Tightly-coupled coordination of multi-robot systems for Mars exploration

Terry Huntsberger, Paolo Pirjanian, Ashitey Trebi-Ollennu, Hari Das, Hrand Aghazarian, Anthony Ganino, Mike Garrett, Sanjay S. Joshi, Paul S. Schenker

Abstract— **Site construction operations by autonomous robotic systems are essential for a sustained robotic presence and human habitation on Mars. We report on the development of a software/hardware framework for Cooperating multiple robots performing such tightly coordinated tasks. This work builds on our earlier research into autonomous planetary rovers and robot arms. Here, we seek to closely coordinate the mobility and manipulation of multiple robots to perform an example of a site construction operation -- the autonomous deployment of a planetary power station. There are numerous technical challenges in the task including the mobile handling of extended objects, as well as cooperative transport/navigation of such objects over natural, unpredictable terrain. In support of this work we have developed an enabling distributed control architecture called CAMPOUT (Control Architecture for Multi-robot Planetary Outposts) wherein integrated multi-robot mobility and control mechanisms are derived as group compositions and coordination of more basic behaviors under a task-level multi-agent planner. CAMPOUT includes the necessary group behaviors and communication mechanisms for coordinated/cooperative control of heterogeneous robotic platforms. In this paper, we describe CAMPOUT, and its application to ongoing physical experiments at the Jet Propulsion Lab in Pasadena, CA where two rovers carry an extended payload (transport phase of a photovoltaic tent deployment mission) over uneven, natural terrain. To our knowledge, this is one of the first efforts that have been successfully undertaken in such an environment.**

Index Terms—**tight coordination, distributed control architecture, multiple mobile robots, robot outposts**

I. Background

Future robotic exploration of Mars will likely entail cooperative activity of multiple robots such as those described in [1]. These missions will probably include the use of multiple, heterogeneous, mobile robotic platforms for infrastructure deployment. A high degree of autonomy is necessary, since the delays of the long communication path to Mars limit the amount of teleoperation that is possible. The cooperating robots will work as "crews" of coordinated intelligent agents, carrying out site preparations, site maintenance functions, and remote science investigations, eventually in partnership with human co-habitants of such planetary outposts. We report the preliminary development and experimentation with such robotic system concepts, building on prior JPL work in autonornus planetary rovers and robots, e.g., our recent development of the MarsArm, LSR, SRR, and FIDO platforms [2-4].

We are currently investigating the robotic needs for the deployment of a modular solar photovoltaic (PV) tent array such as that specified by Colozza [5]. Colozza's study demonstrated that a nearly constant power profile can be realized by a tent array of standard silicon PV cells. Such a PV tent array would be difficult to deploy using a solitary robot, since the modules are 5 meters long and would represent a considerable challenge for precision placement. Two cooperating robots can perform the task using the sequence of steps shown in Figure 1. These steps were designed keeping in mind the mass and power constraints consistent for a mobile robotic platform on the Martian surface.

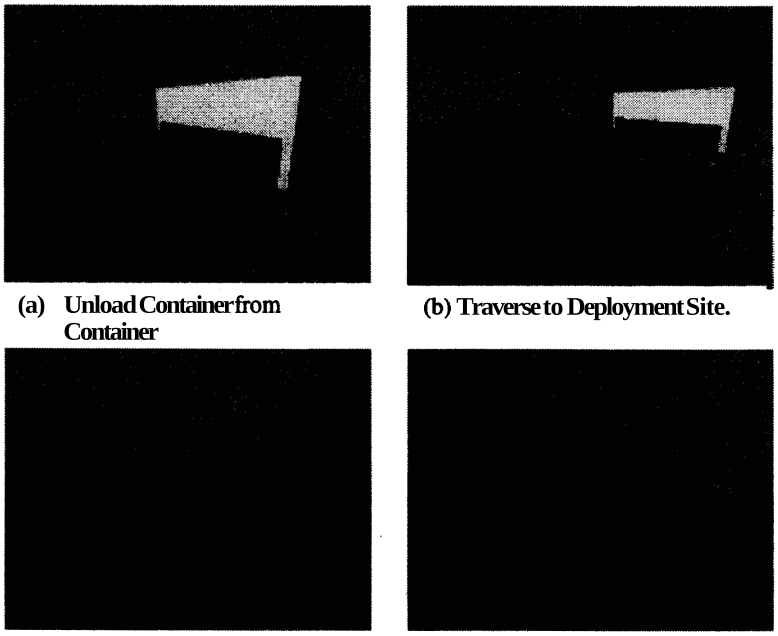
II. Introduction

Our research focuses on multi-robot cooperation for tasks that inherently require tight coordination under strict physical constraints. Such tightly coupled coordination tasks are characterized by constraints *imposed* on the activities of one robot as a function of the state of others. Most work on multi-robot system has to date been limited to tasks such as collective estimation [6, 7] (e.g., mapping and localization) cooperative foraging [9-11,521, and cooperative box pushing [12-14], where tight coordination of the activities of the robots is not required. Collective estimation and foraging tasks can be performed independently by each robot and usually do not require a tight coordination of activities. Cooperative box pushing requires tighter cooperation but can be accomplished by turn-taking schemes where each robot can alternate in pushing one end of the box towards a goal. But since the box rests on a surface, the activities of the robots do not need to be closely orchestrated simultaneously. The task of cooperative mobile object grasping, manipulation, and handling [15-26] (e.g., lifting and carrying, not pushing, a piano up the stairs) requires tight and simultaneous (vs. turn-taking) coordination of each robot in order to maintain grip of the object while manipulating/handling it. State of the art is currently limited to indoor lab demonstrations on a level floor often using omni-directional mobile manipulators with multiple degrees of freedom. There are numerous challenges in this prototypical task including the cooperative manipulative

Manuscript received April 5,2001. The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

All authors are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. (address all correspondence to Terry Huntsberger, telephone: 818-354-5794, e-mail: Terry.Huntsberger@jpl.nasa.gov).

acquisition of extended objects from a container storage depot, the cooperative transport of such a container to the power array construction site, and the physical deployment of the container into the array. The main robotics requirements for this task include coordinated grasping and navigation over open terrain by two or more cooperating **robots.** Navigation over unconstrained terrain will prove **to** be a significant challenge, especially with Mars-like rovers with severe holonomic constraints. In addition, there must be accurate localization of the **robots** as the PV tent containers **are** unloaded from a container storage unit (CSU) and delivered to the site, since damage to the solar tents could otherwise occur.



(a) Unload Container from Container

(b) Traverse to Deployment Site.

(*c*) Position and Open Container.

(d) Deploy Tent.

Figurel. Four step sequence for a PV tent array deployment. PV tent storage container is 5 m in length and is not well handled by a single robot.

Two features of **this** scenario **are** particularly salient in our ongoing work, which emphasizes the ***"traversephase"*** of the task **1)** cooperative sensor-based autonomous traverse of two kinematically linked rovers across natural, uncertain terrain; and 2) distributed force-motion control of **this** non-holonomic extended platform (each rover having a gimbal-mounted gripper that is instrumented for force-position in all axes, and compliance in one). We have developed a distributed control architecture for tightly-coupled operation of multiple robots, wherein mobility and control functions are derived as group compositions and coordination of more basic behaviors under the downward task decomposition of a multi-agent planner. The architecture is extensible and scales freely with regard to the behavioral mechanisms and protocols it can host and fuse, re-mappable inter-robot communications it can support, and the overall ability to functionally integrate heterogeneous, multi-purpose platforms.

We report on **this** Control Architecture for Multi-robot Planetary **Outposts** ***(CAMPOUT)*** **[28],** and physical experimentation to date with two **rovers** carrying a model payload over natural terrain. Robotic **outposts, as** based in ***robot work crews*** **(RWC), require** close integration of mechanical subsystems, rich multi-sensory **data streams, and** networked control architectures. Such **an** outpost will, by definition, be a collection of evolving heterogeneous robotic platforms, under frequently varying control and communications protocols due to the wide range of **tasks** (some unforeseeable) that they will be required to do. The control architecture must therefore not be a "point design," but rather, extensible and expandable. Tasks **may** include not only site preparation and maintenance functions, but also support of science goals (instrument deployments, sample transport, in-field rendezvous, etc.). ***Section*** *III* contrasts our research to related work; ***Sections*** ZV and V overview our control architecture, Section VI details the distributed control used for container transport and **some** experimental studies; and we conclude with ***Section*** VZZ **.**

III. RELATED WORK

A key requirement for a robotic outpost is the capability of manipulating/transporting extended structural elements necessary for construction and maintenance tasks. These elements will be **of** a size that is not easily handled by a single mobile platform. For example, the container length for a single element of a **PV** tent array **as** detailed by Colozza **[5]** is projected **to** be 5 meters.

Recently, robotics researchers have investigated transportation of large extended objects using autonomous cooperating or coordinated multiple robots (wherein the latter term, ***coordinated,*** infers tight coupling of the physical platforms' kinematics and dynamical parameters). Compliant control for multiple mobile robots is very different from that of a single mobile robot. First, the compliance frame is implicitly time varying, and second, the environment is not static because the contact occurs or is maintained while all robots are in motion. In general, we note that many approaches reported for cooperative robot motion do not generalize; they may not consider activity within a natural terrain, versus an idealized environment (lab floor), and/or fail to maintain an explicit continuous closed loop coordination of joint robot activities under physical constraints (rather, using time-sequenced, iterative actions of the independent robots to partially address global task constraints). Activities may be cooperative in a spatial sense, but not necessarily coordinated below a strategic level as to platform kinematics and inertial/dynamical interactions. In the more specific literature noted **below,** several researchers describe decentralized approaches, **with** centralized, supervisory control schemes for transportation **of** large objects using multiple mobile robots.

Vinay et al. **[26]** presented simulation results of two mobile robots transporting a long object. Lagrange techniques were utilized to develop a state space model for two wheeled mobile robots compliantly coupled to a common payload. State feedback control technique was used to decouple the system into five subsystems, thus simplifying and facilitating the supervisory control design.

Hisashi et al. **[19]** also presented simulation and experimental results of two cooperative mobile manipulators transporting a payload on an uneven ground. In the reported experiments, the robots and the payload consisted of three moving tables driven by ball screws. Mechanical compliance is achieved by locking some of the joints of the manipulator and making the rest free. Simple joint position control laws are employed to accomplish compliant control between. the mobile manipulators without the need for explicit communication.

Khatib et al. **[15]** proposed a somewhat more general decentralized cooperative control algorithm for multiple mobile manipulators using an augmented object and a virtual linkage model. The augmented object is used to describe the system's closed chain dynamics. The virtual link model is used to characterize and synthesis control laws for internal forces in a multi-arm systems. However, the algorithm requires an explicit and not always realistically achieved communication between the platforms. The experimental results presented demonstrate the potential effectiveness of the control scheme.

Hara et al. **[20]** presented a cooperative transportation control scheme for two quadruped robots transporting a long payload. The quadruped robot locomotion **is** based on a vibration model in walking. A decentralized control scheme is developed based on a "leader-follower." Several experimental results are presented, such as transporting the load over stairs. In reflecting on these developments and motivation for our own work, we note that previous studies of robotic requirements for Mars robotic outposts **[38]** indicate that increased levels of autonomy and more generalized payload handling capabilities than have been reported to date will be needed for habitat construction and surface infrastructure support on planetary surfaces. The applications challenge is further exacerbated by the unstructured nature of the planetary surface environment (often unpredictable with respect to **both** character of perceptual artifacts and poorly modeled nature **of** vehicle-surface interactions), and the extended duration and changing goals/priorities of such missions. **A** generalized ***behavior-based control,*** as described next, appears to offer a practical level of flexibility, autonomy, and computational economy **[32, 39]** for preliminary design of such space-targeted technologies and systems.

IV. CAMPOUT

A control architecture defines the abstract design **of** a class of agents: the set of structural components in which perception, reasoning, and action occur; the specific functionality and interface of each component, and the interconnection topology between components. This definition identifies a **number of** architectural issues (such as perception action components, interfaces and topology between components etc.) that are useful in specifying and describing a particular architecture. Robot control architectures can be broadly characterized as deliberative (based on .planning), reactive (tight coupling between sensing and actuation), or a hybrid blend of the two.

In a nutshell, CAMPOUT is a distributed control architecture based on a multi-agent or behavior-based methodology, wherein higher-level functionality is composed by coordination of more basic behaviors under the downward task decomposition of a multi-agent planner (see Figure **2). In** its current implementation, task decomposition is done by hand and encoded in a script/plan, which is then executed by the agents. We are currently working towards extending CAMPOUT with automated planning of joint team activities.

Robotics is a highly multidisciplinary field, and requires efficient integration of many components (perception, mapping, localization, control, learning, etc.) that use different representations, frameworks, and paradigms (classical control theory, AI planners, estimation theory, data fusion, computer vision, utility theory, decision theory, fuzzy logic, multiple objective decision making etc.). CAMPOUT provides the infrastructure, tools, and guidelines that consolidate a number of diverse techniques to allow the efficient use and integration of these components for meaningful interaction and operation. This is facilitated through a few elementary architectural mechanisms for ***behavior representation, behavior***

composition, and ***group coordination,*** and the interfaces between these. These mechanisms and a framework with guidelines for describing systems define the core of **CAMPOUT. CAMPOUT** is thus extensible and scales freely with regard to behavioral mechanisms and protocols it can host and **fuse,** re-mappable inter-robot communications it can support, and the overall ability to functionally integrate heterogeneous, multi-purpose platforms.

A. Behavior representation

In **our** architectural methodology we formalize a behavior, ***b,*** as a mapping, $b: P^* \times X \rightarrow [0; 1]$, that relates each percept sequence $p \in P^*$ and action $x \in X$ pair, (p, x), to a preference value that reflects the action's desirability. The percept describes possible (processed or raw) sensory input and the N-dimensional action space is defined to be a finite set of alternative actions. The described mapping assigns to each action $x \in X$ a preference, where the **most** desired actions are assigned **1** and undesired actions are assigned 0, from that behaviors point **of** view. Note that **this** definition of a behavior does not dictate how the mapping is to be implemented but provides a general recipe for a behavior with a well-defined interface (useful when composing behaviors regardless of their roles in a behavior hierarchy). This representation does not exclude implementation using a look-up-table,,a finite state machine, a neural network, an expert system, control laws (such **as** PID etc.), or any other approach for that matter. Note also that **this** representation does not restrict us to reactive behaviors since it could have internal state. In CAMPOUT **this** representation is implemented using an N-dimensional array, which contains the desirability values recommended by a behavior.

B. Behavior composition

Behavior composition refers to the mechanisms **used** for building higher-level behaviors by combining lower-level ones. **A** major issue in the design of behavior-based control systems is the formulation of effective mechanisms for coordination of the behaviors' activities into strategies for rational and coherent behavior. In order to do this, it **is** necessary to address the problem of the coordination of the activities of the behaviors using behavior coordination mechanisms (BCMs) ***so*** as coordinate the activities of lower-level behaviors within the context of a high-level behavior's task and objective. ***An*** explicit design goal of CAMPOUT has been to support not one but **an** arbitrary number of BCMs. BCMs can be divided into two main classes: arbitration and command. For a detailed overview, discussion, and comparison of behavior coordination mechanisms **see** [28].

C. Behavior coordination mechanisms

If behaviors are viewed as operands, then **BCMs are** the operators used to combine behaviors into higher-level behaviors. In **this** section we describe the BCMs that **are** readily available in CAMPOUT for behavior composition. BCMs can be divided into two complementary classes: arbitration and command fusion. ***Arbitration mechanisms*** select one behavior, from a group of competing ones, and.give it ultimate control of the system (the robot) until the next selection cycle. **This** approach is suitable for arbitrating between the set of active behaviors in accord with **the** system's changing objectives and requirements under varying conditions. It can focus the use of scarce system resources (sensory, computational, etc.) on **tasks** that **are** considered to be relevant. ***CAMPOUT*** implements the following arbitration mechanisms:

- Priority-based arbitration: which is a subsumptive-style, priority-based arbitration **mechanism,** where behaviors with higher priorities are allowed **to**

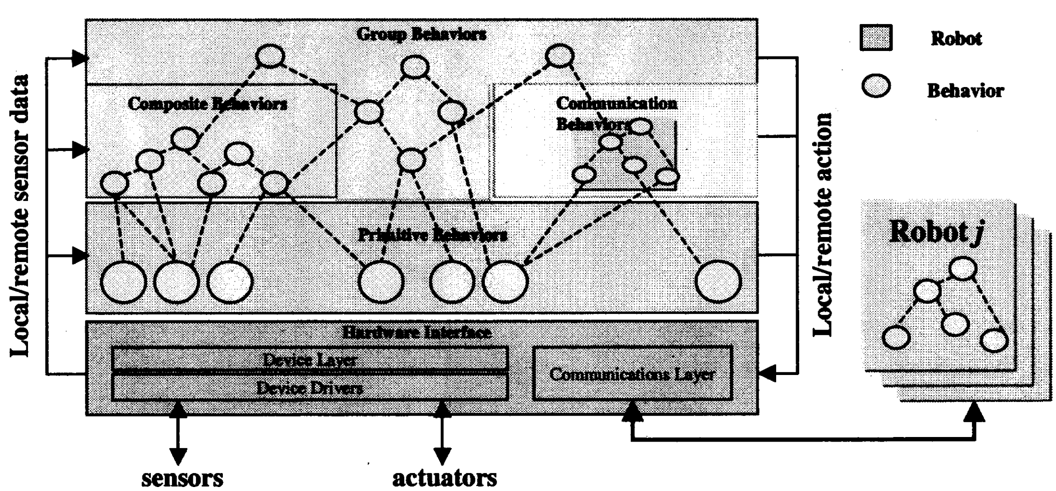


*Figure 2 Schematic overview of **CAMPOUT** and its hierarchical organization in terms of primitive behaviors, composite behaviors built from primitive behaviors and group behaviors that are composed from coordination of behaviors across multiple robots. Each robot runs an instance of this architecture and coordinates activities through group behaviors, which is facilitated through the communications behaviors.*

suppress the output of behaviors with lower priorities.

- State-based arbitration: which is based on the Discrete Event Systems **(DES)** formalism [30], and is suitable for behvior sequencing.

Command fusion mechanisms combine recommendations from multiple behaviors to form a control action that represents their consensus. This approach provides for a coordination scheme that allows all behaviors to simultaneously contribute to the control of the system in a cooperative rather than a competitive manner, which makes them suitable for tightly coupled tasks that require spatio-temporal coordination of activities. CAMPOUT provides a number of complementary mechanisms for fusion:

- Voting techniques interpret the output of each behavior as votes for or against possible actions and the action with the maximum weighted sum of votes is selected. CAMPOUT implements a DAMN-style [34] voting algorithm based on BISMARC [52]
- Fuzzy command fusion mechanisms (see [40-41]) use fuzzy logic and inference to formalize the action selection processes. In addition, fuzzy approaches enable a new class of Coordination mechanisms denoted context-dependent blending, introduced to robotics by Saffiotti, Ruspini, and Konolige in [40], which allow for weighted combination of behaviors. The implementation in CAMPOUT follows that described in [40].
- Multiple objective behavior fusion provides a formal approach to behavior coordination based on multiple objective decision theory [42]. Action selection consists of selecting an action that makes the best trade-off between the task objectives and which satisfies the behavioral objectives as much as possible.

D. Group coordination

In order to cooperate and collectively contribute to a common task, the robots must cooperate and coordinate their activities. Behavior coordination is basically concerned with resolving or managing conflicts between mutually exclusive alternatives and between behavioral objectives. Group coordination in CAMPOUT (see Figure 3) is treated as ***the coordination of multiple distributed behaviors, across a network of robots,*** where more than one decision maker is present.

Behavior coordination in multi-robot systems has received relatively little attention. One approach proposed in [14] uses inhibition and suppression across a network of heterogeneous robots augmented with motivational behaviors that can trigger behavior invocation based on some internal parameters that measure progress. A similar approach was proposed in the ALLYU architecture **[REFS]** which uses port arbitration as the main mechanism for multi-robot behavior coordination. Both these approaches can be viewed as the extension, of subsumptive-style arbitration to multi-robot coordination.

Recently, work in progress is investigating the extension of the 3T architecture to multi-robot coordination [59]. The above approaches as well as most multi-robot architectures including ACTRESS, GOFER, SWARM [56, 57, 58] invariably have two things in common. First, multi-robot coordination mechanisms are limited to only one approach, and second this approach mostly tends to be arbitration rather than a command fusion scheme. Arbitration limits cooperation to execution of tasks that are either independent/parallel or loosely coupled, turn-taking tasks. We maintain that arbitration and command fusion mechanisms are complementary and a system implementation will typically make use of both.

The philosophy in CAMPOUT is that the architecture should support both arbitration and fusion. Further, we favor mechanisms that are based on formal theories to support a sound approach to description and validation of system behavior. This is an important characteristic of CAMPOUT, since it enables us to provide certain performance guarantees. We have chosen to support, but not limit the architecture to, arbitration using ALLIANCE and ALLYU's subsumptive-style and the discrete event system. Additionally, multi-objective behavior coordination is supported by CAMPOUT for command fusion [60].

The view taken in CAMPOUT is thus that multi-robot cooperation arises from coordination of multiple behaviors that reside on not one but a group of robots (see Figure 3). In order to support this view, BCMs must be extended to support multi-robot coordination. In [60] the multiple objective behavior coordination approach was extended to multi-robot applications.

CAMPOUT provides the infrastructure by which the distributed behaviors can interact through communication. The behaviors and hence the robots can communicate implicitly by interaction through the environment or explicitly using sensory feedback or explicit communication. The first

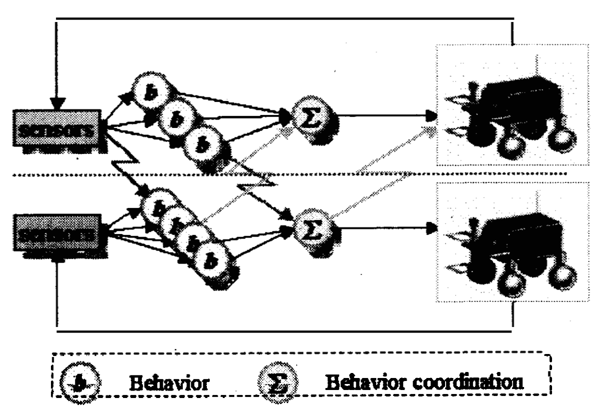


Figure 3 Networked robotics and resource sharing elements of CAMPOUT that enable definition of group coordination behaviors.

two approaches, interaction through the environment and sensory feedback, do not require any explicit form for architectural support as long as the robots have the necessary sensing capabilities to facilitate such interaction. These forms for interaction can be difficult and often computationally demanding, that is why most multi-robot systems resort to a form of explicit communication. CAMPOUT provides a rich and efficient infrastructure for explicit communication to facilitate multi-robot cooperation. Using this infrastructure, behaviors on one robot can interact with behaviors on other robots. In general the infrastructure defines a network of resources that can be shared among the robots. These resources include behaviors, sensors, and actuators. Thus a behavior on one robot can be driven by a sensor on another robot or even contribute to the control of a different robot. This idea is depicted in Figure **3,** where behavior composition can be achieved across several robots.

E. Communication Behaviors

In order to facilitate a group of robots to coordinate their activities and cooperate towards the accomplishment of a common task they may be required to communicate to share resources (e.g., sensors or, actuators), exchange information (e.g., state, percepts), synchronize their activities etc. The primitive and composite behaviors constitute the skill set that enable a robot to interact with and accomplish **tasks** in its environment. The skill set of the robot can be augmented by adding new primitive and/or composite behaviors. For cooperation and interaction with each other the robots are required to communicate thus they must have a set of basic behaviors for communication. Communication is not necessarily limited to explicit exchange of information via **some** soft of a data link but can also include visual, auditory, tactile, and other types of communication. For instance a robot can determine the relative position of another robot using cameras. Alternatively, the other robot could explicitly transmit its position within a global coordinate system.

CAMPOUT provides a broad set of facilities to foster such collaborative effort by offering a communications infrastructure. The current implementation of communications in CAMPOUT are provided using UNIX-style sockets. Another approach would be to base the communications on some general-purpose message-passing package such as MPI. However, such generality comes at significant overhead cost in efficiency, which we intend to avoid for the **types** of applications that CAMPOUT is designed for. The communications facilities consist of the following core functions:

- **Synchronization:** two main functions Signal (destination, sig) and Wait (source, sig) are used to send and wait for a signal to and from a given robot. **This pair** constitutes the facilities for synchronizing the activities of robots and/or behaviors.
- **Data exchange:** SendEvent (destination, event) and GetEvent (source, event) are used to send and receive an event, structure to and from a particular robot. The event structure can contain arbitrary data packages as contracted between the sender (source) and receiver (destination). For instance, it can be used to transmit a percept or raw sensor data from one robot to the other etc. E.g., robot **2** will be able to have a behavior that is being fed by the position of robot **1** (to, e.g., follow it).

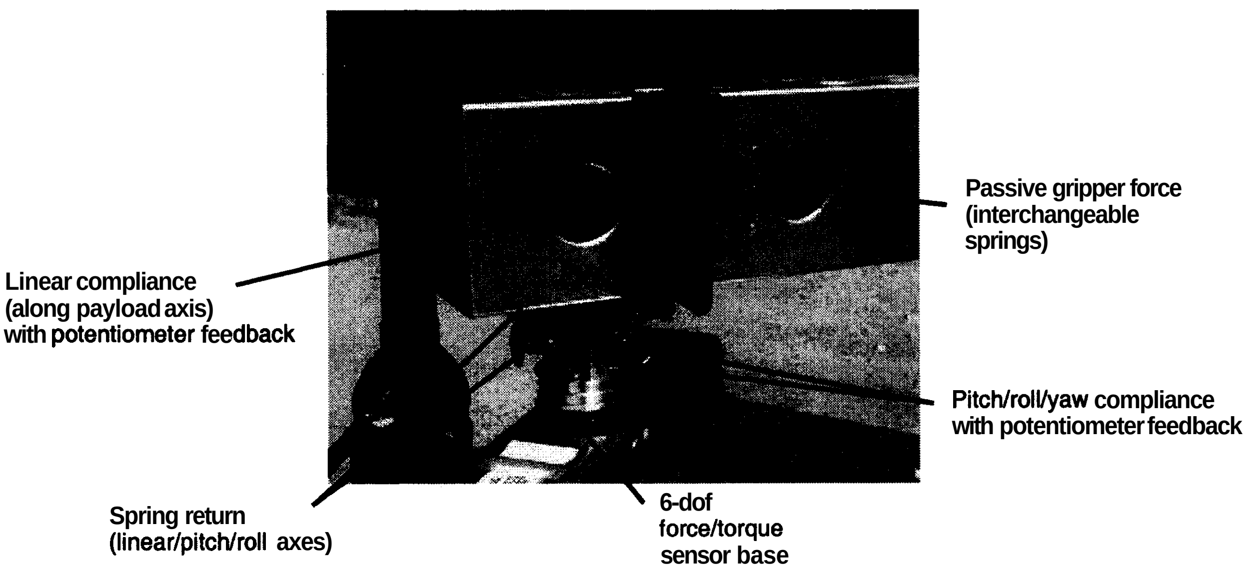


Figure 4 Compliant gimbal instrumented with position, angle, and force feedback sensors used to hold and sense the container.

Behavior exchange: SendObjective (destination, objective) and GetObjective (source, objective) are used to send and receive objective functions (multivalued behavior outputs) to and from a robot. Using these functions one can form a network of behaviors across a distributed group of robots.

These core set of communications facilities (and other convenience functions) support distributed sharing of resources such as sensors and state, as well as providing the necessary tools to form a network of behaviors spanning a group of physically distributed (but informationally connected) robots. The state of one robot (e.g., sensor readings or output from a behavior) can be used to affect/determine the behavior of another robot. All these facilities are showcased in the following coordinated transport task.

V. COORDINATED OBJECT TRANSPORTATION

We have selected a PV tent deployment scenario as our experimental test-bed for **CAMPOUT** (see Figure **4). A** study was done on the viability of a PV tent array for the power needs of a human habitat on Mars. The individual containers of the PV tent elements are 5 meters in length, 'so it would be **difficult** for a single **mobile** platform to manipulate **and** transport one to a deployment site. A four step process for the deployment of a single PV tent by two rovers is shown in Figure 1.

Our studies, this year, have concentrated on Step **2,** the traverse to the deployment site. We have retrofitted two of our Sample Return Rovers **(SRR** and SRR2K) with a gimbal **mounted** on a cross-brace between the shoulders. The gimbal is not actuated but is fully instrumented with **6** DOF force-torque sensors and **pots** and offers **some** mechanical compliance. The **gimbal** arrangement and two of **the** coordinated transport formations are shown in Figures **4** and 5 respectively.

The coordinated transport task in open, uneven terrain requires a tightly-coupled, close coordination of the activities of the two robots. This is accomplished by **some 20** behaviors, organized in a hierarchy as shown in Figure **3.** These behaviors are implemented and tied together using the mechanisms provided by **CAMPOUT.**

We have developed a **finite** state machine (FSM) description of the transport phases to emulate the planning level in **CAMPOUT,** since our first year task is not developing a planner. The four phases **are:** clear the container storage unit and assume the column transport formation, traverse to the staging area, survey the deployment area for a clear site, and traverse to the deployment site and align the container.

There are two main group behaviors used in **CAMPOUT** for this **task** ***Assume Formation*** and ***Approach Target.*** The ***Assume Formation*** group behavior is used to turn the formation to face the deployment target area and is invoked each time the heading error relative to the target is larger than a preset threshold. The ***Face Target*** behavior uses a visual target finding algorithm based on color-segmentation to localize the rovers for heading adjustments during the traverse step in the sequence. The ***Approach Target*** group behavior is used to safely **carry** the container towards the deployment area.

Key to these group behaviors is the notion of compliance by implicit communication through the shared container (payload carried by the two rovers) and explicit communication through communication behaviors for distributed resource sharing. The compliance behaviors assure safe handling of the container during turn and carry operations by constraining **and** adjusting the movements of the two rovers.

Note in Figure **3** that we distinguish between **some** of the behaviors as belonging to the leader and **some** to the follower. This is due to the fact that the rovers have heterogeneous capabilities; the lead rover **(SRR)** has color stereo cameras on an actuated **mast** that can be used for target detection and tracking. This capability is used by the ***Face Target*** behavior

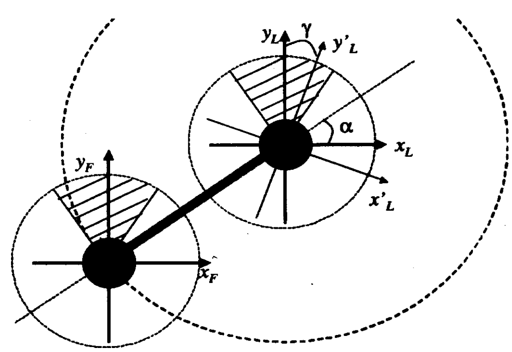


Figure 6 Formation between the two robots with follower on left and leader on right. The formation is defined by the angle α between the two robots. Desired heading is given by the relative heading angle γ. The shaded area on the lead rover is a safety zone where the container beam should not enter to prevent collisions with its mast.

to align the robots to the deployment area. The follower rover (SRR2K) does not posses this capability and can thus not achieve the goal of facing the target on its own and thus uses shadow behaviors to retrieve target information **from SRR.** In the current implementation, the roles of the rovers are pre-assigned as opposed to being dynamically determined through **some** negotiation or task allocation mechanism. **CAMPOUT** does not have any explicit mechanism for dynamic task allocation but its existing mechanisms could be used to provide **this** (although having dedicated mechanisms would make this less tedious).

In the following we describe the implementation of the main group behaviors using CAMPOUT.

A. *Assume Formation*

The Assume Formation group behavior is invoked to configure the two robots into a given formation, defined by the relative angle between them, *a,* and the relative angle towards the target, y *(see* Figures 7 and **8).** The *Face* Target behavior provides the angle to the target then the *Turn* group behavior reconfigures the formation to a desired one. Two constraints make this a challenging task. First, transformation between the current and target formations must ensure that the container is handled safely, i.e., the distance between the robots, ***d*,** should always remain within some tolerance margin, d_{min} ■ $d \le d_{max}$, determined by the distance between the grip points of the rovers, **L** (200 cm), and the longitudinal translation in the gimbal T_{gimbal} (±2 cm). I.e., L-$2T_{gimbal}$ ■ $d \le L+2T_{gimbal}$, which implies that the distance between the two rovers should be

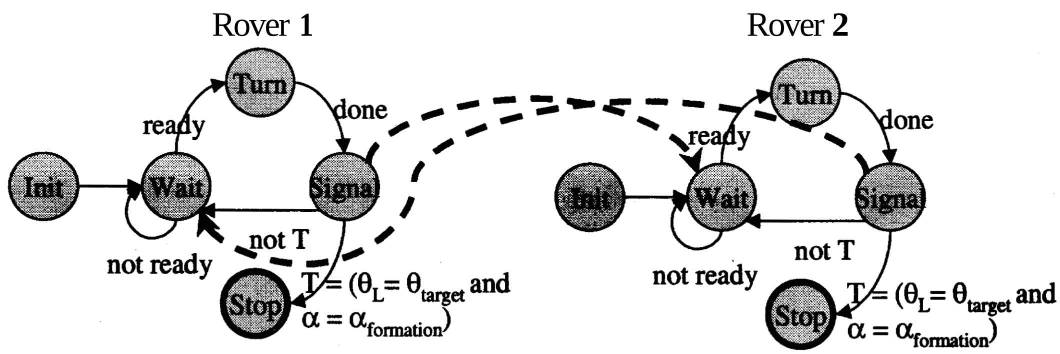


Figure 7 Distributed plan used for the compliant formation- keeping task. The arrow s represent events that cause transitions, and the dashed curves represent events caused by explicit communication of signals.

maintained within a margin of 8cm ($4T_{gimbal}$). A set **of** compliance behaviors, described later, **monitor** the state of the load and constrain the movement **of** the rovers to guarantee **this** requirement.

Second, it is required that the container does not collide **with** the mast on the lead rover *(see* Figures 2 and **4),** which could lead to damaging the mast, the gripper/gimbal, or the container, and/or dropping the container. The shaded area around the lead rover, in Figure *6,* indicates the safety zone (-35° to +35°) where the container beam cannot enter because it will then collide with the mast.

1) Centralized motion planning

This problem can be formulated as a constraint satisfaction search problem, with the following description:

- Configuration space is possible states of the formation defined by (θ_L, θ_F, *a),* where θ_L and θ_F are the absolute heading angle of the leader and follower respectively and *a* is the formation angle. The configuration space will exclude states where the beam intersects with the safety zone.
- The goal configuration is (e, θ_{target}, $\alpha_{formation}$), where θ_{target} is the heading angle to the target and $\alpha_{formation}$ is the desired formation angle.
- The operators for search correspond to the actions that the robots can perform and include: TurnInPlace(ϕ) and Ackerman(β) for each of the rovers. I.e., four **types** of operators exist, two for each rover. However, due to the strategy we have chosen for the compliance behaviors *(see* next section) we have constrained the motion of the leader to only TurnInPlace movements. Hence only three **types** of operators exist. LeaderTurnInPlace, FollowerTurnInPlace, and FollowerAckerman. Ackerman causes the follower to pivot around' the lead rover.

Using **some** search algorithm, a centralized module/planner can generate the sequence of actions (operators) that bring the system to the goal configuration. The execution of the sequence must command and synchronize the motion of each of the robots. A main advantage **of this** approach is that it is complete and it can generate **optimal** (e.g., shortest sequence of movements) solutions. **This** is, however, outweighed **by** its many disadvantages including it's. polynomial computational complexity due to a three-dimensional configuration space and a large branching factor determined by **the** number of operators. Further, **this** approach requires a centralized module/planner, which generates commands to control each of the **robots** and monitors their state during execution. This adds to the communication overhead. In short, this approach does not scale well in terms of computational complexity and communication overhead.

Figure 5 Transport of an extended object requiring the tightly-coupled coordination of multiple robots. (left) Column (diagonal) formation for long traverse. (right) Row formution for precision placement.

2) Decentralized motion generation and coordination

A more appropriate approach would be a decentralized scheme that scales well to other similar problems.

By a careful inspection of results generated by the above search method we observed a pattern in the action sequences, which was used to design a decentralized solution. The optimal solution generated, by the search algorithm had the following pattern:

1. The lead rover turns as far as possible until either θ_{target} is reached or it cannot move further due to the safety zone constraint. It turns in the direction that minimizes the difference between its current heading angle and the desired heading, θ_{target}.
2. The follower pivots until either $\alpha_{formation}$ is reached or it cannot move further due to the safety zone constraint. It pivots in the direction that minimizes the formation angle error.

This sequence will alternate the two until the goal configuration is reached. Note that once one rover moves it also frees the other rover from being constrained by the safety zone. In this way, incremental progress is made towards the goal configuration. Using this insight we constructed a distributed solution to the problem where the lead and follower rovers alternate in turning in place and pivoting until the goal configuration is reached:

- Lead rover performs:

TurnInPlace (max (min (α_{left}, θ_{left}), min (α_{right}, θ_{right}))),

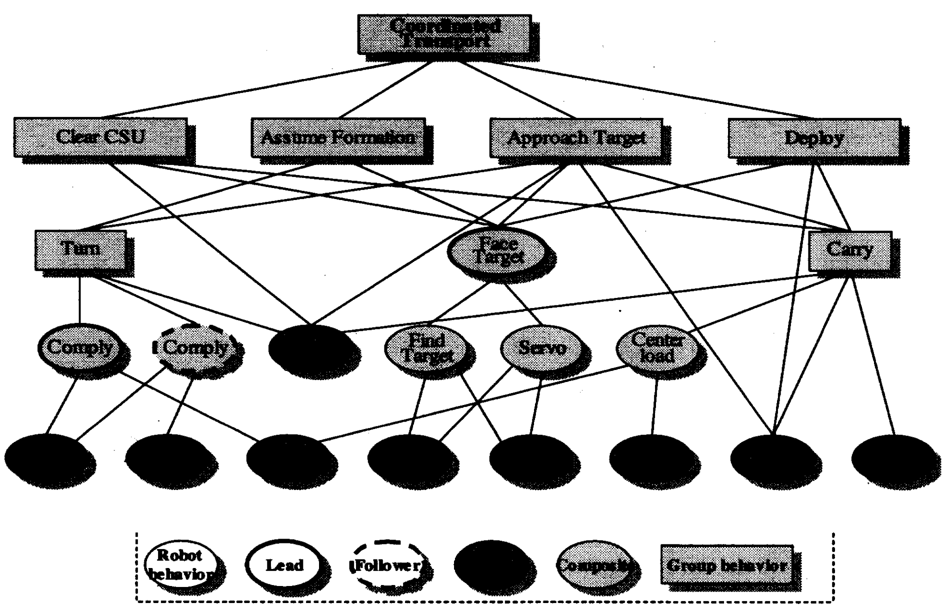


Figure 8 Behavior hierarchy describing the coordinated transport task Bubbles represent single robot behaviors and boxes represent group behaviors. The hierarchy shows how the behaviors are composedfrom lower-level behaviors.

where α_{left} (-35 degrees) and α_{right} (35 degrees) are the limit angles of the safety zone and θ_{left} and θ_{right} are the relative angle to θ_{target} in clock-wise and counter-clock-wise direction respectively.

- Follower rover performs:

Ackerman (max (min (α_{left}, $\alpha_{formation}$), min (α_{right}, $\alpha_{formation}$))),

where α_{left} and α_{right} are the limit angles of the safety zone and $\alpha_{formation}$ is the desired formation angle.

It can be shown that this strategy is complete, i.e., it will reach a solution if one is found. However, the strategy does not guarantee an optimal solution (minimum steps) although its solutions are typically close to optimal.

The lead and follower rovers need to synchronize their activities for two purposes: 1) termination of formation configuration and 2) turn-taking between leader turning in place and follower pivoting. The communication behavior Signal is used to perform this synchronization. The termination condition is when $\theta_L \doteq \theta_{target}$ and $a = \alpha_{formation}$. The lead rover can measure a locally from the gimbal pots and its heading θ_L based on visual feedback and position encoders. The follower can access θ_L using communication constructs of CAMPOUT and it can measure a locally. This behavior is implemented using a discrete event system or finite state machine action selection mechanism as shown in Figure 9. Note that the ***Turn*** behavior in the figure is either a ***TurnInPlace*** or an *Ackerman* for the leader and follower, respectively (see also behavior hierarchy in Figure 3).

The group turn behavior has two separate, distributed pieces one that runs on the leader and one on the follower *(see* Figure 7). Each of these consists of a composition of a set of primitive behaviors on each of the rovers, which use local sensory feedback for control. These distributed pieces of the group turn behavior are synchronized in part using explicit communication for invoking either the leader part or the follower part. This basically corresponds to a finite state machine with two states where state transitions are triggered by signal events using explicit communictions. Each state represents the activation of a (set of) behavior(s) and transitions between the behaviors are triggered by events as indicated on the arrows. These events can be generated either by perceptual feedback or explicit communication between robots. Note that a finite state machine is a behavior coordination mechanism that is supported by CAMPOUT for behavior arbitration. The states, represented by bubbles in the figure, correspond to primitive or composite behaviors implemented within CAMPOUT. *Also* the robots coordinate their activities through resource sharing, where the follower's behavior is driven by the leader's visual sensing of the target. The behavior coordination and communication mechanisms provided by CAMPOUT enable a seamless integration and coordination of behaviors across the robots.

Revisiting the formation keeping strategy, we see in the figure that one rover turns (using Turn behavior) until it is done (i.e., cannot turn further) then hands the token to the other rover by a signal and waits. The ***Turn*** behavior has different implementation for each of the rovers; for the follower it consists of Ackerman turns and for the leader it consists of turn-in-place (see Figure 8). The ***Wait*** behavior in each of the rovers consists of a number of behaviors including compliance behaviors. I.e., when the other rover starts moving/turning the waiting rover monitors the state of the load (through the sensors of the gimbal) and then triggers a compliance behavior to assure that the container is handled safely in accordance with the distance constraint describe above. This is accomplished by crabbing in the direction of the container in order to center the load (based on pot-meter readings) and to reduce the forces on the gimbals (based on the force-torque sensor readings). These compliance behaviors are explained in more detail in the next section.

B. Approach Target

The ***Approach Target*** behavior's objective is to safely carry the container towards the deployment area. It is composed of two main group behaviors ***Carry*** and ***Turn*** (see Figure 4). The main challenge of this behavior is to prevent the container from falling, which is achieved by active compliance. The compliance behaviors consist of ***Center Load*** and ***Comply*** composite behaviors, which must comply to any external and internal disturbances caused by the rovers or the uneven terrain.

1) Center load

The ***Center*** *Load* behavior is activated when the force in the gimbal on either of the rovers exceeds a specified threshold. Figure 23 illustrates the sequence of motions that occur to center the load on both rovers and reset the force. In Figure 23 we assume a scenario where the rovers are in column formation in group transport behavior when the center behavior is triggered. The corrective procedure is for each rover to center the load with respect to the center of its gimbal. The misalignment is illustrated by the arrows on Figure 23a. In the corrective procedure, the lead rover performs its correction while the follow rover waits. When the lead rover has completed its correction, the rovers reverse roles and the follow rover performs its correction. The following steps occur in sequence during the center load behavior:

- Step 1: A synchronization occurs between the rovers to indicate triggering of the center load behavior. Both rovers then halt and enter the group center load behavior. (Figure 23a illustrates the rovers in this configuration)
- Step 2: The lead rover turns its wheels to align them with the load (as illustrated on Figure 23b). The distance to drive to correct the misalignment is determined by reading the displacement from the gimbal translate sensor (the sign indicates the direction to drive in).

- Step 3: The lead rover then drives the appropriate distance to correct for the misalignment (as shown on Figure 23c), Upon completion of the correction, the lead rover straightens its wheels (as shown on Figure 23d).
- Step 4: The rovers reverse roles. The follow rover also performs Steps 2 and 3 as shown on Figure 23e, f and g respectively.

2) *Group Formation*

The group formation behavior change the formation of the rovers between any arbitrary start and end formation. Figure 24 illustrates the sequence of motions that occur to change formation. In Figure 24a we assume a scenario where the rovers are in row formation in group transport behavior when a change formation command is received. Each rover has a specific role and their actions occur simultaneously. The role of the lead rover is to drive a pre-determined trajectory along an arc to change the formation. At the same time, the follow rover wheels are continuously aligned with the load and it simultaneously drives forwards or backwards to ensure that the load is centered in its gimbal and load forces are minimized. The following steps occur in sequence to change the formation:

- Step 1: The follow rover aligns its wheels with the load and the lead rover waits (as shown on Figure 24b).
- Step 2: The lead rover turns its wheels to drive along the pre-determined arc trajectory (Figure 24c).
- Step 3: As the lead rover drives along an arc, the follow (pivot) rover continuously aligns its wheels with the load and drives forwards or backwards based on sensory inputs from its gimbal to compensate for the lead rover's deviations from the arc (that inevitably occur due to ground slippage, terrain effects, etc.). (Figure 24d).
- Step 4: When the lead rover has traversed the arc, the lead rover steers its wheels into a turn-in-place (point turn) configuration. At the same time, the follow rover straightens its wheels back to its original wheel configuration. (Figure 24e).
- Step 5: The lead rover turns in place until the load is at the commanded formation angle (Figure 24f).

o

3) *Group Transport*

The group transport behavior coordinates the motion of the two rovers in a desired formation. During a traverse, both rovers continuously modify their heading (i.e. steering trajectories) and velocity trajectory profiles to ensure that the formation is maintained , the load is centered in their gimbals and gimbal forces do not exceed a specified threshold. The following steps occur in sequence during group transport:

- Step 1: The rovers get into the commanded formation using the Group Formation behavior.
- Step 2: The rovers synchronize to initiate driving.
- Step 3: During driving, the state information (force, torque, and translation) from the gimbal on each rover is used to continuously *modify* velocity and heading of the rovers.
- Step 4: During transport, excessive force in the load on either rover may trigger a Center load behavior. The rovers perform the Center load behavior. Upon completion of the Center load behavior, the Group transport behavior resumes (Steps 1, 2 and 3) until the transport distance is completed.

4) *Comply*

At the lowest level, the ***Comply*** behavior performs coordinated turns and straight-line formation motion of the rover pair with minimal explicit communication between the rovers. Utilizing the gimbal sensory information and the known physical constraint between the rovers imposed by the PV tent container, each rover can partially estimate its physical relationship with respect to the other rover. Using this information and knowing its role in achieving the current goal (turn or move in formation in a straight line), each rover can operate independently until the terminal condition indicating goal achievement or an exception condition occurs.

In the coordinated turn of an arbitrary angle one rover acts as the pivot in the turn and the other rover drives in an arc to cover the turn angle. Since the length of the container is known, the arc length and its radius to be traversed are pre-computed before execution of the turn. During the turn, the rover at the pivot turns in place to maintain alignment with the container. As the other rover drives in an arc, the rover at the pivot drives forwards or backwards based on sensory inputs from its gimbal to compensate for the other rover's deviations from the arc (that inevitably occur due to ground slippage, terrain effects, etc.). The terminal condition to end this activity is when the rover on the arc has completed driving its arc length and the rover at the pivot has either turned the appropriate angle in place or the time allotted for the turn expires. The exception condition is when the force in the gimbal exceeds a specified threshold. This usually occurs almost simultaneously on both rovers because of reaction forces on the container. Should an exception occur, the rovers stop the activity, synchronize and re-acquire the target to re-initialize their current locations with respect to the target location. This is also done upon successful completion of the turn because the actual angle turned will differ from the desired. Another turn to correct for the difference will probably be needed.

In coordinated formation driving, the rovers attempt to drive in a straight line. Each rover attempts to maintain its orientation with respect to the container (and *so* its orientation with respect to the other rover) using local sensory data from its gimbal. Depending on the formation (column, row or something in between), each rover uses its speed and its heading to compensate for deviations from the formation and for force build-up (compression or extension) of the container. The terminal condition for this activity is the achievement of the distance traversed as determined from wheel odometry.

Thresholds on force and formation angle error trigger exceptions that abort the activity.

5) *Compliance behaviors*

The compliance behaviors ensure safe handling of the load by the two rovers.

The tightly coupled multi-robot system depicted in figure **4** can be considered **as** a single vehicle system. **This** system is composed of two rovers with independent all-wheel drive and steering. The two rovers are mechanically coupled through a 2.5 meter long hollow beam with a 0.25 meter by 0.25 meter square cross-section (mockup of a PV tent). This payload is held at its ends by the rovers with four degree-of-freedom (d.o.f.) passive compliant gimbals (**see** Figure 2). The gimbals are mounted to the top of the rovers at a point along the centerline of their turn-in-place rotation. This allows each rover to turn in place under ideal conditions without affecting the payload position. The four potentiometer-instted d.o.f. of the gimbals **are** pitch, yaw, roll and sliding along the longitudinal **axis** of the payload. The pitch and yaw d.o.f. have springs that return the gimbals to the vertical positions. The slider allows translation **of the beam** within its grippers **of** plus or **minus** 0.02 meters (see figure 11). Force and torque in **the** gimbals can also be *sensed* using a **6** d.o.f. force/torque sensor **mounted** at the base of the gimbals.

The low-level decentralized comply control behaviors are:

- ***Formation Controller Behavior***
- ***Minimize Forces/Torques on Payload Behavior***
- ***Center Payload in Longitudinal Slider Behavior***

The control inputs for each of these three controllers are (1) rover **speed** and (2) heading control (steering). The ***Formution Controller Behavior*** receives a desired formation angle command from the ***Group Formation Behavior.*** The desired formation angle is mapped into the corresponding gimbal yaw angles on each rover. The ***Formation Controller Behavior*** is then tasked with driving and steering the rover to achieve and maintain the desired gimbal yaw angle on each rover.

The ***Minimize Forces/Torques on Payload Behavior*** is tasked with minimizing the forces on the payload or compliant linkage on each rover. The forces on the payload can be **high** if the relative **speed** between the two rovers is greater than a set threshold. The magnitude of the 3-D force vector along the payload longitudinal axis is the input for **this** behavior. The predominant control output for ***Minimize Forces/Torques on Payload Behavior*** **is** rover speed, supplemented with some steering. The ***Center Payload in Longitudinal Slider Behavior*** is tasked with minimizing deviations of the payload from longitudinal slider center on each rover. A linear potentiometer is used to measure the gimbal slider position. The control output for ***Center Payload in Longitudinal Slider Behavior*** is rover speed and heading (steering) control.

The ***Formation Controller Behavior, Minimize Forces/Torques on Payload Behavior,*** and ***Center Payload in Longitudinal Slider Behavior*** controllers have conflicting goals. In actual operation, one may encounter a situation were ***Center Payload in Longitudinal Slider Behavior*** will request an increase in rover speed and ***Formation Controller Behavior*** would command a reduce **speed.**

In order to solve this problem we developed a priority-based, weighted PD controller scheme for rover **speed** and heading trajectory modifications that satisfies, ***Formation Controller Behavior, Center Payload in Longitudinal Slider Behavior,*** and ***Center Payload in Longitudinal Slider Behavior*** under steady state conditions. For each rover, we compute the formation error ϑ_{Lerror} (gimbal yaw angle error), the translation error T_{error} (deviation from gimbal slider center), and the force error F_{error} (magnitude of gimbal force vector along the payload longitudinal axis) **as** follows:

$$\vartheta_{Lerror} = {}'desired - [illegible] \qquad \textit{Equation 1}$$

where ϑ_{Lerror} is the **gimbal** yaw angle error, $\vartheta_{desired}$ is the desired gimbal yaw angle, and ϑ_{actual} **is** the actual **gimbal** yaw angle;

$$T_{error} = T_{desired} - T_{actual} \qquad \textit{Equation 2}$$

where T_{error} is the gimbal slider translation position error, $T_{desired}$ is the desired gimbai slider translation position, and T_{actual} is the actual gimbal slider translation position;

$$F_{error} = F_{desired} - F_{actual} \qquad \textit{Equation 3}$$

where F_{error} is the force error, $F_{desired}$ is the desired force error, and F_{actual} is the actual force reading.

First we define PD controllers that will maintain formation angle (desired gimbal yaw angle), center payload and minimize payload forces as follows:

$$'output = K_{p\vartheta}\vartheta_{error} + K_{d\vartheta}\frac{d}{dt}(error\ ) \qquad \textit{Equation 4}$$

where ϑ_{output} is the output of the PD gimbal yaw angle controller, $K_{p\vartheta}$ is the proportion gain of the PD gimbal yaw angle controller, and $K_{d\vartheta}$ is the derivative gain of the PD gimbal yaw angle controller;

$$F_{output} = K_{pF} F_{error} + K_{dF} \frac{d}{dt}(F_{error}) \qquad \textbf{Equation 5}$$

where F_{output} is the output of the PD force controller, K_{pF} is the proportional gain of the PD force controller, and K_{dF} is the derivative gain of PD force controller;

$$T_{output} = K_{pT} T_{error} + K_{dT} \frac{d}{dt}(T_{error}) \qquad \textbf{Equation 6}$$

where T_{output} is the output of the PD gimbal slider translation position controller, K_{pT} is the proportional gain of the PD gimbal slider translation position controller, and K_{dT} is the derivative gain of the PD gimbal slider translation position controller.

The PD controllers defined above independently achieve their respective goals but when implemented simultaneously will result in conflicting speed and heading corrections. To resolve these conflicts, we combined the outputs of each of the PD controllers into a single function using a weighting scheme to compute the desired speed and heading corrections for each rover. The weighted functions are defined as follows:

Lead Rover:

$$AVel = -W_{\vartheta} \vartheta_{output} + W_F F_{output} - W_T T_{output} \qquad \textbf{Equation 7}$$

such that

$$W_{\vartheta} + W_F + W_T = 1.0 \qquad \textbf{Equation 8}$$

Follow Rover:

$$AVel = W_{\vartheta} \vartheta_{output} + W_F F_{output} - W_T T_{output} \qquad \textbf{Equation 9}$$

such that

$$W_{\vartheta} + W_F + W_T = 1.0 \qquad \textbf{Equation 10}$$

where $AVel$ is the required **speed** correction factor, W_{ϑ} is the weight assigned to the PD gimbal angle controller output, W_F is the weight assigned to the PD force controller output, and W_T is the weight assigned to the PD gimbal translation position controller;

$$\Delta Heading = W_{H\vartheta} \vartheta_{output} + W_{HF} F_{output} + W_{HT} T_{output}$$

Equation 11

such that

$$W_{H\vartheta} + W_{HF} + W_{HT} = 1.0 \qquad \textbf{Equation 12}$$

where $\Delta Heading$ **is** 'the required rover heading correction factor, $W_{H\vartheta}$ is the weight assigned to the PD gimbal angle controller output, W_{HF} is the weight assigned to the PD force controller output, and W_{HT} is the weight assigned to the PD gimbal translation position controller.

The combined control laws for these controllers are best characterized by considering the follow cases:

Case 1: Rovers in a row transport formation (zero formation angle).

Ideally, in a row formation the gimbal angles of both rovers are zero degrees (i.e. the rovers are aligned and the longitudinal axis of the beam is perpendicular to their heading). During a traverse, both rovers will deviate **from** their path due to differences in velocities, ground slippage and terrain effects, etc. These will result in two undesirable consequences: (1) the payload will not be centered in the gimbals, and (2) the forces on the payload will exceed the desired threshold.

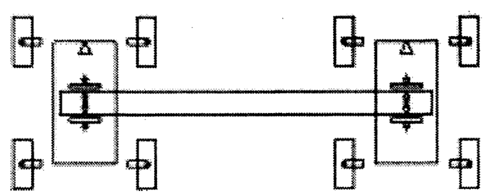


Figure 12

The heading correction equation(11) was very difficult to implement in the row formation due to the slow response **of** the steering actuators. Therefore a force threshold was set, and if the threshold is exceeded **on** either rover, both rovers stopped, synchronized, and **took turns** to center the payload. The speed corrections proved to be very effective. It was used in a traverse of over **30** meters. In the row formation the following weights were used

$$W_{\vartheta} = 0.6, W_T = 0.3, W_F = 0.1$$

$$W_{H\vartheta} = 0.0, W_{HF} = 0.0, W_{HT} = 0.0$$

Case 2: Rovers in a column transport formation , lead-follower scheme (formation angle greater 10°but less or equal to 85°).

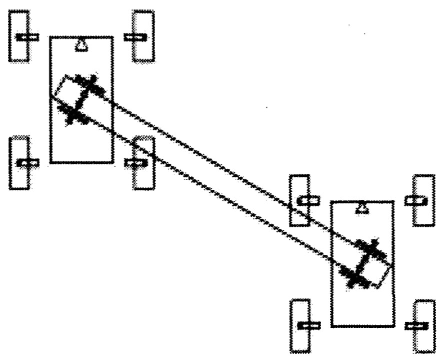


Figure 13

Similar to the row formation during traverse, both rovers will deviate from their path due to difference in speeds, ground slippage and terrain effects, and other disturbances. These will result in the payload not being centered in the gimbals and forces on the payload exceeding the desired threshold. Here also we use the same priority based weighted PD controller scheme for rover speed and heading trajectories modifications that satisfies ***Formation Controller Behavior, Center Payload in Longitudinal Slider Behavior,*** **and** ***Center Payload in Longitudinal Slider Behavior*** **forces** under **steady state** conditions. However the weights are different **from** the **row** formation scheme.
In the column formation the following weights were used:

$$W_{\theta} = 0.05, W_T = 0.90, W_F = 0.05$$

$$W_{H\theta} = 0.85, W_{HF} = 0.1, W_{HT} = 0.05$$

For the column formation case, the heading correction **is** implemented. Most of the traverses reported in the experimental studies used the column formation.

C. Experimental results

Our experimental **setup** is shown in Figure **4** where two of our SRRs (sample Return Rover) have been retrofitted with gimbals to carry a load. For our experimental study, we demonstrated under closed loop control the second **step** of the **4** step deployment scenario shown in Figure **1** in the Arroyo Seco at **JPL** in September of 2000. **A** number **of** coordinated motion behaviors are required for transport of an extended container using the sequence illustrated in Figure **10.** The **four** phases are: **(1)** clear the **CSU** in preparation for a turn, **(2)** traverse to a staging area, **(3)** survey the deployment area for a clear site, **(4)** traverse **to** docking site. The two main behaviors required for these four phases are ***Assume Transport Formution,*** a group behavior that autonomously guides the two rovers into a specific formation such as row (side-by-side) or column (leader-follower) as **shown** in Figure **4,** and ***Coordinated Transport,*** a group behavior that autonomously controls the system during any traversal.

Both of these group behaviors rely on compliant control of the extended container ***so*** as not to drop it during the movement. Minimizing the communication between rovers is also a goal, since there are power and bandwidth restrictions for systems deployed on planetary surfaces such **as** Mars. The gimbal shown Figure **2** has ***6*** DOF force/torque sensors and **pots.** Monitoring of the pot settings is used for control of the angular offset between the rovers, and monitoring of the force sensors is used for side-slip and terrain offsets between the rovers. ***Comply*** is a group behavior that uses communication between the rovers only at the beginning of a sequence and when there is a need to stop in order to re-center the load.

Our experimental runs in the Arroyo Seco at JPL were used to determine the number of actuators that will be needed for robust control of the transport sequence. Since the gimbals were not actuated, all repositioning of the container had to be done using the robots. This is potentially a problem on slopes of more than **10°,** where the load will tend to **shift** backwards onto the follower robot. A row formation can be used to partially offset this shift, but obstacle detection **in this** formation is more complicated. Actuation of the gripper mechanism is the best compromise, in that it allows a grip/re-grip process to be used to keep the load balanced.

Our experimental **studies** demonstrated:

- **40-to-50** meter autonornus traverses **of outdoor** irregular **terrain** (maximal **slope** of **9")** by two rovers (SRR/SRR2k) in the tightly coupled transport of an extended container,
- autonomous change of formation by two rovers **carrying** an extended container under compliant control, and
- continuous visual guidance to a designated deployment site from 50 m, with a heading error **< 1";** and a distance error < ***5%*** at 8m by use of a visual template.

Section VII Conclusions

We have presented a control architecture called CAMPOUT for the system level coordination of multiple mobile robots. The design is three-layer, with a behavior-based middle layer. The behavior hierarchy is built on earlier work by Pirjanian. **It** uses a multiple objective decision making paradigm in order to select actions that are satisficing in the Pareto **optimal** sense. The lowest level in **CAMPOUT** is built using legacy device drivers from previous rover tasks such as **SRR** and FIDO. During the next fiscal year we will concentrate on the development of the grasp and manipulate behaviors that are necessary for the first, third and fourth steps in the PV tent deployment scenario.

REFERENCES

[1] Robot Colonies, Special Issue of Autonomous Robots (Eds. R. C. Arkin and *G.* A. Bekey), Vol. 4, No. 5,1997.

[2] P. S. Schenker, E. T. Baumgartner, S. Lee, H. Aghazarian, M. S. Garrett, R. A. Lindemann, D. K. Brown, Y. Bar-Cohen, S. S. Lih, B. Joffe, and S. S. Kim, B. H. Hoffman, T. L. Huntsberger, "Dexterous robotic sampling for Mars in-situ science," in Proc. SPIE Conf. Intelligent Robotics and Computer Vision XVI, vol. 3208, Pittsburgh, PA, Oct. 14-17,1997.

[3] P. S. Schenker, E. T. Baumgartner, R. A. Lindemann, H. Aghazarian, D. Q. Zhu, A. J. Ganino, L. F. Sword, M. S. Garrett, B. A. Kennedy, G. S. Hickey, A. S. Lai, and L. H. Matthies, B. D. Hoffman, T. L. Huntsberger, "New planetary rovers for long range Mars science and sample return," in Proc. SPIE Conf. Intelligent Robotics and Computer Vision XVII, Boston, MA, vol. 3522, Nov. 1-5, 1998.

[4] T. L. Huntsberger, E. T. Baumgartner, H. Aghazarian, Y. Cheng, P. S. Schenker, P. C. Leger, K. D. Iagnemma, and S. Dubowsky, "Sensor-fused autonomous guidance of a mobile robot and applications to Mars sample return operations," in Proc. SPIE Conf. Sensor Fusion and Decentralized Control in Robotic Systems II, Boston, MA, vol. 3839, pp. 2-8, Sep. 19-22, 1999.

[5] C. R. Weisbin, G. Rodriguez, P. S. Schenker, E. Baumgartner, R. Volpe, S. Hayati, and H. Das, "Autonomous rover technology for Mars sample return," in Proc. (5th Inteanational Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '99), 1-3 June 1999, Noordwijk, The Netherlands.

[6] R. Volpe, E. Baumgartner, P. Schenker, and S. Hayati, 'Technology development and testing for enhanced Mars rover sample return operations," in Proc. IEEE Aerospace 2000, Big Sky, MT, March 2000.

[7] A. Colozza, "Design and optimization of a self-deploying PV tent array," NASA CR187119, Lewis Research Center Group, Brook Park, OH, 1991.

[8] S. I. Roumeliotis and G. A. Bekey, "Collective localization: A distributed Kalman filter approach to localization of groups of mobile robots," in 'Proc. 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, April 22-28, pp. 2958-2965.

[9] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. "A probabilistic approach to collaborative multi-robot localization," Auonomous Robots, vol. 8, no. 3, 2000.

[10] B. B. Werger and M. J Mataric, "Robotic "food" chains: Externalization of state and program for minimal-agent foraging," in proceedings, From Animals to Animats 4, Fourth International Conference on Simulation of Adaptive Behavior (SAB-96), Pattie Maes, Maja Mataric. Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson, eds, MIT Press/Bradford Books, pp. 625-634, 1996.

[11] R. C. Arkin, "Cooperation without communication: Multi-agent schema based robot navigation," Journal of Robotic Systems, vol. 9, no. 3, April 1992, pp. 351-364.

[12] D. Goldberg and M. J Mataric, "Coordinating mobile robot group behavior using a model of interaction dynamics," in Proc. Autonomous Agents '99, Seattle, WA, May 1-3, 1999, 100-107.

[13] M. J. Mataric, "Behavior-based control: Examples from navigation, learning, and group behavior," Journal of Experimental and Theoretical Artificial Intelligence, Special Issue on Software Architectures for Physical Agents, vol. 9, no. 2-3, H. Hexmoor, I. Horswill, and D. Kortenkamp, eds., pp. 323-336, 1997.

[14] D. Rus, B. Donald, and J. Jennings. "Moving furniture with teams of autonomoujs robots," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 235-2442, 1995.

[15] B. Gerkey and M. J. Mataric, "Principled communication for dynamic multi-robot task allocation," in Proc. of the International Symposium on Experimental Robotics 2000, Waikiki, Hawaii, December 10-13, 2000.

[16] L. E. Parker, Heterogeneous Multi-Robot Cooperation, Massachusetts Institute of Technology Ph.D. Dissertation, January 1994. Available as MIT Artificial Intelligence Laboratory Technical Report 1465, February 1994.

[17] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holberg, and A. Casal, "Decentralized cooperation between multiple manipulators," 5th IEEE Int. Workshop on Robot and Human Communication, 1996.

[18] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, "Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation," IROS'96.

[19] A. Agahand G. A. Bekey, "In a team of 'robots the loudest is not necessary the best," Intelligent Systems for the 21st Century., IEEE Int. Conf. on Systems, Man and Cybernetics, 1995.

[20] J. Borenstein, "Control and kinematic design of multi-degree-of-freedom mobile robots with compliant linkage," IEEE Trans. on robotics and Automation, vol. 11, no. 1, pp. 21-35, 1995.

[21] M. Hara, M. Fukuda, H. Nishibashi, Y. Aiyama, Y., J. Ota, and T. Arai, "Motion control of cooperative transportation system by quadruped robots based on vibration model walking," in Proc. IROS'99.

[22] Hisashi Osumi, M. et al , "Cooperative control of multiple mobile manipulators on uneven ground," IEEE Int. Conf. on Robotics and Automation (ICRA'98), Leuven, Belgium, 1998.

[23] H. Hoshino, E. Muro, T. Fukuda, and Y. Adachi, "Cooperative motion control of six-legged walking robot equipped with a manipulator," IEEE Intl. Conf. on Robotics and Automation (ICRA'98), 1998.

[24] H. Kimuraand H. Katano, "Vision based motion recognition of the hexapod for autonomous assistance," IROS'98, 1998.

[25] H. Kimura G. Kajiura, "Motion recognition based cooperation between human operating robot and autonomous assistant robot," in Proc. ICRA'97, 1997.

[26] D. Kurabayashi, J. Sasaki, and Y. Aiyama, "Cooperative transport with regrasping of torque-limites mobile robots," in Proc. IROS'96, 1996.

[27] N. Miyata, J. Ota, Y. Aiyama, J. Sasaki, J. and T. Arai, "Cooperative transportation system with regrasping car-like mobile robots," in Proc. IROS'97, 1997.

[28] T. Sugarand V. Kumar, "Multiple cooperating mobile manipulators," IEEE Int. Conf. on Robotics and Automation, 1999.

[29] T. Viney et al., "Dynamic modeling and feedback control of a mobile robot pair cooperatively transporting a long object," Third Biennial Engineering Mathematics and Applications Conference, The University of Adelaide, South Australia, 1998.

[30] P. Pirjanian, T. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, H. Das, S. S. Joshi, and P. S. Schenker, "CAMPOUT: A control architecture for multirobot planetary outposts," in Proc. SPIE Conf. Sensor Fusion and Decentralized Control in Robotic Systems III, Boston, MA, Nov 2000.

[31] P. Pirjanian, "Behavior Coordination Mechanisms - State-of-the-art," Tech-report IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, October, 1999.

[32] R. A. Brooks, "A robust layered control system for a mobile robot," IEEE Journal of Robotics and Automation}, 2(1):14--23, March 1986.

[33] J. Kosecka and R. Bajcsy, "Discrete Event Systems for autonomous mobile agents," in Proc. Intelligent Robotic Systems '93 Zakopane, pages 21--31, July 1993.

[34] R. C. Arkin and D. MacKenzie, 'Temporal coordination of perceptual algorithms for mobile robot navigation," IEEE Transactions on Robotics and Automation, vol. 10, no. 3, pp. 276-286, June 1994.

[35] Sensor Planning with Bayesian Decision Theory In Proceedings of Third International Symposium on Intelligent Robotic Systems '95, Pisa, Italy, July 1995.

[36] P. Maes, "How to do the right thing," Connection Science Journal, vol. 1, no. 3, pp. 291-323, 1989.

[37] J. Rosenblatt, 'The Distributed Architecture for Mobile Navigation," Journal of Experimental and Theoretical Artificial Intelligence, vol. 9, no. 2/3, pp.339-360, April-September, 1997.

[38] J. Riekki and J. Roning, "Reactive task execution by combining action maps," International Conference on Integrated Robots and Systems (IROS'97), Grenoble, France, pp. 224-230, September 1997.

[39] J. Hoff and G. Bekey, "An architecture for behavior coordination leaming," in IEEE International Conference on Neural Networks, Perth, Australia, vol. 5, pages 2375--2380, November 1995.

[40] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots,"The International Journal of Robotics Research, vol. 5, no. 1, pp. 90 -- 98, 1986.

[41] R. C. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior," in IEEE International Conference on Robotics and Automation (ICRA'87), pp. 264--271, 1987.

[42] E. W. Large, H. I. Christensen, and R. Bajcsy. "Scaling the dynamic approach to path planning and control: Competition among behavioral constraints," International Journal of Robotics Research, vol. 18, pp. 37--58, January 1997.

[43] A. Saffiotti, K. Konolige, and E.-H. Ruspini, "A multivalued logic approach to integrating planning and control," Artificial Intelligence, vol. 76, pp. 481--526, March 1995.

[44] J. Yen and N. Pfluger, "A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation,"

IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, no. 6, pp. 971 -- 978, June 1995.

[45] P. Pijanian, "Multiple objective behavior-based control," Journal of Robotics and Autonomous Systems, vol. 31, no. 1-2, pp. 53-60, Apr 2000.

[46] P. Pirjanian and M. Mataric, "Multiple Objective vs. Fuzzy Behavior Coordination", Lecture Notes in Computer Science on Fuzzy Logic Techniques for Autonomous Vehicle Navigation, eds. D..Drainkov and A. Saffiotti. 2001.

[47] B. G. Drake, "Reference Mission Version 3.0, Addendum to the Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team," SP-6107-ADD, NASA Lyndon B. Johnson Space Center, Houston, TX, 1998.

[48] E. Gat, "Three-layer Architectures," in Artificial Intelligence and Mobile Robots, eds. D. Kortenkamp, R. P. Bonasso, and R. Murphy, MIT Press, Cambridge, MA, 1998.

[49] S. J. Hoffman and D. L. Kaplan, "Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team," SP-6107, NASA Lyndon B. Johnson Space Center, Houston, TX, 1997.

[50] T. Huntsberger, M. Mataric, P. and Pirjanian, "Action selection within the context of a robotic colony," in Proc. SPIE Sensor Fusion and Decentralized Control in Robotic Systems II, vol . 3839, Boston, MA, 1999.

[51] T. L Huntsberger, G. Rodriguez, and P. S. Schenker, "Robotics challenges for robotic and human Mars exploration," in Proc. ROBOTICS 2000, Albuquerque, NM, March, 2000.

[52] T. L. Huntsberger, H. Aghazarian, E. Baumgartner, and P. S. Schenker, "Behavior-based control systems for planetary autonomous robot outposts," in Proc. IEEE Aerospace 2000, Big Sky, MT, March, 2000.

[53] L. E. Parker, "Current State of the Art in Distributed Robot System," Distributed Autonomous Robotic System 4, Lynne E. Parker, George Bekey, and Jacob Barhen (eds.), Springer, pp. 3-12, 2000.

[54] Y. Cao, A. S. Fukunaga and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," Autonomous Robots, vol. 4, no. 1, pp. 7-27, 1997.

[55] T. L. Huntsberger and J. Rose, "BISMARC," Neural Networks, vol. 11, no. 7/8, pp. 1497-1510, 1998.

[56] P. Caloud, W. Choi, J.-C. Latombe, Le C. Pape, and M. Yin, "Indoor automation with many mobile robots," in Proc. IEEE/RSJ IROS'90, 1990, pp. 67-72.

[57] H. Asama, A. Matsumoto, and Y. Ishida, "Design of an autonornus and distributed robot system ACTRESS," in Proc. IEEE/RSJ IROS'89, 1989, pp. 283-290.

[58] K. Jin, P. Liang, and G. beni, "Stability of synchronized control of discrete swarm structures," in Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA'94), 1994, pp. 1033-1038.

[59] R. Simmons, S. Singh, D. Hershberger, J. Ramos, and M.F. Smith, III, "First results in the coordination of heterogeneous robots for large-scale assembly," in Proc. of the ISER '00 Seventh International Symposium on Experimental Robotics, December, 2000.

[60] P. Pirjanian and M. Mataric, "Multi-robot target acquisition using multiple objective behavior coordination," in Proc. IEEE Intl Conf. on Robotics and Automation (ICRA2000), San Francisco, April 2000, pp 101-106.

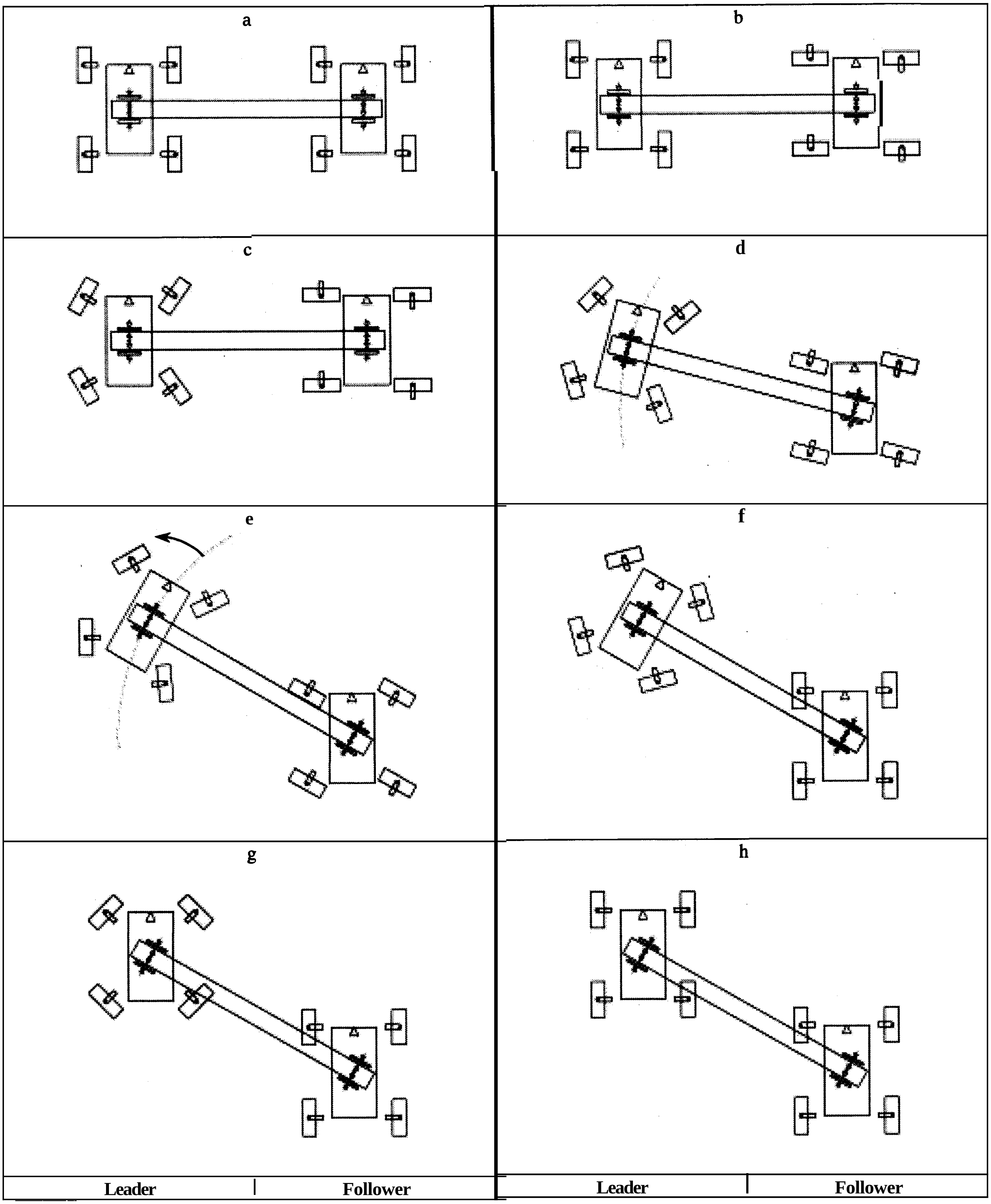


Figure 9 Distinct phases of the Center Load group behavior.

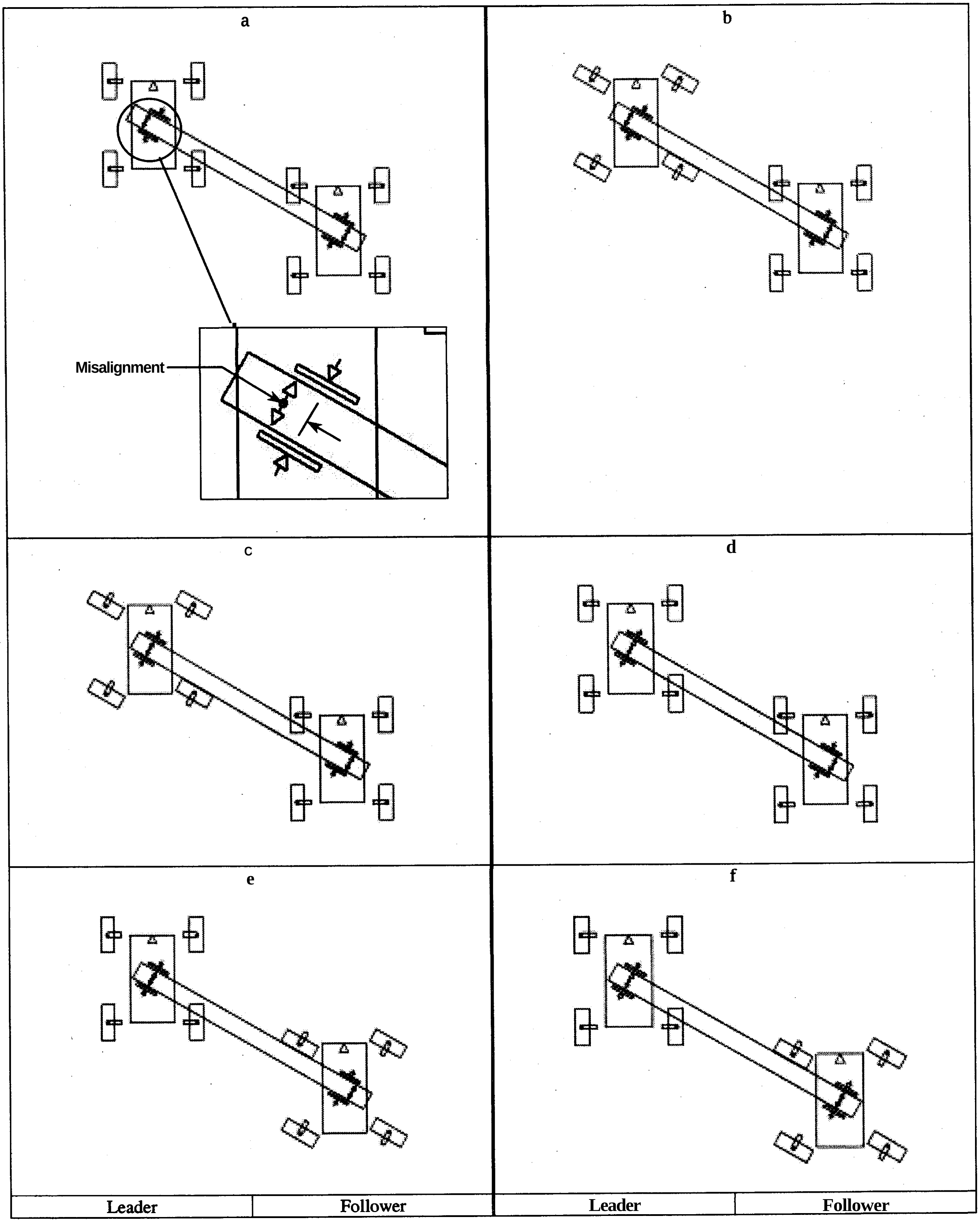


Figure 10 Distinct phases of the Load Centering behavior.