The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

IQ-ASyMTRe: Synthesizing Coalition Formation and Execution for Tightly-Coupled Multirobot Tasks

Yu Zhang and Lynne E. Parker

Abstract— **This paper presents the IQ-ASyMTRe architecture, which is aimed to address both coalition formation and execution for tightly-coupled multirobot tasks in a single framework. Many task allocation algorithms have been previously proposed without explicitly enabling the sharing of robot capabilities. Inspired by information invariant theory, ASyMTRe was introduced which enables the sharing of sensory and computational capabilities by allowing information to flow among different robots via communication. However, ASyMTRe does not provide a solution for how a coalition should satisfy sensor constraints introduced by the sharing of capabilities while executing the assigned task. Furthermore, conversions among different information types[1] are hardcoded, which limits the flexibility of ASyMTRe. Moreover, relationships between entities (e.g., robots) and information types are not explicitly captured, which may produce infeasible solutions from the start, as the defined information type may not correspond well to the current environment settings. The new architecture introduces a complete definition of information type to guarantee the feasibility of solutions; it also explicitly models information conversions. Inspired by our previous work, IQ-ASyMTRe uses measures of information quality to guide robot coalitions to satisfy sensor constraints (introduced by capability sharing) while executing tasks, thus providing a complete and general solution. We demonstrate the capability of the approach both in simulation and on physical robots to form and execute coalitions that share sensory information to achieve tightly-coupled tasks.**

I. INTRODUCTION & RELATED WORK

Tightly-coupled multirobot tasks (MTs) [3] refer to tasks that require the cooperation of multiple robots. The underlying assumption is that individual robots may not have all the capabilities required to accomplish a task; instead, multiple robots need to share their capabilities. In order to determine appropriate coalitions of robots from the robot team, many architectures [2] [9] [4] reason about coalitions by dividing tasks into subtasks or roles that individual robots can perform. For example, in a robot insertion task [8], a supervisor robot provides visual information to guide the implementor robot to execute the insertion. One issue with this approach is that subtasks or roles have to be predefined for different tasks, as it is not feasible to define all possible subtasks or roles for arbitrary tasks. Another issue is that robots sometimes need to share sensory and computational capabilities in order to accomplish a task. Defining robot capabilities in terms of subtasks or roles is often too coarse, and is often not helpful for enabling capability sharing.

The work of [2] achieves fault-tolerant task allocation with uncertain task specifications using a cooperative backoff adaptive scheme. It defines robot capabilities in terms of basic high-level tasks and allows robots to use task execution histories to detect and adapt to device imperfections and malfunctions. However, it does not explicitly enable capability sharing among robots and hence the flexibility of the approach is limited. The work of [4] introduces a market-based framework for tight coordination in multirobot tasks. Passive coordination is used to quickly produce solutions for local robots, while active coordination is used to produce complex solutions via coordination among teammates. Although the robots need to tightly coordinate through communication, robot capabilities are still not shared. The work of [9] adapts a multiagent coalition formation algorithm to the multirobot domain and aims at maximizing the overall utility of the system. Again, robot capabilities are not shared among the robot team members. Inspired by information invariant theory [1], ASyMTRe [5] was proposed which divides robot capabilities into sensory and computational level schemas and motor schemas, and forms coalitions by reasoning about the information required for activating these schemas. Connections among communication schemas enable information to flow through the system to where it is required and capability sharing is implicitly achieved. The ASyMTRe architecture enables a finer resource sharing among heterogeneous robots through communication and is hence more flexible. Thus, we build upon this approach in the research described in this paper.

Another issue with tightly-coupled multirobot tasks is how to physically execute the tasks while sharing capabilities. This is especially challenging when the sharing of capabilities within the coalitions introduces sensor constraints. In a cooperative robot navigation task for example, the leader robot shares its localization capability by communicating its global position to the follower robot. In order to localize, the follower robot is constrained to keep the leader within its field of view (FOV) so that it can compute the relative position of the leader. Only with both pieces of information would the follower be able to compute its own global position and then navigate to a desired goal position. Our previous work [10] develops an information quality based approach that provides a flexible way for maintaining sensor constraints within the coalitions while executing tasks.

In order to achieve a complete solution for tightly-coupled multirobot tasks, we need both ASyMTRe for forming coali-

This material is based upon work supported by the National Science Foundation under Grant No. 0812117.

Yu Zhang and Lynne Parker are with the Distributed Intelligence Laboratory in the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996-3450, USA, {yzhang51,parker}@eecs.utk.edu.

[1] Information types differ from data types (e.g., integer) in that they have semantic meanings (e.g., a robot's global position).

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

tions as well as our previous approach for sensor constraint satisfaction to physically execute the task. However, the combination of the two approaches is not a simple addition. Several links are missing and challenging. First of all, the definition of information type in ASyMTRe is not sufficient for task execution. Knowing that a robot can sense the relative positions of other robot teammates and that a teammate has a localization capability does not necessarily imply a solution, since the two robots may not be within the line of sight of each other. The definition of information type has to be complete in the sense that the presence of certain information types guarantees a solution. This requires an extended architecture in which the reasoning process can distinguish among different environment settings (e.g., where the robots are located) and provide feasible coalition solutions accordingly. Furthermore, information conversion, which is one of the key points of information invariant theory, is not explicitly modeled. Without explicit modeling, application-specific code needs to be designed in advance for different tasks. Moreover, during the task execution, when the information quality model identifies the situations in which the satisfaction of sensor constraints may potentially fail for certain coalitions, the robots must be able to search and reform coalitions in a timely manner. Thus, dynamic environments should also be taken into account.

This paper presents a new architecture that combines ASyMTRe and our previous information quality based approach into a complete solution for tightly-coupled multirobot tasks. This approach, which we call IQ-ASyMTRe, introduces the capability of dynamic-environment reasoning for dynamic coalition formation. To the best of our knowledge, this is the first attempt to create a complete and general solution for tightly-coupled multirobot tasks addressing both the formation of coalitions, as well as the execution of those coalitions to address a specific task. After a review of ASyMTRe and our previous work (Sections II and III), we explain the new IQ-ASyMTRe architecture in detail (Section IV). Afterwards, we present experiments in simulation and with physical robots to demonstrate some aspects of the new architecture (Section V). Finally, we conclude with some discussion and plans for future work (Section VI).

II. ASyMTRe

The ASyMTRe architecture [5] defines basic building blocks of robot capabilities to be collections of environmental sensors (ESs), perceptual schemas (PSs), motor schemas (MSs), and communication schemas (CSs). A robot, R_i, can be represented by $R_i = (ES^i, S^i)$, where ES^i is a set of environmental sensors installed on R_i, and S^i is the set of schemas that are pre-programmed into R_i. According to a set of rules, connections are created among the schemas on the robots to allow information to flow through the system. A set of information types $F = \{F_1, F_2, ...\}$ is introduced to label the inputs and outputs of each schema. A schema can be activated if its inputs are satisfied either by sensors or the outputs of other schemas with the same information types. The ultimate goal is to activate the required MSs on the robot coalition team members to accomplish the task.

For reasoning about coalitions, ASyMTRe uses an anytime algorithm to search the entire solution space and return the best solution found so far according to predefined cost measures. One of the most important contributions of ASyMTRe is that it enables a finer resource sharing by dividing robot capabilities into smaller chunks (i.e., schemas), and reasons about how these schemas can be connected. Information can flow through the system to where it is required such that capability sharing is implicitly enabled through communication. ASyMTRe has been shown to provide more flexibility for achieving tightly-coupled multirobot tasks. (Please refer to [5] for more details of ASyMTRe.) However, several limitations of ASyMTRe prevent its application to arbitrary physical multirobot tasks. One of the most obvious limitations is that ASyMTRe does not provide a solution for how a coalition should satisfy the sensor constraints introduced by the sharing of capabilities while executing the assigned task. This issue has been addressed in our prior work [10], which introduces an information quality based approach that models sensor constraints explicitly and provides a general method for satisfying the constraints, such that the coalitions can be maintained during the execution.

III. THE INFORMATION QUALITY APPROACH

In order to flexibly control robots while maintaining sensor constraints through the environment, a mechanism is needed to quantify the utility of these constraints being satisfied. For this purpose, inspired by [6], our previous work [10] uses measures of *information quality* to describe the fitness for using the information in the current environment settings, including the retrievability and usefulness of the information. We divide the computation of the information quality measurement into two parts. First, the sensor quality measurement is used to describe the retrievability and/or usefulness of the information using the sensor based on sensor characteristics; second, environmental influence is modeled as a weight to the sensor quality measurement to capture how the environment affects the retrievability and/or usefulness of the information. For example, in a robot tracking task, the information quality for tracking is higher when the target is in the center of the tracker robot's FOV than if the target is on the edge of the FOV; in a box pushing task, when the box is pushed too close to an obstacle, the information quality for tracking the box is reduced since the obstacle can potentially block the view of the box. To handle the application-specific environmental influence, instead of exact geometric reasoning, we compute the influence based on environment samples. To compute a motion command for execution, we sample the motion space into motion vectors using a motion model. Then for each motion vector, we predict the information quality measurement for the robot's predicted configuration after executing the vector. Afterwards, we simply choose the motion vector that leads to the best information quality (for more details, please see [10]). A constraint model is also implemented to provide alternative coalitions when certain sensor constraints are unsatisfiable. For example, in a cooperative robot navigation task, when the view of the leader robot is blocked, the follower robot can relax the original constraint to follow another robot which is also following the leader. The relaxation of constraints enables robot coalitions to dynamically change formations. This capability is important for robots to execute tasks in different environment settings. For example, to navigate through narrow hallways, the robots need to change from the original formation into a line formation while maintaining the relaxed constraints for achieving the same task.

Experimental results have illustrated that this approach works for applications such as robot tracking and cooperative navigation tasks. However, one obvious limitation of this approach is that the constraint model has to be designed in advance to control how the constraints should be relaxed for alternative coalitions. The dynamic coalition formation capability of ASyMTRe can reason about alternative coalitions based on the information required; the alternative coalitions can be used to control the constraint relaxation process. Hence, a general solution for constraint relaxation is achieved by combining the two approaches.

IV. THE IQ-ASyMTRe ARCHITECTURE

We next describe the IQ-ASyMTRe architecture, including a discussion of the incorporation of information quality. We then present the solution space for the new architecture and discuss the advantages and disadvantages of this approach. Coalition formation and execution are discussed afterwards.

A. Extensions of Representation

The incompleteness of the definition of information type in ASyMTRe is due to the fact that the relationships between entities and information types are not specifically captured. Entities can be locations, agents or anything that can be identified in the environment. Intuitively, information must be specified with a set of referents. For example, the information of *r_A's global position* is meaningless without specifying r_A. Furthermore, ASyMTRe does not explicitly model conversions among information types. Our approach to overcome these limitations is discussed in the following.

1) Referents for Information Types: For a particular information type $F_i \in F$ (see section II), there are N_i referents associated with it. We represent the information type by $F_i(\mathit{Ref}_{1:N_i})$, where Ref_j is used to refer to the jth referent for the information type F_i. Each referent, Ref_j, can be instantiated to a particular entity or remain uninstantiated for future instantiations. Fully instantiated instances of information types represent actual information that can be used. Partially instantiated instances of information types represent a class of information. For example, $F_G(X)$ can be the global position information of any entity that X is instantiated to. Specifying referents for the information types makes the definition of information type complete. For example, the relative position information type F_R has two referents. After instantiating the two referents to robot r_A and r_B respectively, the reference of information $F_R(r_A, r_B)$ has a unique meaning (i.e., r_A's position relative to r_B), no matter how the information is retrieved or used.

TABLE I
RULE PS USED IN COOPERATIVE ROBOT NAVIGATION TASK

Rule PS	Description
$F_G(X) + F_R(Y, X) \rightarrow F_G(Y)$	global + relative → global
$F_R(Y, X) \rightarrow F_R(X, Y)$	relative → relative
$F_R(X, Z) + F_R(Y, Z) \rightarrow F_R(X, Y)$	relative + relative → relative

The advantage of a complete definition for information type is that infeasible coalitions can be excluded from consideration. For example, one way for robot r_A to compute its own global position is to acquire another robot's global position and the relative position of the robot to itself. Given that r_A has a camera to detect robot teammates and robot r_B has a localization sensor, ASyMTRe would believe that a coalition with r_B is a solution even though r_B is not in r_A's camera's FOV. IQ-ASyMTRe would exclude the coalition with r_B since the information $F_R(r_B, r_A)$ is not retrievable.

2) Information Rule PS: Information rule PS is a special kind of perceptual schema that specifically models the conversions among information types. ASyMTRe handles the conversions by hardcoding them into different perceptual schemas. The introduction of these rule PSs is especially useful when information types are specified with referents. With referents instantiated to different entities, two instances of the same information type represent two different pieces of information, while they are considered to be the same in ASyMTRe. This distinction is important when multiple pieces of information of the same information type can be used in the conversions to create other information. For example, from the relative position of robot r_B to robot r_A and robot r_C to robot r_A, we can compute the relative position of r_B to r_C. The rule PS representation for this conversion is $F_R(r_B, r_A) + F_R(r_C, r_A) \rightarrow F_R(r_B, r_C)$. ASyMTRe cannot represent such conversions, which may lead to the loss of possible solutions. Table I shows some basic rule PSs. The conversions are general since the referents can be instantiated to different entities. However, we require that the same referent labels be instantiated to the same entities.

B. Incorporation of Information Quality

Our previous work [10] introduces measures of information quality to describe the fitness for using the information in the current environment settings. Information quality measurements are used to guide motion selections such that the robots can execute tasks through the environment while maintaining certain levels of information quality for sensor constraints within the coalitions. To incorporate information quality, a function is defined for each information type that a sensor can produce, which returns the information quality measurement with respect to the current environment settings in the sensor's FOV. $Q_i(\mathit{Conf}_{1:N_i})$ represents the information quality measurement for F_i given the current environment settings (i.e., work space settings and the current configurations for entities $\mathit{Ref}_{1:N_i}$) in the sensor's FOV, whereas *Conf*$_j$ is the current configuration of the entity that *Ref*$_j$ is instantiated to. The information type must be fully instantiated for the measurement to return non-zero values.

C. Solution Space and Potential Solutions

Potential solutions represent the possible alternative ways that schemas can be connected to achieve a certain task. Potential solutions can be extracted from the solution space. In order to create the solution space, the IQ-ASyMTRe reasoning algorithm starts from the goal motor schemas (MSs) and checks all schemas that can provide inputs for the MSs. The algorithm then checks recursively for the inputs of those schemas until the path either ends in a conflict state (i.e., the same referent is instantiated to different entities) or in a terminal state (i.e., communication schema (CS) or environmental sensor (ES) that can be the source of the required information). One important difference between our IQ-ASyMTRe reasoning algorithm and the reasoning algorithm in ASyMTRe is that we allow multiple activations of the same schemas for individual potential solutions. This is due to the fact that in IQ-ASyMTRe, information can be different for the same information types with different instantiations of the referents. Hence, the same schemas can be used to accept and produce different information having the same information types. This may seem at first to lead to a clumsy representation of the solution space. However, the fact that the information types can be partially instantiated enables us to represent the space more concisely; meanwhile, instantiations of the information types can also be specific enough to exclude infeasible solutions.

As an example, Figure 1 shows a typical solution space for computing the global position information. Each schema or sensor acts as a connection node in the solution space. The algorithm processes from left to right, from top to bottom; while information flows from right to left, from bottom to top as the arrows indicate. Two special connection nodes are created to indicate the *AND* and *OR* relationships among nodes in different levels. This solution space encodes two potential solutions. While the ASyMTRe architecture needs to introduce two schemas that distinctly work for these two scenarios, the rule PS we introduce in IQ-ASyMTRe provides a more comprehensive and extensible approach. Figure 1 also demonstrates the representation power of the partially instantiated information types. In this figure, the referent X can be instantiated to any entity as long as all references of X are instantiated consistently.

A solution space is usually composed of more than one potential solution. The potential solutions can be extracted and represented as:

$$PoS_h^k = (S_1^k, S_2^k, ..., F_1^k, F_2^k, ..., Map^k) \quad (1)$$

where PoS_h^k is the hth potential solution for robot r_k. $S_1^k, S_2^k, ...$ are the schemas that need to be activated and $F_1^k, F_2^k, ...$ are the pieces of information that need to be communicated. Map^k records the current instantiations for how the referents are instantiated to entities, and is used to determine whether future instantiations are valid or not.

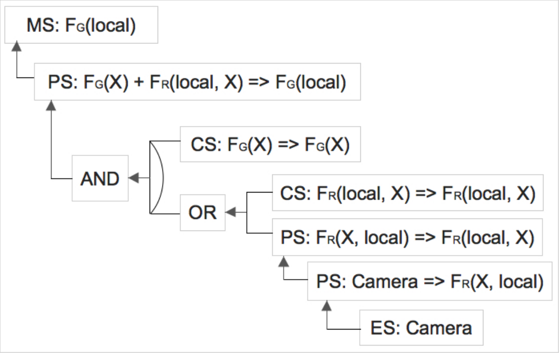

MS: FG(local)
PS: FG(X) + FR(local, X) => FG(local)
AND
CS: FG(X) => FG(X)
OR
CS: FR(local, X) => FR(local, X)
PS: FR(X, local) => FR(local, X)
PS: Camera => FR(X, local)
ES: Camera

Fig. 1. A typical solution space for a robot to obtain its global position with only a camera sensor. The referent *local* refers to the robot itself. The solution space encodes two solutions. One solution is to have another robot send over its global position (CS: $F_G(X) \Rightarrow F_G(X)$) and (AND) use the camera sensor to sense the relative position of the robot to itself (PS: Camera $\Rightarrow F_R(X, local)$). A rule PS (PS: $F_R(Y, X) \Rightarrow F_R(X, Y)$) is used to convert $F_R(X, local)$ to $F_R(local, X)$. The other solution (OR) is to have both pieces of information (CS: $F_G(X) \Rightarrow F_G(X)$ and CS: $F_R(local, X) \Rightarrow F_R(local, X)$) sent over by another robot.

In order to simplify the solution space, apart from the basic connection constraint that the inputs and outputs of the connecting schemas must have the same information types, additional constraints are created. First, we require that information of the same type appearing further to the right of a reasoning path must not have fewer or the same uninstantiated referents than the set of uninstantiated referents from the information further to the left. Secondly, the *Localness in Reasoning* constraint states that no schema connection except for CS should be created if none of the referents for the information type of concern is instantiated to the local entity. Given this constraint, for example, robot r_A would not directly provide the relative position of robot r_B to robot r_C, even though r_A can compute the information from the relative positions of r_B and r_C to itself. However, the required information for the computation would be available upon request. The *External Communication* constraint states that a CS can be used only when not all referents are instantiated to the local entity for the information type. For example, if robot r_A needs its global information, $F_G(r_A)$, it cannot request it directly. Instead, it must first seek other ways (e.g., computing it from another robot's global position and the relative position of the robot) to obtain the information.

D. Coalition Formation & Execution

Coalition formation is performed in a distributed manner. We use the same request-and-wait negotiation protocol for coalition formation as the distributed version of ASyMTRe uses. However, this method often comes with latency. (Analysis of the latency will be performed in our future work.) For robots to maneuver through dynamic environments, this latency is especially hazardous since environments can change unexpectedly and rapidly. When sensor constraints are unsatisfiable in certain situations, the robots must react in real time to search for alternative coalitions. The fact that the IQ-ASyMTRe architecture enables dynamic-environment reasoning by eliminating infeasible coalitions may alleviate the problem, since the delay for processing these infeasible coalitions is avoided. Dynamic-environment reasoning is not possible in ASyMTRe since entities related with the information types are not specified. Upon request, if the robot being requested has the information, it will notify the request sender of the availability. When there are multiple possible coalitions for a potential solution, the possible coalitions can be sorted by the quality of the coalitions (which we call *coalition quality* henceforth), which can be computed using the information quality measurements for the information types related to the coalitions. After the coalition is determined, the requesting robot would send a request to set up the coalition (for details, see [5]). During the execution, in cases that the coalition quality drops continuously due to changes in the environment that may potentially lead to unsatisfiable sensor constraints, the coalition formation process can be activated dynamically to search for alternative solutions.

For every robot in a certain coalition, a goal command vector is computed. When the coalition quality is above a risk threshold, the robots directly execute the goal command vector. When the coalition quality falls below the risk threshold, the robots would try to improve the quality. If the coalition quality deteriorates below a danger threshold, the coalition formation process would be re-activated to search for alternative solutions.

E. The Algorithm

The overall algorithm for IQ-ASyMTRe is shown in Algorithm 1. The algorithm starts with collecting sensory information related to the current potential solution and information communicated to it. It then checks the current potential solution against the information collected and sets up a coalition when possible. If the information is not sufficient, then the robot will send out requests for the missing pieces of information and check the solution again later. One important characteristic of the algorithm, which differs from ASyMTRe, is that the information collected using sensors and through communication is used in the planning (coalition formation) phase to exclude infeasible coalitions when relevant information required is not retrievable.

V. SIMULATION & EXPERIMENTAL RESULTS

In simulation, we first show the solution spaces for different robot configurations in a cooperative robot navigation task. Afterwards, we present the capability of dynamic-environment reasoning enabled by the IQ-ASyMTRe architectue and explain how such a capability is used to eliminate infeasible coalitions. We then show the architecture in action for a robot navigation task. Physical robot experiments are provided afterwards. For all the following experiments, unless specifically mentioned, we use only the first and second rule PSs given in Table I.

A. Simulations

In a robot navigation task, each robot needs to localize in order to navigate to the goal position. For this purpose, robots can use sensors onboard and/or request help from other more capable robots. The robot can also help other robots upon request. We next show the solution spaces in the robot navigation task for different robot configurations, and using different rule PSs as listed in Table I. To order the potential solutions in the solution space, we assign costs to schemas that reflect the complexity of using particular schemas. Due to space limitations, we omit the listing of perceptual schemas that are required for each solution. We assign a cost of 0.5 to PSs (including rule PS) since they solely involve computation. The costs of ES and CS are

Algorithm 1 The Algorithm for IQ-ASyMTRe

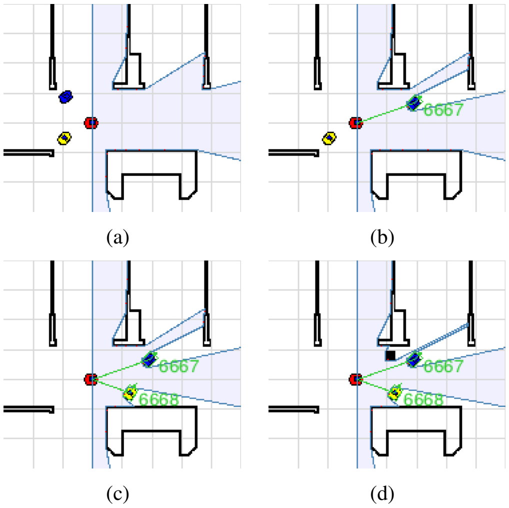
```
Reason about the MSs required for the task and create the
solution space using the reasoning algorithm.
Extract potential solutions and order solutions according
to the predefined cost measures for schemas.
while true do
  COLLECT: collect sensory information related to the
  current potential solution.
  Collect information communicated by other robots.
  Process requests from other robots.
  Share requested information when available.
  if a coalition is set up then
    goto EXECUTION.
  end if
  Retrieve the next solution from the ordered list.
  Check the solution against all information collected.
  if the information collected is sufficient then
    if some information needs to be communicated then
      Request to set up a coalition with related robots.
      if the coalition is set up then
        Reset the next solution to be the first solution.
        goto EXECUTION.
      else
        goto COLLECT.
      end if
    else
      Set up a coalition by itself and goto EXECUTION.
    end if
  else
    Send requests for the missing information and set the
    next solution to be the current solution and wait for
    some time, then go to COLLECT.
  end if
  EXECUTION:
  if coalition quality is below the danger threshold then
    Break the coalition and goto COLLECT.
  else if coalition quality is below the risk threshold then
    Execute the motion vector produced by the informa-
    tion quality model and goto COLLECT.
  else
    Execute the goal command.
  end if
end while
```

TABLE II
SOLUTION SPACE USING FIRST TWO RULE PS'S IN TABLE I

Fiducial Only	Fiducial & Laser
	1. ES:$F_G(local)$
1. CS:$F_G(X)$, ES:$F_R(X, local)$	2. CS:$F_G(X)$, ES:$F_R(X, local)$
2. CS:$F_G(X)$, CS:$F_R(local, X)$	3. CS:$F_G(X)$, CS:$F_R(local, X)$
3. CS:$F_G(X)$, CS:$F_R(X, local)$	4. CS:$F_G(X)$, CS:$F_R(X, local)$
4. CS:$F_G(X)$, CS:$F_R(local, X)$	5. CS:$F_G(X)$, CS:$F_R(local, X)$

TABLE III
SOLUTION SPACE USING ALL RULE PS'S IN TABLE I

Fiducial & Laser
1. ES:$F_G(local)$
2. CS:$F_G(X)$, ES:$F_R(X, local)$
3. CS:$F_G(X)$, CS:$F_R(local, X)$
4. CS:$F_G(X)$, CS:$F_R(X, local)$
5. CS:$F_G(X)$, CS:$F_R(local, X)$
6. CS:$F_G(X)$, CS:$F_R(local, Y)$, CS:$F_R(X, Y)$
7. CS:$F_G(X)$, CS:$F_R(X, Y)$, CS:$F_R(local, Y)$



(a) (b) (c) (d)

Fig. 2. (a) No robots are within the FOV of the red robot. (b) Only the blue robot is within the FOV of the red robot. (c) Both the blue and yellow robots are within the FOV of the red robot. (d) The same robot configurations as in (c) with an obstacle, shown in black.

assigned to 1 and 2 respectively. The cost of a CS is set higher to discourage the robots from requesting help.

Tables II and III list the ordered potential solutions for robots with different configurations and using different rule PSs in the robot navigation task. In these tables, ES represents the use of environmental sensors. Fiducial sensors can sense the relative positions to special markers and are often used to retrieve the relative position information between robots, while laser sensors can support localization using a Monte Carlo method. CS represents the use of communication schemas to transfer information from one robot to another. The ultimate goal in such tasks is to obtain the global position, $F_G(local)$, such that the robot can localize and navigate to the goal. Table II shows the potential solutions for robots with a fiducial sensor only and with a laser and a fiducial sensor. Table II shows that the reasoning process generates different solution spaces for different robot configurations. ASyMTRe needs to build a PS for each potential solution listed in the tables, since conversions among information types are not specifically modeled and must be hardcoded into different PSs. Such an approach would work for cases with few potential solutions as in Table II. However, for cases with more complicated solution spaces or more potential solutions, as in Table III, a more flexible solution as our approach is required. One note is that the ASyMTRe architecture actually cannot be used to produce solutions 6 and 7 in Table III. This is because ASyMTRe cannot distinguish information of the same information type, which is due to the lack of referents associated with the information types. One of the disadvantages of the new solution space, however, is that there can potentially be duplicate solutions. Notice that solutions 3 and 5 (as well as solutions 6 and 7) in Table III represent the same solution even though they are generated from different branches in the solution space and have different costs. In future work, we will analyze the impact of the duplicates on the searching performance. To resolve the issue, we can always design a post-processing algorithm to remove the duplicates.

To illustrate the capability of dynamic-environment reasoning for coalition formation, we provide four scenarios for a robot in a cooperative robot navigation task to determine which coalitions to set up, as shown in Figure 2. In all scenarios, we assume that the red robot has only a fiducial sensor for sensing relative positions of other robots. Figure 2(a) shows a scenario in which no other robot is within the FOV of the red robot. The reasoning algorithm searches the potential solutions from the left column in Table II, from top to bottom, until it finds a feasible solution. For solution 1, the algorithm successfully reasons about the current situation and moves to solution 2, since no other robot is within its FOV. ASyMTRe, on the other hand, cannot identify that one related entity (i.e., another robot) for the information type F_R is not available, as it does not associate entities with the information types. (Actually, ASyMTRe would treat all four scenarios equally, assuming that all other robots are within the red robot's FOV.) Solution 2 then requires the robot to broadcast a message requesting the information of $F_G(X) + F_R(r_{red}, X)$. The referent *local* is instantiated to the red robot itself. In Figure 2(b), the blue robot is within the FOV of the red robot and the reasoning process identifies the feasibility of solution 1 and broadcasts a request for $F_G(r_{blue})$. In Figure 2(c), the red robot understands that both the blue and yellow robots are within its FOV, so it broadcasts a message requesting $F_G(r_{blue})$ or $F_G(r_{yellow})$. If both the blue and yellow robots reply with their global positions, the red robot would choose the one with the better coalition quality (refer to section IV.D). The robots in Figure 2(d) are in the same configurations as in Figure 2(c), except that an obstacle is added such that the blue robot can potentially be blocked. The reasoning process ignores the coalition with the blue robot due to the low coalition quality and requests only $F_G(r_{yellow})$. This experiment shows the dynamic-environment reasoning capability for coalition formation enabled by the combined approach.

We next show in simulation a cooperative robot navigation task that combines the previous perspectives and demonstrates how IQ-ASyMTRe completely solves a tightly-coupled multirobot task from the beginning to end. In this experiment, three robots are to navigate to the same goal position. Among them, only the red robot can use the laser sensor to localize. The other two robots have a fiducial sensor to sense the relative positions of other teammates and a laser sensor for environment sampling. They cannot use the laser sensors for localization since the perceptual schema for localization (i.e., software for Monte Carlo localization) is disabled. Figure 3 shows snapshots from the experiment with the execution time shown in seconds. The robots start with the initial configurations as in Figure 3(a). Both follower robots search for solutions from the left column of Table II, from top to bottom. As they both can sense the leader robot's relative position using the fiducial sensors, they both set up a coalition with the leader using solution 1. The coalitions navigate through the environment in a triangle formation as shown in Figure 3(b). When the environment becomes narrower due to obstacles on both sides of the corridor, there is risk of collision for the yellow robot to navigate through without adjusting its moving direction. Furthermore, the obstacle nearest to the yellow robot also reduces its coalition quality with the leader due to potential blocking risk. The blue follower robot goes through first as the obstacles are further away. The yellow robot, on the other hand, first tries to improve the coalition quality after it drops to below the risk threshold, while avoiding obstacles. However, as the coalition quality deteriorates due to the potential blocking risk from the blue robot and obstacles, the yellow robot breaks the coalition with the leader robot and searches the solution space from the start again. It then realizes that the blue robot is in its FOV and tries to set up a coalition with it. Since the blue robot knows that it can localize with the help from the leader robot, the coalition is confirmed and the blue robot becomes the leader of the yellow robot as Figures 3(c) and 3(d) show. One interesting note is that the yellow robot does not need to know how the blue robot achieves localization. The rest of the snapshots show that the robot coalitions navigate through the environment in a line formation and reach the goal position successfully. Notice that after they start navigating in a line formation, unless the yellow robot's coalition with the blue robot is at risk and it regains sight of the leader, the yellow robot would not try nor is it necessary to rebuild coalition with the leader robot. A supplementary video file for this entire task execution is also available for downloading.

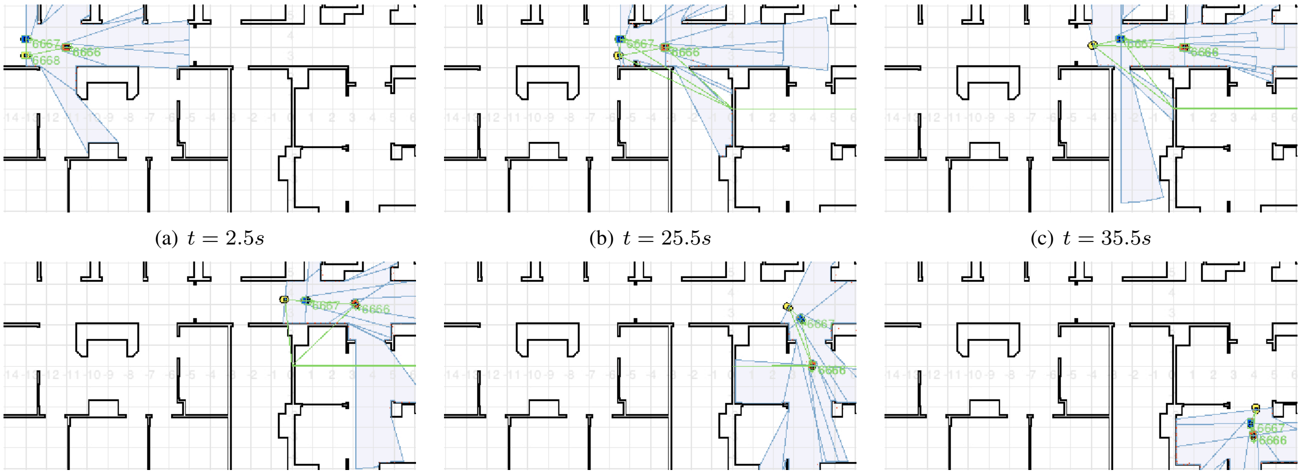
This experiment presents an important capability which we name dynamic-environment reasoning for dynamic coalition formation. Our previous work [10] achieves dynamic coalition control by hardcoding the alternative coalitions in the constraint model. The IQ-ASyMTRe architecture uses the dynamic coalition formation capability of ASyMTRe for reasoning about alternative coalitions based purely on the information required, hence providing a more general and flexible solution. The information quality model is essential since it helps the robots maintain the sensor constraints and determines when a coalition needs to be reformed.

B. Physical Experiments

To demonstrate the flexibility and robustness of the IQ-ASyMTRe architecture with physical robots, we design two scenarios for the cooperative robot navigation task in which two robots have the same goal position. Both robots have a camera sensor pointing forward to determine the relative position of other robots and only one of them (the helper, labeled '1' in Figures 4 and 5) has a localization capability. The motivation is that the helper robot may be in front of the requesting robot (i.e., the robot that needs help), or the helper robot may be behind the requesting robot, in which case the requesting robot cannot detect the helper in order to compute its relative position. ASyMTRe would consider these two cases equal and hence would provide the same coalition solution, which is for the helper robot to communicate its global position to the requesting robot. However, in the latter case, even with the helper robot's global position, the requesting robot cannot compute its global position since the helper's relative position is not retrievable, as the helper robot is behind it. The snapshots for the execution of these two scenarios are shown in Figures 4 and 5. To show that the robots can set up coalitions in a timely manner, we start the two robots without delay for the initial coalition setup. For both scenarios, the helper robot starts execution first since it is self-sufficient for this task. In Figure 4, the helper starts in the front and the coalition is set up for the helper robot to communicate its global position to the requesting robot. In Figure 5, the helper starts at the back and the IQ-ASyMTRe architecture successfully reasons about the situation; the coalition is set up for the helper robot to communicate both its global position (via laser) and the relative position (via camera) to the requesting robot. After the coalitions are set up, the robots execute the coalitions while maintaining the sensor constraints and successfully reach the goal positions. The elimination of infeasible coalitions through dynamic-environment reasoning makes the coalition formation more robust to various situations. The coalition formation and execution are all performed dynamically and in real time. This experiment shows that the dynamic-environment reasoning capability can distinguish among different environment settings (including robot configurations) and provide feasible coalitions accordingly.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents the IQ-ASyMTRe approach, which is aimed for a complete solution for tightly-coupled multi-robot tasks in which robots share capabilities. This approach combines an extended ASyMTRe architecture and our previous information quality based approach such that coalition formation and coalition execution are accomplished within



(a) $t = 2.5s$

(b) $t = 25.5s$

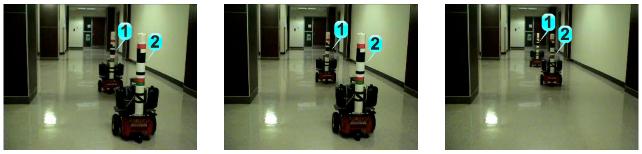
(c) $t = 35.5s$

(d) $t = 49.0s$

(e) $t = 65.5s$

(f) $t = 89.5s$

Fig. 3. (a) Initial configurations. (b) Both follow robots set up a coalition with the leader (red) and robots navigate in a triangle formation. (c) The environment becomes narrower due to obstacles and influences the yellow robot. The yellow robot breaks the coalition with the leader and sets up a coalition with the blue robot. (d) Robots navigate through the environment in a line formation. (e) and (f) Robots navigate around the corner and reach the goal position.

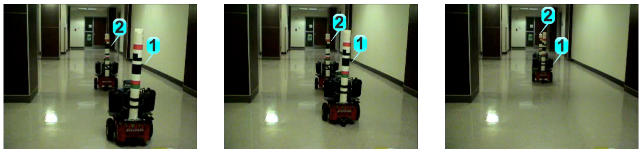


(a)

(b)

(c)

Fig. 4. (a) Initial configurations with the helper robot (labeled '1') in the front. (b) The helper robot goes to the goal while the requesting robot is trying to set up a coalition with the helper. (c) The coalition is set up and the robots navigate through the environment with the helper robot communicating its global position to the requesting robot at the back.



(a)

(b)

(c)

Fig. 5. (a) Initial configurations with the helper robot (labeled '1') at the back. (b) The helper robot starts first while the requesting robot is trying to set up a coalition with the helper. (c) The coalition is set up and the robots navigate with the helper robot communicating both the global information of itself and the relative position to the requesting robot in the front.

the same architecture, with sensor constraints introduced by capability sharing within the coalitions. This approach introduces the capability of dynamic-environment reasoning for dynamic coalition formation and demonstrates how such a capability enables autonomous and dynamic cooperative control. To the best of our knowledge, this is the first attempt to create a complete and general solution for tightly-coupled multirobot tasks that involves capability sharing.

In future work, we plan to apply IQ-ASyMTRe to more complex tasks and with more robots, in order to further prove the feasibility and show the scalability of this approach. The properties of the new architecture will be analyzed in detail to include the real-time computational requirements of this approach. We will also provide a formal analysis of the performance of coalition formation and compare it with other approaches. Other interesting aspects include developing more complete definitions of information quality, as well as addressing sensor fusion [7].

References

[1] B.R. Donald, J. Jennings, and D. Rus. Information invariants for distributed manipulation. In *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 431–459, 1997.

[2] 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.

[3] B.P. Gerkey and M.J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, September 2004.

[4] N. Kalra, D. Ferguson, and A. Stentz. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proc. of the IEEE Int'l. Conf. on Robotics and Automation*, 2005.

[5] L.E. Parker and F. Tang. Building multirobot coalitions through automated task solution synthesis. *Proc. of the IEEE*, 94(7):1289–1305, July 2006.

[6] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July and October 1948.

[7] A.W. Stroupe, M.C. Martin, and T. Balch. Distributed sensor fusion for object position estimation by multi-robot systems. In *Proc. of the IEEE Int'l. Conf. on Robotics and Automation*, volume 2, pages 1092–1098 vol.2, 2001.

[8] V. Sujan and S. Dubowsky. Visually guided cooperative robot actions based on information quality. *Autonomous Robots*, 19(1):89–110, July 2005.

[9] L. Vig and J.A. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649, 2006.

[10] Y. Zhang and L.E. Parker. A general information quality based approach for satisfying sensor constraints in multirobot tasks. In *IEEE International Conference on Robotics and Automation*, 2010.