Distributed multi-robot sensing and tracking:  
A behavior-based approach

Lynne E. Parker
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory, P. O. Box 2008
Oak Ridge, Tennessee 37831-6364
Email: ParkerLE@ornl.gov, Telephone: (615) 241-4959, Fax: (615) 574-7860

ABSTRACT

An important issue that arises in the automation of many large-scale surveillance and reconnaissance tasks is that of tracking the movements of (or maintaining passive contact with) objects navigating in a bounded area of interest. Oftentimes in these problems, the area to be monitored will move over time or will not permit fixed sensors, thus requiring a team of mobile sensors — or robots — to monitor the area collectively. In these situations, the robots must not only have mechanisms for determining how to track objects and how to fuse information from neighboring robots, but they must also have distributed control strategies for ensuring that the entire area of interest is continually covered to the greatest extent possible. This paper focuses on the distributed control issue by describing a proposed decentralized control mechanism that allows a team of robots to collectively track and monitor objects in an uncluttered area of interest. The approach is based upon an extension to the ALLIANCE behavior-based architecture that generalizes from the domain of loosely-coupled, independent applications to the domain of strongly cooperative applications, in which the action selection of a robot is dependent upon the actions selected by its teammates. We conclude the paper by describing our ongoing implementation of the proposed approach on a team of four mobile robots.

Keywords: multiple object tracking, mobile sensors, distributed control, sensor coverage, multi-robot systems, behavior-based systems

1 INTRODUCTION

An important issue that arises in the automation of many large-scale surveillance and reconnaissance tasks is that of tracking the movements of (or maintaining passive contact with) objects navigating in a bounded area of interest. One approach to this problem is to fix a group of sensors at pre-specified locations, and fuse the information obtained from each of the sensors into a coherent whole. However, oftentimes the area to be monitored will move over time or will not permit fixed sensors, thus requiring a team of mobile sensors — or robots — to monitor the area collectively. In these situations, the robots must not only have mechanisms for determining how to track objects and how to fuse information from neighboring robots, but they must also have distributed control strategies for ensuring that the entire area of interest is continually covered to the greatest extent possible. Of course, many variations of this dynamic, distributed sensory coverage problem are possible. For example, the relative numbers and speeds of the robots and the objects to be tracked can vary, the availability of communication can vary (such as limited-distance, point-to-point, broadcast, or not available), the robots can...
differ in their sensing and movement capabilities, and so forth.

Much of the previous work in distributed sensory coverage problems falls into the category of the art gallery theorems, which typically describe centralized approaches to the placement of sensors and make a number of simplifying assumptions, such as the use of fixed sensory locations or sensory patrol paths, the use of ideal (noise-free and infinite range) sensors, and the availability of enough sensors to cover the entire area of interest. However, when an object begins to go out of range of a robot, the problem we are addressing here requires the robots to dynamically shift their locations so that as many objects as possible remain under surveillance. This can be quite a complex control problem.

Clearly, this research topic incorporates many subproblems, including the physical tracking of objects (e.g. using vision, sonar, IR, or laser range), prediction of object movements, multi-sensor fusion, distributed navigation, multi-robot communication, selection of target to track, achieving adequate collective terrain coverage, and so forth. Thus, to focus initial efforts in this problem, we concentrate first on the aspects of distributed control, developing distributed mechanisms that allow teams of robots to dynamically shift their monitoring locations so that the group as a whole maintains sensor contact with the objects within an uncluttered, bounded area.

We also emphasize that although the cooperative sensing and tracking application is interesting in its own right, this application domain was selected for the additional purpose of serving as a testbed for developing generalized approaches for the control of cooperative teams. The cooperative tracking problem is attractive for this purpose for a number of reasons. First, it requires a strongly cooperative solution to achieve the goal, meaning intuitively that the robots must act in concert to achieve the goal, and that the task is not trivially serializable. This makes the cooperative control problem much more challenging than a weakly cooperative approach. Additionally, it offers an excellent domain for the comparison of different control strategies, such as centralized versus distributed control approaches. Finally, it allows us to explore the extension of the ALLIANCE cooperative control architecture, which we previously developed for the domain of loosely-coupled, independent tasks, to the domain of strongly cooperative applications.

In this paper, we describe a mechanism for achieving distributed cooperative control in the tracking application domain. Section 2 defines the tracking problem of interest in this paper, and is followed by a brief discussion of related work in this area. We then describe our overall approach in section 4, beginning with a review of the ALLIANCE architecture, followed by a discussion of the subcomponents of our behavior-based approach to cooperative tracking. Section 5 describes the ongoing implementation of our proposed approach on both a simulated and a physical robot team of four Nomad robots. Finally, we offer concluding remarks in section 6, as well as directions of future research.

2 PROBLEM DESCRIPTION

The cooperative tracking problem is defined as follows. Given a two-dimensional, convex polyhedral spatial region $S$, a team of robots $\mathcal{R}$, and a set of objects $O$ whose membership is not known in advance by the robot team $\mathcal{R}$, the goal of the robots is to collectively track the paths of objects in $O$ as they travel through $S$ to the fullest extent possible. A secondary goal is to collectively maximize sensory coverage over as large an area in $S$ as possible and to minimize redundant tracking. To reduce the complexity of the problem initially, we assume that no obstacles are in $S$ except for members of $\mathcal{R}$ and $O$.

Further, let $\text{sensor\_coverage}(r_i)$ return the area visible to robot $r_i$'s tracking sensors, for $r_i \in \mathcal{R}$. Then we assume that, in general, $\bigcup_{r_i \in \mathcal{R}} \text{sensor\_coverage}(r_i) \ll S$. That is, the maximum area covered by the tracking sensors of the robot team can be much less than the total area to be monitored. This implies that fixed robot sensing locations or sensing paths will not be adequate in general, and that, instead, the robots must move dynamically as objects appear in order to maintain tracking contact with them and to maximize the coverage of
We further assume the following:

- The robots have a broadcast communication mechanism that allows them to talk with each other within a given communication range.
- The tracking sensor on each robot is directional, and can only be used to track one object at a time. Further, the directional movement of the tracking sensor can be controlled independently from the direction of movement of the robot.
- For all \( r_i \in \mathcal{R} \) and for all \( o_j \in \mathcal{O} \), \( \max_v(r_i) > \max_v(o_j) \), where \( \max_v(a) \) returns the maximum velocity of entity \( a \).
- The robot team members are able to derive or directly access position information on relative locations of nearby objects and robots.

3 RELATED WORK

Although this author is unaware of previous research directly addressing the autonomous, decentralized, dynamic motion coordination of multiple mobile robots operating in an area that is large relative to the sensory coverage of the robots, much research related to the topic of distributed tracking has been developed. A full review of this work, however, is beyond the scope of this paper; instead, we will mention a few articles that are most closely related to the topic of this paper.

Several authors have looked at the static placement of sensors for object tracking in known polygonal environments. Many of these approaches use variations on the art gallery theorems, addressing many problems related to polygon visibility. The general art gallery problem is to determine the minimum number of guards required to ensure the visibility of an interior polygonal area. Variations on the problem include fixed point guards or mobile guards, which can patrol a line segment within the polygon. For example, Briggs uses art gallery theorems in the development of algorithms for planning the set of placements from which a sensor can monitor a region within a task environment. Her approach uses weak visibility as a model for detectability, in which all points in the area to be monitored are visible from at least one point in the sensor placement region. Her work does not address the dynamic cooperative object tracking addressed in this paper.

Everett et al. have developed a coordinated multiple security robot control system for warehouse surveillance and inventory assessment. The system includes is semi-autonomous, utilizing autonomous navigation with human supervisory control when needed. They propose a hybrid navigational scheme which encourages the use of known “virtual paths” when possible. Wesson et al. describe a distributed artificial intelligence approach to situation assessment in an automated distributed sensor network, focusing on the issues of knowledge fusion. Durfee et al. describe a distributed sensor approach to object tracking using fixed sensory locations.

4 APPROACH

As noted earlier, there are many challenging research issues that must be addressed to fully solve the cooperative tracking application domain. To focus our initial efforts in this area, therefore, we are first concentrating on the issues of distributed control. However, we do advocate the development of complete systems whose subsystems are integrated with each other to the fullest extent possible from the beginning of development. Thus, we
are making temporary approximate solutions to the remaining areas of the cooperative tracking problem (such as the physical sensing and tracking of objects) and incorporating them into a complete system fully implemented on a mobile robot team. This allows us to study our cooperative control algorithms on the physical robot testbed early in the system development, to reduce the risk of mismatching interface assumptions later in our research.

An early decision in developing solutions to this type of problem involves consideration of either a centralized or a distributed, decentralized approach. For the application domain under consideration here, it seems unrealistic to expect that complete global information can be made available in real time for a centralized decision maker to control the actions of all robot team members, especially as the task scales to larger surveillance areas, more robots, and more objects to track. Thus, we opted to investigate a distributed, decentralized mechanism for cooperative tracking rather than a centralized approach\(^1\). Our general approach to distributed cooperative control in the tracking problem is to extend the ALLIANCE architecture we have previously developed\(^9,10\) to this additional application domain. We now provide a brief overview of the ALLIANCE architecture and explain why this technique needs to be extended before it can address the cooperative tracking domain. We then describe the overall design of our behavior-based approach to this problem, followed by a discussion of each subsystem in the design.

### 4.1 Overview of ALLIANCE

The ALLIANCE software architecture\(^9,10\) is a behavior-based, fully distributed architecture that utilizes adaptive action selection to achieve fault tolerant cooperative control in robot missions involving loosely coupled, largely independent tasks. Robots under this architecture possess a variety of high-level functions (modeled as behavior sets) that they can perform during a mission, and must at all times select an appropriate action based on the requirements of the mission, the activities of other robots, the current environmental conditions, and their own internal states. Since cooperative robotic teams often work in dynamic and unpredictable environments, this software architecture allows the team members to respond robustly and reliably to unexpected environmental changes and modifications in the robot team that may occur due to mechanical failure, the learning of new skills, or the addition or removal of robots from the team by human intervention. This is achieved through the interaction of mathematically modeled motivations of behavior, such as impatience and acquiescence, within each individual robot. These motivations allow robots to take over tasks from other team members if those team members do not demonstrate their ability — through their effect on the world — to accomplish those tasks. Similarly, it allows a robot to give up its own current task if its sensory feedback indicates that adequate progress is not being made to accomplish that task. The primary mechanism for achieving adaptive action selection in this architecture is the motivational behavior. The output of a motivational behavior is the activation level of its corresponding behavior set, represented as a non-negative number. When this activation level exceeds a given threshold, its corresponding behavior set becomes active.

To enhance the robots' perceptual abilities, ALLIANCE utilizes a simple form of broadcast communication that allows robots to inform other team members of their current activities. Thus, at some pre-specified rate, each robot broadcasts a statement of its current action. This one-way broadcast communication could be replaced by the use of passive action recognition, if such a capability were provided to the team members. No two-way conversations are employed in this architecture.

Previous limitations of ALLIANCE, however, made it difficult to apply to application domains such as cooperative tracking, due to the earlier assumption in ALLIANCE that missions are composed of loosely-coupled, largely independent subtasks. This assumption does not hold for the cooperative tracking task, since a robot's selection of object to track is dependent upon the object tracking selection made by the remaining robot team members. Since we want the robot team to collectively track as many objects that come through the monitoring area as possible and to maintain sensory coverage over as wide an area as possible, an individual robot cannot

\(^1\)In the future, however, we plan to compare our distributed approach to a centralized technique to determine which approach provides the best results for this application domain.
select actions independently of the actions selected by its teammates. Thus, we describe in this paper an extension to ALLIANCE that allows the action selection decision to be made as a function of the action selections of a robot’s teammates.

4.2 Behavior-based control for cooperative tracking

Figure 1 shows the ALLIANCE organization of behaviors for an individual robot team member that is used in our cooperative tracking approach. For each object $o_j$ that is within tracking range of $r_i$, a motivational behavior is spawned within $r_i$ to compute the motivation of robot $r_i$ to track object $o_j$. These behaviors compete with each other as described in section 4.4 to result in the distributed selection of an object to track. The identity of the object selected to be tracked, given by its $(x, y, velocity)$ triple, is passed to the track_object behavior set, which uses a sensor-based algorithm to track the selected object. This behavior set outputs a control command to the independent turret that controls the directional tracking sensor, as well as velocity and steering commands to the locomotion effectors (assumed here to be wheels). The commands to the wheels from the track_object behavior set are combined (as described in subsection 4.7) with the output of the terrain_coverage behavior (which is responsible for ensuring the coverage of as much area in $S$ as possible, as a function of the locations of neighboring robots and objects — see subsection 4.6) to result in a commanded movement vector for the robot. This commanded movement vector is sent to the wheels unless it is temporarily suppressed by the avoid_obstacles behavior, which activates when the robot is dangerously close to some obstacle. The following subsections describe the subcomponents of this approach.
4.3 Inter-robot communication

In previous work,\textsuperscript{1,6,11} it has been shown that communication and awareness of robot team member actions can significantly improve the quality of a distributed solution for certain task domains. Thus, in the approach described in this paper, we supplement a robot's knowledge of object movements gained from direct sensing with position and derived velocity information on object sightings that is communicated by robot team members within a given communication range.

To clarify this idea, figure 2 depicts three ranges that are defined with respect to each robot $r_i$. The innermost range is the \textit{sensing range} of $r_i$, within which the robot can use a sensor-based tracking algorithm to maintain passive contact with an object. The middle range is the \textit{predictive tracking range} of the robot $r_i$, which defines the range in which objects localized by other robots $r_k \neq r_i$ can affect $r_i$'s selection of object to track. The outermost range is the \textit{communication range} of the robot, which defines the extent of the robot's communicated messages. For simplicity, we assume that the sensing, predictive tracking, and communications ranges are the same, respectively, for each robot team member.

When a robot receives a communicated message regarding the location and velocity of a sighted object that is within its predictive tracking range, it begins a predictive tracking of that object's location. (At present, the prediction assumes the object will continue linearly from its current state.) If the communicated information indicates that an object is within robot $r_i$'s predictive tracking range, that information is combined (as described in section 4.4) with information on all other objects $r_i$ knows about (i.e. within the predictive tracking range) to effect its selection of the object to physically track.

Note that, of course, the accuracy of the tracking of an object $o_j$'s movements is much higher when a robot is physically tracking $o_j$ than when a robot is predictively tracking $o_j$. Thus it is important that each robot also communicate to its teammates the location and velocity of the object it is currently physically tracking. To refer to this communicated tracking message in the ALLIANCE formalism, we define the function $\text{comm\_received}$ as follows:

\[
\text{comm\_received}(i, k, j, t_1, t_2) = \begin{cases} 
1 & \text{if robot } r_i \text{ has received a message from robot } r_k \text{ in the time span } (t_1, t_2), \text{ where } t_1 < t_2, \text{ which indicates that } r_k \text{ is tracking object } o_j \\
0 & \text{otherwise} 
\end{cases}
\]

and the related function $\text{comm\_received\_all}$ is defined as:

\[
\text{comm\_received\_all}(i, R, j, t_1, t_2) = \begin{cases} 
1 & \text{if } \exists k ((k \neq i) \text{ and } (\text{comm\_received}(i, k, j, t_1, t_2) = 1)) \\
0 & \text{otherwise} 
\end{cases}
\]

4.4 Selection of object to track

Throughout the cooperative tracking mission, each robot team member must decide which object to track and where to move to best track that object. Since we are dealing with a potentially large space relative to the sensing range of the robots, it is unrealistic to expect that all robots will have complete global information on the location and movement of all objects present in $S$. Thus, the robots make their decisions based upon local information within their predictive tracking range.

As a robot discovers (either through its own sensors or through a communicated message) an object $o_j$
within its tracking range, an ALLIANCE motivational behavior is invoked to compute that robot's motivation to physically track \textit{oj}. A robot \( r_i \)'s motivation to track object \textit{oj} at time \( t \) is dependent upon three factors:

- \( r_i \)'s position and velocity at time \( t \)
- which object (if any) \( r_i \) is tracking at time \( t \)
- whether any other robot \( r_k \neq r_i \) is tracking \textit{oj} at time \( t \)

These factors are combined into the ALLIANCE \textit{impatience} motivation for \( r_i \) as follows:

Let the function \( \text{tracking}(r_i, t) \) return the object \textit{oj} that \( r_i \) is physically tracking at time \( t \). We define \( \tau \) as a fault-tolerant feature of ALLIANCE, which gives the period of time robot \( r_i \) allows to pass without receiving a communication message from a specific teammate before deciding that that teammate has ceased to function. (This allows robot \( r_i \) to assume a teammate \( r_j \) that \( r_i \) has not heard from lately has experienced some failure, and is probably no longer tracking the object indicated in its last message.)

Then we define the \textit{impatience} motivation as:

\[
\text{impatience}_i(o_j, t) = f(o_j, t, \text{comm\_received\_all}(i, R, j, t - \tau, t)) \\
= \frac{1}{\text{dist}(r_i, o_j, t)} \times w_i(o_j, t)
\]

where \( \text{dist}(r_i, o_j, t) \) is the distance between \( r_i \) and \textit{oj} at time step \( t \), and:

\[
w_i(o_j, t) = \begin{cases} 
  w_1 & \text{if } (\text{tracking}(r_i, t - 1) = o_j) \\
  w_2 & \text{if } (\text{comm\_received\_all}(i, R, j, t - \tau, t) = 0) \\
  w_3 & \text{if } (\text{comm\_received\_all}(i, R, j, t - \tau, t) = 1)
\end{cases}
\]
Thus, a robot $r_i$'s impatience motivation to track a given object is proportional to its distance from that object, weighted by one of three weights, depending upon whether $r_i$ has been tracking $o_j$, and, if not, whether any robot is tracking $o_j$. The relative values of the weights $w_1$, $w_2$, and $w_3$ will impact the global characteristics of the cooperative team, from placing priority on nearby objects to placing priority on the object a robot is already physically tracking. We expect that hysteresis is preferred in the system, so that $w_1$ will be weighted more heavily than the other weights.

In the original ALLIANCE architecture, the motivation level to select an incomplete task grows over time until one of the motivations crosses a threshold, at which time that motivation becomes active and suppresses the remaining motivational behaviors corresponding to other tasks. Additionally, a number of factors besides impatience fed into the calculation of the motivation to perform a given task. In the tracking application domain discussed in the present paper, however, the motivation calculation is much simpler, and is in fact the same as the impatience level described above. We make one other modification for the current application domain to eliminate the risk of idle time between object tracking selections, by using a max operation instead of a thresholding operation to select the motivational behavior which gains control. Thus, the impatience levels of robot $r_i$ to track each object $o_j$ are computed independently, and the object with the highest corresponding impatience level is the object selected to be tracked. When this selection is made, that object's $(x, y, velocity)$ values are passed as input to the track_object behavior set.

### 4.5 Tracking the selected object

In the initial phase of research in this problem, which concentrates on the cooperative control issues of distributed tracking, we reduce the scope of the problem by using an indoor global positioning system as a substitute for vision- or range-sensor-based tracking. Under this approach, each object to be tracked is equipped with a global position sensor, and broadcasts its current $x, y$ position via radio to the robots within communication range. Each robot team member is also equipped with a positioning sensor, and can use the objects' broadcast information to determine the relative location of nearby objects.

The track_object behavior set controls the movement of the directional sensor (mounted on an independent turret) to maintain passive contact with the object being tracked. It also computes a desired speed and steering command to maintain a desired distance from the object being tracked. The speed and steering command is combined with the output of the terrain_coverage behavior as described below to derive the resulting commands that are output to the wheels.

In future work, we plan to develop and/or implement vision- or range-sensor-based tracking algorithms to reduce or eliminate our dependence upon the global positioning system.

### 4.6 Terrain coverage / dispersion

The secondary goal of the robots in this application is to maintain coverage of as much of the area $S$ as possible. Of course, if all of the objects are concentrated in a small area of $S$, then the robots should place a higher priority on tracking the objects than on maintaining terrain coverage. The terrain_coverage behavior thus performs a combination of dispersion from neighboring robots with attraction toward objects in the area. This combination should result in robots concentrating more heavily in areas with more objects, and less in areas with fewer objects or with adequate existing robot coverage of the area.

Using the notation of Mataric, we define the dispersion of robot $r_i$ due to neighboring robots as:
where $\delta_{\text{disperse}}$ equals the desired minimum distance between neighboring robots, $p_i$ is the position of robot $r_i$, and $C(r_i, \delta_{\text{disperse}})$ is the centroid operator that, given a robot $r_i$ and a distance threshold $\delta_{\text{disperse}}$, returns the local centroid of robots in that area.

Similarly, we define the attraction towards nearby objects as (again, using the notation of Mataric):

$$
\frac{-v_i}{\|C(r_i, \delta_{\text{disperse}}) - p_i\|} (C(r_i, \delta_{\text{disperse}}) - p_i)
$$

where $\delta_{\text{attract}}$ equals the desired maximum distance to nearby objects.

These values are combined as a weighted sum into a vector which gives the desired direction of movement of $r_i$ from the perspective of the terrain.coverage behavior set.

### 4.7 Combination of behavior output

The vector output from the track.object behavior set is combined as a weighted sum with the output from the terrain.coverage behavior set to result in the motion command to the robots wheels. The relative weights between the object tracking behavior and the terrain coverage behavior provide the opportunity to vary the characteristic performance of the collective robot team from concentrating primarily on object tracking to concentrating largely on terrain coverage.

### 5 EXPERIMENTAL TESTBED

The proposed approach to cooperative distributed tracking described in this paper is currently being implemented on the team of four Nomadic Technologies robots shown in figure 3. These robots (named Ada, Grace, Alexandra, and Edith, whose namesakes were female pioneers in computer science) are wheeled vehicles with tactile, infrared, ultrasonic, compass, 2D laser sensing, and indoor global positioning systems. In addition, the robots are equipped with a voice synthesizer and radio ethernet for inter-robot communication and communication with a development workstation. The robots will soon be upgraded to include a simple manipulator which will allow us to perform experiments involving a higher degree of robot interaction with the environment. The Nomad robots use a shared memory multiprocessor system, with a PC 486 master processor and two 6811 controller processors to handle the motor and sensor drivers. The robots are programmed using C.

A multi-robot simulator provided as part of the development environment from Nomadic allows us to test and debug our algorithms prior to executing them on the actual robots. The code generated during the simulation is then ported directly to the robots for experimentation in the "real world". Figure 4 shows this simulator.

At the time of this writing, we are completing the implementation of the supporting software for inter-robot communication, the global positioning system, and the routines for dispersion and attraction. Implementation and experimentation are ongoing to test and refine our proposed approach to cooperative distributed tracking.
CONCLUSIONS AND FUTURE WORK

We have proposed a behavior-based approach for cooperative tracking by a team of mobile robots. This approach is based upon extensions to the ALLIANCE architecture to allow for execution of tasks in strongly cooperative application domains. Empirical investigations of our proposed cooperative control approach are ongoing on both the simulated robots and the physical robot team. As we fine-tune the distributed control approach, we will be addressing additional issues of the cooperative tracking problem, such as the development and/or implementation of a physical object tracking algorithm used by individual robots and multi-sensor fusion, seeking to reduce or eliminate the dependence of our support routines on a global positioning sensor.

As stated earlier, a primary goal in our cooperative robotics research is to develop methodologies that are generally applicable to given application domains, rather than addressing cooperative robot missions on a case-by-case basis. Thus, we plan to use our distributed tracking results to attempt to extend our methodologies to the broader class of generalized cooperative coverage problems, where generalized cooperative coverage is defined as having some spatial region(s) of interest addressed by the appropriate robots on the team. Examples of generalized coverage problems include planetary exploration, search and rescue missions, some types of cooperative manipulation, and, of course, distributed tracking and surveillance. We expect that our approach will readily extend to this wider application domain, and intend to study these areas in the future.

ACKNOWLEDGEMENTS

I extend my thanks to Vic Baker for installing the global positioning system on the robots and for developing the inter-robot communication support modules, to Cindy Maher for implementing the multi-robot dispersion and attraction routines, and to Shashank Shekar for early discussions on the cooperative tracking problem. This research was funded by the Office of Engineering Research, Basic Energy Sciences (directed by Dr. Oscar Manley), of the U.S. Department of Energy, under contract No. DE-AC05-84OR21400 with Lockheed Martin Energy Systems, Inc.
Figure 4: The multi-robot simulator used in our cooperative robotics research.
REFERENCES


DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.