in a laboratory, use onboard sensors for perception of environment features, and validate complex planning algorithms.

The above capabilities were demonstrated in a heterogeneous multi-agent learning setting, with an application to forest fire management [27]. In this work, a discretized 12×30 forest environment consisting of varying terrain and vegetation types (such as trees, bushes, rocks) was constructed and projected in MAR-CPS. A particle system was used to simulate fires, with OpenGL blending functions used to render dynamic fire effects. Seed fires of varying intensities were initiated on the terrain. A fire propagation model (from [28]) was used for dynamically updating the intensities and distribution over the terrain. A quadrotor used an onboard camera (Sony 700 TVL FPV Ultra Low Light Mini Camera) to wirelessly transmit analog video frames to a perception CPU, which created a segmented panorama of the complete forest environment. Fig. 18 shows a perspective view of the MAR-CPS environment, as well as associated image capture obtained from the quadrotor. The radio transmitter used to stream the onboard camera frames added noise and transmission artifacts to the resulting images, which in this case was desirable since noisy measurement models were being investigated. If a more robust image

transmission scheme, such as Wi-Fi, was utilized, then artificial noise could be injected either directly into the projection or during post-processing.

Hue-saturation values of the images were used for classification of fire intensities throughout the environment, resulting in an intensity matrix. Repeated applications of this process produced spatio-temporally varying intensity matrices, from which state transitions of fire intensity distributions were modeled. This information was used to predict future fire propagation, allowing targeting of firefighting efforts on more constructive regions of the environment.

This experiment highlights the use of MAR-CPS to create dynamic counterparts of real-world situations in an augmented reality environment, where noisy measurement systems can be used to assess the real-world effectiveness of CPS prior to deployment.

Real Time Vision-based Path Planning

MAR-CPS allows testing of machine vision systems using noisy data obtained from real-world sensors, enabling numerical modeling of observation noise. This feature of the augmented reality laboratory space distinguishes it from previous work in this field. This capability was demonstrated in a multi-agent planning experiment, shown in Fig. 19.

In this scenario, a ground vehicle navigates a stochastic obstacle course while a quadrotor provides overhead reconnaissance. The ground vehicle traverses over various grids in the domain, each dictating different dynamical model constraints, while planning a minimum-time path to a goal destination. The vehicle speed is nominal on white grids, 200% of nominal on green

grids, 50% speed of nominal on red grids, and 5% of nominal on black grids. On blue grids, the vehicle loses turning capability. All sensing capabilities are provided by the quadrotor, which has an onboard camera. Photographs of the grids are taken by the quadrotor and a belief-map of the world state is compiled, allowing real-time path planning for the ground agent.

Experiments conducted in this domain highlight the advantage of using MAR-CPS for doing data processing on noisy, real-world measurements. Visualizations were restricted to simple color grids to allow better analysis of the impact of camera noise on the perceived world map. Despite the grid colors being discrete and easily distinguishable by the human eye, analyzing hue-saturation-value plots revealed a notable level of noise in the data (Fig. 20) due to lighting condition and sensor noise. Calibrations similar to those which would need to be done in an outdoor flight test were conducted using these noise statistics, improving the accuracy of the perceived map.

Modeling Dynamic Agents

In many applications, autonomous vehicles operate in dynamic environments that are populated with other mobile agents, such as pedestrians, cyclists, and automobiles. For safe navigation in such environments, it is important to understand the other agents' intentions; that is, to predict their future behavior patterns (trajectories). These behavior patterns can be learned from past observations (a set of trajectories) by statistical inference, such as using Gibbs sampling on a Dirichlet Process Gaussian Process (DPGP) mixture model [29].

More precisely, in a set of N trajectories, each trajectory t^i is a sequence of position

measurements $\{(x_1, x_2), \ldots, (x_l, y_l)\}$ taken at fixed time interval Δt. There exist M behavior patterns, b_m, unknown a priori and modeled as pairs of Gaussian Processes (GPs), which specify the 2D velocity flow fields for predicting an agent's future motion (speed and trajectory). Each behavior pattern consists of two GPs (for each trajectory dimension, x and y), characterized by observations $\{t^k\}$ (a subset of trajectories from the dataset) and hyperparameters θ^{GP}_{xm}, θ^{GP}_{ym}. The following generative model specifies the probability of generating a trajectory t^i from the m-th behavior pattern,

$$\begin{aligned} l(b_m; t^i) &= p\left(t^i | z_i = m, b_m\right) \\ &= p\left(\frac{\Delta x_{1:l}}{\Delta t} \,\middle|\, \{t^k : z_k = j\}, \theta^{GP}_{xm}\right) p\left(\frac{\Delta y_{1:l}}{\Delta t} \,\middle|\, \{t^k : z_k = j\}, \theta^{GP}_{ym}\right), \end{aligned} \tag{1}$$

where the cluster assignment $z_k \in \{1, \ldots, M\}$ is a categorical random variable specifying which behavior pattern k-th trajectory belongs to. The terms on the right of Eq. (1) are the GP likelihoods [30], with other trajectories currently assigned to this behavior pattern as observations. Further, a Dirichlet Process prior is placed on the cluster assignments,

$$p(z_i = j | z_{-i}, \alpha) = \frac{n_j}{N - 1 + \alpha} \tag{2}$$

$$p(z_i = M + 1 | z_{-i}, \alpha) = \frac{\alpha}{N - 1 + \alpha}, \tag{3}$$

where $n_j = \sum_{k, k \neq i} 1(z_k = j)$, is the number of trajectories currently assigned to cluster j, N is the number of trajectories in the dataset, and α is a concentration parameter [29]. Eq. (2) is the probability of assigning label i to an existing cluster j, and Eq. (3) is the probability of assigning label i to a new cluster. Combining Eq. (1) and Eq. (2), the probabilities of assigning

a trajectory t^i to an existing motion pattern b_m and a new motion pattern b_{M+1} are

$$p\left(z_i = m | t^i, \alpha, b_m\right) = l(b_m; t^i) p(z_i = j | z_{-i}, \alpha) \tag{4}$$

$$p\left(z_i = M+1 | t^i, \alpha, b_m\right) = \int l(b_m; t^i) d\theta_x^{GP}, d\theta_y^{GP}\, p(z_i = M+1 | z_{-i}, \alpha). \tag{5}$$

The goal of the inference process is to find the number of clusters M, the cluster assignments z_i for each of the N trajectories, and the hyperparameters $\theta_{xm}^{GP}, \theta_{ym}^{GP}$ for each of the M behavior patterns. This learning process is typically carried out off-line using Gibbs sampling.

During the online planning phase, upon detecting a mobile agent in the environment, the autonomous vehicle can predict the mobile agent's intention by (i) determining which behavior pattern the mobile agent is currently following, and (ii) predicting the mobile agent's future trajectory using the GP model. Step (i) can be computed in a probabilistic sense using Eq. (5). More sophisticated methods have also been developed to detect changes in an agent's intention [31]. Step (ii) can be carried out by propagating the current observation forward in time using the GP model, thus generating a trajectory into the future.

Fig. 21 illustrates a hardware demonstration of the above process, involving a ground agent (circled in green) planning a trajectory to a goal destination (yellow circle projected on the floor) while avoiding collisions with two pedestrian vehicles. Trajectories protruding from the pedestrian vehicles indicate projected paths that the ground agent believes they might take. This can be interpreted as the belief of paths they plan on taking (although not in the POMDP sense). In Fig. 21a, the ground vehicle plans a short trajectory (shown as a green line), terminating before the goal destination because the agent's belief in the pedestrians' future trajectories indicates

that a collision may occur if it continues a path towards the goal. After the agent waits several seconds, it obtains further observations of the pedestrians, allowing it to modify its belief of their paths and plan a trajectory to the goal (Fig. 21c).

Through trajectory and belief information (as in Fig. 21b and Fig. 21d), MAR-CPS allows real-time visualization of planning algorithms directly on the hardware testbed. In this specific instance, the experiments conducted also reveal the usefulness of MAR-CPS for hardware debugging. Prior to MAR-CPS, trajectory validation was being conducted separately from the hardware testbed, on a monitoring computer. For some time, the performance on hardware was found to be notably lower than simulated performance. Upon visualization using MAR-CPS, the cause was found to be a hardware calibration issue causing pedestrian vehicles to not follow trajectories accurately. The simultaneous visualization of planned and executed trajectories on hardware test-beds are highly useful for debugging such issues.

Motion Planning Under Uncertainty

Robot motion planning is the problem of navigating a moving robot from a start location to a goal location. In real-world applications, often a robot's motion and its sensory measurements are subject to noise. To plan motions for a robot under uncertainty, the current configuration of the robot based on noisy measurements needs to be inferred. The result of such an inference is a probability distribution over all possible configurations of the robot, which is captured by the belief state $b(s)$ [32]. With MAR-CPS, these probability distributions can be projected on the physical environment alongside the robots. Moreover, the "intent" of the robot (the trajectory it

decides to follow or the task it aims to complete) can be projected to the physical environment.

Usage of MAR-CPS in a motion planning under uncertainty domain has been demonstrated [31]. Fig. 22 includes a few photographs of this demonstration. A chance-constrained rapidly exploring random tree (CC-RRT) method is used to plan the robot's motion from its current location to goal (shown in yellow) in the presence of moving obstacles (people or smaller robots such as iRobot Creates). The robot's perception of these moving/static obstacles is projected onto the ground in purple. The generated tree for planning is also projected in green, where the best trajectory is highlighted. This application demonstrates usage of MAR-CPS in an autonomous driving scenario, where visualization of planned paths are useful for debugging and calibration purposes. By visualizing all the considered paths of the robot, MAR-CPS provides researchers insight into the reasoning behind the robot's executed trajectory. The contrast between planned and executed trajectories is especially useful in chance-constrained path planning, since the robot considers uncertainty in obstacle positions while simultaneously choosing a reasonably short path. Details involved in this consideration may be difficult to understand on a computer monitor, since the observer must perform a mental transformation of the display onto the space of the hardware experiment. With MAR-CPS, no such mental transformation is required, as the projected visual is automatically-aligned onto the hardware domain.

A multi-agent sequential convex path planning algorithm has also been implemented in MAR-CPS [33]. Planned paths for a team of two physical and two virtual quadrotors were visualized using the projection system, with the vehicles' states being displayed and updated in real-time (Fig. 23). Comparisons between planned and actual vehicle trajectories are made

easier using the augmented reality space, which is especially useful for large teams of high-speed vehicles.

Human-Robot Interactivity

Taking advantage of the motion capture system used in MAR-CPS, human-robot interactivity can also be demonstrated. Specifically, tracking sensors can be placed on a human to allow interaction with autonomous vehicles from a safe distance. Props can be used for representation of objects within the simulated world. One demonstration involves a quadrotor landing due to a simulated onboard fire (see Fig. 24a), and a human operator subsequently using a water spout prop to quench the vehicle (see Fig. 24b). Simultaneously, the projection system in MAR-CPS displays both fire and water animations using a particle system, conveying the impact of the interactive process to spectators. Another example involves human spectators with motion capture helmets 'intruding' a domain where virtual vehicles plan paths around them (Fig. 10b). This allows path-planning in uncertain environments, while guaranteeing no danger of collisions with the human obstacles. Such examples of human-robot interactivity allow demonstrations of scenarios which would otherwise not be possible to perform in an enclosed laboratory space.

Visualizing Multi-agent Communication Links

A domain involving visualization of communication links in a multi-agent setting has been developed (Fig. 25a). In this experiment, a ground vehicle (the 'leader') contains a communications beacon that can be used to assign tasks to a fellow agent (a 'follower'). Task

assignment only occurs when the agents are within the beacon's communication range. MAR-CPS allows visualization of this range, as well as the moment at which the communication link-up occurs (Fig. 25b). After link-up, the 'follower' agent immediately begins its assigned task (for instance, movement to a target destination).

Task allocation using the Hybrid Information and Plan Consensus algorithm [34] has been visualized in MAR-CPS (Fig. 26). In this demonstration, 6 agents are servicing tasks (shown as colored crosses in Fig. 26) throughout the domain. Orange lines indicate local information being shared between agents, and purple lines indicate the communication network. Once a task has been serviced, its indicator is made transparent. Using this visualization, spectators can easily observe previously-serviced tasks at a glance, while also being informed of communication and decision-making networks amongst teams of agents. Such information would otherwise be invisible to spectators.

Challenges and Limitations

In practice, certain domains may require special consideration when integrated into MAR-CPS. Domains which require visual feedback of the simulated environment from onboard vehicle cameras must be careful not to corrupt this data by including latent state information in the image frame. In such a case, the vehicle state with respect to the ground can be obtained using motion capture tracking and combined with the camera intrinsic parameter matrix to calculate the camera's field of view in real-time. This field of view is then used to define a 'no-state-visualization' zone within MAR-CPS. All latent state or belief visualizations are then displayed

outside this footprint. Fig. 7a shows an example of this type of visualization, where the vehicle health belief is shown as a donut chart, allowing the center of the camera frame to observe the simulated environment underneath the quadrotor.

The above method was found to work well for sparse visuals which follow the vehicles. This idea can be modified for denser visuals, such as a particle field atop a simulated environment, using a layered visualization approach. To do so, the environment layer is rendered underneath the particle field layer, and the camera footprint is used to mask the particle layer out of the final rendered image. For domains where this is not possible (specifically, when environment visualization is needed for the entire laboratory space), one alternative is to run experiments with the state visualization layer disabled, record vehicle trajectories using the motion capture system, and then re-run the experiment using the recorded trajectories (but this time with state visualization enabled). Another idea for future consideration is the use of dual polarized projectors, with opposite-handed polarized lenses used for vehicle cameras and spectators, respectively. This would allow simultaneous display of simulated environments for vehicle cameras, while allowing display of high-density state information for spectators.

Another class of challenges stems from the transfer of outdoor experiments to indoor laboratories, where sources of uncertainty present in outdoor settings must be simulated indoors. Position estimation of an outdoor vehicle using GPS can be replicated indoors by injecting artificial noise into the motion capture system's measurements. This requires careful modeling of noise in the outdoor system, which may not be possible in some scenarios. A similar challenge occurs for vision-based systems. The close vicinity of vehicles to the MAR-CPS floor typically

results in fairly high-resolution photographs of the projected environment using onboard cameras. On real systems, photographs of distant objects can suffer from blur and noise. Some domains, therefore, might require modeling of such sources of uncertainty and artificial injection of blur and noise into the final projected environment.

Conclusion

Measurable Augmented Reality for Prototyping Cyber-Physical Systems (MAR-CPS) is an indoor architecture for testing and debugging of autonomous vehicles by providing a window into the underlying decision-making space. This work combines a motion capture system, ground projectors, autonomous vehicle platforms, and a communications network to allow researchers to gain a low-level understanding of the performance of perception, planning, and learning algorithms in real time. The work extends previous capabilities of MIT's RAVEN testbed to allow display of latent information and uncertainty, allow sensor-perception of simulated environments, and serves as a communication and teaching tool for spectators. Various experiments have been conducted using MAR-CPS, including forest fire management, planning for large-scale multi-agent systems, motion planning under uncertainty, and visualization of communication links between vehicles. Future work includes further investigation of applications in human-robot interactivity, as well as more complex applications in real-time perception and learning of the augmented reality environment.

Acknowledgments

This work is supported by Boeing Research & Technology. The authors gratefully acknowledge the anonymous reviewers for their insightful comments and feedback.

References

[1] F. A. Administration, "Press release – dot and faa propose new rules for small unmanned aircraft systems," 2015. Online `http://www.faa.gov/news/press_releases/news_story.cfm?newsId=18295`.

[2] K. J. Astrom, "Optimal control of Markov decision processes with incomplete state estimation," *Journal of Mathematical Analysis and Applications*, vol. 10, pp. 174–205, 1965.

[3] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," vol. 28, pp. 51–64, April 2008.

[4] D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control: A multivehicle platform for research in networked embedded systems," *IEEE Control Systems Magazine*, vol. 27, pp. 58–78, June 2007.

[5] G. Hoffman, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (starmac)," in *Proceedings of the IEEE Digital Avionics Systems Conference*, (Salt Lake City, UT), November 2004.

[6] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV

testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[7] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014.

[8] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *Control Systems, IEEE*, vol. 34, no. 4, pp. 46–64, 2014.

[9] E. N. Johnson and D. P. Schrage, "System integration and operation of a research unmanned aerial vehicle," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 1, pp. 5–18, 2004.

[10] A. Ulusoy, M. Marrazzo, K. Oikonomopoulos, R. Hunter, and C. Belta, "Temporal logic control for an autonomous quadrotor in a nondeterministic environment," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 331–336, IEEE, 2013.

[11] F. Ghiringhelli, J. Guzzi, G. A. D. Caro, V. Caglioti, L. M. Gambardella, and A. Giusti, "Interactive augmented reality for understanding and analyzing multi-robot systems," in *IROS*, 2014.

[12] F. Leutert, C. Herrmann, and K. Schilling, "A spatial augmented reality system for intuitive display of robotic data," in *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction*, HRI '13, (Piscataway, NJ, USA), pp. 179–180, IEEE Press, 2013.

[13] O. VR, "Oculus vr," 2014. Online `http://www.oculus.com/`.

[14] SteamVR, "Steamvr," 2015. Online `http://store.steampowered.com/`

universe/vr.

[15] M. Daily, Y. Cho, K. Martin, and D. Payton, "World embedded interfaces for human-robot interaction," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 5 - Volume 5*, HICSS '03, (Washington, DC, USA), pp. 125.2–, IEEE Computer Society, 2003.

[16] V. M. Systems, "Motion capture systems from Vicon," 2008. Online http://www.vicon.com/.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, 2009.

[18] S. D. Technologies, "Scalable display technologies," 2014. Online http://www.scalabledisplay.com/.

[19] C. Berger and B. Rumpe, "Engineering autonomous driving software," *Experience from the DARPA Urban Challenge*, pp. 243–271, 2012.

[20] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[21] N. K. Ure, G. Chowdhary, J. P. How, and J. Vian, *Planning Under Uncertainty*, ch. Multi-Agent Planning for Persistent Surveillance. MIT Press, 2013.

[22] I. K. N. Nigam, S. Bieniawski and J. Vian, "Control of multiple uavs for persistent surveillance: Algorithm and flight test results," vol. 20, pp. 1236–1251, September 2012.

[23] A.-a. Agha-mohammadi, N. K. Ure, J. P. How, and J. Vian, "Health aware stochastic planning for persistent package delivery missions using quadrotors," in *IEEE/RSJ International*

Conference on Intelligent Robots and Systems (IROS), (Chicago,IL), 2014.

[24] M. Cutler and J. P. How, "Actuator constrained trajectory generation and control for variable-pitch quadrotors," (Minneapolis, Minnesota), pp. 1–15, August 2012.

[25] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *Proceedings of the International Conference on Robotics and Automation, Seattle, WA, USA*, 2015.

[26] Y. F. Chen, N. K. Ure, G. Chowdhary, J. P. How, and J. Vian, "Planning for large-scale multiagent problems via hierarchical decomposition with applications to uav health management," (Portland, OR), June 2014.

[27] N. K. Ure, *Multiagent Planning and Learning Using Random Decompositions and Adaptive Representations*. PhD thesis, Cambridge, MA, February 2015.

[28] D. Boychuk, W. J. Braun, R. J. Kulperger, Z. L. Krougly, and D. A. Stanford, "A stochastic forest fire growth model," *Environmental and Ecological Statistics*, vol. 16, no. 2, pp. 133–151, 2009.

[29] J. Joseph, F. Doshi-Velez, A. Huang, and N. Roy, "A Bayesian nonparametric approach to modeling mobility patterns," *Journal of Autonomous Robotics*, vol. 31, pp. 383–400, 2011.

[30] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[31] S. Ferguson, B. Luders, R. C. Grande, and J. P. How, "Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions," in *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, (Istanbul, Turkey), August 2014.

[32] A.-a. Agha-mohammadi, S. Chakravorty, and N. Amato, "FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements," *International Journal of Robotics Research (IJRR)*, vol. 33, no. 2, pp. 268–304, 2014.

[33] Y. Chen, M. Cutler, and J. P. How, "Decoupled multiagent path planning via incremental seqential convex programming," (Seattle, WA), IEEE, May 2015.

[34] L. Johnson, H.-L. Choi, and J. P. How, "The hybrid information and plan consensus algorithm with imperfect situational awareness," (Daejeon, Korea), November 2014.

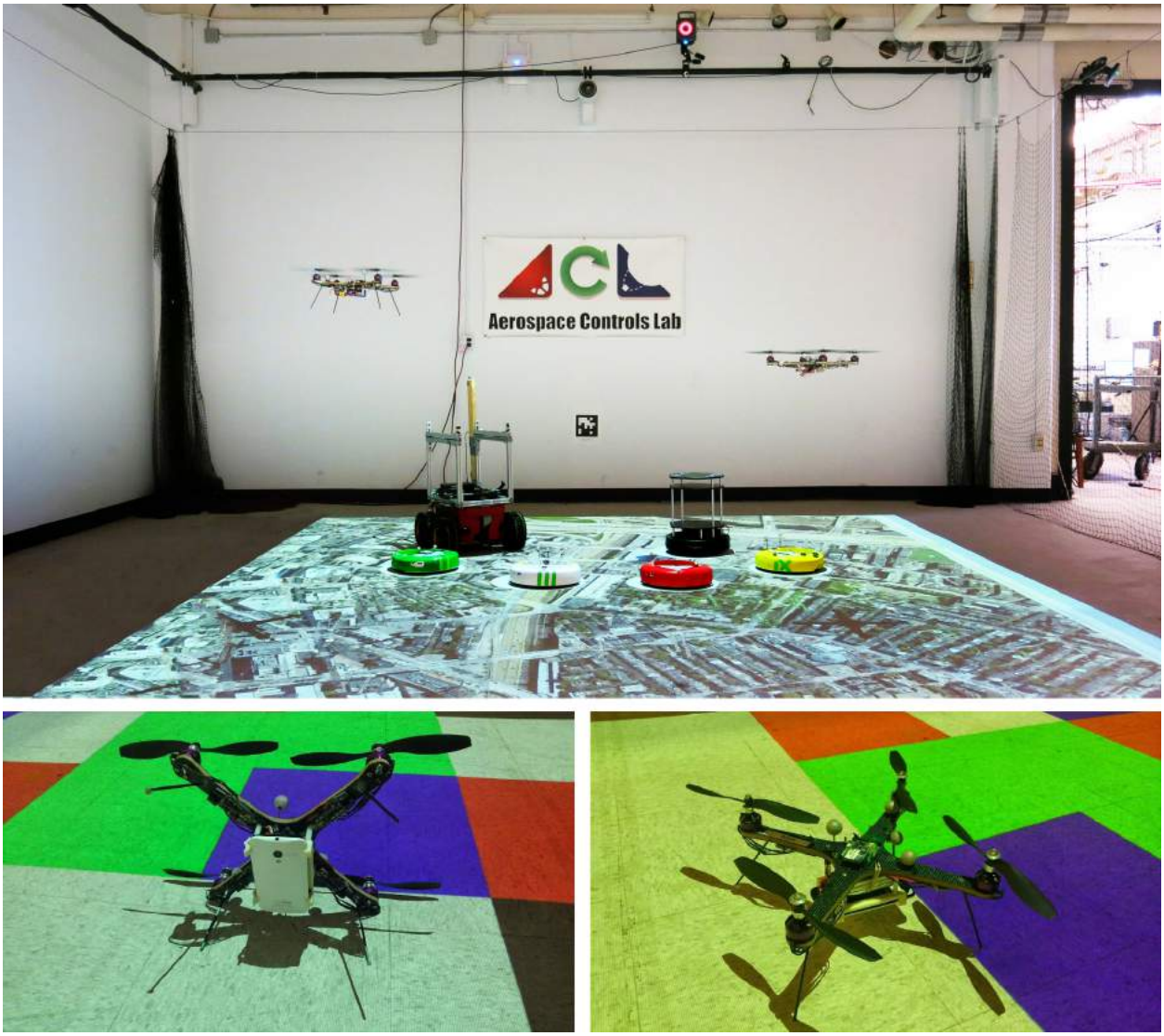


Figure 1: Components of Measurable Augmented Reality for Prototyping Cyber-Physical Systems (MAR-CPS). MAR-CPS includes physical vehicles and sensors such as cameras, a motion capture technology, a projection system, and a communication network. The role of the projection system is to augment a physical laboratory space with 1) autonomous vehicles' beliefs and 2) a simulated mission environment, which in turn can be measured by physical sensors on the vehicles.

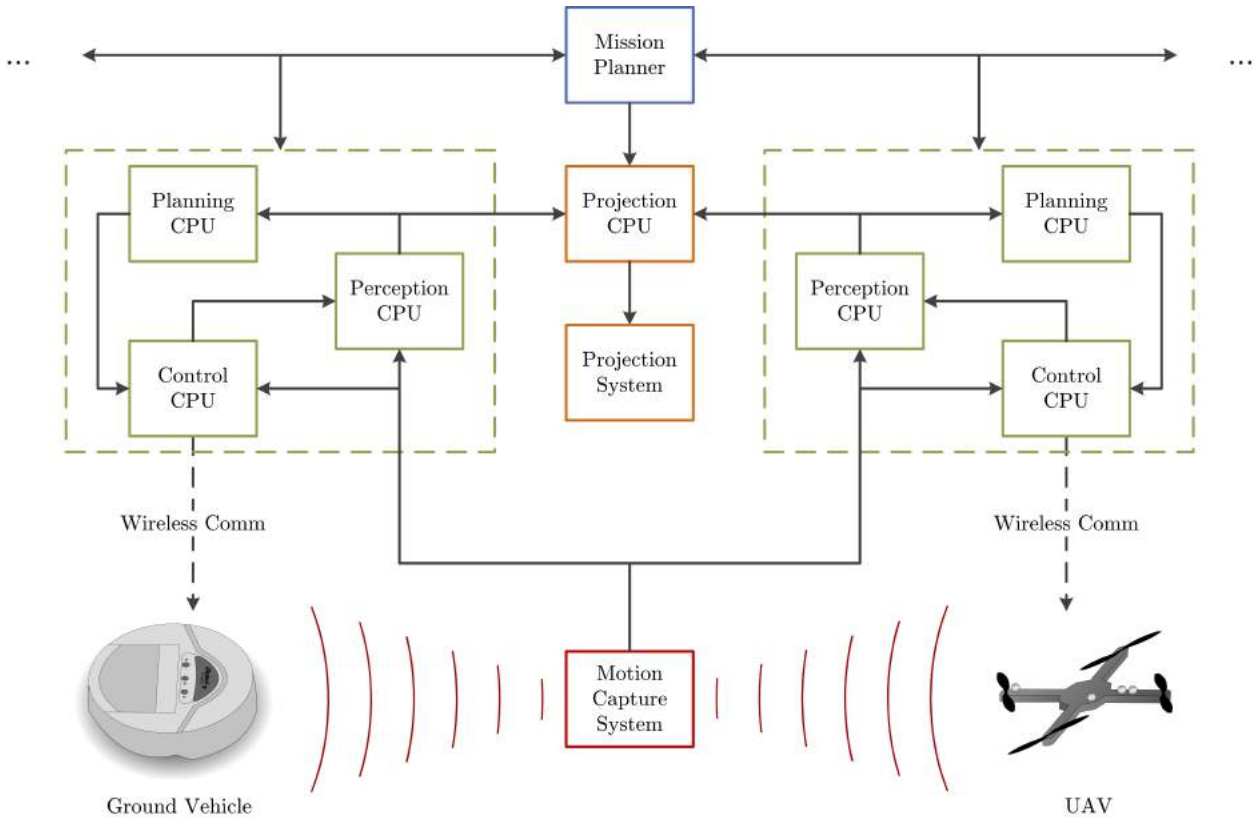


Figure 2: Architecture overview for MAR-CPS, for a mission involving both ground and air vehicles. Each vehicle communicates with a designated CPU for planning, perception, and low-level control. Given a task, the planning CPU defines a valid trajectory for the vehicle. The trajectory is relayed to a control CPU, which defines low-level control inputs to the vehicle using feedback from the motion capture system. Note that the planning CPU also has knowledge of the controllability of the vehicle in question, and its role can be combined with the control CPU if desired. The Control, Planning, and Perception CPU can be moved onboard vehicles.

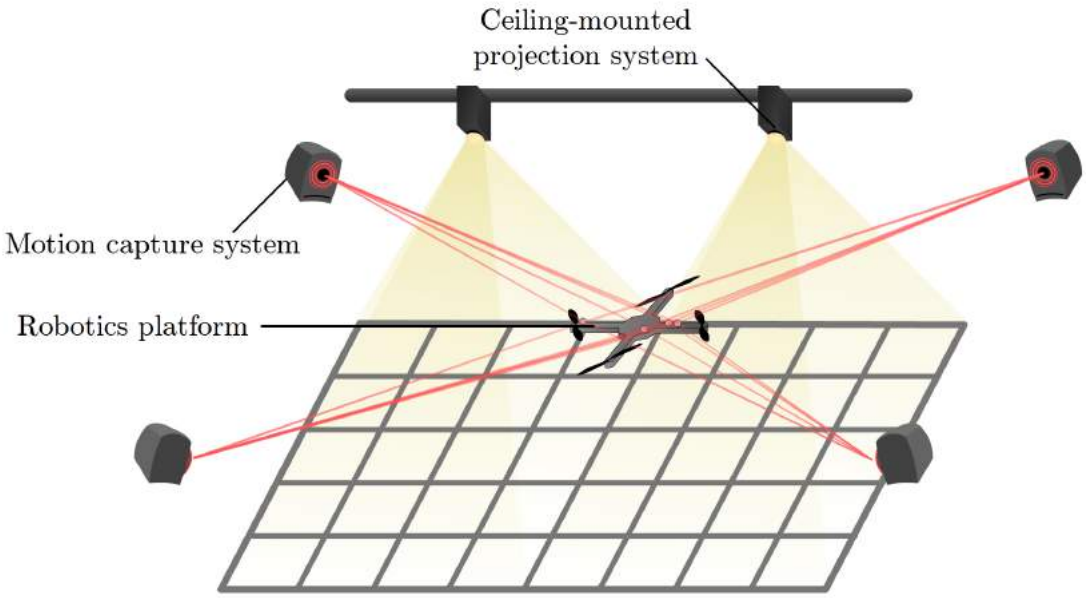


Figure 3: Hardware overview for MAR-CPS, involving autonomous vehicles, a motion capture system, and a ceiling-mounted projection system. The visualization system is implemented in MIT Aerospace Controls Laboratory's Real-time indoor Autonomous Vehicle test ENvironment (RAVEN) [3] flight testbed. This system uses 18 Vicon T-Series motion capture cameras allowing tracking of heterogeneous teams of autonomous vehicles [16]. A unique pattern of reflective motion capture markers is affixed to each vehicle, allowing the motion capture system to determine the position and orientation of the vehicles, as seen in Fig. 3. The motion capture system transfers each vehicle's pose to a projection processing CPU, which augments the laboratory space with environmental and mission information.

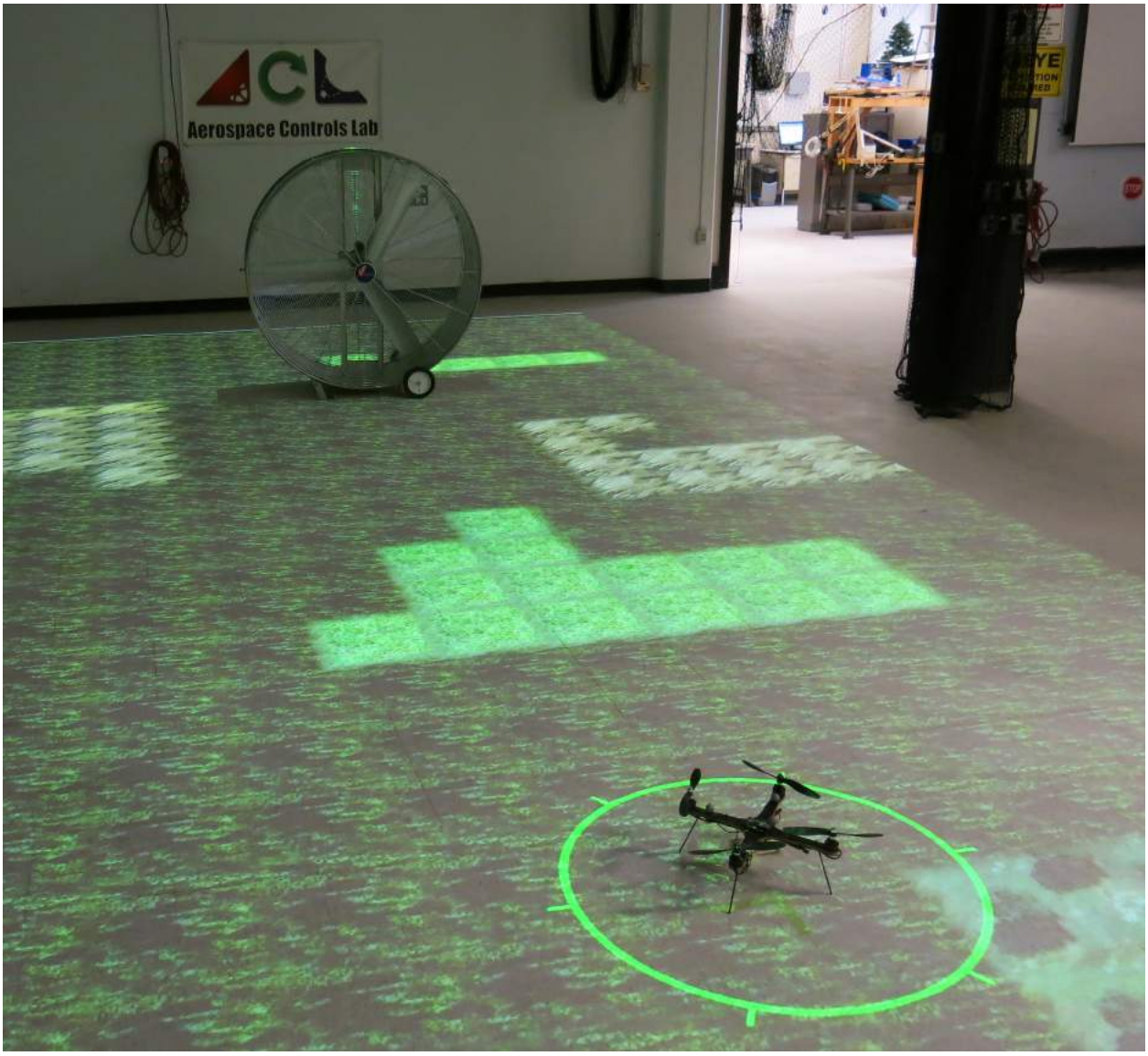


Figure 4: Props in MAR-CPS. Usage of physical props, such as fans, allows simulation of environmental factors within MAR-CPS.

(a) Pixel space to Vicon space mapping is performed using a reference wand to map a calibration grid's vertices to motion-capture coordinates. Delaunay triangulation is then used to perform the mapping.

(b) Calibration grid with reference wand coordinates projected.

Figure 5: Multi-projection system coordinate transform calibration. To map the pixel space to the Vicon space, a Delaunay triangulation is constructed in the pixel space using motion capture markers. Two piecewise-linear surfaces are constructed on top of the Delaunay triangulation with the height of the surface at each marker set to the x and y coordinates of the marker in the physical space respectively. A point in the pixel space can then be mapped to the Vicon space efficiently by querying the two piecewise-linear surfaces.

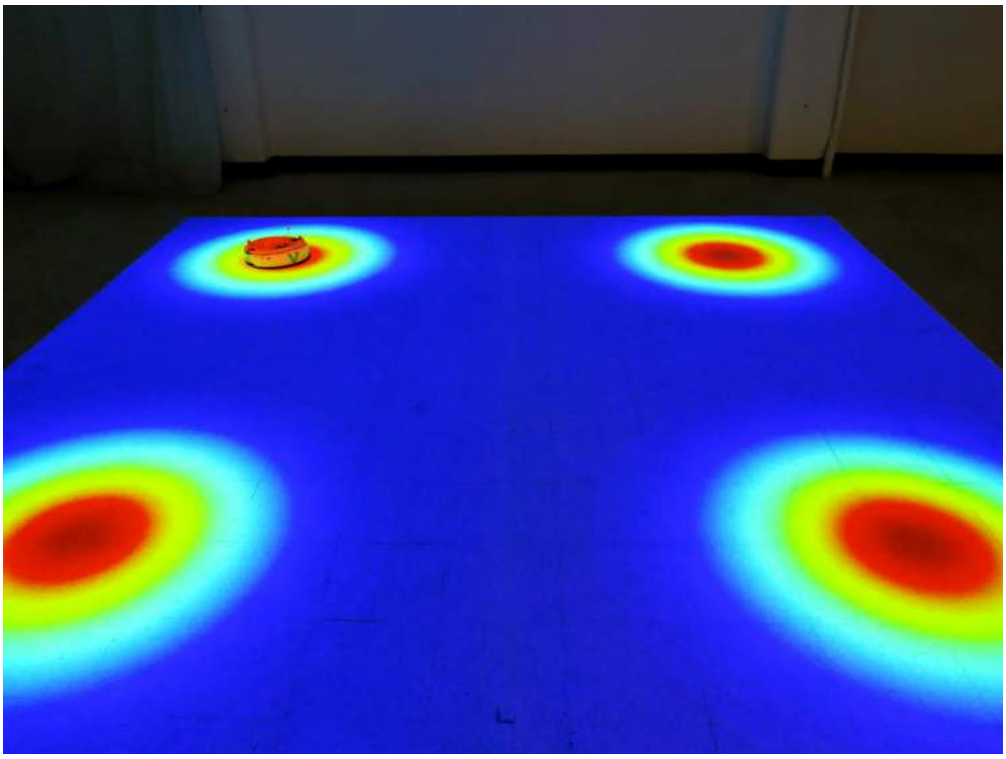
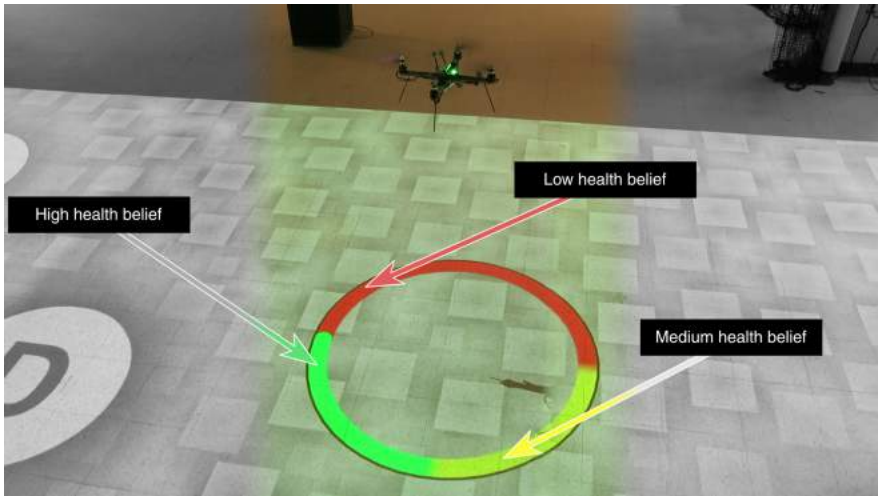
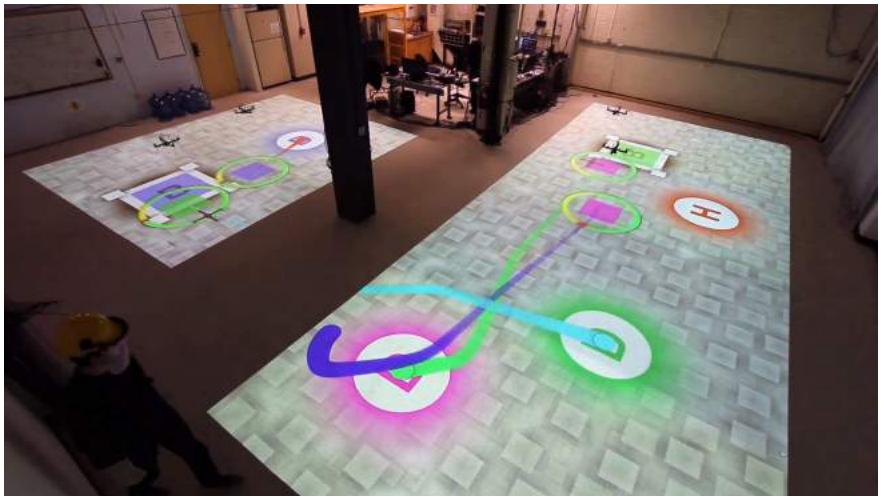


Figure 6: Belief space visualization for robot localization. In such domains, robots observe features of their surroundings in order to determine their location. However, observations are typically made using noisy sensors, leading to a posterior distribution on the robot's location. MAR-CPS allows visualization of this distribution. In the above, regions with high posterior probability for the robot's location are highlighted in red. The robot in this figure uses a Gaussian mixture model to represent belief on its location, and MAR-CPS prominently indicates the underlying multi-modal belief state. Using MAR-CPS, spectators can visualize and intuitively understand the concept of 'belief space'. They can also see the transformation of the robot's belief state in real-time (and with the correct scale in the laboratory) as it performs actions and gathers observations.



(a) In this domain, a quadrotor maintains a belief over its health state, which is partitioned into low, medium, and high health. As the quadrotor executes its mission, it uses noisy observations from a simulated health sensor to update its health belief, which is projected underneath it as a donut chart. The visualization follows the quadrotor and is dynamically updated as the mission progresses.



(b) Full mission domain with 4 quadrotors, each with an associated health belief. Also indicated are the planned paths for each vehicle, as well as labeled locations where each vehicle can perform specific tasks.

Figure 7: Quadrotor health belief visualization in MAR-CPS.

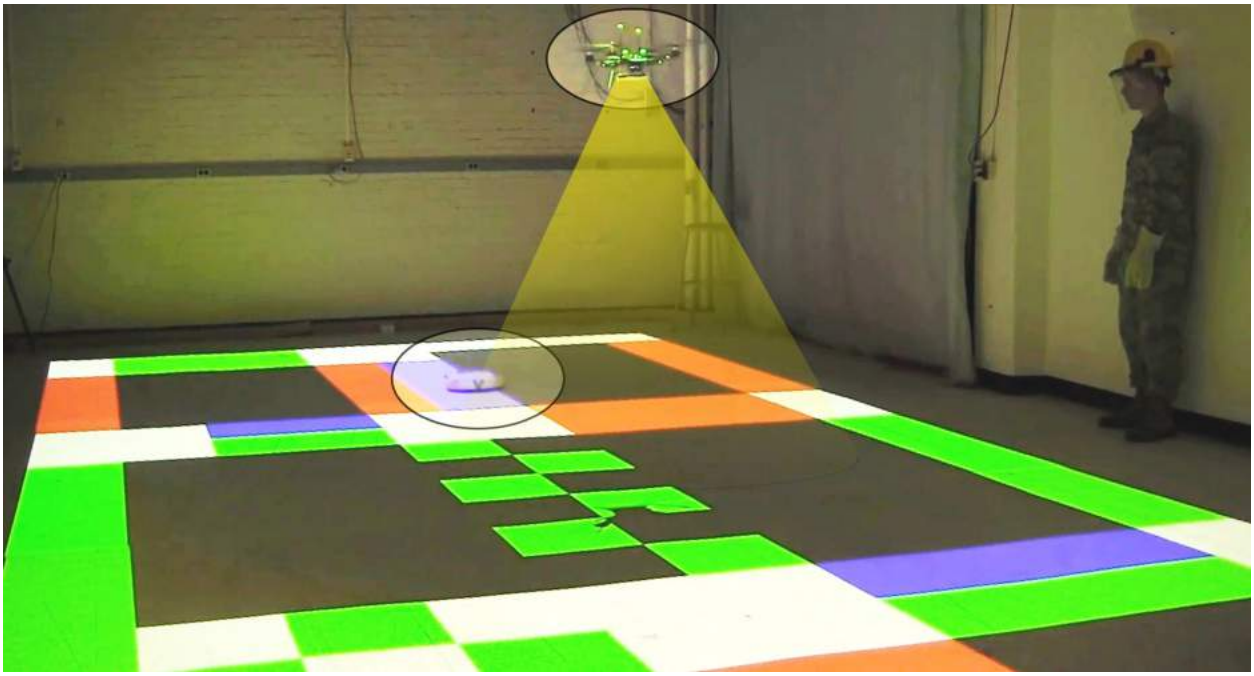


Figure 8: Closed-loop vision feedback in MAR-CPS. The combination of a projected simulated environment and sensors observing it creates a lab environment which is essentially a replacement for the outdoor world. Using MAR-CPS, noisy observations of the state space can be obtained and state observation probabilities $P(o|s', a)$ can be modeled directly in the lab space. In this demonstration, a quadrotor provides vision information to a ground robot navigating an obstacle course (note that the yellow field-of-view cone for the quadrotor camera was added in post-processing).

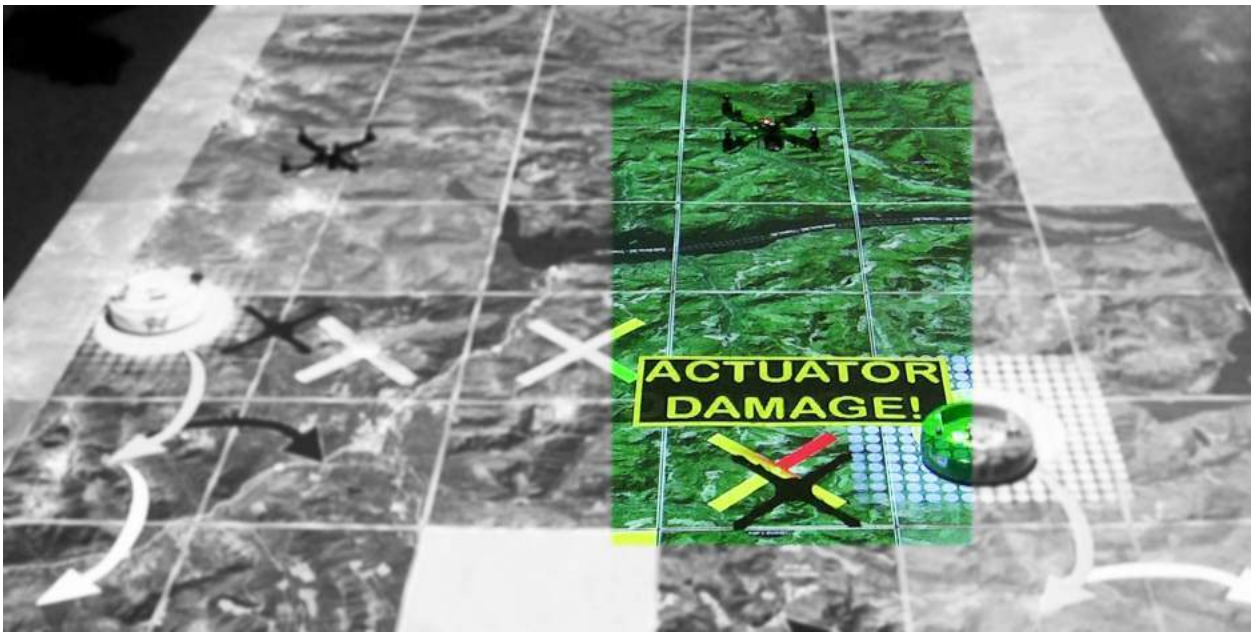
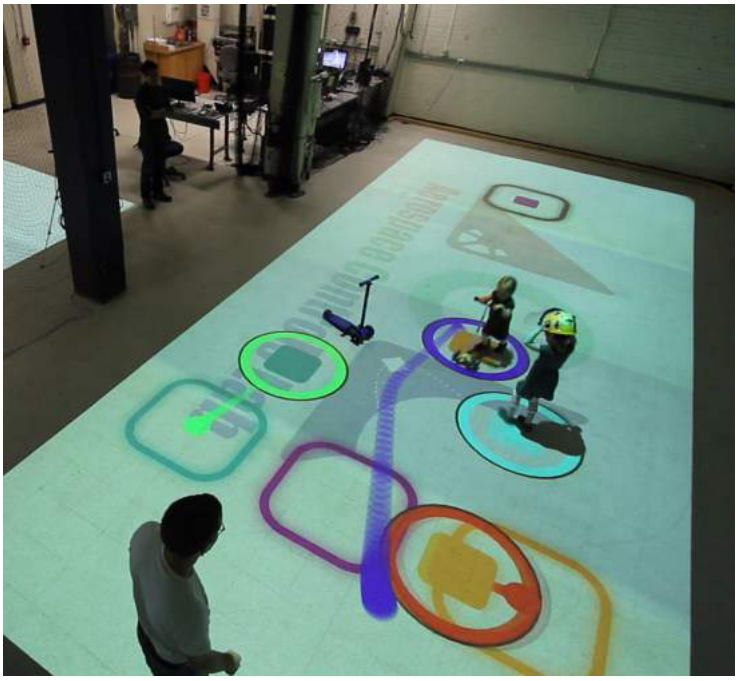


Figure 9: Vehicle health monitoring messages in MAR-CPS. The cross projected under each quadrotor indicates its pose. One of the arms of each cross is dedicated for representing vehicle health state (red for low health, in this example). In the above scenario, a vehicle undergoes actuator damage and must leave the mission premise until it is repaired. Using MAR-CPS, spectators gain an understanding of such events without need for additional explanation.



(a)

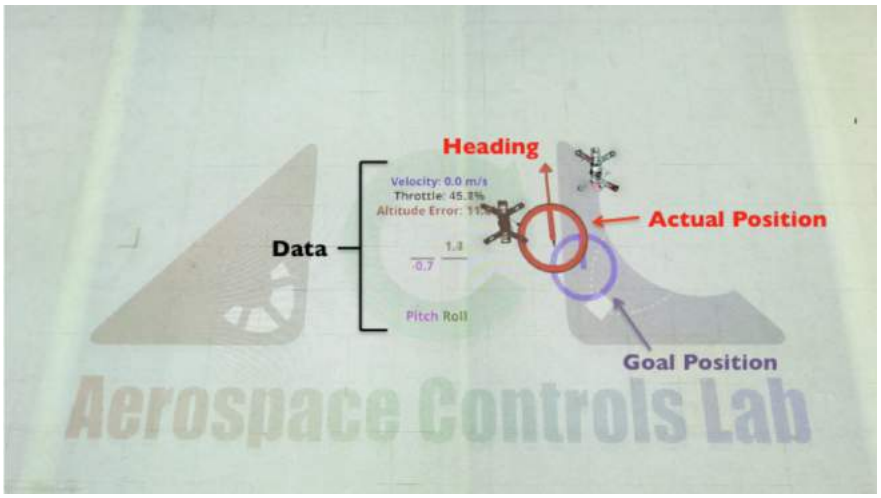
(b)

Figure 10: Human-robot interaction in MAR-CPS. A team of virtual quadrotors plan paths to randomly-generated goal destinations while avoiding collisions with each other and with a human wearing a motion-capture helmet.

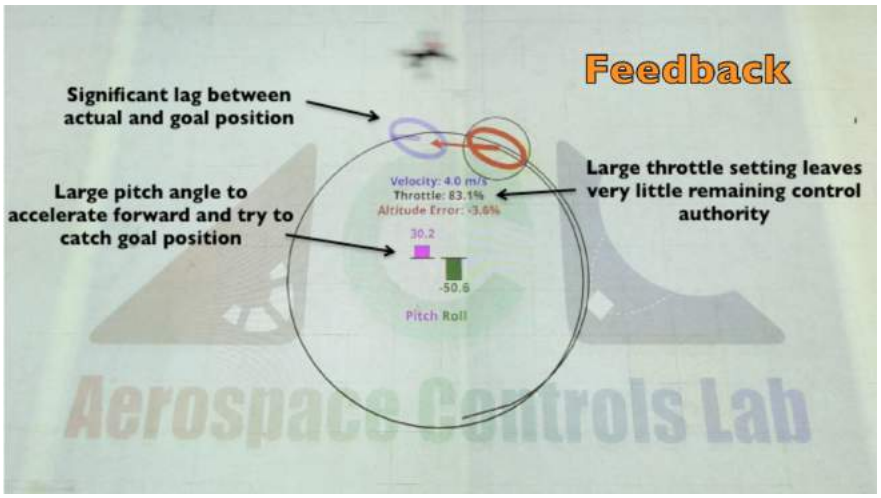
(a)

(b)

Figure 11: Photographs of quadrotor banking 60° during aggressive turn. For high-speed maneuvers, it is difficult for researchers to extract information about position and altitude error in real-time, while simultaneously observing vehicles.

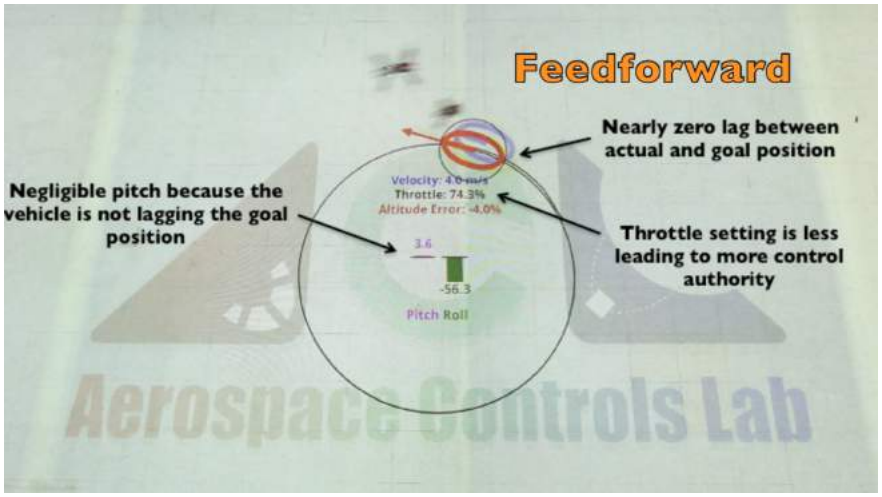


(a)

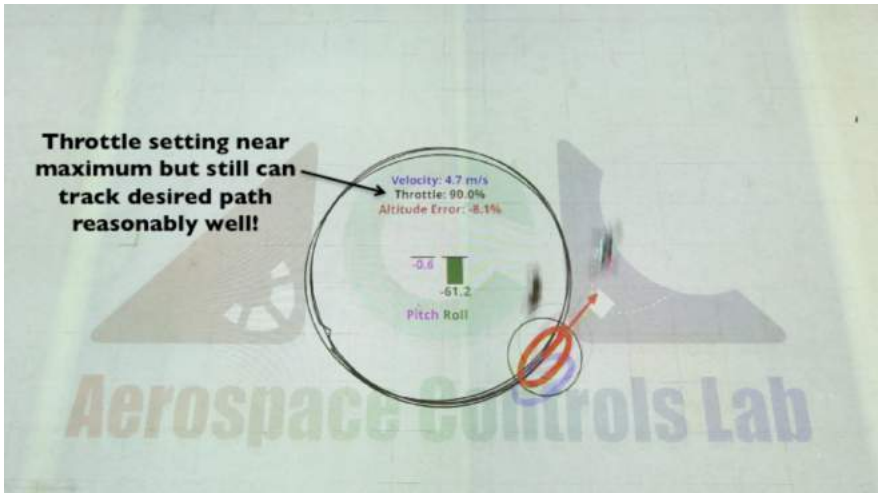


(b)

Figure 12: Photographs of MAR-CPS displaying real-time data during aggressive turn experiments. The blue circle is the desired position, the red circle is the actual position, the red arrow protruding from the red circle is desired heading, and relevant flight data is shown in the center.

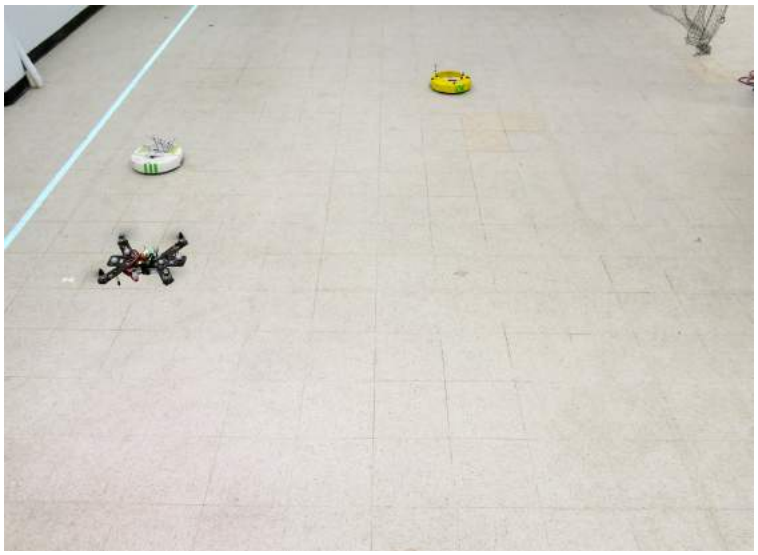


(a)

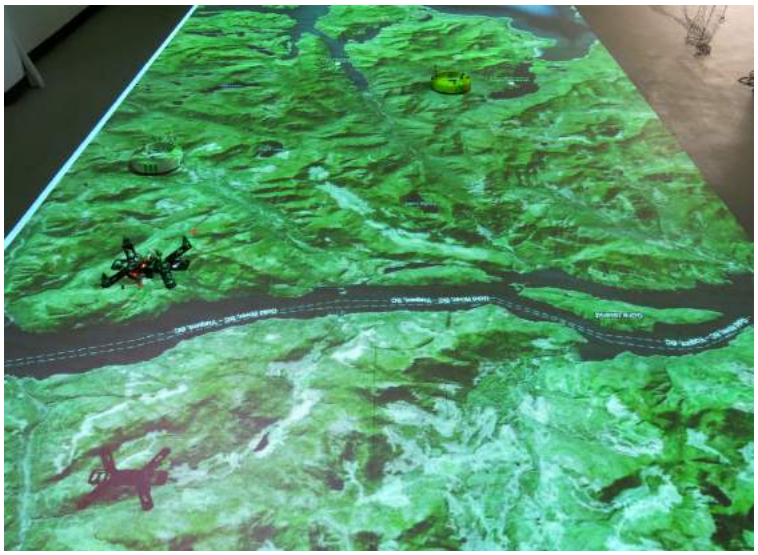


(b)

Figure 13: Photographs of MAR-CPS displaying real-time data during aggressive turn experiments. The vehicle has a notable pitch of 30.2° because the lag between its actual and desired position is significant, shown by the offset between the blue and red circle. Further, the circular contours beneath the quadrotor indicate the history of its trajectory. Without MAR-CPS, it would be difficult to convey this error in real-time on the experiment platform.

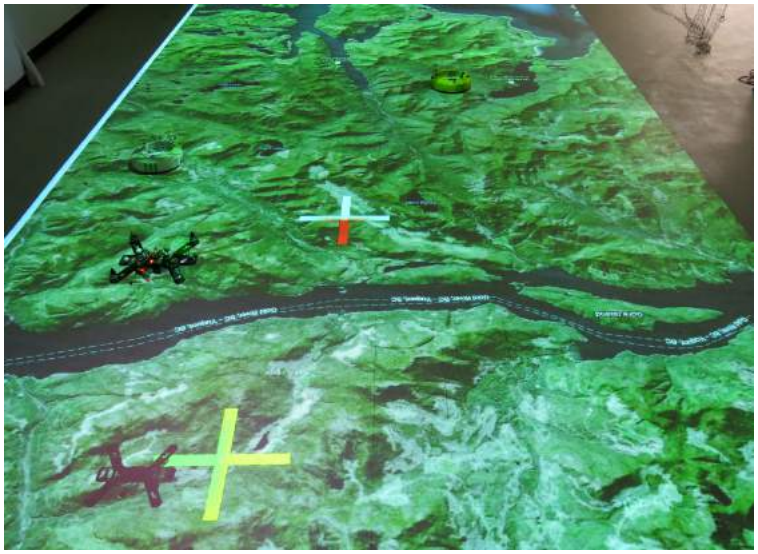


(a) The domain with no projected visualizations, with a team involving 1 physical and 1 virtual quadrotor. Without use of MAR-CPS, only the physical quadrotor is visible to spectators, making it difficult to understand the underlying decision-making framework.

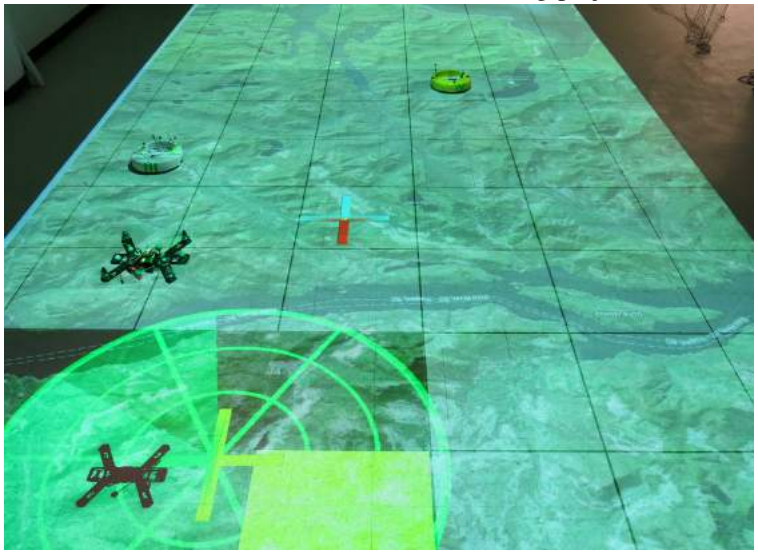


(b) Since the state of ground vehicles is initially unknown to the quadrotors, satellite imagery is projected to convey a sense of 'camouflaged ground intruders' to spectators.

Figure 14: Sequential addition of visualization elements in multi-agent intruder monitoring domain.



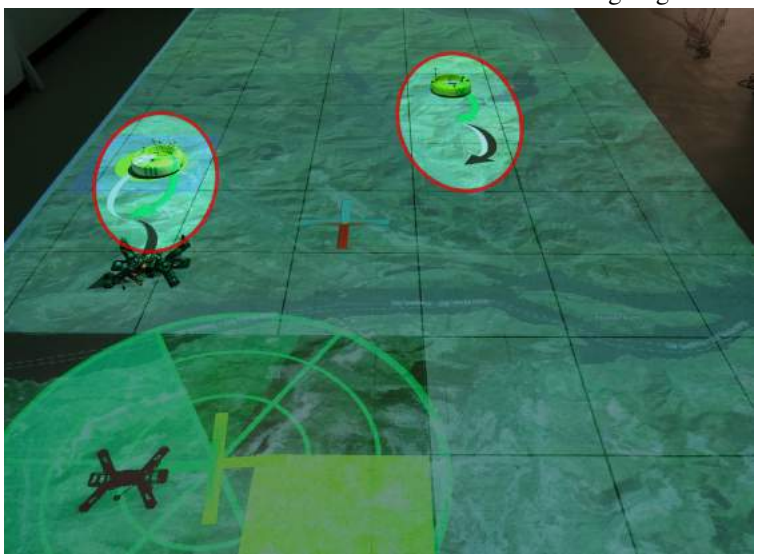
(a) The perceived state of the vehicle (or mean of the 'belief' state) can be projected directly in the hardware domain to indicate discrepancies with actual vehicle location. This is done using projected crosses underneath vehicles.



(b) The sensor footprint of a given vehicle can be visualized as seen above, where the radar footprint of the physical quadrotor is indicated. This visual also adds the underlying discretized state space of the domain using grid cells. Grid cells which have been visited by a surveying quadrotor are removed, and the remaining are represented as foggy regions. A yellow grid square is also added to indicate the goal destination of the ground vehicles.

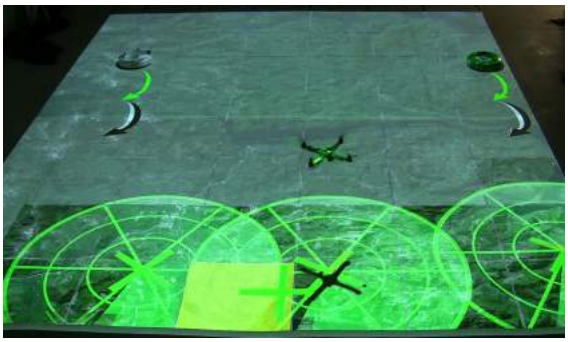
Figure 15: Sequential addition of visualization elements in multi-agent intruder monitoring domain.

(a) A ground vehicle which has been previously detected by a surveillance quadrotor is given a highlighted color. This creates an easily-discernible contrast between detected and still-camouflaged ground vehicles.

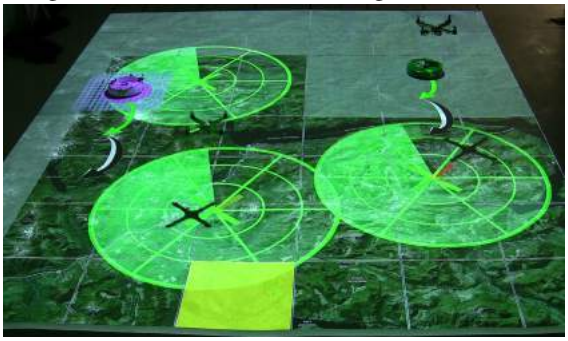


(b) Green arrows protruding from each ground vehicle indicate their most likely next state, while black arrows indicate their most likely next-next state had the quadrotors not been there. Likewise, white arrows indicate their next-next most likely state given the quadrotors' current positions. These serve as real-time visualizations for the transitions of vehicle state, $P(s'|s, a)$, giving spectators an immediate and intuitive understanding of the impact of the quadrotors' actions on the overall objective of keeping the ground vehicles away from their target destination.

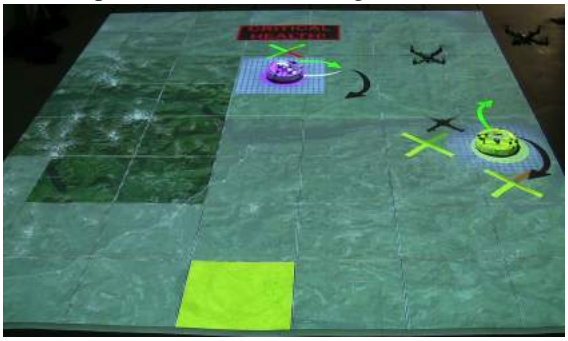
Figure 16: Sequential addition of visualization elements in multi-agent intruder monitoring domain.



(a) Initiation of planning problem, with quadrotors using radar surveillance to detect ground vehicles.

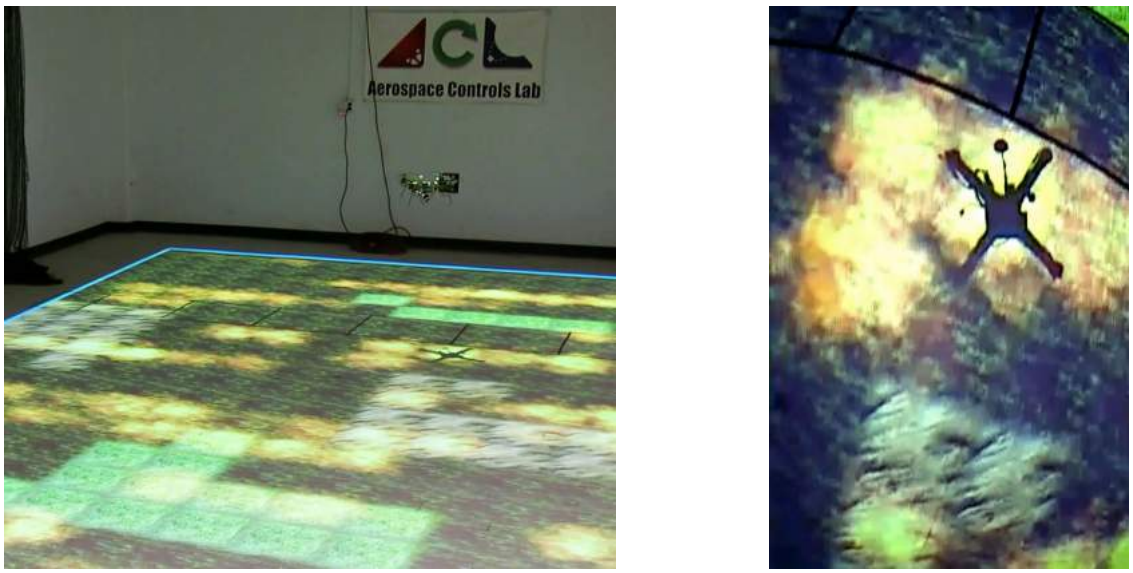


(b) Detection of a ground vehicle by the quadrotor in the top left corner of the image.



(c) Quadrotors can work cooperatively (right) or individually (top) to herd away ground vehicles.

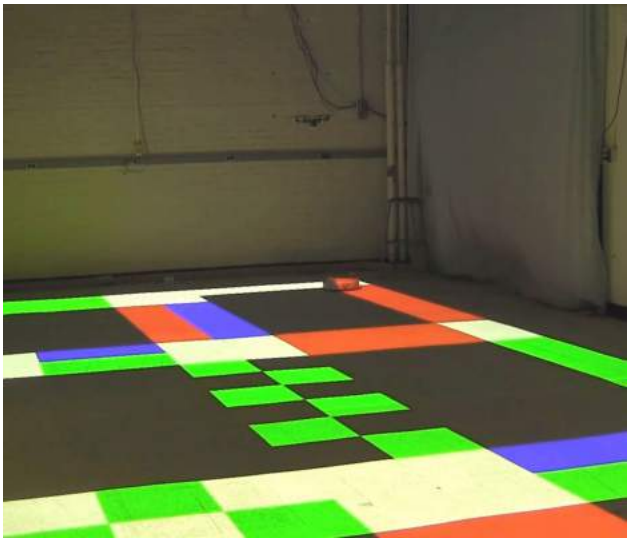
Figure 17: Multi-agent intruder monitoring mission in MAR-CPS. A team of autonomous ground vehicles attempts to reach a goal location in a discretized world while a competing team of quadrotors attempts to push or "herd" them away.



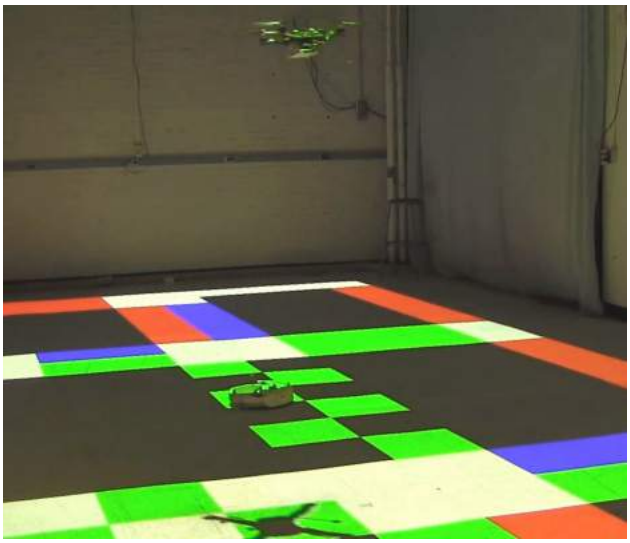
(a) Perspective view

(b) Onboard camera view

Figure 18: Quadrotor firefighting demonstration in MAR-CPS. A discretized 12×30 forest environment consisting of varying terrain and vegetation types (such as trees, bushes, rocks) was constructed and projected in MAR-CPS. Seed fires of varying intensities were initiated on the terrain, with a fire propagation model used for dynamically updating the intensities and distribution over the terrain. A quadrotor used an onboard camera (Sony 700 TVL FPV Ultra Low Light Mini Camera) to wirelessly transmit images to a perception CPU, which created a segmented panorama of the complete forest environment.



(a) Ground agent attempts to navigate a stochastically-changing environment projected in MAR-CPS.



(b) Noisy images from quadrotor-mounted camera provide an estimated environment map to the ground agent.

Figure 19: Real-time vision-based planning under uncertainty in MAR-CPS. A ground vehicle navigates a stochastic obstacle course while a quadrotor provides overhead reconnaissance. Specifically, the ground vehicle traverses over various grids in the domain, each dictating different dynamical model constraints, while planning a minimum-time path to a goal destination.

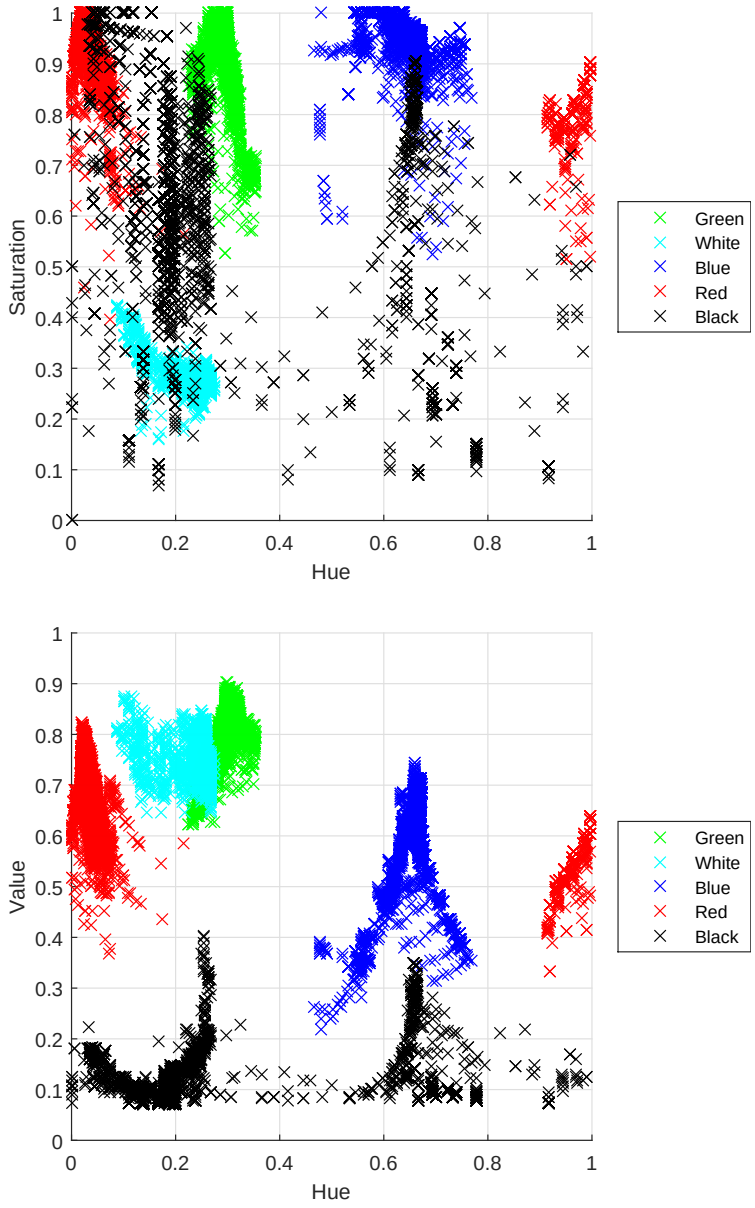
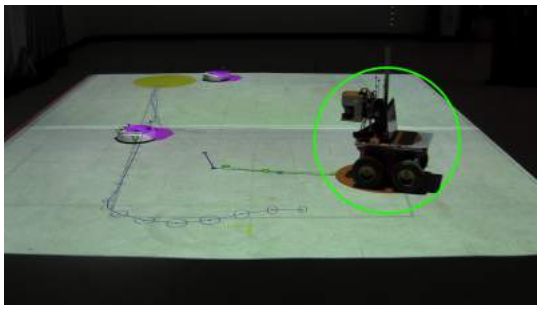
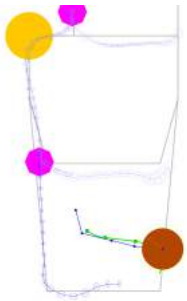


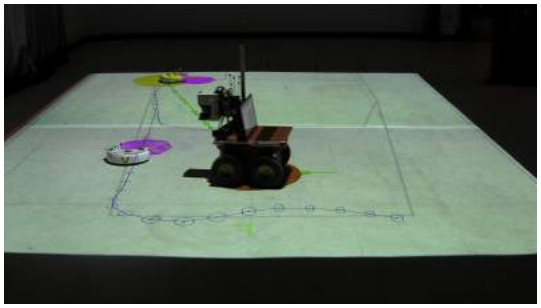
Figure 20: Noisy hue-saturation-value image dataset captured in MAR-CPS for real-time closed-loop perception planning domain (shown in Fig. 19). Grid cells which appear a uniform color to the human observer appear as noisy clusters when captured by a quadrotor-mounted camera. Using MAR-CPS, indoor calibration of sensor systems and robustness against noise can be conducted prior to outdoor flight tests.



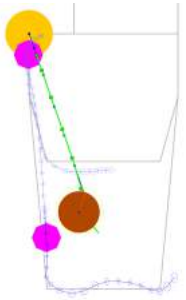
(a) A ground agent (highlighted in green) maintains a "belief" of the future trajectories of two other vehicles (pedestrians) as it tries to reach a goal destination (yellow circle in the top-left). The agent plans a trajectory (shown in green) up to the point where a collision may be imminent.



(b) Overhead visualization of the scenario on left. In this view, the brown circle represents the ground agent, while the pink circles represent pedestrian vehicles. The goal region is shown in yellow. Trajectories protruding from the pedestrian vehicles indicate projected paths that the ground agent believes they plan on taking. This can be interpreted as the belief of paths they plan on taking (although not in the POMDP sense).

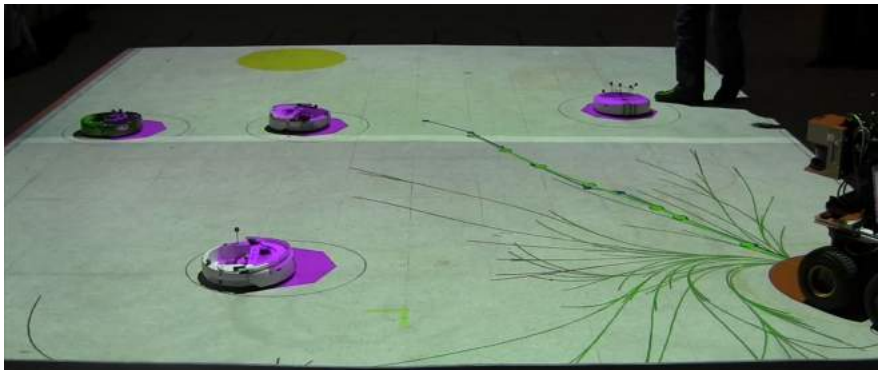


(c) As the vehicles traverse the environment, the ground agent's perception of the 2 pedestrian agents dynamically updates. Once its belief of the pedestrians confirms no collisions are imminent, it plans a direct path to its goal.

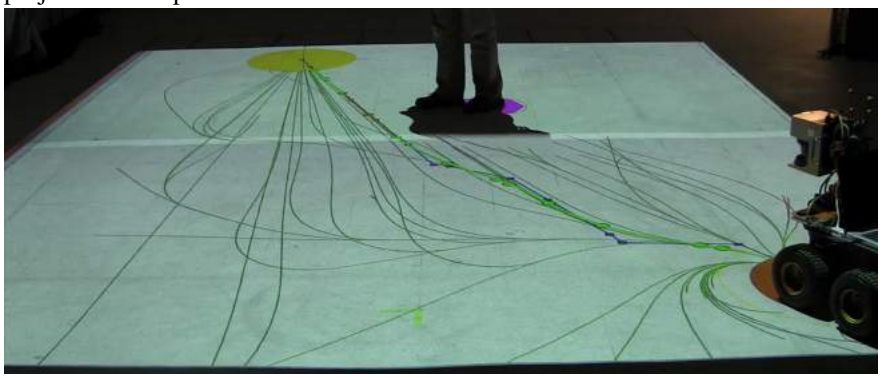


(d) Overhead visualization of the left experiment domain.

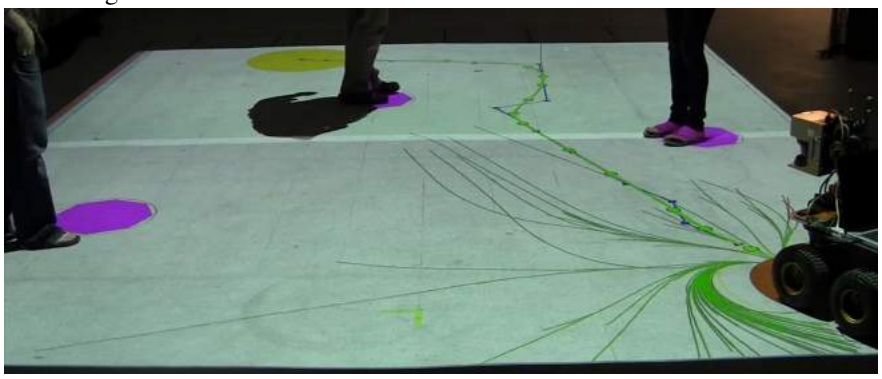
Figure 21: A ground agent (circled in green) plans a trajectory to a goal destination (yellow circle projected on the floor) while avoiding collisions with two pedestrian vehicles.



(a) A ground agent's planned trajectory based on perception of obstacles is projected and updated in real-time.



(b) Detection of human pedestrians for self-driving cars is visualizable in real-time using MAR-CPS.



(c) MAR-CPS enables visualization of complex scenarios such as path planning in multi-pedestrian environments.

Figure 22: Demonstration of motion planning under uncertainty in MAR-CPS.

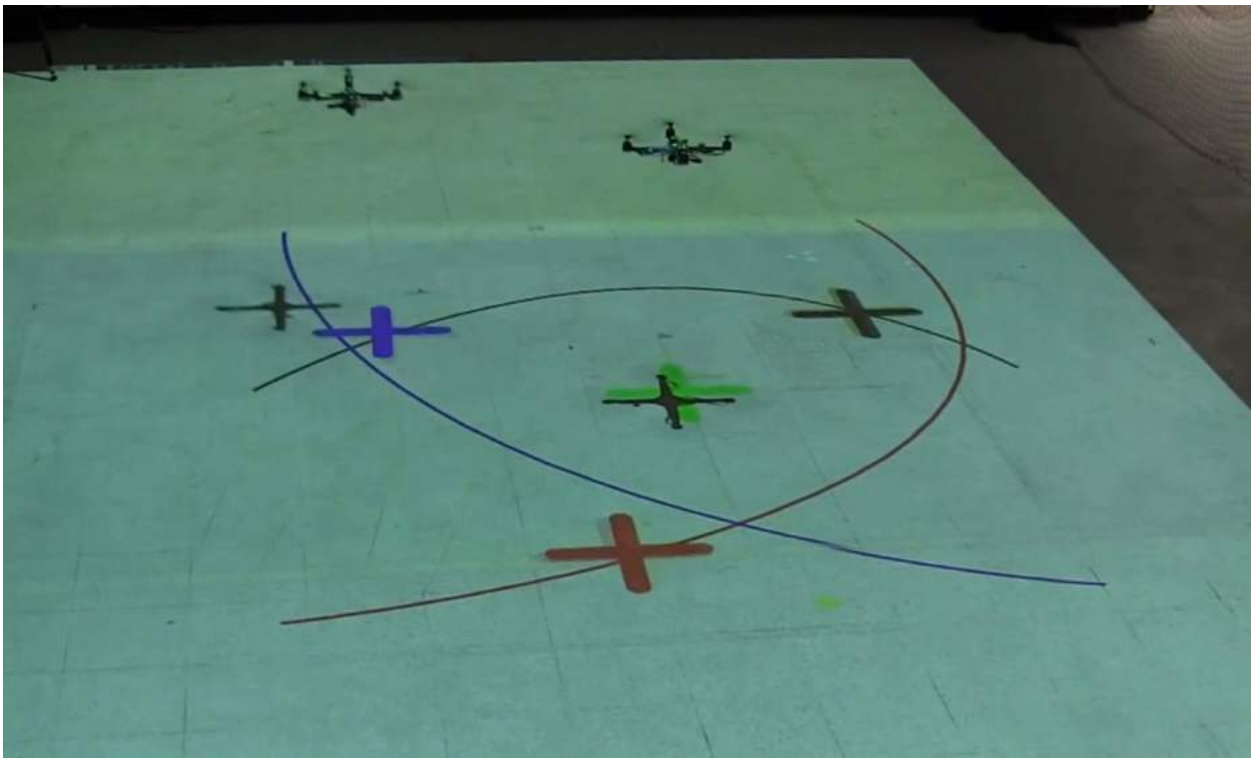
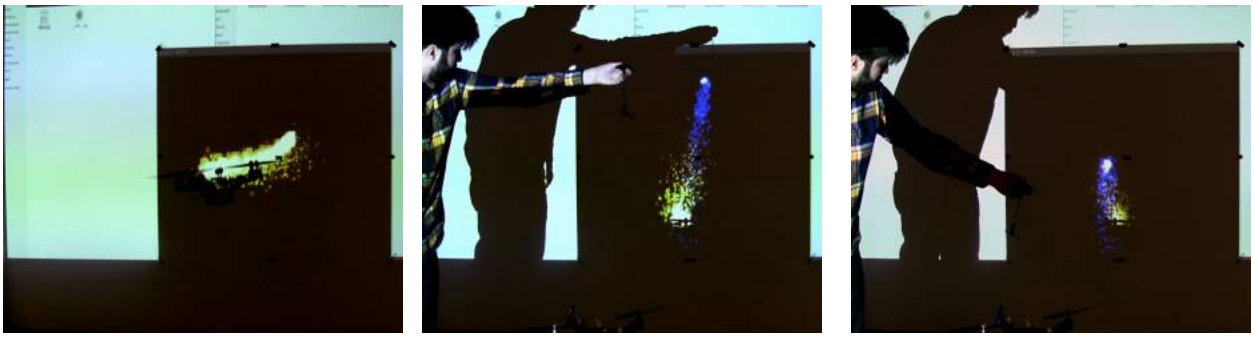


Figure 23: Multi-agent path planning implemented in MAR-CPS. The planned trajectory for each vehicle is projected as a color-coded path in real-time. In situations where physical vehicles are expensive, use of real-time projection in MAR-CPS allows integration of virtual vehicles in experiments. Physical and virtual vehicle interactions can be modeled in software, allowing inexpensive testing of complex, multi-agent missions.

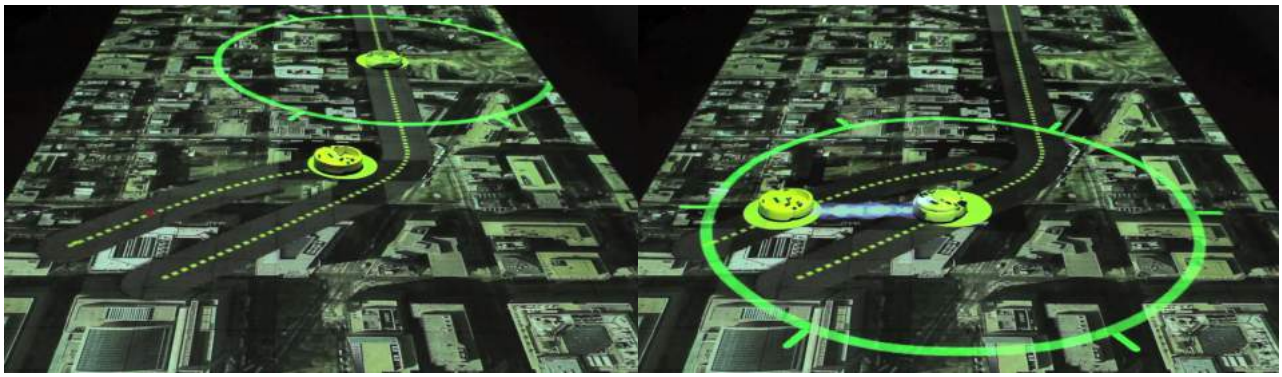


(a) Visualization of quadrotor on fire using a particle simulator and projection system.

(b) Use of a water spout motion capture prop to quench fire is visualizable in MAR-CPS.

(c) MAR-CPS allows supports implementation of interesting human-robot interactivity demonstrations.

Figure 24: Human-robot interactivity using motion capture props in MAR-CPS. Tracking sensors can be placed on a human to allow interaction with autonomous vehicles from a safe distance. Props can be used for representation of objects within the simulated world. This demonstration involves a quadrotor experiencing a simulated onboard fire.



(a) A "leader" agent's radius of communication is indicated by a green circle surrounding it.

(b) When agents are within each other's communication radius, a link between them is established, and information transfer is visualized in MAR-CPS.

Figure 25: Visualization of communication networks in multi-agent systems. In this demonstration, a ground vehicle (the "leader") contains a communications beacon that can be used to assign tasks to a fellow agent (a "follower"). Task assignment only occurs when the agents are within the beacon's communication range. MAR-CPS allows visualization of this range, as well as the moment at which the communication link-up occurs. Images courtesy of Melanie Gonick, MIT News.

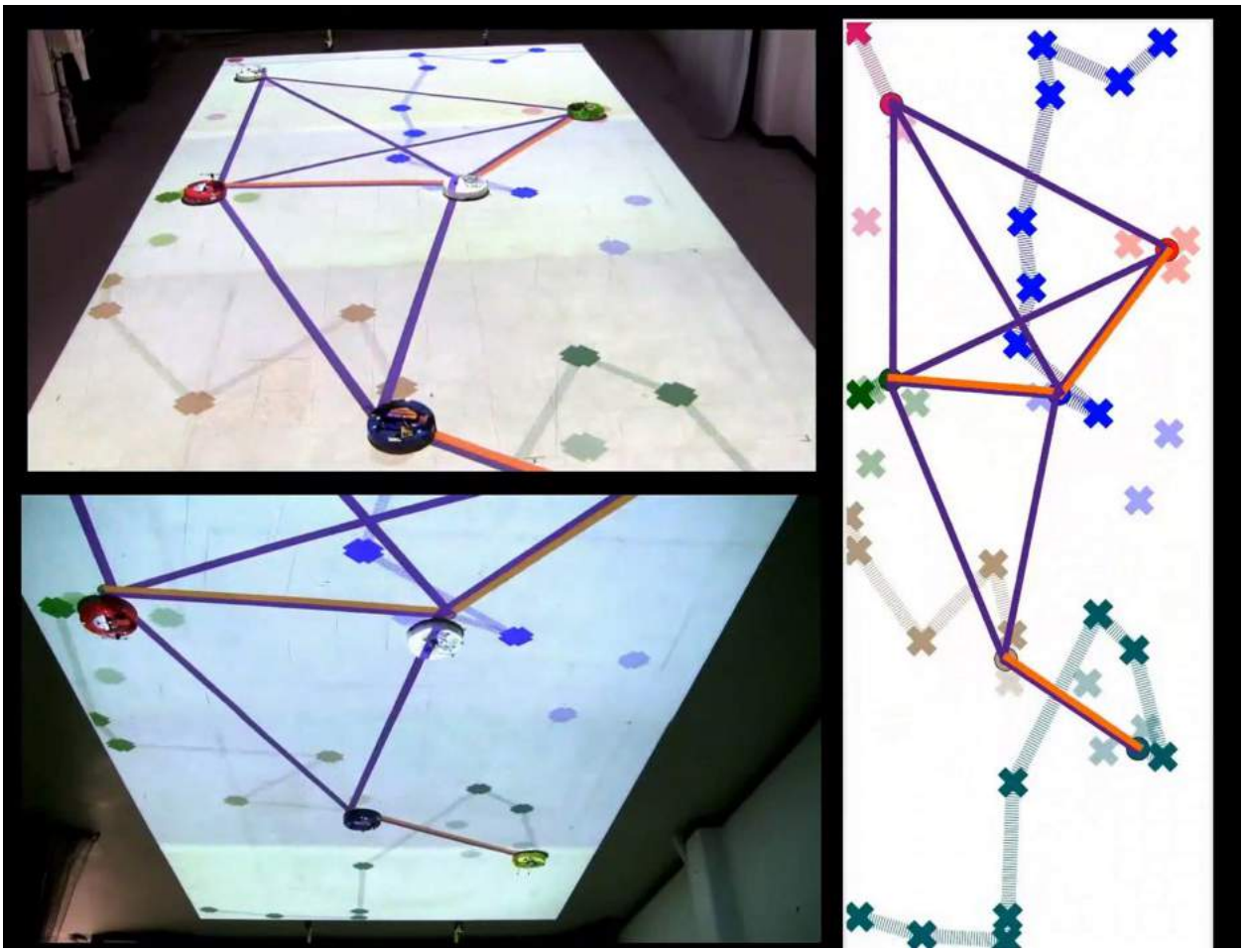


Figure 26: Multi-agent path planning implemented in MAR-CPS. The planned trajectory for each vehicle is projected as a color-coded path in real-time. In this demonstration, 6 agents are servicing tasks (shown as colored crosses above) throughout the domain. Orange lines indicate local information being shared between agents, and purple lines indicate the communication network. Once a task has been serviced, its indicator is made transparent. In these scenarios, MAR-CPS enables visualization of key mission features such as communication networks, which would be otherwise invisible to spectators.

Author Information

Shayegan Omidshafiei is a Ph.D. student at the Laboratory for Information and Decision Systems (LIDS) and Aerospace Controls Laboratory (ACL) at MIT, Cambridge, MA. He received the B.A.Sc. degree from the University of Toronto in 2012 and the S.M. degree from MIT in Aeronautics and Astronautics under supervision of Professor Jonathan How. His research interests include multi-agent planning under uncertainty, autonomous vehicles, reinforcement learning, and data visualization.

Ali-Akbar Agha-Mohammadi is a staff research engineer at Qualcomm Research, San Diego, CA. Prior to Qualcomm, he was a post-doctoral researcher at LIDS and ACL at MIT. He has received the Ph.D. degree in Computer Science and Engineering from Texas A&M, and also holds B.S. and M.S. degrees in Electrical Engineering (Control Systems). His research interests include robotics, stochastic systems, control systems, estimation, and filtering theory. The focus of his current research is on vision-based motion planning under uncertainty for autonomous robots.

Yu Fan Chen is a Ph.D. student at LIDS and ACL at MIT, under supervision of Professor Jonathan How. His research interests include multi-agent planning under uncertainty, multi-agent motion planning, machine learning, and autonomous vehicles.

Nazim Kemal Ure is an assistant professor at the Aeronautical Engineering Department, Istanbul Technical University and affiliated with the Aeronautical Research Center (ARC). He received the Ph.D. degree in Aeronautics and Astronautics from MIT at 2015. His research

interests include development of learning and planning under uncertainty algorithms for multi-agent systems, unmanned aerial vehicles and automated air traffic management systems.

Shih-Yuan Liu is a postdoctoral associate at LIDS and ACL at MIT. He received the Ph.D. degree in Mechanical Engineering in Controls from University of California, Berkeley in 2014. His research interests include control, path-planning, and coordination of autonomous ground and aerial vehicles in dynamic environments.

Brett T. Lopez is an S.M. candidate in ACL at MIT under the supervision of Professor Jonathan How. His research interests include control, path planning, and state estimation for autonomous vehicles performing aggressive maneuvers.

Rajeev Surati Ph.D. is a serial entrepreneur and inventor. His MIT Ph.D. thesis established the field of projector camera calibration. He is currently chairman of Scalable Display Technologies, a company commercializing the projector de-warping and edge-blending technology used in MAR-CPS.

Dr. Jonathan P. How is the Richard C. Maclaurin Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology. He received a B.A.Sc. from the University of Toronto in 1987 and the S.M. and Ph.D. degrees in Aeronautics and Astronautics from MIT in 1990 and 1993, respectively. Professor How currently serves as the head of the Information sector within the Department of Aeronautics and Astronautics, and is the Director of the Ford-MIT Alliance. He is the Deputy Editor-in-Chief of the IEEE Control Systems Magazine and an Associate Editor for the AIAA Journal of Aerospace Information Systems. Professor How was the recipient of the 2002 Institute of Navigation Burka Award, a Boeing Special Invention award

in 2008, the 2011 IFAC Automatica award for best applications paper, Recipient of the AIAA Best Paper Award from the 2011 and 2012 Guidance Navigation and Control Conferences, and he is an Associate Fellow of AIAA and a senior member of IEEE.

John Vian is a Technical Fellow at Boeing Research & Technology. He has over 30 years of experience in flight controls and autonomous systems. He has contributed to control system design on multiple aircraft programs and served as Principal Investigator on NASA, FAA, Navy, and Air Force research contracts. He holds a B.S. in mechanical engineering, M.S. in aeronautical engineering, and Ph.D. in electrical engineering.