Intelligent Water Drops Algorithm for Coordinating Between Cluster Spacecraft in a Communications-Denied Environment

Jeremy Straub
Intelligent Water Drops Algorithm for Coordinating Activities Between Cluster Spacecraft in a Communications-Denied Environment

Jeremy Straub
Department of Computer Science, University of North Dakota, Grand Forks, ND, 58202, USA

This paper presents a modification of Shah-Hosseini’s Intelligent Water Drops (IWD) technique that can be utilized for collaborative control of multiple spacecraft in environments where communications are limited, intermittent or denied. It presents Shah-Hosseini’s base IWD algorithm as well as refinements thereof, which simplify it, making it more suitable for more computationally constrained environments (such as small spacecraft and UAVs). A framework for testing the proposed approach as well as several implementation impediments are discussed.

I. Introduction

This paper presents a modification of Shah-Hosseini’s Intelligent Water Drops (IWD) technique [1, 2] that can be utilized for the collaborative control of multiple spacecraft in an environment where communications are intermittent or intentional regular or irregular denial of node-to-node communications is predicted.

The IWD algorithm is a swarm-style control approach, which is based (loosely) on the movement of water through a network of streams. During this process, the water drops pick up and deposit sediment, based on factors such as the amount of sediment present and the speed of the stream. The amount of sediment that a drop is carrying changes its speed (thus, impacting its ability to pick up or deposit sediment from successor locations). Thus, the network affects the drop and the drop affects the network. The approach utilized also incorporates a probabilistic decision making mechanism for deciding how to move through the stream network. Through the use of this probabilistic mechanism, a shared network and a random (or pseudo-random) number generator, collaborative interaction between the different cluster spacecraft can be attained with limited communication. This communication consists primarily of network updates and status updates about the transmitting spacecraft.

The paper starts with a presentation of Shah-Hosseini’s base algorithm and the presents several prior modifications that have been made to it. The impact of these modifications is summarized. One of these modifications, a simplified version (Simplified Intelligent Water Drop, SIWD) that has significantly (order-of-magnitude) lower computational requirements is then discussed in greater detail, as it serves as the basis for the rest of the work.

With the SIWD algorithm presented, several communications impaired / denied scenarios are presented and discussed. A simulation that has been designed to model each, utilizing a typical command approach, is discussed.

The paper continues by discussing several prospective impediments to implementation. The most pronounced is the lack of traceability when utilizing a swarm-style control approach to command expensive assets. Because of the variation introduced by the real world, it is not necessarily possible to gain a post-occurrence or post-problem understanding of the state of the decision making process prior to the occurrence. This can be problematic when trying to derive best practices and ‘lessons learned’ from failure. The lack of certifiable and verifiable performance is also problematic as it makes the performance of the system unpredictable. As one prospective approach to satisfying these concerns, the use of the IWD approach as a backup system to be used only in the event of protracted communications outages is proposed. The benefits, drawbacks and technical feasibility of this approach are evaluated.

The paper concludes with a discussion of key challenges in aerospace software, in particular, the field of autonomous control. It considers how solutions like IWD can be embraced in a risk averse industry, while still ensuring the safe performance of critical software systems.
II. Intelligent Water Drops

The IWD algorithm is based on an approximation (with some modifications) of water movement in nature. With IWD, water drops flow through a network created to represent a problem or decision making framework. What makes IWD of particular interest is that the drops are constantly modifying the network concurrent with their movement. This algorithm was initially developed by Shah-Hosseini [1-6] and has been refined by numerous other researchers. In [1], its utility is considered for the common problem of solving for the traveling salesman’s ideal routing. To do this, it uses a process of iterative refinement. The basic IWD algorithm is depicted in Figure 1.

Figure 1. Basic IWD Algorithm [7].
Other problems have also been solved using the IWD (or an IWD-based) approach. IWD was applied to a mutation algorithm in [6]. In this work, the IWD water drop path network is altered by the random removal and/or replacement of edges, changing the connectivity between various nodes. In [4], IWD is utilized (with the incorporation of a thresholding mechanism), to enhance the quality of various images. The improvement generated by the IWD algorithm is assessed using subjective assessment as well as a signal to noise ratio. The IWD algorithm’s usefulness in deriving a solution for the n-queen and the multidimensional knapsack problems is discussed in [3]. While demonstrating the versatility of the algorithm, these papers – problematically – fail to compare IWD’s performance to other approaches that could (and/or have been) used for similar challenges. Aljila, et al. [8], however, begin to fill this assessment gap through benchmarking the IWD approach.

Since its initial introduction by Shah-Hosseini, use of the IWD algorithm by others has been significant. It has been used for network security [9, 10], irrigation system development [11], supply chain management [12], and trajectory design for unmanned aerial combat vehicles (UACVs) [13].

III. Simplified IWD Algorithms

The IWD algorithm (that was initially proposed by Shah-Hosseini [1-6] and which was presented in Section II) was used as a template to create a simplified version. This simplified IWD (SIWD) algorithm is shown in Figure 2. To simplify the algorithm, a modified version (which was introduced, and the similarity of solution produced and network coverage provided was assessed, in [7]) was created. This version effectively removes one loop, resulting in a significant reduction in the complexity of the algorithm and, thus, lower computational requirements.

![Figure 2. Simplified IWD Algorithm](image-url)
Both approaches (IWD and SIWD) commence with network initialization. Once network initialization has been completed, a specific water drop is arbitrarily selected. For the selected drop, its path for its entire movement for a single run is determined. In the SIWD algorithm, all movement for a particular drop for the a given run is performed in a single loop iteration. Note that this differs from the approach used in the basic IWD algorithm.

The speed and sediment levels (the average of the specific values) across the path that was selected (for the selected drop and run) are determined and stored (for use in sediment level modification). The final location of the drop is recorded and the sediment level changes are calculated (based on the average speed and sediment levels) and applied to the nodes in the traversed path. A change value, based on the average value (determined from the average speed and sediment levels), is applied to each node. This change value is calculated using the proportion of a node’s local sediment level to the average level. The algorithm then selects another drop to process (or if no more exist, returns its result).

The use of the IWD/SIWD algorithms for routing purposes was discussed in [14]. To plan air and ground routes, the applicable decision making process was developed to be a network of IWD paths which could represent actual physical routes, a decision making process or a combination of the two. An application-specific route evaluation metric was also required. With the networks created, an arbitrary number of simulation runs could be performed and a ‘best performing’ approach identified for use (or a few identified for human review and final selection).

IV. Cluster Control

In [15], the utility of the IWD (and its derivative algorithms) for aerospace applications was considered. Therein, it was suggested that the IWD algorithm (and its derivatives not be used) for problems for which a deterministic solution is possible. Instead, it is well suited problems without a direct solution. Therein it was used for routing and, while it is not a general-purpose path-finding algorithm, the potential benefits offered were discussed. These included the ability to use network-incorporated heuristics to short cut the search process and the ability to change the network to allow new decisions to be affected by previous ones.

The cluster control application has many similarities to the routing application (in that both are controlling movement). For spacecraft, a given level of (orbital) movement is always present; thus, the algorithm primarily controls changes to the orbital parameters (deviation-correction is performed by a lower-level control system). In addition to controlling movement, the SIWD algorithm is also utilized to coordinate activity. Basically, each spacecraft is assigned to be a water drop on an a priori shared network. All spacecraft know what the other ones are likely to be doing (based on the network and probability functions and coefficients assigned) and they, thus, base their own decision making on that. This has an emergent effect of a well-coordinated constellation of craft being produced by numerous craft making their own probabilistic decisions using a shared network.

V. Testing Framework

Work is ongoing to test this approach. Several simulations have been designed that will be used to characterize IWD-commanded craft performance. In each simulation, nodes transmit change that they have made to the network (i.e., sediment added or removed) as well as state information (location deviation, etc.). Simulation topics include communications intentionally denied scenarios, intermittent communications scenarios (with different levels of network availability) and regular, but limited, communications scenarios (as discussed, in the context of an orbital service model, in [16]). Each is now, briefly, discussed.

The communications intentionally denied scenario begins with a fully connected network with all nodes having shared and up-to-date IWD network and state information. Then, at an arbitrary point in the simulation, the communications capabilities between nodes are blocked. The performance of the constellation is assessed at regular intervals, over time.

In the intermittent communications scenarios, a shared IWD network is available to all nodes; however, no initial shared state exists. The nodes communicate with each other to exchange state and network change data over time, while the network continues to evolve (from changes made by movement on various nodes). As with the previous scenario, the performance of the constellation is assessed at regular intervals, over time.

Finally, in the regular, but limited, communications scenarios, the constellation (similarly) starts with a shared IWD network but no initial shared state data. This, again, is changed over time. Again, the performance of the constellation is assessed at regular intervals, over time.
VI. Implementation Impediments

Several prospective impediments to implementation are now considered. These include the lack of traceability and the lack of certifiable / verifiable performance.

The most pronounced problem with the proposed approach (and many similar approaches) is the lack of traceability. This is, of course, a byproduct of using a swarm-style control that is dependent on an emergent behavior; however, it is unlikely to be immediately embraced as an approach suitable for commanding expensive assets. Over time operating, demonstration of the approach’s continued satisfactory performance may overcome this initial concern; however, this creates a ‘chicken or the egg’ style problem, whereby adoption is required to create the confidence required for adoption.

The fear of relying on emergent behaviors is not unjustified. A post mortem assessment of this type of system is also problematic (making it difficult to explain any failure) as the variation introduced by the real world makes it problematic to gain a post-occurrence / post-problem understanding of the state of the decision making process prior to the occurrence. This can be problematic when trying to derive best practices and ‘lessons learned’ from failure.

The lack of certifiable and verifiable performance is also problematic as it makes the performance of the system unpredictable. While the performance of the IWD network (and network-of-networks) can be accurately predicted, certain conditions can result in long-running processes (making the system unable to produce a completed solution by the time-of-need and forcing reliance (in these limited conditions) on a fallback mechanism. Again, this does not necessarily inspire confidence in prospective system users.

One prospective approach to satisfying these concerns is to install the IWD-based control software, but configure it for use as a backup system which is engaged only in the event of protracted communications outage. This may be a more palatable solution; however, it does little to advance the acceptance of IWD or swarm-style control approaches. Alternatively, the IWD-based system could be utilized as a primary (but initially supervised) control system, with other software receiving and approving the commands issued by the IWD-based system. This could drive confidence in the IWD-based system (and/or identify corrections needed in the approach), while still ensuring suitable command performance of the expensive hardware.

VII. Key Challenges in Aerospace Software & Autonomous Control

The aerospace industry is confronted with several, nearly diametrically opposed challenges. On one hand, when controlling expensive assets, it is undesirable to leave control decisions to software when human control is possible (and uncomfortable, even when it is not). The recent ‘seven minutes of terror’ during Mars Science Laboratory’s entry, descent and landing [17, 18] was, in fact, attributable to having to rely on fully autonomous control during these seven minutes. On the other hand, there are many cases where autonomy is desirable (and a subset where it is required). Autonomous response may, for example, be required to achieve a requisite response level to an enemy’s (cyber or physical) attack to ensure a mutual assured destruction-style deterrence. It has also been shown to increase performance of exploration (as compared to human control) [19]. Other work has demonstrated the limited acceptance of autonomous control in the planetary sciences [20].

Creating stakeholder understanding and obtaining acceptance of the capabilities (and risks) of fully autonomous systems is, of course, an ongoing challenge. The growing acceptance of the previously regarded as nearly discredited faster, better, cheaper [21] approach suggests that this may simply be a matter of time. The use of autonomy in systems (such as those proposed in [22, 23]) with resilience from redundancy may also aid in gaining this acceptance.

VIII. Conclusions & Future Work

This paper has provided an overview of the IWD algorithm and a derived SIWD algorithm which increases the performance of the IWD algorithm and, thus, its suitability for control applications. It also eliminates some complexity of the algorithm. The use of this SIWD approach for autonomous spacecraft cluster control has been discussed. Finally, the acceptance (and barriers thereto) has been considered and a discussion of the key challenges to acceptance of these type of control approaches has been discussed.

Acknowledgments

Some facilities and equipment for the work on Intelligent Water Drops have been provided by the University of North Dakota Department of Computer Science. Additional hardware has been supported by North Dakota EPSCoR (NSF Grant # EPS-814442) and the John D. Odegard School of Aerospace Sciences.
Small satellite development work at the University of North Dakota is or has been supported by the North Dakota Space Grant Consortium, North Dakota NASA EPSCoR, the University of North Dakota Faculty Research Seed Money Committee, North Dakota EPSCoR (NSF Grant # EPS-814442), the Department of Computer Science, the John D. Odegard School of Aerospace Sciences and the National Aeronautics and Space Administration.

Presentation of this paper at the AIAA SciTech 2015 conference has been partially supported by the University of North Dakota School of Graduate Studies’ Doctoral Inventive Program Fund.

References


