North Dakota State University--Fargo

From the SelectedWorks of Jeremy Straub

March, 2015

Swarm Intelligence, a Blackboard Architecture and Local Decision Making for Spacecraft Command

Jeremy Straub

Available at: https://works.bepress.com/jeremy_straub/250/
Swarm Intelligence, a Blackboard Architecture and Local Decision Making for Spacecraft Command

Jeremy Straub
Department of Computer Science
University of North Dakota
3950 Campus Road, Stop 9015
Grand Forks, ND 58202-9015
701-777-4107
jeremy.straub@my.und.edu

Abstract—Control of a multi-spacecraft constellation is a topic of significant inquiry, at present. This paper presents and evaluates a command architecture for a multi-spacecraft mission. It combines swarm techniques with a decentralized / local decision making architecture (which uses a set of shared blackboards for coordination) and demonstrates the efficacy of this approach. Under this approach, the Blackboard software architecture is used to facilitate data sharing between craft as part of a resilient hierarchy and the swarm techniques are used to coordinate activity. The paper begins with an overview of prior work on the precursor command technologies and then presents five command architectures for comparison purposes. Then, it presents a qualitative analysis of these techniques, followed by a quantitative analysis which characterizes the constellation’s performance across a variety of prospective scenarios including normal operations, several mission scenarios which limit communications, operations in an intentionally communications-denied environment and operations across a variety of craft failure scenarios. From performance analysis, the utility of the techniques is analyzed.

TABLE OF CONTENTS
1. INTRODUCTION ..............................................1
2. BACKGROUND .............................................2
3. COMMAND ARCHITECTURES .........................2
4. QUALITATIVE ANALYSIS OF TECHNIQUES ......4
5. QUANTITATIVE ANALYSIS OF TECHNIQUES ....5
7. CONCLUSIONS AND FUTURE WORK ..........6
REFERENCES ...............................................6
BIOGRAPHY ..................................................7

1. INTRODUCTION

Control of a multi-spacecraft constellation is a topic of significant inquiry, at present. Prior work has suggested that spacecraft can be combined to create synergistic outcomes. This is particularly true in the case of small spacecraft which can provide increased temporal coverage, at the expense of individual craft capabilities.

A wide variety of command and control approaches for multiple types of spacecraft constellations have been proposed. These range from human-in-loop systems which require significant bandwidth for control purposes, to systems that rely on humans for command and not control to systems with a high degree of autonomy.

Systems with high levels of autonomy take several forms. These include top-down control, limited control delegation and centralized approaches. However, even a decentralized approach is still highly reliant on command data transmissions to facilitate coordination with peer craft. Providing resiliency across spacecraft loss and temporary or longer-term craft or communications failures is highly desirable for a multitude of applications.

This paper presents and evaluates a command architecture for a multi-spacecraft mission. It combines swarm techniques with a decentralized / local decision making architecture (which uses a set of shared blackboards for coordination) and demonstrates, via comparison to centralized and non-swarm approaches, the efficacy of this approach. Under this approach, the Blackboard software architecture is used to facilitate data sharing between craft as part of a resilient hierarchy (which reforms if a leader’s heartbeat signal is not detected for a period of time) and the swarm techniques are used to coordinate activity.

This approach has two key benefits. First, the use of swarm techniques (specifically derivatives of the Intelligent Water Drops – IWD – approach), reduces communications requirements for control. Second, this approach allows a constellation to continue operating for an extended period of time in a communications denied (intentionally or inadvertently) environment.

The paper begins with an overview of prior work on the precursor command technologies. Then, it presents five command architectures for comparison purposes. The first is a typical top-down command approach where decisions are made centrally. The second is a highly distributed command approach, where delegation decisions are made through a hierarchy and task-performance decisions are made onboard the task-performing craft. The third approach augments the second with a Blackboard architecture. The forth approach considered augments the distributed (second) control approach with swarm (IWD) techniques and the fifth does the same to the Blackboard distributed (third) control approach.

Next, the paper presents a qualitative analysis of these techniques: it discusses the particular strengths and weaknesses of each technique. Then, a quantitative analysis is performed which characterizes the constellation’s performance across a variety of prospective scenarios including normal operations, several mission scenarios.
which limit communications, operations in an intentionally communications-denied environment and operations across a variety of craft failure scenarios. From performance analysis, the utility of the various techniques is analyzed.

2. BACKGROUND

The work presented herein draws from significant prior work in the control of collections of heterogeneous robotic craft. First, an overview of work is presented. Then, a review of prior work in this particular line of research is provided.

Multi-Craft Control

A variety of techniques for the control of multiple craft (and multi-craft combined with fixed sensors) have been proposed. Fink, et al. [1-3] have proposed a “tier-scalable” approach as well as a decision making process [4] for prioritizing targets for it. This approach utilized a top-down command approach, centralizing most decision making in a key orbital spacecraft. Others [5, 6] have proposed similar concepts.

Communications are critical to the operation of multi-craft control and, potentially, a key bottleneck for operations. Mathews, et al. [7] demonstrated the ability to use only slow-speed links for coordination. Rauch, et al. [8] attempted to emulate animals’ decision making and Marsh, et al. [9] used the model of an “environmental nervous system”.

Work has also been performed regarding controlling particular craft types and their interaction. Parker and Agogino [10] demonstrated a technique for allocating tasks between multiple spacecraft, while Pinciroli [11] performed work on controlling a collection of small spacecraft. Vladimirova [12] created an adaptive response, also for spacecraft control. In UAVs, work has included command with only local communications [13], coordination [14] and planning [15]. Ground vehicles have, in addition to basic command, control and communications considerations, also been utilized to demonstrate the efficacy of swarm movement control [16] and used the ant colony optimization technique to drive exploratory efforts [17].

Prior Work

The proposed work is a combination of several previous lines of prior work. It draws from an initial focus of trying to reduce spacecraft communications requirements [18, 19] which demonstrated the utility of having pre-shared data. It also directly incorporates work on the development of a Blackboard architecture-based command and control framework [19-22]. Adding to this area of work is the introduction of the intelligent water drop algorithm. This algorithm, originally created by Shah-Hosseini [23], has been refined through prior work to create a simplified version that is better-suited to robotic control [24].

3. COMMAND ARCHITECTURES

As part of the discussion of the proposed system, which combines the Blackboard Architecture approach with the Simplified Intelligent Water Drop (SIWD) decision making technique (adapted from [24]), multiple control architectures are compared. The following subsections present a description of the five control architectures under consideration: top-down, highly distributed, highly-distributed with Blackboard, highly distributed with Intelligent Water Drops and highly distributed with Blackboard and Intelligent Water Drops.

Top-Down

The top-down control technique centralizes decision making at a single node (in the context of a heterogeneous craft mission, typically the most capable orbital craft; however, a fixed ground control node could also be utilized). Commands can either be relayed through intermediate tiers of craft (as shown in Figure 1) or conveyed directly from the orbital craft to all commanded craft (as shown in Figure 2).

![Figure 1. Top-down Control Structure with Relayed Commands.](image1)

![Figure 2. Top-down Control Structure with Direct Command.](image2)

Data that is collected by the subordinate craft is transmitted back to the core node for processing and decision making. Craft have limited onboard autonomy which is used to respond to communications failure and for auctioning assigned tasks; however, most computing power is
centralized in the core node. Resiliency is created through the robustness and resiliency of the core node.

**Highly Distributed**

The highly distributed control approach is characterized by craft having multiple ad-hoc relationships with other craft. These relationships are established on an as-required basis to complete various assigned tasks. With the highly distributed approach, goals are provided by ground controllers and tasking is determined collaboratively, using a bid system where the group coalesces on a task-performance approach based on optimizing performance metrics. This approach is highly resilient; however, it may have single points of failure from key components (such as long range communications capabilities) being housed on only a single or small subset of craft. Aside from these special capabilities, the cluster can survive craft loss down to a single remaining craft (with deterioration proportionate to loss). This approach requires significant onboard computational capabilities on each node. A diagram of the highly-distributed approach (showing a snapshot of ad hoc connections) is presented in Figure 3.

![Figure 3. Highly Distributed Control Structure Showing a Snapshot of Ad hoc Communications.](image)

**Highly Distributed with Blackboard**

One problem with the highly distributed approach presented in the previous section is a fragmentation of information. Because communications are ad hoc in support of arranging collaboration, information that may be of use to multiple craft (e.g., conditions in a particular area or status information) may end up ‘trapped’ on a single craft. The addition of a central blackboard, a concept from Hayes-Roth’s [25] Blackboard Architecture, is added as a solution to this problem. Under this approach, a central blackboard stores all information about the environment, current goals and pathways to arrive at those goals. A solver [26] is used to identify and choose between pathways to goal completion and the bidding process (described in the previous subsection) is used to determine how to best allocate the available tasks that need to be performed to individual cluster members. This approach is presented in Figure 4.

![Figure 4. Highly Distributed Control Approach with Blackboard Architecture.](image)

**Highly Distributed with Intelligent Water Drops**

Where the incorporation of the Blackboard architecture solved the data fragmentation problem present in the highly distributed control approach, the use of IWD presents a prospective solution to the data exchange requirements of ad hoc collaboration. The IWD algorithm and SIWD algorithm (presented in [24]) allow simulated drops to be probabilistically distributed through prospective solution pathways and for those pathways to be reshaped by the movement of the drops. This approach can be utilized to distribute work between multiple collaborating craft, with pathway modifications serving to represent changing local and regional conditions. Under this approach, a shared network is distributed by the lead node (the orbital craft is presumed) and local calculations are undertaken by each subordinate node to determine what tasks to undertake.

![Figure 5. Highly Distributed Control Structure with IWD.](image)
Because of the need to run IWD simulations, significant computational resources are required on all craft. The exact level of resources depends on the network size and is thus application dependent; however, in all cases, inter-craft communication needs are reduced somewhat.

**Highly Distributed with Blackboard and IWD**

Finally, an approach combining the solutions presented by Blackboard and IWD incorporation is presented. Like with the solutions presented in the two previous subsections (which presented just IWD and just Blackboard incorporation), the use of the Blackboard architecture solves the data fragmentation problem, while the IWD reduces communications through localized decision making. The combination, however, is synergistic. The previous subsection described how IWD’s simulated drops were run through solution pathways to determine what action to take and probabilistically distribute workload across multiple craft. However, these pathways were limited by the local data store. With the blackboard data available, the solution space to search (and through which to flow simulated drops) is significantly expanded. Effectively, the simulated drops can be run flown through the entire blackboard network, examining and prospectively modifying the blackboard-derived solution pathways in light of local conditions and other factors. This approach is depicted in Figure 6.

4. **QUALITATIVE ANALYSIS OF TECHNIQUES**

Each of the discussed techniques has particular benefits and limitations. This section examines the tradeoffs made by technique selection.

The top-down approach benefits from its conceptual simplicity, ease of understanding and troubleshooting and the ability to use more limited computational hardware in the lower level craft (as they are, effectively, just following the script created by the lead craft). The approach is very resilient to the failure of any node, other than the lead node. However, it has virtually no resiliency to failure of this lead node. Notably, this may not differ much from other approaches in scenarios where the lead node is required for long distance transmission and reception (e.g., to and from Earth in the context of a planetary mission). The approach is also not well suited to applications which place the subordinate craft out of contact with the lead node for extended periods of time (or require other independent operation of the subordinate nodes).

The highly-distributed approach solves the problem of the reliance on a single node (though, perhaps not for communications, as enabling all of the craft for long distance to/from Earth communication is likely cost, mass and volume prohibitive). This approach introduces several problems, however. First, it is very difficult to troubleshoot as, to understand the bidding-based decisions that have been
made, you need access to state data from multiple craft at the time the bidding was undertaken. Second, the approach may result in sub-optimal resource allocation via a bidding process in which bids are based on inaccurate data (particularly in the case where another robot has more accurate information) or where bidding results in a craft ending up in a sub-optimal position relative to future needs. Third, the approach requires significant (albeit local) communication to facilitate the bidding process and numerous ad-hoc node-to-node collaborations required for task performance. Benefits, in addition to the resiliency of the approach, include its suitability for missions which will spread the craft out over a wide area (where they would be outside of the range of a central controller, but still need to communicate to collaborate locally) and the ability to consider local condition knowledge in planning, via the bidding process.

The approaches that add the use of the Blackboard Architecture and IWD to the highly distributed approach each mitigate part of the problems that have been identified therein. The Blackboard Architecture addition creates a consistent central knowledge store that can be used to inform the bidding process to prevent (or raise the cost of) bids that place the craft in a non-ideal position for future work. It also provides shared information about conditions throughout the known operating area which can reduce the potential of bids being artificially low or high due to invalid assumptions about local conditions (when another craft in the collection already has experience with the particular area). Problematically, the addition of the Blackboard architecture does add additional communications overhead from having to move Blackboard data between craft.

The approach that adds IWD focuses on the communications problem as well as the problem with the bidding process optimizing for a single task and creating longer-term problems. The IWD approach utilizes the IWD flow process to create optimized assignments for each craft that fulfill the requirements dictated by mission objectives. Because communication is limited to the transmission of network updates (and some limited status and initialization data), the communications needs of the mission are reduced. IWD use also facilitates control decision making by craft which are spread out beyond their communications capabilities (albeit with growing potential for optimality problems as the decision making networks diverge overtime between non-communicating groups). The inclusion of IWD, problematically, increases computational requirements by requiring local resources to flow the simulated drops for decision making purposes, which is more computationally intensive than the limited calculations required for the bidding process.

The approach combining the addition of IWD and the Blackboard Architectural elements to the highly distributed approach is not a panacea. While resolving limitations in both of the single-addition approaches, it introduces other considerations. For example, the combined Blackboard-IWD approach requires more communications than the IWD-only-added approach (though less than the addition of Blackboard by itself). The combined approach also creates greater computational requirements than the Blackboard-only-added approach via the need for additional resources for flowing the simulated drops for decision making.

### Table 1. Impact of Control Paradigm on Operations.

<table>
<thead>
<tr>
<th></th>
<th>Top-Down</th>
<th>Highly Distributed</th>
<th>Distributed + BB IWD</th>
<th>Distributed + BB + IWD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tasks</td>
<td>%</td>
<td>Tasks</td>
<td>%</td>
</tr>
<tr>
<td>Normal Operations</td>
<td>200</td>
<td>16</td>
<td>200</td>
<td>16</td>
</tr>
<tr>
<td>Limited Communications</td>
<td>141</td>
<td>22</td>
<td>141</td>
<td>22</td>
</tr>
<tr>
<td>Communications Denied</td>
<td>73</td>
<td>11</td>
<td>73</td>
<td>11</td>
</tr>
<tr>
<td>Hardware Failure</td>
<td>Res. Unl. Core</td>
<td>Resilient</td>
<td>Resilient</td>
<td>Resilient</td>
</tr>
</tbody>
</table>

5. **Quantitative Analysis of Techniques**

This section provides a quantitative comparison of the four techniques across four different scenarios (normal operations, limited communications, a communications denied environment and a hardware failure). Each is now discussed. The results are summarized in Table 1.

The data presented is based on a scenario with the following characteristics. There are 200 total tasks that are scheduled to be run. Of these, 20% require long-range communications or pre-planning/re-planning capabilities, 50% require local communications or pre-planning/ re-planning capabilities, 10% require instantaneous communications and 30% benefit from shared knowledge.

**Normal Operations**

In the normal operations scenario, it is presumed that all communications links are working. Random link failure is excluded from all scenarios to prevent the introduction of arbitrary random noise to the data presented.

**Limited Communications**

The limited communications scenario is taken as interrupting communications 50% of the time. Given the
potential for retrying failed communications, this limitation is taken as, effectively, causing tasks that suffer repeated impairments (which also equates to 50%) to fail. Others are impaired by the failure to transmit the requisite local data, but this doesn’t cause outright failure.

Communications Denied

The communications denied scenario adds persistent blocking of all long-range communications to the intermittent blocking of the local communications in the limited communications scenario.

Hardware Failure

The resiliency to hardware failure is characterized in this final scenario. All approaches are fairly resilient to failure (as four of the five use the resilient highly distributed approach, in three cases with improving additions). The top-down approach is also resilient, in most cases; however, it is completely non-resilient in the case of loss of the head craft.

Analysis of Results

The data (presented in Table 1) demonstrates the different impact of the various scenarios on the performance of the different control approaches. All five complete all tasks in normal operations, while three (top-down, highly distributed and the distributed with IWD) suffer limited impairment under this scenario. In the limited communications and communications denied scenarios, the versions including the IWD capabilities perform the best, completing 163 tasks. The addition of the blackboard capabilities reduces the percent impaired from 16% to 6%. Other approaches perform much less satisfactorily (ranging from worst performance of 73 tasks completed and 11% impairment, for communication denial and 136 tasks completed and 10% impairment, for communications limitation). Thus, while performance is similar for unimpaired situations, the incorporation of either the Blackboard Architecture or IWD (and ideally both) significantly improves performance under less-than-ideal conditions, increasing mission resilience. Using the Blackboard Architecture reduces the percent impairment under all scenarios, including normal operations.

7. CONCLUSIONS AND FUTURE WORK

This paper has presented a concept for the combination of the Blackboard architecture and IWD to enhance the highly distributed control approach. It has, through a limited computational simulation of multiple prospective scenarios, characterized areas where the two augmentation techniques are able to add value to the base distributed control algorithm and compared the four distributed control approaches to the top-down control approach.

This initial work suggests that the combination of these two techniques may hold significant promise for improving the control of multiple heterogeneous craft. Future work will involve a more elaborate simulation and, prospectively, a real-world trial to validate and refine the assumptions used relative to the various actual conditions.

REFERENCES


**Biography**

Jeremy Straub conducts research into the autonomous control of robots for air and space applications at the University of North Dakota, where he is a PhD candidate in the Department of Computer Science. His work spans the gauntlet between technical development and answering policy questions of technology development and use. Jeremy has published over twenty journal articles and over 75 full conference papers. He has also authored more than 55 national or international conference presentations and more than 80 at local or regional ones. Jeremy was a founding member of the AIAA Small Satellite Technical Committee and serves as the chair of its conferences sub-committee. He’s also a full member of Sigma Xi and several national honors societies.