

Using Critical Junctures and Environmentally-Dependent Information for Management of Tightly-Coupled Cooperation in Heterogeneous Robot Teams

Lynne E. Parker, Christopher M. Reardon, Heeten Choxi, and Cortney Bolden

Abstract—This paper addresses the challenge of forming appropriate heterogeneous robot teams to solve *tightly-coupled, potentially multi-robot tasks, in which the robot capabilities may vary over the environment in which the task is being performed. Rather than making use of a permanent tightly-coupled robot team for performing the task, our approach aims to recognize when tight coupling is needed, and then only form tight cooperative teams at those times. This results in important cost savings, since coordination is only used when the independent operation of the team members would put mission success at risk. Our approach is to define a new semantic information type, called *environmentally dependent information*, which allows us to capture certain environmentally-dependent perceptual constraints on vehicle capabilities. We define locations at which the robot team must transition between tight and weak cooperation as *critical junctures*. Note that these critical juncture points are a function of the robot team capabilities and the environmental characteristics, and are not due to a change in the task itself. We calculate critical juncture points by making use of our prior ASyMTRe approach, which can automatically configure heterogeneous robot team solutions to enable sharing of sensory capabilities across robots. We demonstrate these concepts in experiments involving a human-controlled blimp and an autonomous ground robot in a target localization task.*

I. INTRODUCTION

In many heterogeneous robot teams, some robot team members are unable to perform certain tasks without direct help from other teammates. This limitation in team member capabilities leads to the need for robots to form coalitions to help each other solve tasks. Typically, these tasks are called *multi-robot tasks* (per the taxonomy of Gerkey [4]) that must be solved using *tightly-coupled* cooperation [7]. However, this categorization of a task as a *multi-robot task* presumes that the designation is independent of the robot team member capabilities. In reality, whether a task can be performed by a single robot, or requires multiple robots, is in fact dependent on the capabilities of the robot team members. Prior work (e.g., [11]) has resulted in approaches that can automatically construct solutions for tasks that are dependent upon the capabilities of the team members.

This work is funded in part by Lockheed Martin Advanced Technology Laboratories and in part by the National Science Foundation under Grant No.IIS-0812117. L. E. Parker and C. M. Reardon are with the Distributed Intelligence Laboratory, Department of Electrical Engineering and Computer Science, The University of Tennessee, 203 Claxton Complex, Knoxville, Tennessee 37996-3450, USA (email: parker@eecs.utk.edu, creardon@utk.edu). H. Choxi and C. Bolden are with Lockheed Martin Advanced Technology Laboratories, 3 Executive Campus, Cherry Hill, NH 08002, USA (email: hchoxi@atl.lmco.com, cbolden@atl.lmco.com).

However, existing techniques are not sufficient when the capabilities of the robot team members not only vary from robot to robot, but also vary within an individual robot depending upon the environmental characteristics. Examples of this type of task include landmark-based localization, in which sufficient landmarks within the field of view are needed to properly achieve localization; camera-based navigation, in which sufficient lighting is needed to avoid collisions; or wheel-based locomotion, which requires a relatively smooth surface for motion. Thus, a task may at times require the tight coupling of multiple robots working together, while at other times, a different coalition of robots, or even a single more capable robot working alone, is needed to achieve the task. By automatically determining when tight cooperation is needed during the task, the robot team can avoid the permanent use of high-cost coalitions of robots working together, restricting those solutions to segments of the task that cannot otherwise be solved.

This paper addresses this class of robot tasks. In particular, we focus on *weakly-decomposable tasks*, which we define to be tasks such as “search”, “coverage”, “forage”, or “path following” that must be performed in continuous space and time, and which cannot be trivially be subdivided into discrete subtasks independently of the environment or robot team. Note that such tasks could be subdivided into separate routes that fully-capable teams could execute independently to achieve the overall task, such as different paths for covering different parts of the workspace; however, these subdivisions are typically made independently of the robot capabilities, and cannot be trivially assigned to robots without considering the characteristics of the environment or of the robots themselves. Our objective is to develop an approach that allows robots with limited, environmentally-dependent capabilities to perform such tasks autonomously by teaming with other robots at the appropriate time and place. In this paper, we focus primarily on robots’ perceptual capabilities that may vary across the environment.

In this paper, we first define the concept of *environmentally dependent information*, which provides a way to describe certain environmentally-dependent perceptual limitations of the robots. We next define the concept of *critical junctures*, which are those environmental regions of the task where the robot team must transition between tight and weak cooperation. To reason with this information to determine appropriate teams of robots at each critical juncture, we make use of our prior ASyMTRe architecture [11] for configuring

robot team solutions that make use of sensor sharing when needed.

We illustrate these new ideas in a target localization task, which requires the team to explore an indoor area and localize targets found in the environment. We present experiments in which a human-piloted Unmanned Aerial Vehicle (UAV) (specifically, a blimp), and an autonomous mobile (ground) robot work together in this target localization task. Coordination occurs between the autonomous mobile ground robot with sophisticated localization capabilities (but limited target detection capabilities) and the human-controlled blimp with only limited localization capabilities (but more capable target detection capabilities). Our experiments compare independent robot operations with approaches that either (1) pre-plan the critical juncture regions, or (2) reactively recognize critical juncture points at run time. Our results show that it is indeed possible to automatically identify the critical junctures for when tightly-coupled coordination is needed between vehicles, and that teams employing these tightly-coupled cooperative behaviors during critical juncture regions exhibit improved performance over non-cooperative teams. Our results further indicate that pre-planning critical regions is not always preferred over a reactive approach.

II. RELATED WORK

A significant body of research has addressed the subject of coordination in heterogeneous entity teams (see [10] for an overview of the field). Research specific to heterogeneous robots often focuses on the issue of *task allocation*, which is the problem of determining a suitable mapping between robots and tasks. A formal analysis of this problem is presented in [4]. Several approaches to robot team task allocation have been developed. Typically, a task is decomposed into independent subtasks [9], hierarchical task trees [18], or roles [6], [13], [15] either by a general autonomous planner or by the human designer. Independent subtasks or roles can be achieved concurrently, while subtasks in task trees are achieved according to their interdependence. An example of a behavior-based approach to multi-robot task allocation is ALLIANCE [9]. After Smith [16] first introduced the Contract Net Protocol (CNP), many market-based approaches addressing multi-robot cooperation through negotiation were developed, including M+ [1], TraderBots [2], [18], and Hoplites [7]. In these approaches, a task is divided into subtasks or hierarchical subtask trees (in the case of [18]) for the robots to bid and negotiate to carry out the subtasks. In other work, Fua and Ge [3] also address the problem of multi-robot tasks using the COBOS cooperative backoff adaptive scheme. However, these prior approaches do not include a concept of environmentally dependent information or capability.

Other work related to this research effort is work in coalition formation (e.g., [14]). In these systems, agents are organized into coalescent teams to achieve a higher-level goal [5]. Shehory [14] describes a method of allocating a set of interdependent tasks to a group of agents by forming

coalitions. However, as noted in [17], many of the multi-agent approaches to coalition formation cannot be directly transferred to multi-robot applications, since robot capabilities and sensors are situated directly on the robots and are not transferable between robots. Lundh, et al., [8] present an approach to configuring groups of robots based on their *functionalities*, and using knowledge-based techniques. Their work is different from ours, in that they plan for a series of discrete actions to achieve a goal, rather than addressing a weakly-decomposable task.

In prior work, we have developed the ASyMTRE approach [11] for automatically configuring heterogeneous robot team solutions for solving a given task. Rather than pre-defining a task to be a single-robot task or a multi-robot task, the ASyMTRE approach determines what the best solution to the task would be by exploiting the sensing, computation, and motor schema capabilities of the robot team members. The resulting solution for a task could either require only a single team member, or it could require multiple team members working together to share their sensing, computational, and effector capabilities to achieve the given task. Our ASyMTRE approach differs from other prior work in that it abstracts the problem at the *schema* level, rather than the task level, permitting more flexibility in the solution approach. ASyMTRE can use this abstraction to generate alternative combinations of robot capabilities (i.e., schemas that generate required information) that can accomplish the task, and does not assume a fixed list of capabilities (or even information types) that are needed to accomplish a task. By abstracting the task using schemas and information requirements, rather than specific solutions based on specific sensors, we believe ASyMTRE generates more flexible solution strategies for multi-robot coalition formation that are not dependent upon a fixed list of capabilities or resources required for each task. The work described in this paper extends this prior work of ASyMTRE by defining robot capabilities that are dependent upon the local environment, rather than being fixed for that robot. Further, we show how the critical regions requiring tight cooperation between robots can be determined automatically.

III. APPROACH TO SYNERGISTIC CONTROL

Our fundamental ideas for achieving synergistic control are based on two concepts. First, we develop a representation for environmentally-dependent, sensor-generated information that robots must use to achieve certain tasks; we call this information *environmentally dependent information* (or EDI). Second, we define *critical junctures* in which environmental characteristics change the perceptual capabilities of robot team members, thus requiring a new team composition. The following subsections describe these concepts in more detail.

A. Environmentally Dependent Information

In defining the perceptual capabilities of a robot, our approach is to focus on the *information* that is obtained from a combination of sensors plus computation, rather than focusing on the sensor resource alone. This focus on

information flow emphasizes the use of the sensor toward some objective, rather than just the presence of the sensor alone. Thus, to achieve a task, a robot (or coalition of robots) must have a required flow of information from sensor to motor control. As an example, consider the task of localizing a target. The required information needed to localize a target might be a scanned map of the environment, the current global position of the robot within the environment, and the relative position of the target within the field of view. In turn, some of this information may itself be dependent upon yet other information. For example, determining the current global position of the robot may require a combination of a laser scanner and a Monte Carlo Localization algorithm for finding the most likely position of the robot in the given map. Or, alternatively, the current global position could be calculated using a combination of a camera and two or more known landmarks within the field of view. The particular combination of sensor and perceptual computation necessary to generate the required information will vary from robot to robot. Indeed, some information can come from other robots, thus enabling coalitions of robots to work together to solve the given task. In any case, as long as the required information is available, regardless of the source, the robot can calculate the global position of the target and mark its position on the map of the environment.

This perspective of the problem in terms of the information flow through the robot system, regardless of the source of the information, is the basis of the ASyMTRe approach in our earlier work [11]. To simplify the automatic reconfigurability of the approach, *schemas* are used as basic building blocks, consisting of environmental sensors (ES), perceptual schemas (PS), communication schemas (CS), and motor schemas (MS). The inputs and outputs of these schemas are defined in terms of their semantic information content, where the set of possible information types is given by $F = \{F_1, F_2, \dots\}$. For a schema S_i (which is one of the available ES, PS, CS, or MS schemas on a robot R_i), I^{S_i} and $O^{S_i} \subset F$ represent the input and output sets of S_i , respectively. The output of a schema can be connected to the input of another schema as long as their semantic information types match. The n robots on the team are defined as $R = \{R_1, R_2, \dots, R_n\}$. The task to be performed is defined as $T = \{MS_1, MS_2, \dots\}$, which is the set of motor schemas that define the group-level task to be achieved. A robot R_i is represented by $R_i = (ES_i, S_i)$, in which ES_i is a set of environmental sensors that are installed on R_i , where $O^{ES_i^j} \subset F$ is the output of ES_i^j (that is, the j th ES on robot R_i). A solution to the task is given by a combination of schemas that can satisfy the required information flow to meet the task requirements. Different configurations of schemas can solve a given task in different ways, either within an individual robot, or across multiple robots. When alternative solutions exist, the preferred approach is selected using a supplied utility function.

To incorporate the idea of *environmentally-dependent information* (EDI) into this concept, we must add additional descriptive content to the required information; this

additional descriptive content identifies the environmental characteristics necessary to generate the information. These characteristics must then be used in planning the required coalitions for different regions of the environment. More formally, the approach is as follows. Each generated solution for robot R_i will contain a set of environmental sensors and perceptual schema combinations (ES_i^j, PS_i^j) , for all j environmental schemas used in the solution for robot R_i . Each of these combinations generates a required type of information $O^{ES_i^j} \subset F$. It is these combinations of sensors and perceptual schemas that can be dependent upon certain environmental characteristics. Thus, corresponding to these pairs of sensors and perceptual schema processes must be additional information on environmental constraints for using that combination, if any. Thus, we define constraints, $H_{p,q}^{1..r}$, which specify the r environmental constraints on a given (ES_p, PS_q) combination. Environmental constraints are defined as conditions (i.e., propositional variables) that must be either true or false for a set of pre-determined, application-dependent environmental variables. Examples of such H conditions include “*range < 2m*”, “*lighting-level > 60 footcandles*”, “*3 landmarks of type x within range < 3m*”, etc. If the environmental constraints are not met in a given environmental region, then ASyMTRe must discard this combination of sensors and schemas as a possible solution.

As an example of a practical definition of an EDI, consider the task of indoor localization for the purpose of target localization. For autonomous robots in static indoor environments, localization can typically be achieved by making use of a laser scanner, a scanned map of the environment, and a localization algorithm. However, a less-capable vehicle with unsophisticated sensors (such as a human-controlled blimp with only a crude camera onboard) may only be able to perform localization if known landmarks, corresponding to known positions on the map, are within line of sight of the remote camera. In this example, the EDI is defined to be a distinct corner landmark that is useful to the human for performing global localization; this EDI is associated with the (ES, PS) combination of (*camera, global localization schema*). Figure 1 shows the corner EDIs thus defined for an example application environment in the target search task. As long as these landmarks are within sight, and they can be mapped to a global position in the environment, then a trained human should be able to localize targets in the image, within some error bounds. However, if no identifiable landmarks are within view, then the human needs an alternative reference point in order to localize targets in the environment. To illustrate this idea in our experiments, the alternative reference point is the autonomous ground robot itself, which, when needed by the human to localize a seen target, moves to the general proximity of the blimp and reports its precise position to the human (via radio). The human can then approximate the relative position of the blimp to the ground robot, and use this information to localize the visible target. This cooperation between the autonomous ground robot and the human-controlled blimp

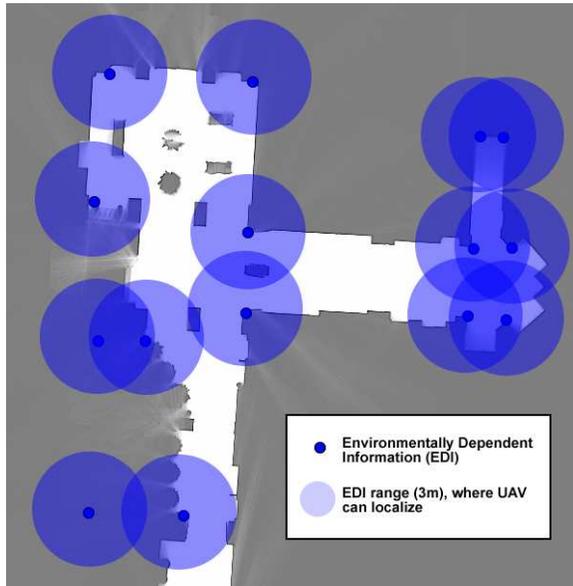


Fig. 1. Environmentally Dependent Information (EDI) (i.e., corner landmarks) in the test environment for a target localization task.

is the type of tightly-coupled cooperation illustrated in our experiments.

B. Critical Junctionures and Regions

Our hypothesis is that the cost of using tightly-coupled coordination can be minimized by using independent behaviors when possible and tightly-coupled coordination only when independent behaviors would otherwise fail. We use the term *critical junctionures* (CJs) to describe the starting point in a weakly-decomposable task where independent behaviors will fail and tightly-coupled coordination is required; this term is also used to describe ending points during a task where tightly-coupled coordination is no longer required. The time between these critical junctionure points is called a *critical region*, or *cooperative region*.

In our current approach, we make some important assumptions affecting critical junctionures. First, we assume that the environment in which the robots operate is known in advance, so that advanced planning techniques can be used for calculating critical junctionures. If this assumption does not hold, meaning that the environment is not known *a priori*, then it is still possible to identify critical junctionures using reactive planning. Second, we assume that robots (or coalitions of robots) performing the tasks of interest (i.e., search, coverage, etc.) do so by following a path (or, more precisely, a set of paths) through the environment that is (are) generated by a separate computational process. These paths define the routes that fully-capable robot teams must take in order to achieve the task in a cooperative fashion. By assuming that robot teams are given paths to follow to achieve the given task, critical junctionure points can be associated with locations along the path at which environmental characteristics change the required composition of the robot team. Finally, we assume that the available robot assets are

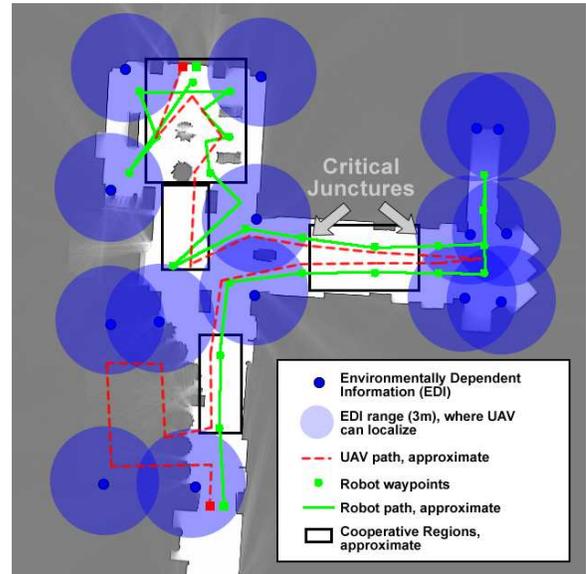


Fig. 2. Critical (or cooperative) regions, demarcated by Critical Junctionures, indicating where tightly-coupled cooperation is needed, due to the lack of EDI information.

sufficiently dense, relative to the workspace, so that robots can provide assistance to each other when needed without incurring an excessive overhead cost. Future research will be aimed at relaxing these assumptions.

C. Using ASyMTRe and EDIs to Compute CJs

Our approach can either pre-compute critical junctionures, or determine reactively when a critical junctionure has been reached. As previously stated, for pre-computed critical junctionures, we assume that a separate process (not addressed in this paper) has preplanned paths for fully capable robot coalitions. As an example, Figure 2 shows example paths for a UAV and a ground robot for a target search task, which requires the robot team to search the entire workspace for targets of interest. These paths are then sampled at regular intervals, with the spacing along the paths dependent upon the sensing range of the robots. At each sampled position, ASyMTRe is run as a subroutine to generate a task solution for that position. A different set of robot schema configurations will be generated depending on whether the area is within range of a required EDI. When the generated task solution changes between an independent solution and a cooperative solution, that point is identified as a critical junctionure. Segments of the paths that are between critical junctionures are identified as critical regions, which require tight cooperation between robots. Figure 2 shows example paths of the UAV and ground robot, with the cooperative regions (boxed in black) overlaid on the EDI map from Figure 1.

Of course, it is clear that the optimal paths of the robots are dependent on which regions of the environment require tightly-coupled cooperation; thus, in the general case, our decoupled approach of considering the paths separately from the cooperative regions will not find optimal paths. However, it is also well-known that the general multi-robot path

planning problem is intractable (specifically, PSPACE-hard), so it is a non-trivial problem to generate optimal solutions to this problem. Nevertheless, in future work we plan to consider tractable methods of coupling the multi-robot path planning step with the consideration of the critical junctures.

For reactive detection of critical junctures, the robots periodically rerun the ASyMTRE configuration algorithm to determine if the current configuration is still valid, or, if not, to determine an alternative configuration of robot capabilities that can address the task in the current area of the environment. When the best alternative solution requires tightly-coupled cooperation between robots, the robots communicate with each other to initiate the tightly-coupled cooperation.

IV. PHYSICAL ROBOT EXPERIMENTS

To illustrate our concepts, we implemented a physical robot solution to the indoor target localization task. The heterogeneous team for this task consisted of a human-controlled aerial blimp and an autonomous ground robot. The following subsections describe the task, the robots, and the test scenarios we used to evaluate our approach.

A. Target localization task

The objective of the target localization task is to visit all of the identified indoor workspace in search of targets, and to note the location of the identified targets on a map. The team is provided with a map of the environment in advance (which, in this case, was autonomously generated by the ground robot in advance of the target localization task). In the physical robot experiments, targets were chosen to be solid green balloons of size 30 cm in diameter, which have the beneficial characteristics of being spherical (and thus present the same two-dimensional profile from any angle), easy to detect in our indoor test area, inexpensive, lightweight, and easy to manage. (The images in Figure 4 show several balloon targets.) Targets are uniformly distributed in the environment at two possible heights (“high” and “low”), using an automated process that selects the target locations, followed by manually placing the targets at the generated locations. A total of 12 targets were used in the experiments reported in this paper, in an environment of size of approximately 450m², yielding approximately a 4-5m spacing between targets. Figure 3 shows an example of the test environment and the selected target positions. Note that in this environment, the area in the lower left of the workspace, marked by a dashed rectangle, is an area accessible only by the blimp; this is a lunch area with many tables and chairs that block the ground robot’s path.

B. Robots and behaviors

The human-operated blimp used in these experiments was a custom-built, 180cm long envelope controlled by a standard, two stick, hobby-type transmitter/receiver for remote control. The blimp carries an Eyecam 2.4GHz color micro wireless video camera, whose output is displayed on a projector, for use by the human controller. More details on the blimp design are available in [12]. The autonomous ground

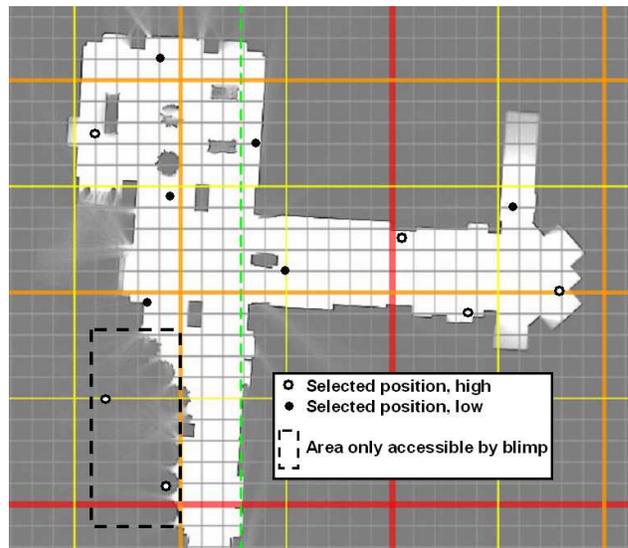


Fig. 3. Example target locations.

mobile robot used for these experiments was a Pioneer 3 DX robot. This robot includes an onboard computer, a SICK laser range finder, a Canon VC-C4 camera (tilted down 15° from horizontal), and 802.11 wireless. Software behaviors operating on the robot include self-localization, navigation, communication, and target detection and localization.

The behavior of the vehicles for target detection is as follows. The autonomous ground robot moves along a pre-planned path, and, at periodic waypoints along that path, it stops and conducts a 360° visual sweep of the area with the camera, looking for targets. The human-controlled UAV, being less-precisely controllable than the autonomous ground robot, follows its predefined path and, as deemed necessary by the human pilot, rotates and changes altitude to visually scan the surrounding area for targets. Upon detecting a target, the robot or UAV then localizes the target to the best of its ability. For the ground robot, this involves using the direction and size of the target in the image, combined with self-location and orientation to determine a precise x, y position of the target. Because the robot’s camera is fixed in a slightly downward-facing position, the robot is not capable of seeing “high” targets; thus, all z -axis values reported by the robot are set at the ground level, or “low”. For the UAV, target localization involves a human operator using self-localization, determined by position and orientation relative to a corner EDI or the ground robot (whose global position is communicated to the UAV operator when requested), combined with the direction and size of the target, to visually estimate and record the position of the target. The human operator identifies the x, y position as well as a z value (either “high” or “low”) per target.

Because of the way the targets are distributed and the information types defined for the environment, the UAV and robot are expected to have different levels of success localizing targets for different target locations. For example, targets placed at a high elevation in the environment are



Fig. 4. On the top, the UAV waits on the ground robot for localization assistance. On the bottom, the robot assists the UAV in localizing a target.

unlikely to be detected by the ground robot, whereas targets in areas outside the range of EDIs are unlocalizable to the UAV without assistance. In these experiments, while there are targets that are detectable only by the UAV, there are no targets only detectable by the robot. While in this special case a solution could be designed where the robot is dedicated to assisting the UAV, such a solution would work only for this special case. The solutions described here are generalizable and as such will also work in the case where there are targets localizable only by the robot. We therefore only present and evaluate the solutions in that context.

C. Scenarios

We validated our proposed approach in three scenarios, each representing an alternative approach to identifying critical regions: Independent, Cooperative Planned, and Cooperative Reactive.

In the *Independent* scenario, the UAV and robot each perform the task independently, without any cooperation. This represents a base case for comparison, where no cooperation is attempted. Note the differing capabilities of the robot and the UAV in this scenario: the robot, with its short height and fixed camera, cannot localize “high” targets, while the UAV cannot localize any targets outside of the range of an EDI.

In the *Cooperative Planned* scenario, the CJs are planned algorithmically in advance at the start of the task. In this scenario, the paths of the robot and UAV coincide at segments where cooperation is required. Critical juncture points in both the robot and UAV path are preplanned using ASyMTRe. When navigating through such a cooperative region, the robot stops at each waypoint and communicates its position to the

UAV, as well as its state as being ready to proceed to the next point at the UAV’s request. The UAV then uses the robot’s communicated position to perform relative localization of any nearby targets. After searching that area for targets, the UAV sends the command to proceed to the robot. When navigating outside of a cooperative region, the robot and UAV proceed searching along their paths independently. If either the robot or the UAV reaches a CJ that denotes the beginning of a cooperative region before the other, they wait for the other team member. This cooperative behavior allows the UAV to localize all targets in the environment.

In the *Cooperative Reactive* scenario, the CJs are identified algorithmically as needed during experiment runtime, in reaction to the EDI information available to the blimp during the experiment. In this scenario, the robot and UAV proceed exactly as they would in the independent scenario, except that when the human-driven UAV realizes that EDI information needed to localize a seen target is missing, the human UAV operator communicates a message to the robot to proceed to a roughly approximated position near the UAV and the detected target¹. Upon receipt of this request, the robot proceeds to the position requested by the UAV. Once there, the robot communicates its presence and exact position to the UAV. Then, the human operator of the UAV is able to visually localize the target relative to the robot. At this point, the UAV communicates a message to resume previous behavior to the robot, and the robot returns to its previous waypoint position and to the independent search behavior. This scenario is included in our experiments to determine whether it streamlines the need for cooperative behavior, by only invoking tightly-coupled cooperation at CJs where targets are detected, rather than at all CJs.

V. RESULTS AND DISCUSSION

Five sets of three experiments each were run. Experiments within the same set used the same randomly generated target positions. The three runs within each set were for the three scenarios outlined in the prior section. Thus, a total of 15 experimental runs on the physical robots were conducted. Figure 4 shows two snapshots of the robots during these experiments. To evaluate the performance of the behavioral approaches, we defined three metrics: Target Localization Accuracy, Task Completion Time, and Aggregate System Run Time. Target Localization Accuracy is the ratio of the number of targets localized to the number of actual targets. Task Completion Time is the maximum of the robot and UAV run times for a single run. Aggregate System Run Time is the sum of the run time of the robot and UAV for a single run.

¹In our experiments, it is assumed that the human UAV operator can roughly identify an approximate position in the vicinity of its current location to which the robot should move. The camera on the blimp has sufficient range that if the robot gets within the vicinity, the human can see it. For more accurate positioning, the human and robot can iterate to move the robot into closer range of the blimp, in case the original approximated position is too far for accurate target localization. For more general applications, we would like to enable the robot to seek out the blimp, based on the roughly approximated position provided by the human.

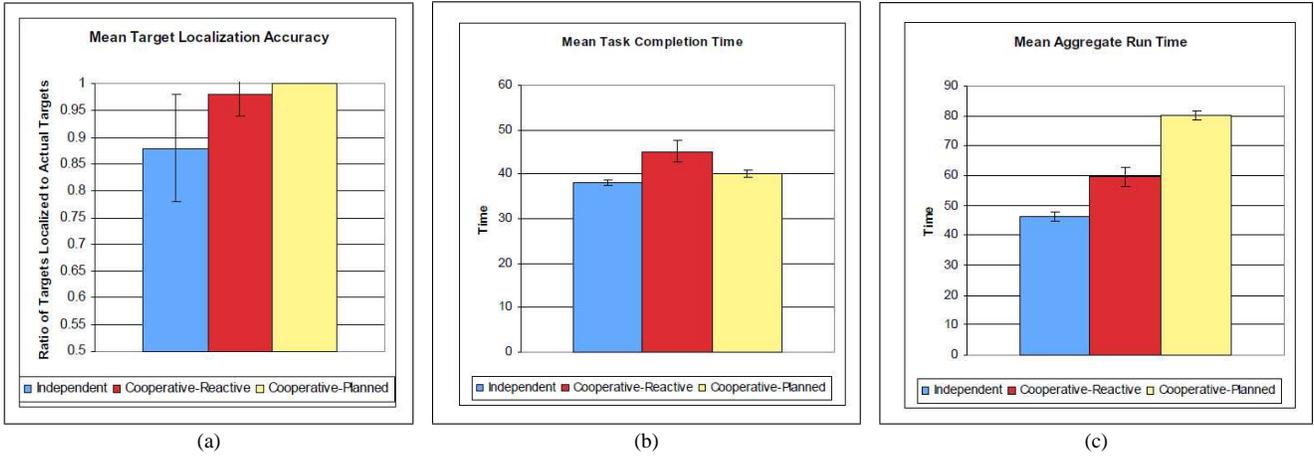


Fig. 5. (a) Mean Target Localization Accuracy, per scenario; (b) Mean Task Completion Time, per scenario; (c) Mean Aggregate System Run Time, per scenario. The error bars indicate the standard deviation of the results

Figure 5(a) shows the results of these experiments measured in terms of mean Target Localization Accuracy². Both of the cooperative approaches performed better as a system, localizing a significantly greater percentage of targets than in the Independent approach. This is as expected, since certain subsets of the overall task can only be accomplished via cooperation. In the general case, the performance of the Independent approach is clearly bounded by the number of tasks that can be accomplished independently. In this case this is total number of targets minus those targets that are located high and outside the range of an EDI, which renders those targets viewable by only the UAV, but not localizable.

Figure 5(b) shows the results measured in terms of Task Completion Time³. In these experiments, the Task Completion Time happens to be equal to the run time of the ground robot, since the UAV can move much faster through the environment than the ground robot. However, while the three approaches are significantly different, the relative differences are small. The Independent approach is the fastest, as the robot only has to navigate and search along its own path without any interaction or interruption to cooperate with the blimp. Interestingly, the Cooperative-Planned approach is only slightly (5% on average) longer than the Independent approach; this can be attributed to the fact that, because the UAV was significantly faster than the robot, the UAV is always at each CJ at the same time as the robot. Thus, the UAV only has to take a few seconds to localize any nearby targets before issuing a “resume” command to the robot, thus impacting the robot run time only a small amount. The Cooperative Reactive approach takes the longest (about 18% longer than Independent), since during each run there are at least a few targets that the UAV requires assistance

localizing; this assistance sometimes requires a long round-trip for the robot from its search path to wherever the UAV requests assistance. It is worth noting that the longer Task Completion Time of the Cooperative Reactive approach is directly related to the maximum speed of the standard Pioneer robot; with a much faster robot, the Task Completion Time of Cooperative Reactive should approach that of Cooperative Planned.

Figure 5(c) shows the results measured in terms of Aggregate System Run Time⁴. This value is interesting because it can be used as a very rough approximation of the energy consumption of the team. In these experiments, the robot is in nearly constant motion, and, while one might think that the UAV would be idle when waiting for the robot in the Cooperative scenarios, the reality of blimp navigation is that in any real-world environment, there is a constant need to expend a non-negligible amount of energy simply to remain in a stationary position, as drafts and temperature changes are constantly affecting the UAV’s position. The Independent value represents the sum of the times that it takes each team member to navigate and search along its respective path. The Cooperative-Planned approach has a much higher Aggregate Run Time than the other approaches (nearly 75% more than Independent), since this approach involves the UAV and robot navigating their paths in synchronization, processing each CJ at the same time. Because of this, the much faster UAV must wait on the robot at each point; therefore, the UAV’s run time is much longer in the Cooperative-Planned scenario. The Cooperative-Reactive approach is around 28% longer on average than the Independent, but much less than Cooperative-Planned. This is because the CJs are generated reactively as the UAV encounters a target outside of the EDI range; thus, while the UAV does have to wait for the robot to arrive and assist at some points, it does not have as many points at which to wait.

²Assuming a normal distribution of these results, the Student’s t-test confirms that the differences between the Independent and the two cooperative approaches are statistically significant, with confidence levels of 95% and 97.5%, respectively. There is no significant difference between the two cooperative approaches.

³The Student’s t-test shows that all pairs of scenarios are significantly different, with a confidence of 99.5%, assuming a normal distribution.

⁴The Student’s t-test shows that the differences between all pairs of scenarios are statistically different, with a confidence level above 99.95%, assuming a normal distribution.

To compare the approaches in terms of these findings, the Cooperative Reactive and Planned approaches both perform the task of target localization equally well in terms of Target Localization Accuracy, and better than the Independent approach. In essence, if there is any benefit at all to be had from cooperation, the cooperative behaviors will obviously realize this benefit whereas the Independent will not. In terms of Task Completion Time, Cooperative Planned performs slightly better than Cooperative Reactive, with the caveat that the magnitude of the performance increase is directly related to the speed difference between the robots. However, if one takes into account the energy expended by the robots by considering the Aggregate Run Time, Cooperative Reactive is superior to Cooperative Planned. When choosing an approach for a particular situation, it may also be necessary to consider the relative values placed on human time and robot time. For example, in some situations, human time may be highly valued, so an approach that minimizes the human-controlled UAV run time may be favored.

These results suggest that in any scenario that (1) involves a task that would benefit from cooperation; (2) human and robot time are valued equally; and (3) there is a demand for the most rapid completion of the task possible, then the Cooperative Planned approach should be employed. On the other hand, if (a) energy consumption or the aggregate run time of the robots is a concern; or (b) in a situation where a planned approach is not possible, then a Cooperative Reactive approach might be best. In a scenario where human time is valued much higher than robot time, a Cooperative Reactive approach might be preferred because it reduces Aggregate Run Time by reducing the human-controlled UAV run time. We also observe that for some mission configurations, especially those with a high number of critical junctures, the increased cost of a reactive cooperation should be taken into account, leading to the possibility that the Planned Cooperative approach is best.

VI. CONCLUSIONS

In this work we defined the concepts of environmentally dependent information and critical junctures. These concepts were used to identify critical points at which tightly-coupled cooperation benefits a heterogeneous team of less-than-fully-capable robots performing a weakly-decomposable task. This work shows how to identify these critical points algorithmically, both *a priori* and reactively during task execution. In this work, the need for tightly-coupled cooperation was driven by environmentally dependent information, which caused certain robot/vehicle perceptual capabilities to vary across the environment. We have validated the benefit of cooperation at these critical junctures in real-world experimentation. While both *a priori* Planned cooperation and Reactive cooperation have been shown to perform identically well in terms of the target localization task, we have examined differences that distinguish Planned versus Reactive, including (1) Task Completion Time, where Planned slightly outperforms Reactive and (2) Aggregate System Run Time, by which Reactive performs better than Planned.

In future work, we plan to extend the concept of Environmentally Dependent Information to a broader class of information constraints that affect robot capabilities. Additionally, we plan to investigate the coupling of the motion planning for multi-robot teams with the need for tightly-coupled cooperation during critical regions, to determine if tractable methods for approximating coupled solutions are possible. We further plan to conduct extensive simulations to confirm our physical robot experiments, and to conduct experiments and simulations with larger, fully autonomous robot teams.

REFERENCES

- [1] S. Botelho and R. Alami. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1234–1239, 1999.
- [2] M. B. Dias and A. Stentz. A free market architecture for distributed control of multirobot system. In *Proceedings of the International Conference on Intelligent Autonomous System*, 2000.
- [3] C.-H. Fua and S S Ge. COBOS: Cooperative backoff adaptive scheme for multirobot task allocation. *IEEE Transactions on Robotics*, 21(6):1168–1178, 2005.
- [4] B. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.
- [5] B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. Technical Report 04-45, University of Massachusetts, 2004.
- [6] N. R. Jennings and C. Kirkwood-Watts. Distributed mobile robotics by the method of dynamic teams. In *Proceedings of Fourth International Symposium on Distributed Autonomous Robotic Systems (DARS 1998)*, 1998.
- [7] N. Kalra, D. Ferguson, and A. Stentz. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1182–1189, 2005.
- [8] R. Lundh, L. Karlsson, and A. Saffiotti. Automatic configuration of multi-robot systems: Planning for multiple steps. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 2008.
- [9] L. E. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.
- [10] L. E. Parker. Chapter 40: Multiple mobile robot systems. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*. Springer, 2008.
- [11] L. E. Parker and F. Tang. Building multi-robot coalitions through automated task solution synthesis. *Proceedings of the IEEE, special issue on Multi-Robot Systems*, 94(7):1289–1305, 2006.
- [12] C. M. Reardon. Using automated task solution synthesis to generate critical junctures for management of planned and reactive cooperation between a human-controlled blimp and an autonomous ground robot, M.S. thesis, Dept. of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, 2008.
- [13] B. Satterfield, H. Choxi, and D. Houston. Role Based Operations. In L. E. Parker, A. Schultz, and F. Schneider, editors, *Multi-robot Systems Volume III: From Swarms to Intelligent Automata*. Springer, 2005.
- [14] O. Shehory. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, 1998.
- [15] R. Simmons, S. Singh, D. Hershberger, J. Ramos, and T. Smith. First results in the coordination of heterogeneous robots for large-scale assembly. In *Proc. of the ISER '00 Seventh International Symposium on Experimental Robotics*, 2000.
- [16] R. G. Smith. The Contract Net Protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12), December 1980.
- [17] L. Vig and J. A. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649, 2006.
- [18] R. Zlot and A. Stentz. Complex task allocation for multiple robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1527–1534, 2005.