Vulnerability Analysis of a Subset of Decentralized Information Consolidation Mechanisms in C2

Nicolas Léchevin
Defence Research and Development Canada, Valcartier Research Centre, 2459 route de la Bravoure, Québec, QC G3J 1X5, Canada
Nicolas.lechevin@drdc-rddc.gc.ca

Camille Alain Rabbath
Defence Research and Development Canada, Valcartier Research Centre, 2459 route de la Bravoure, Québec, QC G3J 1X5,
Camille-Alain.Rabbath@*drdc-rddc.gc.ca*

Abstract

The digitalization of the information space and the decentralization of C2 processes are important steps to accelerate decision and planning cycles, improve situational awareness (SA), and enhance C2 agility in the face of multifaceted threats. While digital technologies are key enablers to generating superiority, their exploitation in hostile environments also present many challenges. Tactical and strategic networks can be attacked kinetically and non-kinetically through cyber and electromagnetic attacks and can propagate malicious codes, faulty signals, and erroneous information, which in turn can detrimentally affect networked C2 nodes, platforms, weapons, and sensors. It is therefore critical to identify the vulnerabilities of networked systems, the direct and indirect effects and the risks posed by non-kinetic threats on overall SA and the conduct of operations. Time-critical systems, such as networked (semi-) autonomous weapons (offensive operations) and battle management threat evaluation and weapon assignment (defensive operations) systems, require special attention owing to the irreversibility of their action. What happens in terms of security if (*i*) networks are intermittently or permanently disabled, (*ii*) information is corrupted with deceptive content, (*iii*) computing processes are corrupted by malicious codes, or (*iv*) control authority is lost by network-enabled weapons and operator (e.g., weapon's mission can no longer be aborted)? To discuss some of these issues, we focus on decentralized consolidation mechanisms (information fusion and aggregation), which serves as interface systems with decentralized SA and planning capabilities and adopt high-level abstraction models to analyze some of their vulnerabilities. Each mechanism leads to an aggregation dynamical system whose structure reflects the communication network topology. Therefore, each dynamical system is a function of parameters, graph structure, and variables that represent endogenous and exogenous signals to C2 nodes. Each of these elements can be subject to the injection of corrupted signals and codes. The high-level abstraction adopted allows hypotheses to be formulated about possible classes of vulnerabilities and resulting faulty behaviour, such as oscillatory behaviour (undesirable changes in recommendations) and bifurcation leading to instabilities. The decentralized aggregations considered consists of the average, maximum and majority operators, whose applications include Bayesian information fusion, best plan selection using decentralized bidding, risk-averse option selection (e.g., aggregation of threat levels and object classes), and majority-based recommendation. An analytical result provides a condition for possible transitions towards instabilities. Simulations results illustrate the potential effects of vulnerabilities on a class of C2 processes as functions of key parameters.

1 Introduction

Information consolidation aims to improve the quality and value of information (e.g., accuracy, confidence, relevance) relative to their exploitation by different services of command and control decision support systems, to exploit informational synergies through aggregation, and to resolve conflicts arising for example when situational awareness and decision-making processes are decentralized (see Table 1).

Information aggregation and fusion are similar concepts although marked differences exist [1]. Aggregation is often related to criteria aggregation in multi-criteria decision analysis, where one seeks to represent agents' preferences, whereas fusion is generally applied to situational awareness (SA) information and planning information. The former includes wireless sensor network used for collaborative detection, localization, object classification (recognition and identification), tracking, and pursuit [2],[3], while the latter includes decentralized optimization, deconfliction, and service bidding [4],[8].

Decentralization of aggregation tries to compensate for, or mitigate, drawbacks of centralization, which include: (*i*) vulnerability to communications failure and asset loss; (*ii*) scalability for which it can be shown that the required capacity in communications and in memory of each agent tends to grow with the size of the network [10]; and (*iii*) lack of awareness and responsiveness. There is a low awareness and responsiveness of nodes (and increased sensitivity to network impairment) when the central

node sends back the consolidated information only to a very limited sets of nodes; on the other hand, every node available to participate in the decentralized aggregation is aware of the consensus value.

Table 1: Decentralized information aggregation. (A: Asympto-tic; FT: Finite time; DV/CV: Discrete/Continuous Variable)

avr	**min/max**	max_{1p} min_{1p}	**maj**	**Set operations**
A	A/FT	FT	A	FT
CV: Probability, belief function[18], score, cost, risk, gain, utility, decision/planning measures of performance			DV: Class, level, rank	DV: time constraint
DV: Class, level, ranks				CV: time interval
Inference [18], [19], [22] Optim [8], [9] Learning[20]	Bidding [4],[5],[17] Deconfliction, Statistics	Plan-ning [6], [9]	Statistics [7]	Scheduling

Several decentralized information aggregation rules or mechanisms have been proposed. In particular, the average, minimum and maximum consensus rules, and randomized (pairwise) gossip rules have been extensively studied and applied (Table I) over the last two decades [2], [11], [12], [13], [14]. With these mechanisms, each C2 node only aggregates local information; that is, a mathematical operator processes the information Y received from adjacent C2 nodes and updates local aggregated information X by applying iteratively a mechanism f: $X^{+} \leftarrow f(X, Y; p)$. See the state diagram in Figure 1(a).

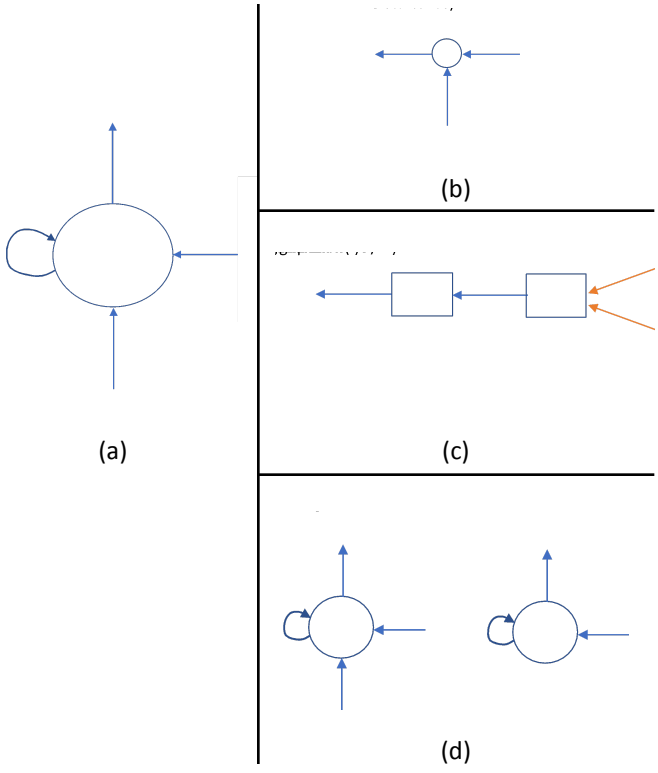


Figure 1: Local aggregation. (a) Update of X as a function of adjacent nodes $Y = \{Y_j, Y_j\}$. (b). Corruption of incoming signals. (c) X as the output of SA process excited by raw data (including disturbance, deception). (d) Loss of edge.

Outcome X^{+} of this mechanism is then propagated in the communication network. After several iterations, under specific connectivity constraints, every local aggregation should reach the same state value as that computed by a central aggregator, should a centralized information consolidation mode be selected. However, local aggregated information can be corrupted by cyber and electromagnetic warfare attacks against tactical networks that exchange information being consolidated. Attacks can corrupt systems and signals of C2 nodes through

- Signal Y received from adjacent C2 nodes (Figure 1(b)), which can be corrupted by jamming (noise) or denial of service (DoS) (delayed signals),
- Endogenous signal X, which, for example, is produced by an impaired SA process internal to the C2 node executing f (Figure 1(c)),
- Parameter p of an algorithm binary code that is corrupted by a malicious code, and
- Operator f, which reflects changes in network structure resulting from denial-of-service attacks (loss of packets), or whose binary code is corrupted (Figure 1(d)).

Corrupted information can then propagate across the network, affecting other C2 nodes in their ability to perform SA and C2 functions in a decentralized manner (e.g., information fusion, plan selection and consolidation). It is thus deemed crucial to analyze the impact of corrupted information on a selection of C2 processes and aggregation mechanisms. A high-level abstraction (in contrast to analyzes carried out at binary code level) is adopted to model (*i*) communication network, (*ii*) information aggregation exploited by a class of C2 processes, and (*iii*) attacks on signals and parameters. In doing so, performance degradation is analyzed and hypothesis about possible vulnerabilities of algorithms (sensitivity to parameter and signal corruption) are formulated for further investigations.

Three decentralized information aggregations are considered here, namely, average aggregation, maximum aggregation, and majority-rule aggregation (e.g., binary voting problem, multiple voting problems [15], [21]). Other aggregation operators, such as the highest and second highest value aggregation (max_{12}/min_{12}), are not discussed here; however, they have similar properties and vulnerabilities. In addition to applications to specific areas of information and decision science and technology (Table I), consensus-based mechanisms have long been

considered to model collective motions and develop control systems that make swarm of mobile systems mimic those behaviours and motions [23], [24]. With this in mind and in the current context of intensive use of ISR/weaponized, air/ground/sea (semi-)autonomous systems in several active theaters of operations, a brief discussion on a class of vulnerabilities of drone formation flight is proposed. It is shown that, similar to some information aggregation scheme, the formation dynamics can undergo a bifurcation depending on the value of a stochastic disturbance parameter.

The paper is organized as follows. Section 2 presents a brief overview of aggregation operators (applications, variables and convergence types) and desired properties. Sections 3 and 4 illustrate and discuss possible vulnerabilities of the three information aggregation mechanisms mentioned above. First, decentralized probability (log-likelihood) average-based aggregation seeking agreement over a network of sensor-classifier for object classification is briefly presented (Section 3). Sensitivities to a class of disturbances (denial of service) and parameter corruption (possibly entailed by hacking action), leading to the occurrence of bifurcation, are analyzed. Second decentralized maximum and majority aggregation mechanisms are presented in the context of category aggregation (e.g., threat level, target classification). Bias and stability issues are analyzed for these mechanisms in Section 4. Section 5 provides an example of collective motion, such as drone formation flight, that undergoes a bifurcation in the event that a digital controller is hacked. The latter can impact operations involving swarms while maintaining the number of operators per vehicle. Section 6 presents concluding remarks. The majority-rule aggregation mechanism is given in the annex to support the discussion of stability issues presented in Section 4.1.2.

2 Aggregation operators: preliminaries

2.1 Common decentralized aggregation operators

Some aggregation operators can be achieved using various algorithms, some of which converges in finite time (i.e., finite number of iterations), such as minimum, maximum and similar operators, or asymptotically, such as the average consensus (e.g., arithmetic and geometric averages, generalized average [16]). It should be noted that the average operator can provide approximation to the min and max operators using the log-sum-exp function; in this case, the min/max aggregation value is reached asymptotically. Aggregation can be applied to continuous or discrete variables. max_{1p} and min_{1p} in Table I are the extensions of max_{12} and min_{12}, which determine the p highest and lowest values of a set of n ($> p$) values, respectively.

2.2 Desired properties

Besides consistency between centralized and decentralized aggregation outcomes (i.e., every node-level aggregation must converge to a common value which is the same as that computed by a central node), decentralized techniques should be stable (Lyapunov stability and input-output stability), and robust to communication delay and time-varying (un)directed graph. The latter may result from the loss of communication links, whether unexpected or planned (to save energy and limit electromagnetic signature), and of nodes (i.e., agents no longer able to operate). A variety of results have been established in the literature regarding, for example, convergence results with random graphs [25], [26], [28] and communication delays with characterization of admissible delay bound [27]. Preserving security and privacy is a critical design specification. For example, security and privacy-preserving maximum and average consensus algorithms based on homomorphic cryptography techniques are proposed in [29]. Privacy-preserving maximum consensus based on random number generation is proposed in [30]. Robustness and resiliency of consensus algorithm through a network under adversarial attacks is another highly desirable property. It is shown in [31] that a malicious agent can compromise a multiagent consensus-based reinforcement leaning by preventing the decentralized algorithm from converging to the desired optimum. In fact, the malicious agent can even select its own optimization criterion to influence the overall optimization criterion designed to serve the purpose of the network of agents in a given environment.

3 Decentralized average aggregation for object classification: bifurcation and noise sensitivity

3.1 Effect of code corruption (positive/negative feedback) and denial of service (delay)

Consider the network of three assets depicted in Figure 2(b) and Figure 2(c), at two time instants, t_1 and $t_2 (> t_1)$, respectively. Each asset i (=1,2,3) (e.g., drone, missile) is equipped with a sensor and a computing capability that executes several processes, including automatic target recognition (ATR), target prioritization (TP) and planning (asset allocation, flight path), as shown in Figure 2(a).

The networked assets {Asset 1, Asset 2, Asset 3} must deliver coordinated effects on a set of targets. The ground truth goes as follows: $O_1 =$ high-value target

(HVT); O_2 = medium-value target (MVT); O_3 = HVT; O_4 = Low-value target. Exchange of information through a tactical network (e.g., Link 16) makes it possible to coordinate maneuvers and effects and to consolidate information generated by the ATR, TP and planning systems. The quality of ATR information has a direct impact on the generation of TP lists, and thus, on the quality of planning service, as suggested in Figure 2(b) and Figure 2(c). In particular, bias or spurious switches in ATR information lead to suboptimal and spurious switches of planning solutions, respectively, which in turn may increase energy consumption of assets and degrade mission performance. In the remainder of this section, focus is put on the object classification (ATR) performed using sensor information; that is, each sensor S_i (onboard Asset i) feeds a classifier. The classifier outputs are consolidated using a decentralized aggregation mechanism presented next. This mechanism involves information exchange through the 3-node network shown in Figure 2(b)-(c).

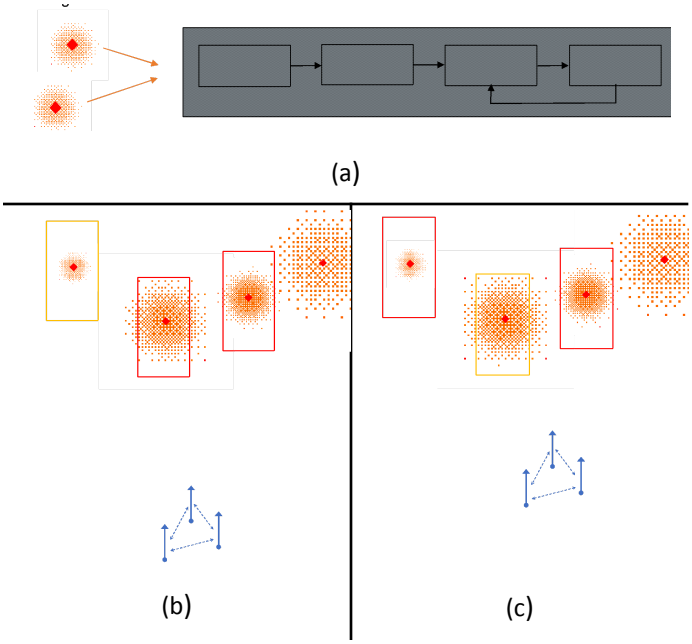


Figure 2: Objected classification for TP and planning (– HVT class; – MVT class). (a) SA and planning processes (Asset i). Classifications and assignment at t_1 (b) and t_2 (c).

A Bayesian approach is adopted to compute a probability distribution about the state of a detected object given observation (i.e., outcome of combined sensor measurement and classification); that is, the posterior at iteration k is a function of the joint likelihood, the prior (i.e., posterior at $k-1$) and a normalization factor. Mathematical details are omitted by lack of space (e.g., see [19] for a similar approach with feature-based classifiers). Assuming that measurements and observation outputs generated by each sensor of the object are independent, the joint likelihood is equal to the product of the likelihood of each node of the network (computed as a function of sensor-classifier statistical models). Taking the log likelihood into account results in a sum that can be computed in a decentralized manner using an average-consensus mechanism as follows

$$X_{i,k+1} = X_{i,k} + \gamma K_{i,k} \sum_{j \in N_i} (X_{j,k} - X_{i,k}), \quad (1)$$

where $i = 1, 2, 3$ and $K_{1,k} = K_{3,k} = 1$. N_i stands for the set of adjacent nodes of node i. $X_{i,k}$ stands for the log likelihood of node i (probability that i observes a class conditioned on the possible object state).

3.1.1 Code corruption ($K_{i,k}$p)

Under nominal operating conditions, gain $K_{i,k}$ ($i = 1, 2, 3$) can be assigned a constant positive value; for example, $K_{i,k} = 1$, which leads to $\gamma K_{i,k} \equiv \gamma$, $\gamma \in R^+$. Therefore, inequality $0 < \gamma < 1/deg_M(G)$, where $deg_M(G)$ stands for the maximum degree of graph G, provides a sufficient condition ensuring that the algorithm converges asymptotically to the network consensus value [32]. This value corresponds to the desired aggregation value. Recall that this value is that a fictitious central aggregation mechanism would have provided. At the next observation iteration, computation of the posterior probability is performed by each sensor using the joint likelihood computed in (1) and by applying the Bayes rule.

Suppose that, under non-nominal operating conditions, node 2 is affected by a malware whose action is modeled using the random variable $K_{2,k}$. This random variable is defined as follows: $P(K_{2,k} = -1) = p$ and $P(K_{2,k} = 1) = 1 - p$, for some probability value $p > 0$. Corruption of the signals received by node 2 may lead to similar variations in gain $K_{2,k}$ and the addition of error signals (considered later). At each iteration k, there is a probability equal to $1 - p$ that the expression in (1), executed by node 2, corresponds to the exact aggregation rule (nominal condition); however, there is a probability equal to p that the rule executed by node 2 is corrupted by simply changing the sign that multiplies the sum.

A sufficient stability condition, which depends on the probability of occurrence of the fault, is provided using a stochastic Lyapunov function candidate, and by applying Theorem 5.2 in [33]. Let P denote a positive definite

matrix. Select the nonnegative function $V_k(X_k) = X_k^T P X_k$. Let $\{F_n\}$ denote a nondecreasing sequence of σ-algebras, where F_n measures at least a sequence of n terms $\{X_k, k \le n\}$. Let $\overline{X} = (X_{1,1} + X_{2,1} + X_{3,1})/3$ (the desired aggregated value), $\widetilde{X}_k = X_k - \overline{X}$, where $X_k^T = [X_{1,k}, X_{2,k}, X_{3,k}]$, and $A_k = (I - \gamma diag(1, K_k, 1)L)$ where L stands for the graph Laplacian. Similar to the standard average-consensus dynamics, consensus value $\overline{X}$ is an invariant quantity (for undirected graphs or balanced directed graphs [32]); that is, $A_k\overline{X} = \overline{X}$ since $L\overline{X} = 0$. Therefore, from (21), the aggregation dynamics in $\widetilde{X}_k$ is given by $\widetilde{X}_{k+1} = A_k\widetilde{X}_k$. The expectation $E[V_{k+1}(X_{k+1}) - V_k(X_k)|F_n]$ can thus be expressed as:

$$E[F_n] = E[X_k^T(A_k^T P A_k - P)X_k|F_n] \quad (2)$$

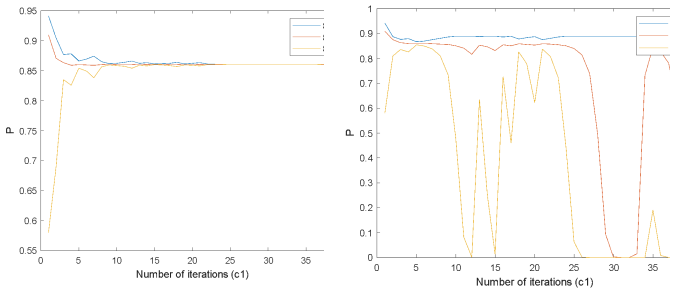
A sufficient condition for the decentralized aggregation dynamics to converge to the consensus value with probability one reads as follows

$$E[A_k^T P A_k] - P <- Q, \quad (3)$$

where Q denotes a positive definite matrix.

Selecting $P = I$ and $Q = I$, the sufficient condition in (3) is satisfied if the eigenvalues of $E[A_k^T P A_k] - I$ are less than one. Matrix $A_k^T P A_k = A(K_k)$ is a function of random variable K_k. Therefore, $E[A_k^T P A_k] = pA(-1) + (1-p)A(1)$. The three-sensor circular network Figure 2(b)-(c) has the following adjacency matrix $[L_1^T L_2^T L_3^T]^T$, with $L_1 = [0\ 1\ 1]$, $L_2 = [1\ 0\ 1]$, $L_3 = [1\ 1\ 0]$. Parameter γ is equal to $0.5/2$ (satisfactory convergence rate). It can be verified numerically that the maximum eigenvalue of $E[A_k^T P A_k] - I$ is equal, in absolute value, to 0.9825 (stable) and 1.0125 (unstable) when $p = 0.64$ and $p = 0.65$, respectively, which provides a rough estimate of the probability that intermittent changes of sign in node 2 may incur potential instability (recalling however that (2) is only a sufficient condition). Calculations are consistent with the trajectories (only class c_1) shown in Figure 3.

In this numerical example, it is assumed that the object to detect belongs to one of the following three class labels $\{c_1, c_2, c_3\}$ and that the sensor-classifier nodes S_1, S_2 and S_3 are characterized by the following probability of detection and classification (two-class statistical discriminator) $P_{D,1} = 0.90$, $P_{D,2} = 0.85$, $P_{D,3} = 0.80$, $P_{C,1} = 0.80$, $P_{C,2} = 0.75$, and $P_{C,1} = 0.7$, which allow computing the likelihood of each sensor.



(a) (b)

Figure 3: Posterior probabilities (c_1). (a) p=0.64. (b) p=0.65.

The Bayesian update and decentralized log-likelihood aggregation are computed using (1) and assuming that the last observation vector, obtained using the sensors, is $\{c_1, c_1, c_3\}$; that is, sensor-classifier S_1, S_2, and S_3 observe c_1, c_1, and c_3, respectively. Given the statistical models and by virtue of the Bayes rule, the steady state of posterior probability trajectories for classes c_1, c_2, and c_3 conditioned on the observation vector is equal to 0.86, 0.03, and 0.11, respectively. Only the posterior probabilities of class c_1 associated with S_1, S_2 and S_3 are shown in the figures. The two other sets of posterior probability trajectories (classes c_2 and c_3) are qualitatively similar but with different steady-state values. Figure 3 highlights changes (bifurcation) in the aggregation dynamics. At critical value $p^* \in (0.64, 0.65)$, the aggregation dynamics becomes unstable (Figure 3(b)).

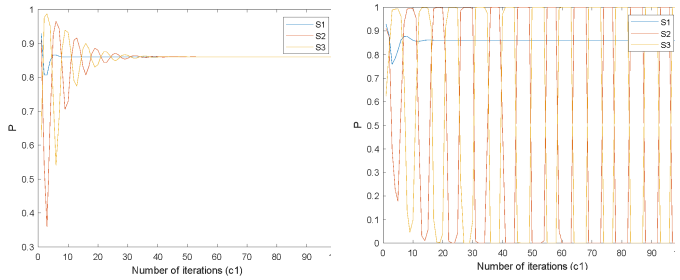
3.1.2 Effect of DoS-induced delay

Denial-of-service attacks, whose effect on dynamical systems is mainly modeled by packet loss or by a constant or stochastic time delay affecting information exchange (see [34] and references therein), can also lead to instabilities, as shown next. Assuming that only node 1 is attacked, the following local aggregation is obtained

$$X_{1,k+1} = X_{1,k} + \gamma \sum_{j \in N_1} (X_{j,k-d} - X_{1,k-d}). \quad (4)$$

Nodes 2 and 3 satisfy (1) with $K_{2,k}$ and $K_{3,k}$ equal to one. The number of steps d in (4) represents the time delay resulting from the DoS attack on node 1. Figure 4 shows

the trajectories of the posterior probabilities for class c_1 (sensors 1,2,3) for two values of d, namely, $d = 1$ and $d = 2$, which correspond to a one-step delay and a two-step delay, respectively.



(a) (b)

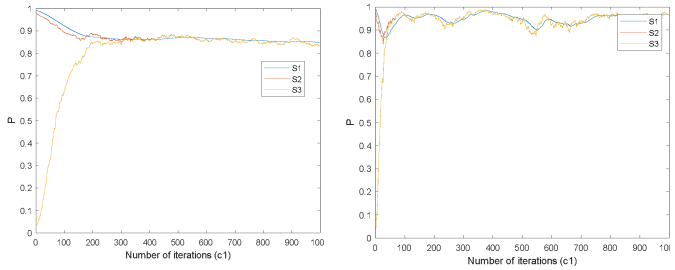
Figure 4: Effect of DoS attack. (a) $d = 1$; (b) $d = 2$.

The decentralized aggregation dynamics becomes unstable when $d \geq 2$, while it remains stable when $d = 1$ despite oscillatory transient. The two other sets of posterior probability trajectories obtained for classes c_2 and c_3 are qualitatively similar; the critical delay is $d = 2$.

3.2 Effect of noise signals (jamming)

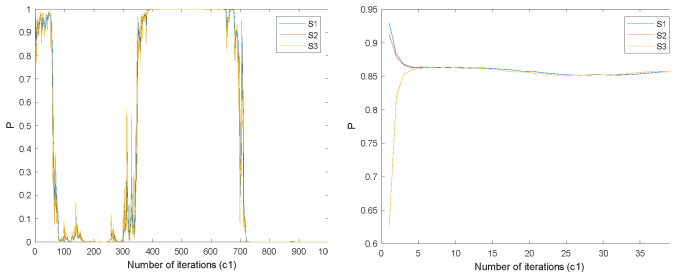
Consider that $X_{i,k}$ and $X_{j,k}$ in (1) are affected by additive noise signals, $e_{i,k}$ and $e_{j,k}$ (Figure 1(b)), respectively. These signals represent exogeneous disturbances caused, for instance, by jamming attacks [35]. Degradation of decentralized aggregation under the action of such disturbances can be assessed using the H_2 norm of the input-output system (i.e., network of local aggregation dynamics) from the disturbance inputs to the steady-state deviation between the outputs of interest and the consensus value $\overline{X}$ for the average aggregation in (1) [36]. Typically, the H_2 norm of a linear dynamical system measures the steady-state covariance of the system output response to unit white noise inputs. Noise signal propagation and amplification in the network depends on gain γ and, under certain conditions, the network struc-ture (representation of process noise and measurement noise in aggregation mechanism such as in (1)).

Assume that only node 1 is under attack; only $e_{2,k}$ and $e_{3,k}$ affect this node. The other nodes are not attacked but can suffer from noise propagation through exchanges of local information. Noise signals are assumed Gaussian and centered on zero with variance σ^2. Tuning parameter γ impacts the trade-off between convergence rate and noise sensitivity, as shown in Figure 5 for class c_1 (σ^2=1).



(a) (b)

Figure 5: Posterior probabilities (c_1). $\sigma^2 = 1$. (a) $\gamma = 0.01/2$. (b) $\gamma = 0.05/2$.



(a) (b)

Figure 6: Posterior probabilities (c_1) (a) $\gamma = 0.5/2$ and $\sigma^2 = 1$. (b) $\gamma = 0.5/2$ and $\sigma^2 = 0.01^2$.

The variance of posterior probability increases with the convergence rate (compare Figure 5(a) with Figure 3(a), noting that the time scales are different). When $\gamma = 0.5/2$ (fast convergence to common aggregate values), the probability of class c_1 becomes meaningless due to oscillations with saturations at 1 and 0, as depicted in Figure 6(a). Gain γ equal to $0.5/2$ yields exploitable probability values when σ^2 is decreased to a low value equal to 0.01^2 in Figure 6(b).

4 Maximum and majority aggregation of class: corrupted information and instability

4.1 Corrupted SA, overkill, underkill

We could think of another type of vulnerability whose malicious exploitation could distort the maximum (risk-averse) aggregation of networked risk assessment capabilities, such as threat evaluation (TE), as explained below. Two rules (finite time, asymptotic) enable the maximum aggregation. The exact maximum rule, given by

$$X_{i,k+1} = \{X_{i,k}, X_{j,k}\}, \quad (5)$$

$k > 0$, $i \in \{1, \ldots, n\}$, $X_{i,k} \in R$, ensures a finite-time convergence [37]. The maximum path length of the graph is an upper bound of the maximum number K of iterations required for every $X_{i,K}$ to maximum aggregation. As already mentioned, the approximate maximum rule is based on the Log-Sum-Exp function [38]

and results in an average consensus based aggregation whose convergence is asymptotic (similar to (1)) with a speed less than or equal to $1 - \gamma\lambda_2$. λ_2 stands for the algebraic connectivity of G [39], which is also the second smallest eigenvalue of the graph Laplacian.

A TE capability, part of a battlespace management system, assesses and categorizes the threats that are detected, recognized and identified within the volume of interest. In doing so, they are prioritized (e.g., threat level and rank) and possibly engaged provided that a set of conditions are satisfied [22]. Assume that networked assets (e.g., ground and airborne sensors) are equipped with computing units allowing TE and other C2 processes, such as those depicted in Figure 9, to be executed in a timely manner. TE outputs (sorted lists, threat levels) are aggregated in a decentralized manner, allowing local decision-makers and planners to quickly react. A malicious code could corrupt the aggregation of such process in, at least, two ways, depending on the aggregation operator.

The malware could frequently inject messages corresponding to a threat level lower than the average aggregated level, such as frequently injecting the lowest threat level. This type of information corruption is likely to result in an aggregated threat level that tends to underestimate an entity's actual threat level, especially if the aggregation is based on the minimum (risk prone), majority, or weighted average operator (when applied to scores instead of levels). Underestimation is likely to generate suboptimal resource planning, such as response delay, insufficient resource matching or even no resource assigned.

On the other hand, the malware could frequently inject messages corresponding to 'high-level threat' instead of 'low level threat', which is likely to result in an aggregated threat level that tends to overestimate the actual threat level of an incoming entity, especially if the aggregation is based on the maximum operator (risk averse). Overestimation can lead to unnecessary expenditure of resources, resulting in premature depletion of those resources.

Anomaly detection is thus necessary to appropriately plan resources. For example, the minimum aggregation of independent exponential distributed variables (interarrival time of Poisson process events) yields an exponentially distributed variable, whose parameter can be computed. Assuming that input signals are emitted by events whose occurrence are independent and exponentially distributed, a statistical test could be designed to detect anomalies resulting from the injection of signals that are not exponentially distributed. This approach could be extended to other aggregation rules and input signal modeling assumptions using machine learning techniques.

Corruption of maximum aggregation may also take place when selecting the best defense/offense plans within a coalition; for example, the authors in [17] proposes bidding strategies to coordinate the engagement plans of national naval forces taking place in an international coalition context. Selecting the best two plans using a modified maximum aggregation, termed max_{12}, may also be desirable either to combine them and improve mission performance or to select a plan based on a conditional maximum operator. The conditional maximum makes it possible to opt for the second-best plan, if the best plan is inappropriate due to a prohibitive secondary criterion (cost, risk) and if the degradation in the first criterion remains acceptable.

Consider two examples of maximum aggregation. Recall that the maximum operator can process discrete input signals such as classes (e.g., object identity, threat level) and continuous input signals such as planning scores (e.g., probability of successful effect). Both types of input signals (class and score) are considered. The objective here is to show that, after inferring the topology of a network of (blue) C2 nodes by applying available topology inference techniques [40], cyber electromagnetic attacks could target specific subsets of nodes to inject packet of information and thus corrupt the maximum aggregation as discussed above.

To simplify the presentation, when considering class aggregation, it is assumed that each node is initialized with one of the two possible classes; an example of a set of two classes is {Low, High}, which characterizes the risk posed by an incoming threat. Concerning the score aggregation, each node is initialized with a score value selected randomly and uniformly over $[0, 1]$; therefore, there are as many initial score values as there are nodes in the graph. Three graphs are considered to assess information degradation as a function of graph. They consist of: (*i*) a 5-node acyclic (chain) graphs G_5^{ac} (maximum path length equal to 4); (*ii*) an 11-node small-world graph G_{11}^4 (Figure 7(a)) with an average degree equal to 4 (average path length equal to 1.57; maximum path length equal to 3); and (*iii*) an 11-node small-world graph G_{11}^6 (Figure 7(b)) with an average degree equal to 6 (average path length equal to 1.27; maximum path length equal to 2). Recall that small-world network properties characterize random networks having

clustering and short average path length. The graphs are obtained using WattsStrogatz function (Matlab).

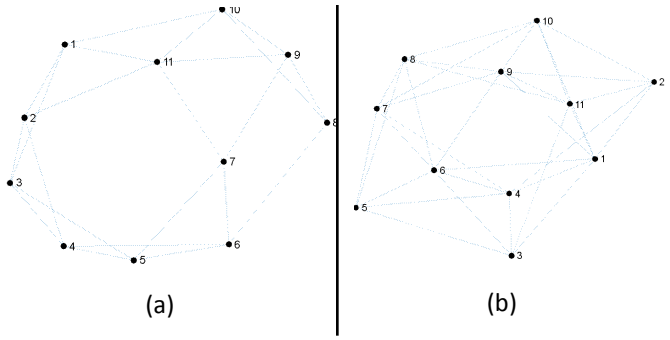


Figure 7: 11-node small-world graph (a) G_{11}^{4}; (b) G_{11}^{6}.

The histogram in Figure 8 showing the frequency of the number of iterations required to reach the maximum value (maximum class level) is obtained by simulating exhaustively every combination of classes for graph G_5^{ac} without edge loss. Similar simulations are carried out (i) for all graphs, (ii) maximum aggregation of class levels and of scores, and (iii) by taking into account edge loss. At each time slot, each link has a probability p to fail equal to 0.2 (independent failures). 500 Monte Carlo simulations are run. Letting n denote the number of C2 nodes, the simulations of score aggregation dynamics are initialized with 2^n random score vectors. Meaningful frequencies are given in Table 2.

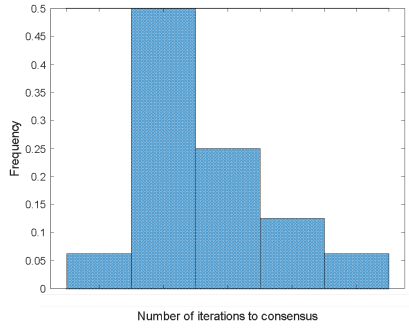


Figure 8: Distribution of iteration number to maximum aggregation (class).

Figure 8 and Table 2 can provide hints on possible attack strategies. Such a strategy could consist in selecting the set of initial class or score values that both lead to the fastest corruption of the aggregated value and involve attacking the fewest nodes possible. First, it can be noticed that the maximum is reached faster for the two-class aggregation than for the score aggregation; one or two iterations (classes) versus two to four iterations (scores) results in probability greater than 0.7. This suggests that, when considering aggregation of a limited number of classes (e.g., threat evaluation of an entity classified as a ground target, a decoy, or something else), contamination of the network of sensor-classifier pairs through decentralized aggregation can quickly spread.

The same applies to scores but with slightly higher numbers of iterations. An experiment similar to that with G_{11}^{4} (s), G_{11}^{6} (s), $p = 0.2$ (500 MC simulation runs) is carried out, except that two elements are modified: (*i*) in every 11×2^n random score vector, a single entry (single node in each network) is set to one; and (*ii*) the stopping rule for the aggregation algorithm is that every node reaches a value equal to one. In other words, this value (one) corresponds to a spurious aggregated value, such as a risk value that is maliciously set to its upper bound. The resulting histograms (not shown) are similar to those in Table 2. This means that a single corrupted node (selected randomly), can propagate, with a probability greater than 0.7, a score of one (i.e., the maximum score value) across the entire network. For example, if scores are measures of effectiveness of plans (e.g., resource allocation) generated by networked C2 nodes, all nodes agree on the same maximum value. The latter is imposed by the infected node independently of the actual value of the plan generated by the node.

Table 2: Distribution of iteration number to maximum aggregation for class (c) and score (s). Edge loss with $p = 0.2$.

	1	2	3	4	5
G_{11}^{4} (c)	0.60	0.35	< 0.05	< 0.01	< 0.01
G_{11}^{6} (c)	0.82	0.17	< 0.01	-	-
G_5^{ac} (s)	0.23	0.41	0.36	-	-
G_{11}^{4} (s)	-	0.05	0.70	0.2	< 0.05
G_{11}^{6} (s)	-	0.76	0.23	< 0.01	< 0.01

4.2 Sensitivity/stability issues

The information to be consolidated (aggregation, fusion) by a subset of C2 nodes, having an appropriate authority level, is produced and processed by different types of C2 nodes, including (*i*) information generators, which process and perform reasoning tasks on observation and measurement signals (e.g., platforms, sensors, troops), (*ii*) information consumers, such as SA and planning processes, and (*iii*) both types of nodes (information generator and consumer). For example, in collaborative network-enabled weapons (NEW) depicted in Figure 9, the information produced by each weapon's automatic target recognition (ATR) and prioritization systems can be fused to improve the overall quality of SA information in support of dynamic targeting, which in turn may lead to collaborative re-tasking (under the authority of the

mission commander) and flight path re-planning [41]. Another example concerns the threat evaluation and weapons assignment (TEWA) [42] of battlespace management systems in a coalition context, where several nations aims to build a common operating picture and coordinate their defense plans (e.g., weapons-target assignment (WTA) and scheduling). The high-level information generated by the coalition TEWA systems (e.g., threat level and rank, resource allocation, action schedule, measures of performance) can be consolidated to increase the coalition quality of information (QoI), value of information (VoI), and quality of C2 services.

ATR and TE are reasoning capabilities that both depend on several other processes, such as data acquisition (e.g. detection, tracking), data association, and object recognition and identification. In principle, the output information generated by each process can be consolidated. The quality of ATR and TE information is thus dependent on these processes and on how uncertainties about threats/targets (e.g., position, identity, intent, risk assessment, maneuver and deception tactics) are managed. For example, are systems and algorithms, such as TE, ATR, and target prioritization, overly sensitive to uncertainties about the entities in the picture compilation, imperfect and incomplete information (e.g., missing tracks, measurement noise, which tends to increase with target-radar distance: long-range surveillance radar versus fire-control radar)? Moreover, the enemy can jam and deceives TE and ATR through cyber and electromagnetic attacks and tactics involving decoy, control of electromagnetic emission, camouflage, and target relocation. Unstable TE and ATR, resulting in oscillatory target and TE lists, could in turn trigger frequent and undesirable replanning, wasting resources and overloading operators' cognition.

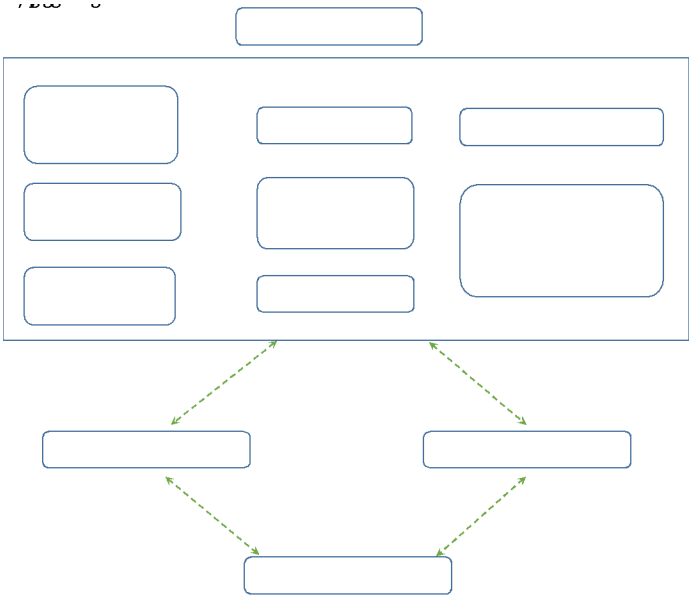
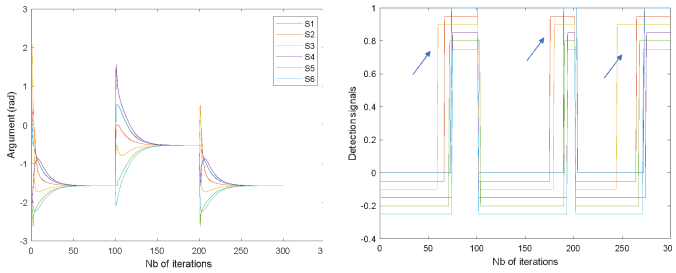


Figure 9: Four collaborative NEWs (e.g., missiles) with some of the core C2 functions.

For the purpose of illustration, consider six NEWs equipped with ATR or six networked TEWA systems, which seeks to agree, in a decentralized manner, on a common assessment about a potential target to strike (e.g., suppression of enemy air defense), or a threat to intercept. More precisely, each ATR generates a class label c_2 (air defense system), c_1 (decoy), or c_0 (something else). A consensus is built using a decentralized majority aggregation, whose mechanism is presented in the annex. In the case of TEWAs, the threat class generated by the TE corresponds to low-level (c_0), mid-level (c_1), or high-level (c_2) threat.

Three graphs of increasing density are considered: (*i*) a 6-node acyclic (chain) graphs G_6^{ac} defined in the annex (maximum path length = 5); (*ii*) a 6-node small-world graph G_6^4 (average degree = 4; average path length = 1; max path length = 2); and (*iii*) a complete 6-node G_6^5 where each node is connected to every other node of the graph (average degree = 5). For all graphs, the majority aggregation dynamics is initialized with $(c_0, c_0, c_1, c_2, c_2, c_2)$; that is, the ith entry corresponds to the object class predicted by node i. After a transient, all six C2 nodes converges to the same numerical value. This value belongs to an interval, which by definition indicates that c_2 is the majority class ($M = c_2$), as shown in Figure 10(a), Figure 11(a), Figure 12(a). This is also shown with the decision signals $d_{f,i,k}$ ($i = 1, \ldots, 6$) in Figure 10(b), Figure 11(b), Figure 12(b) (see the annex for the

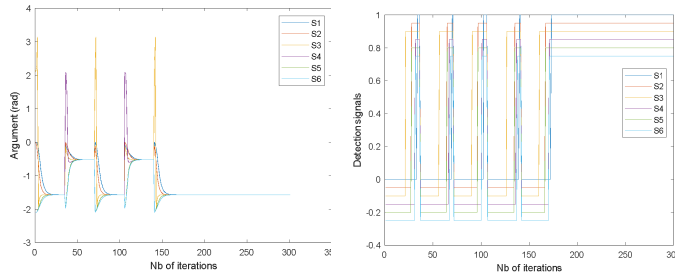
definition of d). Each $d_{f,i,k}$ informs node i that its local aggregation state has reached the majority class.



(a) (b)

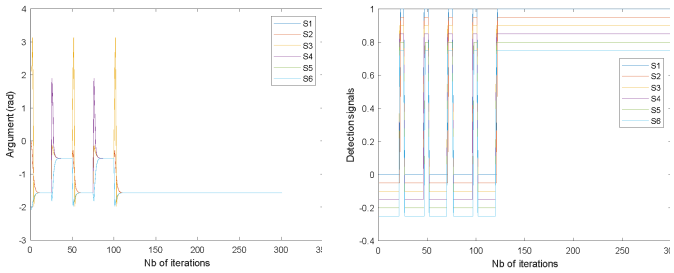
Figure 10: Trajectories of aggregation indicators specifying the identity of the majority class M (G_6^{ac}). (a) Angle trajectories. (b) Detection signals $d_{f,i,k}$.

Convergence gets faster as the algebraic connectivity increases, G_6^5 having a greater connectivity than G_6^4, and G_6^4 than G_6^{ac}. Another consequence is that faster convergence makes the majority aggregation more reactive to the occurrence of events, but also more sensitive to fast undesired class transitions generated by ATR or TE uncertainties, as depicted in the figures. Class transitions occur when $(c_0, c_0, c_1, c_2, c_2, c_2)$ switches to $(c_0, c_0, c_0, c_1, c_2, c_2)$, whose majority class is c_0 $(= M)$, and vice versa. Therefore, fast switching ATR or TE outputs, caused by high noise (jamming) or deceptive enemy tactics, gives rise to frequent undesired transitions of majority class (aggregated information). Complete graph G_6^5 can monitor four transitions occurring in 100 s, while graph G_6^4 can monitor the same number of transitions in 140 s. Chain graph G_6^{ac} takes more time to react to the occurrence of transition.



(a) (b)

Figure 11: Trajectories of aggregation indicators specifying the identity of the majority class M (G_6^4). (a) Angle trajectories. (b) Detection signals $d_{f,i,k}$.



(a) (b)

Figure 12: Trajectories of aggregation indicators specifying the identity of the majority class M (G_6^5). (a) Angle trajectories. (b) Detection signals $d_{f,i,k}$.

Sensitivity of data acquisition, processing and reasoning systems (e.g., ATR, TE) to exogeneous disturbances, whether passive (measurement noise) or active (attacks), can be mitigated by different approaches, such as multi-model and adaptive filtering (e.g., directly applied to sensor signals) and information fusion, hypothesis testing (Is the threat targeting asset A rather than asset B despite uncertainties?), and machine learning (e.g., adversarial reasoning based on learned enemy deception tactics; multiagent reinforcement learning). However, trade-off between (*i*) performance, such as high responsiveness and high probability of detection and correct classification, and (*ii*) robustness, such as stabilized information and low false alarm and misclassification rates, must still be considered. A similar trade-off is illustrated in Section 3.2. where active noise corrupts the average aggregation of log-likelihoods, which depends in particular on the convergence rate of the aggregation dynamics.

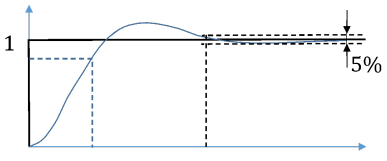
The occurrence of unstable reasoning systems raises the question of when it is best to consolidate information. The quality of TE and ATR information indices is instrumental in deciding when to consolidate information. QoI can be assessed using, for example, statistical and system-theoretic metrics (see [43] for a review of QoI/VoI metrics for wireless sensor networks applications). Statistical QoI includes, for example, uncertainty sets for object classification. Uncertainty sets can be derived, independently of the approach taken to classify objects, using conformal prediction techniques [44]. These distribution-free techniques produce, with formal guarantee, prediction sets that contain the ground truth up to a prescribed level of confidence. A conformal prediction can be associated with each sensor-classifier of a network (e.g., collaborative NEWs with consolidated ATR). A large uncertain set means that the classification model is uncertain or that the object is difficult to classify. In such a situation, the classification outcome should be interpreted and exploited for aggregation with

care.

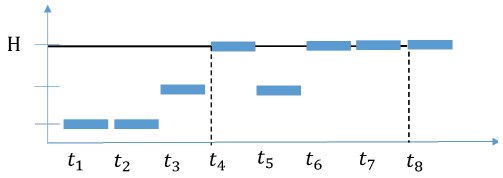
A complementary function to information consolidation is the information dissemination, part of the core C2 functions of collaborative NEWs shown in Figure 9. The information dissemination function aims to control information exchange, and therefore can help mitigate instability by publishing TE or ATR information for consolidation (*i*) when the latter is associated with a satisfactory level of confidence, as determined for example by conformal prediction, or (*ii*) when the risk value is deemed unacceptable. By risk is meant the expected cost of failure computed using probability of failure (e.g., salvo of interceptors fails to intercept a threat that aims at a ground asset having some tactical, operational or strategic value).

The second type of QoI mentioned above is system theoretic and concerns transient analysis. Consider several TEWA systems in a coalition context, where nations exchange and consolidate TE information to coordinate actions. To analyze TE outputs, an analogy can be made with transient response analysis indices used to assess the response time (or rise time) and time to stabilization (or settling time) of continuous-time dynamical systems, as illustrated in Figure 13(a). These metrics are useful for designing a control system such that the closed-loop system behaves within prescribed specification limits.

To simplify, assume that there are three threat levels $\{L, M, H\}$. Consider a situation similar to that of the majority aggregation with classes c_0, c_1, and c_2: due to passive or active uncertainties that corrupt target tracking as well as recognition and identification processes, the TE outcome C_k provided at time t_k by each nation coordinator may not yield the actual threat evaluation, that is, the threat level and rank that would be proposed by a subject matter expert who would have access to perfect and complete information about that threat. In Figure 13(b), the actual threat level is high (H). It is thus expected that, at some time instant $t_{k'}$, $C_{k'} = H$.



(a)



(b)

Figure 13: Transient analysis of TE outcomes. (a) Rise time and 5% settling time for continuous-time system. (b) Response time t_4 and time to stabilization t_8 for discrete space and time systems such as the threat evaluation function.

However, C_k can display undesired behaviour, such as oscillations, after $t_{k'}$. Following the example of continuous-time system transient analysis, an approach consists in defining the time to stabilization as the time $t_{k''}$ such that the following inequality holds

$$max\,(L_{k'\to k''}, M_{k'\to k''}, H_{k'\to k''}) \geq N, \tag{6}$$

where

$$H_{k'\to k''} = card\,(\{H, \ldots, H\} \cap C_{k'\to k''}), \tag{7}$$

$$M_{k'\to k''} = card\,(\{M, \ldots, M\} \cap C_{k'\to k''}, \tag{8}$$

$$L_{k'\to k''} = card\,(\{L, \ldots, L\} \cap C_{k'\to k''}), \tag{9}$$

$$C_{k'\to k''} = \{C_{k'}, C_{k'+1}, \ldots, C_{k''}, C_{k''}\}, \tag{10}$$

$$t_{k'} = t_{k''} - T + 1.$$

$T > 0$ and $N \in N$ denote a prescribed time window length and a decision threshold, respectively. The cardinality ($card$) of each set $\{L, \ldots, L\}$, $\{M, \ldots, M\}$, and $\{H, \ldots, H\}$ used above is $k'' - k' + 1$, which corresponds to the number C_k generated between $t_{k'}$ and $t_{k''}$.

The above inequalities tell that a level is deemed stable if (*i*) the corresponding set defined between time indices k' and k'', has a maximal cardinality (left-hand side of (6)), and if (*ii*) the number of times this level occurs over $[t_{k'}, t_{k''+1})$ is greater than a prescribed threshold N, as specified in (6). $t_{k''}$ is the time instant at which this inequality holds and corresponds to the time to stabilization.

The time response is defined in retrospect to the time to stabilization. Knowing $t_{k''}$ and the stable level as defined above, the response time can be defined as the first time instant t_{k_0} such that $C_{k_0} = C_{k''}$. Of course, the time response alone is not a satisfactory quality index since

the TE outcome may remain oscillatory for an undesirable amount of time after t_{k_0}.

Letting T and N be equal to 5 units of time and to 4, respectively, stability is claimed when four high levels occurred over the past five units of time. Applying these parameters to Figure 13(b), time to stabilization and response time correspond to t_8 and t_4, respectively. It should be noted that the claim about stability and time to stabilization is released at a time subsequent to t_8, that is, once the threat level is determined and published by the TE function.

Decentralized aggregation scheme, such as the majority rule, could exploit the quality indices defined above. For instance, a simple rule might consist in performing aggregation only with nations that provide stable TE outcomes. In doing so, the last available coordination information is taken into account in the sense defined by the time to stabilization; that is, the threat level corresponding to the last time to stabilization is considered. On the other hand, each national TE capability should commit to generating high QoI in TE by ensuring that upstream processes, such as tracking, filtering, object recognition and identification, and information fusion, also provide high QoI, to some extent immune to hacking.

Class, category, and level typically involve comparison of a score function or probability distribution with thresholds. An alternative approach to oscillation mitigation applied to classes is to replace class aggregation by score aggregation, along with oscillation mitigation, followed by comparison with thresholds; that is, 'aggregate and compare' is substituted for 'compare and aggregate'.

5 Networked (semi-) autonomous systems: bifurcation of formation dynamics

Vulnerability of networks of drones to malicious attacks have attracted the interest of industries and militaries. Strategies developed to detect and mitigate security issues could also prove useful for operating formations of collaborative NEWs (C-NEW), as shown in Figure 9. Interestingly, Choudhary et al. propose in [45] a survey of existing intrusion detection systems (IDS) approaches considered to detect abnormal behaviour and illegal activities affecting networks of drones, and identify science and technology challenges, such as IDS detection latency and computational cost (increased implementation overhead impacts negatively power consumption and may decrease drone performance), threat and drone behaviour modeling, and satisfactory network throughput. da Silva et al. propose supervised and unsupervised machine-learning-based IDS techniques (stacked autoencoder, federated learning) to detect in-flight anomaly and network attacks in drone swarm [46]. Pirani et al. provides in [47] an interesting survey of approaches for analyzing the ability of distributed control to mitigate the impact of targeted faulty behaviour resulting from attacks. Interestingly, different types of attacks targeting automotive and intelligent transportation systems (e.g., denial of service attacks in connected vehicles causing delays in vehicle-to-vehicle information communications [34]) as well as swarm robotics are reviewed, indicating potential nuisance of attacks to C-NEW and tactical combat networks.

The outcomes of a NEW health monitoring capability (Figure 9) should be used to trigger self-healing and mitigation strategies (resilient systems) at several levels of control and command, as follows.

1) A NEW should detect and diagnose likely sources of faults, and prognose its impact on (*i*) its ability to reach its objectives (target assignment, time constraints, flight path constraints) and on (*ii*) C-NEW mission objectives; self-healing strategy can be undertaken at several levels; for example, at

- NEW level, if NEW controllability permits it;
- C-NEW level, where an analysis of mission cost, risk, and benefits allows the network to decide for the best option, such as adapting C-NEW plan by modifying constraints so that they reflect the degraded condition of NEW (e.g., in multi-agent coordination problem affected by network failure, a strategy adopted by non-faulty agents could consist in accommodating the faulty agents in the accomplishment of their tasks [48]); and at
- Other C2 nodes, by adapting the mission plan, aborting the mission or at least aborting the degraded NEW.

2) C-NEW should be able to carry out the same diagnosis, prognosis and adaptation functions as in the preceding point, but as a collective reasoning and planning capability, whatever the coordination mode selected.

3) C2 nodes, such as command post, should pre-empt possible plan failures, involving C-NEW and other networked assets, and generate contingency plans, ready to be executed.

Let us illustrate the impact of a specific type of attacks that can lead to abnormal behaviour (instability) at the formation level. The behaviour is qualitatively similar to that analyzed in Section 3.1.1, although the networked

systems below are nonlinear. Health monitoring capability is not further discussed but references are provided (stability monitoring). Consider a chain-type formation of five autonomous systems (e.g., drones) shown in Figure 14. Each vehicle is represented by a 6 degree-of-freedom (DOF), nonlinear state-space model in closed loop with a control system (quadrotor drone model in [49]). To account for sensor imperfection, white Gaussian measurement noise is added to the output vector, which comprises data from GPS and inertial measurement units (IMUs).

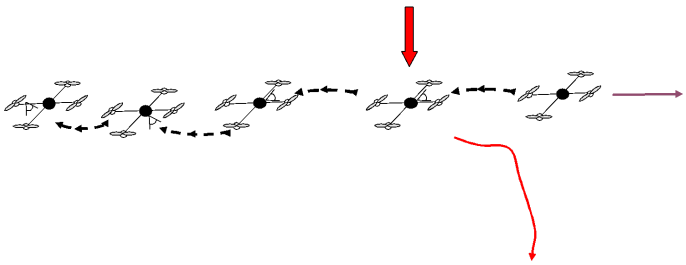


Figure 14: Formation of drone affected by unexpected events. Possible trajectories are drawn following the occurrence of a malware-induced disturbance.

The control system consists of two feedback loops (Figure 15(a)); namely, the autopilot, which leverages the information obtained from the IMUs, and the formation controller [50] (Figure 15(b)), which is a decentralized control law using local information about its neighboring vehicles. The structure of the formation controller is to some extent similar to that of the decentralized aggregation mechanism in (1); indeed, it can be interpreted as a consensus algorithm on information that represents physical quantities, such as angles and distance between neighboring agents. A neighbor of vehicle i is defined as any vehicle that can be sensed by i. Each of the vehicles 2 to 5 of the formation illustrated in Figure 14 has exactly one neighbor (other types of formation flight could be considered).

Each vehicle may be affected by failures whether intended or not. Here, we focus on the effect of the presence of a malicious code that has been inadvertently uploaded on or injected in (through cyber electromagnetic attacks) the electronic board of at least one drone. One can think of an infection similar to Stuxnet propagation through removable disks [51].

The malware consists, as suggested in Figure 15(b), of a simple process that multiplies the data x_i of vehicle i by $+1$ (normal condition) with a probability p^* and by -1 (abnormal condition) with a probability $1-p^*$. The formation controller operating in normal condition feeds back x_i. Therefore, it is expected that using the corrupted data $-x_i$ have a destabilizing effect on the vehicle's dynamics and thus on the formation's dynamics. However, due to the stochastic nature of the malware action, several types of abnormal behaviour can be expected, such as steady-state error, oscillation, and instability.

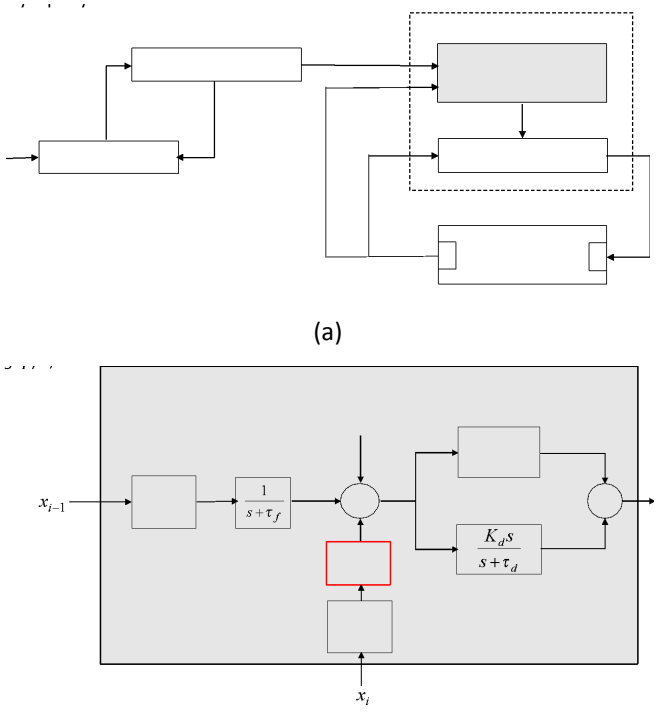


Figure 15: Control system of vehicle i infected by a malware. (a) Inner-loop (autopilot) and outer-loop (formation flight) controllers. (b) High-level representation of the effect of a malware on the formation controller.

Interestingly, it should be noted that this simple malicious mechanism may be interpreted as a model for intermittent losses of communications should this mechanism be applied to data x_{i-1} while assuming that x_{i-1} is obtained by means of a communications network rather than by onboard sensors. The malware could also disable receiver and emitter components onboard the drones. The stochastic stability analysis of the infected formation dynamics is not detailed here. Rather, results of a numerical experiment based on a full 6-dof nonlinear system are presented next. The formation of quadrotor drones, whose speed is about 1 m/s, is required to track a straight-line trajectory of 100 m. At time instant 75 s, the malicious mechanism is triggered with probability p^*. Intuitively, the chance of facing instabilities increases as p^* decreases. This idea is confirmed in Figure 16 and

Figure 17, which can be supported by a stochastic stability analysis similar to that sketched in Section 3.1. Figure 17 shows that, when $p^* = 0.5$, the prescribed straight-line trajectory is unstable for followers 2 to 5. Figure 16 shows that the formation remains stable for larger values of p^*, while vehicles 2 to 5 follow the leader with a non-zero steady-state error.

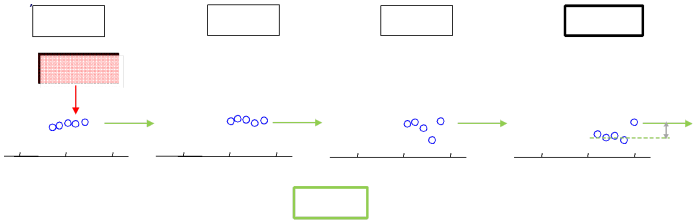


Figure 16: Stable formation with transition to a non-zero steady-state tracking error ($p^* = 0.9$).

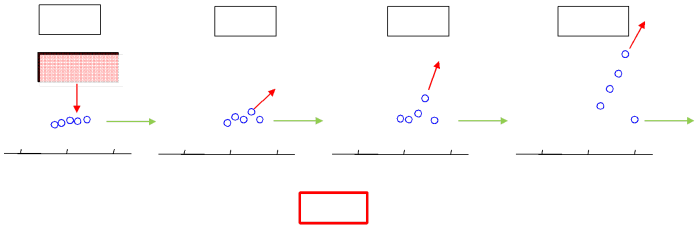


Figure 17: Stability-instability transition ($p^* = 0.5$).

Health monitoring capability, onboard systems and in higher-level authority C2 nodes is necessary to mitigate mission risk, as discussed for NEW applications. In general, fault detection algorithms are based on the derivation of residues, which are designed to be sensitive to faults and failures affecting, for example, actuators or sensors. Regarding formation stability, monitoring the spectrum of some appropriately designed matrix (e.g., resulting from system linearization) is one approach to detect anomalies originating from the activation of a malware, as illustrated above. Stability monitoring of a class of networked dynamical systems has been proposed using physics-based model, linearized models, and statistical hypothesis test (CUSUM) for detecting changes in the behaviour of dynamical systems infected by a malware [52]. This work detects the start of unstable behaviour, assuming that the linearization of dynamics remains valid. It would be interesting to apply machine-learning tools, such as an autoencoder [46], to detect and identify faulty behaviour in a swarm when nonlinearities can no longer be neglected.

6 Conclusion

Decentralized aggregation mechanisms, part of the information consolidation modules interfacing with the SA and planning functions of C2 nodes, present interesting properties compared to centralized aggregation. On the other hand, those mechanisms are not immune to various types of cyber and electromagnetic attacks, which propagate from local aggregators to another, leading to possible instabilities, self-sustained oscillations, and noise amplification. The analysis presented here adopts a system-theoretic, high-level abstraction perspective to identify different classes of vulnerabilities of aggregation mechanisms under the action of attacks. The proposed classes of vulnerabilities are by no means exhaustive. Moreover, attacks are represented at a high level of abstraction under the form of (stochastic) parametric uncertainties and disturbance signals. Further work includes establishing connections between the algorithmic vulnerability analysis presented in Sections 3-5 and low-level representation of attacks (e.g., binary code), disturbance signals (jamming signals) and deceptive tactics.

Risk mitigation strategies are of prime importance and are multi-faceted. Technologies are developed so that systems can operate, with possible graceful degradation, in hostile denied and degraded environment; for example, alternate navigation techniques should allow (semi-) autonomous systems, such as ISR drone, loitering munition, and NEW, to guarantee mission objectives despite global navigation satellite system-denied environments. Battlespace management systems should integrate cyber and electromagnetic attack defensive (e.g., observation, protection, asset planning) and offensive C2 capabilities. For example, such a capability could characterize denied and degraded zones (e.g., location, area of influence, types of attacks, probability of effect on assets) and plan asset motion and maneuver with the appropriate level of autonomy, exploitation, and information sharing strategy. A fully autonomous mode (drone) could be allowed when deemed necessary; for example, a temporary loss of communications due to crossing a jamming zone may justify full autonomy.

As suggested in the analysis, design trade-offs are unavoidable. Some of them, involving the exploitation information, are listed below.

Performance vs robustness. In control theory, there is a waterbed effect involving performance and robustness of linear, time-invariant systems, expressed by the Bode sensitivity's integral. Robust optimization can desensitize planning to certain classes of uncertainties, but may come at the expense of resource expenditure. For decentralized information consolidation algorithms, convergence rate can be improved with dense network at the expense of increase complexity cost (increased number of connections) and increased sensitivity to the

occurrence of spurious events.

Performance versus cost. To achieve objectives in changing environments, adaptive systems are developed with the aim of dynamically controlling their behaviour using estimation, identification and learning mechanisms. However, adaptivity is not a zero-cost strategy. Adaptive mechanism can be sensitive to uncertainty, such as noisy signals, possibly resulting in undesirable costly changes. For example, a high rate of false alarms can unexpectedly trigger dynamic targeting process, possibly overloading the cognitive capacities of operators. More fundamentally, networking weapons and sensors is expected to create SA and decision-making opportunities, yet creating new vulnerabilities such as network and C2 service fault/failures caused by electronic warfare and malwares. Faulty C2 service and network can be dramatically costly, especially when dealing with weapons.

Information temporality versus exploitability. Fast communication of information (e.g., fast detection) may come at the expense of information accuracy, precision, and credibility (e.g., poorly characterized threats). It is expected that networking sensors can improve this trade-off provided gain in information is significant enough during a common opportunity window. If the availability of sensors is limited, the dilemma between speed and QoI persists.

Information exploitability versus control authority. Along the same line, waiting for a high QoI may limit the controllability of systems in achieving their goals (e.g., reach targets), in particular, in delivering effects under specific time and space constraints. Sharing information through tactical networks can however improve the reachability potential of collaborative network-enabled weapons provided that networks and information consolidation mechanisms continue to operate efficiently during critical decision periods. For example, sharing information may help increase the reachable set of the formation by dynamically re-allocating weapons (besides maneuvering) at the right decision points. A formation-level approach is expected to exceed the reachability performance obtained when each weapon decides and plan individually.

References

[1] D. Dubois, W. Liu, J. Ma, H. Prade, “The basic principles of uncertain information fusion. An organised review of merging rules in different representation frameworks,” Information Fusion, vol. 32, P. A, pp. 12-39, 2016.

[2] O. Hlinka, F. Hlawatsch, and P.M. Djuric, “Distributed particle filtering in agent networks: A survey, classification, and comparison,” IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 61-81, 2013.

[3] X. Ge, Q.-L. Han, X.-M. Zhang, L. Ding and F. Yang, “Distributed Event-Triggered Estimation Over Sensor Networks: A Survey,” IEEE Transactions on Cybernetics, vol. 50, no. 3, pp. 1306-1320, 2020.

[4] H.-L. Choi, L. Brunet, and J. How, “Consensus-Based Decentralized Auctions for Robust Task Allocation,” IEEE Transactions on Robotics, vol. 25, no. 4, pp. 912-926, 2009.

[5] L. Johnson, H.-L. Choi and J. P. How, “Convergence analysis of the Hybrid Information and Plan Consensus Algorithm,” 2014 American Control Conference, Portland, OR, USA, 2014, pp. 3171-3176.

[6] T. Li, H.-S. Shin and A. Tsourdos, “Efficient Decentralized Task Allocation for UAV Swarms in Multi-target Surveillance Missions,” 2019 International Conference on Unmanned Aircraft Systems, Atlanta, GA, USA, 2019, pp. 61-68.

[7] Y. Cai, J. He, W. Yu and X. Guan, “Consensus-Based Data Statistics in Distributed Network Systems,” 2018 IEEE Conference on Decision and Control, Miami, FL, USA, 2018, pp. 4206-4211.

[8] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, “DLM: Decentralized Linearized Alternating Direction Method of Multipliers,” IEEE Transactions on Signal Processing, vol. 63, no. 15, pp. 4051-4064, 2015.

[9] Kia, S. (2017). Distributed optimal in-network resource allocation algorithm design via a control theoretic approach. Vol. 107, pp. 49-57.

[10] M. Rabbat, and R. Nowak, “Distributed optimization in sensor networks,” Third International Symposium on Information Processing in Sensor Networks, Berkeley, CA, USA, 2004, pp. 20-27.

[11] J. Tsitsiklis, Problems in Decentralized Decision Making and Computation PhD dissertation, Department of Electrical Engineering and Computer Sciences, Massachusetts Institute of Technology, Boston, Massachusetts, 1984.

[12] S. Farahmand, S.I. Roumeliotis and G.B. Giannakis, “Set-Membership Constrained Particle Filter: Distributed Adaptation for Sensor Networks,” IEEE Transactions on Signal Processing, vol. 59, no. 9, pp. 4122-4138, 2011.

[13] S.S. Kia, B. Van Scoy, J. Cortes, R.A. Freeman, K.M. Lynch, and S. Martinez, “Tutorial on Dynamic Average Consensus: The Problem, Its Applications, and the Algorithms,” IEEE Control System Magazine, vol. 39, no. 3, pp. 40-72, 2019.

[14] A. Tahbaz-Salehi, and A. Jadbabaie, “A One-Parameter Family of Distributed Consensus Algorithms with Boundary: From Shortest Paths to

Mean Hitting Times," 45th IEEE Conference on Decision and Control, pp. 4664-4669, 2006.

[15] M. Bénézit, P. Thiran, and M. Vetterli, "Interval consensus: From quantized gossip to voting," IEEE International Conference Acoustics, Speech and Signal Processing, 2009, pp. 3661–3664.

[16] F. Garin, L. Schenato, "A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms," In: Bemporad, A., Heemels, M., Johansson, M. (eds) Networked Control Systems. Lecture Notes in Control and Information Sciences, vol. 406, Springer, 2010.

[17] W.M. van der Wiel, A. Bloemen, H. Janssen, "Concepts for coalition engagement coordination," Maritime Theatre Missile Defense (MTMD) FTEWA Workshop, Halifax, Canada, 2017.

[18] T. Denœux, "Distributed combination of belief functions," Information Fusion, vol. 65, pp. 179-191, 2021.

[19] R. Olfati-Saber, E. Franco, E. Frazzoli, and J.S. Shamma, "Belief Consensus and Distributed Hypothesis Testing in Sensor Networks," In: Antsaklis, P.J., Tabuada, P. (eds) Networked Embedded Sensing and Control. Lecture Notes in Control and Information Science, vol 331, Springer, pp 169–182, 2006.

[20] K. Zhang, Z. Yang, H. Liu, T. Zhang, T. and Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," 35th International Conference on Machine Learning, vol. 80 of Proceedings of Machine Learning Research, pp. 5872–5881, 2018.

[21] F. Bénézit, P. Thiran, and M. Vetterli, "The distributed multiple voting problem," IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 4, pp. 791–804, 2011.

[22] L. Hammond, "An evidential network approach to threat evaluation in above water warfare," DST Group-TR-3349, 2017.

[23] C. Reynolds, "Flocks, birds, and schools: A distributed behavioral model," SIGGRAPH '87: 14th Annual Conference on Computer Graphics and Interactive Techniques, Vol. 21, pp. 25–34, 1987.

[24] T. Vicsek, A. Czirok, E. Ben Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," Physical Review Letters, Vol. 75, pp. 1226–1229, 1995.

[25] Y. Hatano, and M. Mesbahi, "Agreement over random networks," IEEE Transactions on Automatic Control, vol. 50, no. 11, pp. 1867-1872, 2005.

[26] B. van Scoy, R.A. Freeman, and K.M. Lynch, "Asymptotic mean ergodicity of average consensus estimators," American Control Conference, pp. 4696–4701, 2014.

[27] H. Moradian, and S.S. Kia, "On Robustness Analysis of a Dynamic Average Consensus Algorithm to Communication Delay," IEEE Transactions on Control of Network Systems, vol. 6, no. 2, pp. 633-641, 2019.

[28] A. Tahbaz-Salehi, and A. Jadbabaie, "A One-Parameter Family of Distributed Consensus Algorithms with Boundary: From Shortest Paths to Mean Hitting Times," 45th IEEE Conference on Decision and Control, pp. 4664-4669, 2006.

[29] M. Ruan, H. Gao, and Y. Wang, "Secure and privacy-preserving consensus," IEEE Transactions on Automatic Control, vol. 64, no. 10, pp. 4035-4049, 2019.

[30] X. Duan, J. He, P. Cheng, Y. Mo, and J. Chen, "Privacy Preserving Maximum Consensus," 54th IEEE Conference on Decision and Control, Osaka, Japan, 2015, pp. 4517-4522.

[31] M. Figura, K.C. Kosaraju, and V. Gupta, "Adversarial attacks in consensus-based multi-agent reinforcement learning," 2021 American Control Conference, New Orleans, LA, USA, pp. 3050-3055, 2021.

[32] R. Olfati-Saber, and R.M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1520–1533, 2004.

[33] H.J. Kushner and G.G. Yin, Stochastic Approximation Algorithms and Applications, Springer, 1997.

[34] Z. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 12, pp. 3893–3902.

[35] K.A. Almahorg and R.H. Gohary, "Maximum Likelihood Detection in Single-Input Double-Output Non-Gaussian Barrage-Jammed Systems," IEEE Transactions on Signal Processing, vol. 71, pp. 979-994, 2023.

[36] M. Siami and N. Motee, "Performance analysis of linear consensus networks with structured stochastic disturbance inputs," 2015 American Control Conference, Chicago, IL, USA, 2015, pp. 4080-4085.

[37] B.M. Nejad, S. A. Attia and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies," 2009 XXII International Symposium on Information, Communication and Automation Technologies, Sarajevo, Bosnia and Herzegovina, 2009, pp. 1-7.

[38] A. Tahbaz-Salehi, and A. Jadbabaie, "A One-Parameter Family of Distributed Consensus Algorithms with Boundary: From Shortest Paths to Mean Hitting Times," 45th IEEE Conference on Decision and Control, pp. 4664-4669, 2006.

[39] N.M.M. de Abreu, "Old and new results on algebraic connectivity of graphs," Linear Algebra and its Applications, Vol. 423, No. 1, pp. 53-73, 2007.

[40] J.D. Brown, M. Salmanian, T.J. Willink, "Analysis and Performance of Topology Inference in Mobile Ad Hoc Networks," In: Foschini, L., El Kamili, M. (eds) Ad Hoc Networks 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecomm. Engineering, vol 345. Springer, 2021.

[41] NIAG SG 257 (2021). NATO's Targeting and Interoperability challenges for Networked Enabled Weapons (NEW) in the 2035 Threat Environment. NATO Industrial Advisory Group (NIAG), Study Group 257.

[42] S. Paradis, A. Benaskeur, M. Oxenham and P. Cutler, "Threat evaluation and weapons allocation in network-centric warfare," 7th International Conf. on Information Fusion, Philadelphia, PA, 2005, pp. 1078-1095.

[43] F. Alawad, and F.A. Kraemer, (2022), "Value of Information in Wireless Sensor Network Applications and the IoT: A Review," IEEE Sensors Journal, vol. 22, no. 10, pp. 9228-9245.

[44] A. Angelopoulos, S. Bates, J. Malik, and M.I. Jordan, Uncertainty sets for image classifiers using conformal prediction, International Conference on Learning Representations, 2021.

[45] G. Choudhary, V. Sharma, I. You, K. Yim, I.-R. Chen, and J.-H. Cho, "Intrusion Detection Systems for Networked Unmanned Aerial Vehicles: A Survey," 14th International Wireless Communications & Mobile Computing Conf., Limassol, Cyprus, pp. 560-565, 2018.

[46] L.M. da Silva, I.G. Ferrão, C. Dezan, D. Espes, and K.R.L.J.C. Branco, "Anomaly-Based Intrusion Detection System for In-Flight and Network Security in UAV Swarm," 2023 International Conference on Unmanned Aircraft Systems, Warsaw, Poland, pp. 812-819, 2023.

[47] M. Pirani, A. Mitra, S. Sundaram, "Graph-theoretic approaches for analyzing the resilience of distributed control systems: A tutorial and survey," Automatica, vol. 157, Article 111264, 2023.

[48] P.B. Sujit and J.B. Sousa, "Multi-UAV task allocation with communication faults," American Control Conference, Montreal, QC, Canada, 2012, pp. 3724-3729.

[49] P. Castillo, R. Lozano, and A.E. Dzul, Modelling and Control of Mini-Flying Machines, Springer, 2005.

[50] N. Léchevin, C.A. Rabbath, and P. Sicard, "Trajectory Tracking of Leader-Follower Formations Characterized by Constant Line-of-Sight Angles," Automatica, Vol. 42, No. 12, pp. 2131-2141, 2006.

[51] N. Falliere, L.O. Murchu, and E. Chien, *W32.Stuxnet Dossier*, Version 1.4, Symantec Corporation, 2011.

[52] N. Léchevin, C. A. Rabbath and P. Maupin, "Toward a stability monitoring system of an asset-communications network exposed to malicious attacks," American Control Conference, San Francisco, CA, USA, 2011, pp. 2210-2215, 2011.

[53] E. Goles, P. Medina, P. Montealegre, and J. Santivañez, "Majority networks and consensus dynamics," Chaos, Solitons & Fractals, vol. 164, 112697, 2022.

[54] V.X. Nguyen, G. Xiao, X.J. Xu, Q. Wu, and C.-Y. Xia, "Dynamics of opinion formation under majority rules on complex social networks," *Nature*, Sci. Rep., no.10, 456, 2020.

ANNEX

Decentralized majority aggregation mechanisms, whereby a majority is determined locally (node level), may not always propagate through the network in such a way that the state of every node is equal, after some iterations, to the network-level majority consensus value [53],[54]. Majority dynamics are often modeled to study social opinion evolution. When considering binary systems, a typical gossip-type opinion formation majority rule found in the literature consists, at each iteration, in (*i*) selecting a node randomly and in (*ii*) applying the majority rule, whereby the selected node adopts the opinion that occupies the majority of its adjacent nodes [54]. It is shown that the convergence of such binary-state majority dynamics depends on the network structural properties. More precisely, local majority rules in dense networks, where density is measured by the average node degree, result in consensus propagation dynamics that tend to converge to the global majority consensus, while sparse network do not. This property is empirically verified for scale-free networks in [54]. Bénézit et al. derives in [21] an automaton allowing extension of gossip pairwise algorithm to n classes under some specific conditions. Gossip algorithms are not considered below.

The approach proposed here consists in applying the average-consensus algorithm to the network with initial condition $X_1 \in \{c_1, c_2\}^n$. To simplify the presentation, only two classes, c_1 and c_2, are first considered. The majority aggregation mechanism is thus based on equation (1), assuming that $\gamma K_{i,k} \equiv \gamma$, that is, $K_{i,k} = 1$. Recall that this algorithm converges asymptotically to

$1/n \sum_{i=1}^{n} X_{i,1} = X_\infty$ (i.e., steady-state value). Therefore, intuitively, the sign of X_∞ specifies the class in majority; that is, if $X_\infty > 0$, class c_2 is in majority; if $X_\infty < 0$, class c_1 is in majority; if $X_\infty = 0$, $\#c_1 = \#c_2$. When the number of classes and nodes are such that X_∞ is close to zero, discriminating between $\#c_1 = \#c_2$ and a majority might not be possible, however.

The average-consensus-based majority-rule aggregation algorithm can be extended to $n > 2$ classes by providing a version that exploits sums of complex numbers, as explained below.

Let $z_\alpha = e^{j\alpha\frac{2\pi}{n}}$ be the complex number associated with class c_i, $i = 0, \ldots, n-1$. Starting with n classes c_i, as with $\{c_1, c_2\}$ in the 2-class situation, one is interested in determining the class in majority from the sum $\sum_{\alpha=1}^{n-1} z_\alpha$. This sum is the consensus value, up to a normalization factor, of the following consensus dynamics, which, for each node S_i, consists of

- A complex state $z_{i,k}$ or, equivalently, two real states $x_{i,k} = Re(z_{i,k})$ and $y_{i,k} = Im(z_{i,k})$, and
- Two output variables, namely, the argument of $z_{i,k}$, $\varphi_{i,k} = \arg arg\, z_{i,k} = (x_{i,k}, y_{i,k})$, and the modulus of $z_{i,k}$, $r_{i,k} = |z_{i,k}|$.

The average-consensus algorithm for node S_i is given by:

$$\begin{aligned} z_{i,k+1} &= z_{i,k} + \gamma \sum_{j\in N_i} (z_{j,k} - z_{i,k}), \\ \varphi_{i,k} &= \arg arg\, z_{i,k}, \\ r_{i,k} &= |z_{i,k}|. \end{aligned} \quad \text{(A.1)}$$

As k tends to infinity, $\varphi_{i,k}$ and $r_{i,k}$ tend to the consensus values φ_∞ and r_∞, respectively, which allows inferring the class or subset of classes in majority. This inference is now explained. For example, modulus $r_{i,k}$ is easily interpreted when it converges to zero. This limit corresponds to the case where $\#c_0 = \#c_1 = \ldots = \#c_i$, for all $i = 0, \ldots, n-1$. It is therefore important to always monitor $r_{i,k}$ to detect whether or not this particular case occurs.

Let $\delta_{\alpha,\alpha+1}$, $\alpha \in [0, n-1]$, denote the angle $(2\alpha + 1)\frac{2\pi}{n}$. Parameter α, through condition $\varphi_\infty \in (\delta_{\alpha-1,\alpha}, \delta_{\alpha,\alpha+1})$ or $\varphi_\infty = \delta_{\alpha,\alpha+1}$, is instrumental in determining the class or subset of classes in majority, as illustrated in Figure 18 with three classes.

To simplify the presentation, one considers the case where the number of classes is equal to three, $n = 3$, which gives rise to the parameters defined above, namely, z_0, z_1, z_2, $\delta_{01} = \pi/3$, $\delta_{12} = -1$, and $\delta_{02} = -\pi/3$. The following seven cases define the conditions associated with the classes in majority, if any, allowing to interpret rule (A.1) after some iterations.

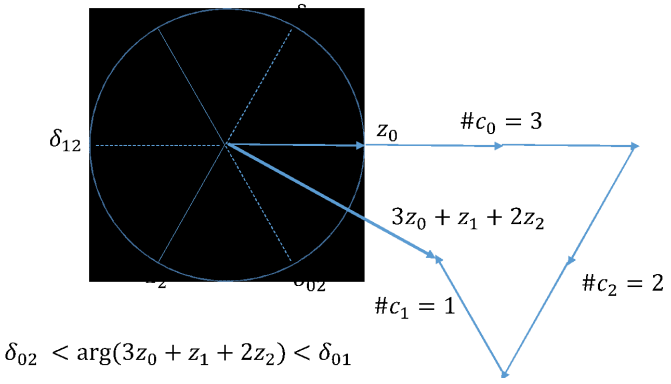


Figure 18: Example of the determination of $maj(c_0, c_0, c_0, c_1, c_1, c_2)$.

Case 1: c_0 is in majority if

$$\varphi_\infty \in (\delta_{02}, \delta_{01}). \quad \text{(A.2)}$$

Case 2: c_1 is in majority if

$$\varphi_\infty \in (\delta_{01}, \delta_{12}). \quad \text{(A.3)}$$

Case 3: c_2 is in majority if

$$\varphi_\infty \in (\delta_{12}, \delta_{02}). \quad \text{(A.4)}$$

Case 4: c_0 and c_1 are in majority ($\#c_0 = \#c_1 > \#c_2$) if

$$\varphi_\infty = \delta_{01}. \quad \text{(A.5)}$$

Case 5: c_0 and c_2 are in majority ($\#c_0 = \#c_2 > \#c_3$) if

$$\varphi_\infty = \delta_{02}. \quad \text{(A.6)}$$

Case 6: c_1 and c_2 are in majority ($\#c_1 = \#c_2 > \#c_0$) if

$$\varphi_\infty = \delta_{12}. \quad \text{(A.7)}$$

Case 7: No class is in majority ($\#c_0 = \#c_1 = \#c_2$) if

$$r_\infty = 0. \quad \text{(A.8)}$$

Case 4 comes from the following equality

$$z_0 + \ldots + z_{0\smile_n} + \quad \text{(A.9)}$$

$$z_1 + \ldots + z_{1\curvearrowright_n} + z_2 + \ldots + z_{2\curvearrowright_{m<n}} = (n-m)$$

and from $\arg arg\,(z_0 + z_1) = \delta_{01}$. The same applies to *Case 5* and *Case 6*.

A.1 Numerical example and consensus awareness

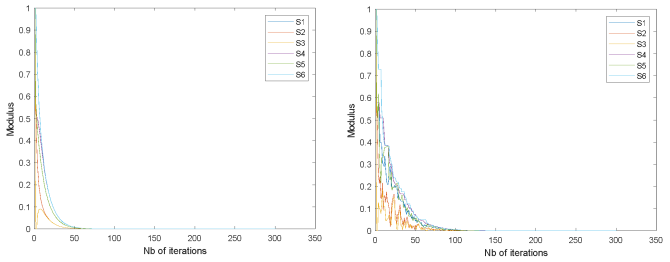
Figure 19(a) illustrates the situation where $\#c_0 = \#c_1 = \#c_2$ that is, $z_0 + z_0 + z_1 + z_1 + z_2 + z_2 = 0$, obtained with a six-node chain graph, whose edge set is given by $\{(1,2),(2,3),(3,4),(4,5),(5,6)\}$. This equality is equivalent to the consensus value of the modulus being equal to zero (angle trajectories are not shown). Figure 19(a) shows the decentralized aggregation dynamics ($\gamma = 0.5/6$) obtained when, at each iteration, each edge may fail to communicate data with a probability equal to 0.5. Figure 20 illustrates the situation where $\#c_2 > \#c_0 > \#c_1$, which corresponds to the consensus value of the argument being equal to $-\pi/2 \in (\delta_{12}, \delta_{02})$.

One approach to locally detect the proximity of aggregated states to the consensus value consists in

- Computing the relative error $\tilde{\varphi}_{f,i,k} = |\varphi_{f,i,k} - \varphi_{f,i-1,k}|/\varphi_{f,i,k}$ of the filtered angle $\varphi_{f,i,k}$ (e.g. moving average), and in
- Executing a detection rule $d_{f,i,k}$ that switches from 0 to 1 (vicinity of the consensus value) when $\tilde{\varphi}_{f,i,k} < \varepsilon \wedge \tilde{\varphi}_{f,i-1,k} < \varepsilon \wedge \ldots \wedge \tilde{\varphi}_{f,i-I,k} < \varepsilon$.

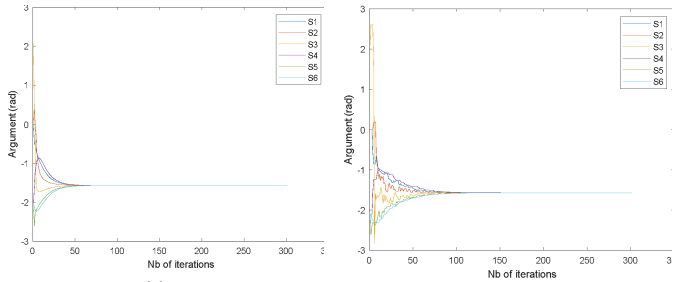
Parameters $\varepsilon > 0$ and $I \in N$ stand for the decision threshold and the length of the sliding window.

Figure 21(a)-(b) show the decision outputs of detection rule $d_{f,i,k}$, $i = 1, \ldots, 6$, computed at each iteration k by nodes 1 to 6 with or without edge failures ($\varepsilon = 0.1\%$, $I = 9$). The low-pass filter time constant is equal to one. Note that each $d_{f,i,k}$ is offset along the y-axis by some nonnegative value so that all detection signals can be clearly identified. As expected, degradation in detection (time response and variability) increases with p. The detector thus provides only a local indication of the proximity of the aggregated information to the consensus value.



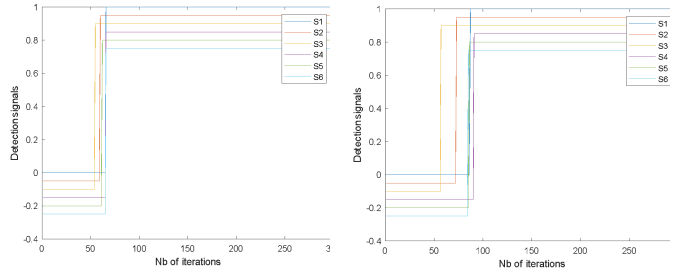
(a) (b)

Figure 19: Modulus trajectories for majority aggregation with $(c_0, c_0, c_1, c_1, c_2, c_2)$. *(a) No edge loss. (b) Edge loss (*$p = 0.5$*).*



(a) (b)

Figure 20: Angle trajectories for majority aggregation with $(c_0, c_0, c_1, c_2, c_2, c_2)$. *(a) No edge loss. (b) Edge loss (*$p = 0.5$*).*



(a) (b)

Figure 21: Detection signals $d_{f,i,k}$ *obtained with* $(c_0, c_0, c_1, c_2, c_2, c_2)$. *(a) No edge loss. (b) Edge loss with* $p = 0.5$.