Incremental Learning Through Graceful Degradations in Autonomous Systems

Ganapathy Mani*, Bharat Bhargava†, Basavesh Shivakumar‡
Department of Computer Science & CERIAS
Purdue University
West Lafayette, USA
manig@purdue.edu, bbshail@purdue.edu†, bammanag@purdue.edu‡*

Jason Kobes
NGC Research Consortium
Northrop Grumman Corporation
McLean, USA
Jason.Kobes@ngc.com

***Abstract*—Intelligent Autonomous Systems (IAS) are highly cognitive, reflexive, multitasking, trustworthy (secure as well as ethical), and rich in knowledge discovery. IAS are deployed in dynamic environments and connected with numerous devices of different types, and receive large sets of diverse data. They often receive new types of raw data that was not present in either training or testing data sets thus they are unknown to the learning models. In a dynamic environment, these unknown data objects cannot be ignored as anomalies. Hence the learning models should provide incremental guarantees to IAS for learning and adapting in the presence of unknown data. The model should support progressive enhancements when the environment behaves as expected or graceful degradations when it does not. In the case of graceful degradations, there are two alternatives: (1) weaken the acceptance test of data object (operating at a lower capacity) or (2) replace primary system with a replica or an alternate system that can pass the acceptance test. In this paper, we provide a combinatorial design—*MACROF* configuration—built with balanced incomplete block design to support graceful degradations in IAS and aid them to adapt in dynamic environments. The architecture provides stable and robust degradations in unpredictable operating environments with limited number of replicas. Since the replicas receive frequent updates from primary systems, they can take over primary system's functionality immediately after an adverse event. We also propose a Bayesian learning model to dynamically change the frequency of updates. Our experimental results show that *MACROF* configuration provides an efficient replication scheme to support graceful degradations in autonomous systems.**

***Keywords*-cognitive autonomy; incremental learning; graceful degradations; combinatorial design; Bayesian learning;**

I. Introduction

Smart and autonomous systems are taskable, rich with knowledge, reflective, and ethical [1]. IAS should be cognizant of their dynamic operating environment, interactions with the different types of devices in that environment, history of these interactions, and knowledge derived from that history of interactions. Based on this database of rich knowledge discovery, IAS should be able to understand and predict scenarios and reflect to adapt to those new scenarios. IAS should perform these tasks for an extended period of time in unpredictable, partially observable, dynamic, and approximately modeled environment with no or very little human intervention.

One of the most important properties of IAS is their reflexivity. With live monitoring their actions, identifying current and potential problems, they should be able to optimize, reconfigure, and repair autonomously as a *reflexive* response. The system should understand, learn, and adapt to advance their reflexive response overtime. This includes acquiring new knowledge and transforming their current operational behaviors by augmenting their knowledge discovery on how to operate on uncertain and dynamic environments to perform specific tasks. These advances can be the culmination of reflexive actions based on history of transactions in and with the operating environment, observation of other entities (including humans) with the system and operating environment, or manual instructions set by human entities. As a result, IAS should continually make changes to their operating behavior and adapt to the new contexts in which they are operating, even if the particulars of that operating context are not specified in the initial model, and the new data points were not present in the training and testing sets. The system needs to incrementally learn and adapt to process and integrate these unknown items into the system or block them as anomalies in case of hostile objects.

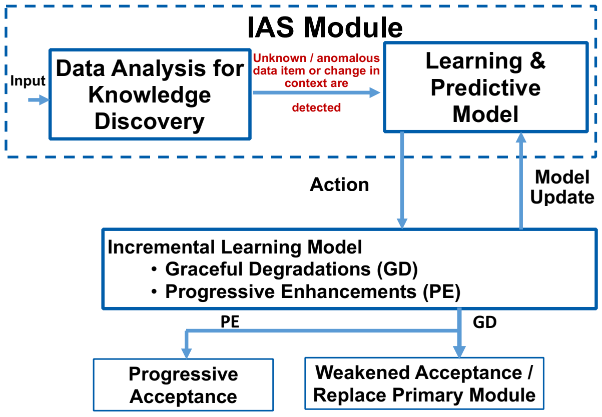
IAS Module
Input
Data Analysis for Knowledge Discovery
Unknown / anomalous data item or change in context are detected
Learning & Predictive Model
Action
Model Update
Incremental Learning Model
• Graceful Degradations (GD)
• Progressive Enhancements (PE)
PE
GD
Progressive Acceptance
Weakened Acceptance / Replace Primary Module

Figure 1. Reflexivity workflow with incremental learning in Intelligent Autonomous Systems (IAS)

There are several definitions of incremental learning in literature. Incremental learning is referred as a methodology that allows systems to gracefully integrate new type of information (objects or classes) without incurring the loss of accuracy in the model and without having to retrain the model from the scratch [2]. It is also referred to as gradual extension of current classes with new types of classes to recognize and classify new objects i.e. incremental multi-class learning [3]. Incremental learning allows an autonomous system to retrain itself based on the new information received from neighboring entities, its new interactions with those entities, changes occurred in the operating context, and changes in its own operating behavior due to the influence of hostile entities such as attacks.

One of the most important constraints of reflexivity properties of IAS is that they need to update their learning model and adapt to the new scenario without disrupting the underlying critical processes. Based on this criteria, reflexive workflow (Figure 1) can be designed with incremental learning that is accomplished by graceful degradations and progressive enhancements. Graceful degradation is nothing but a design principle that allows the system to continue its operations in lower capacity i.e. gradually degrade to accommodate dynamic changes that occur in the system or in operational contexts [4] [5]. Progressive enhancement is also a design principle where the system gradually accepts and enhances the role of new scenarios and data objects without shutting down the system [6] [7].

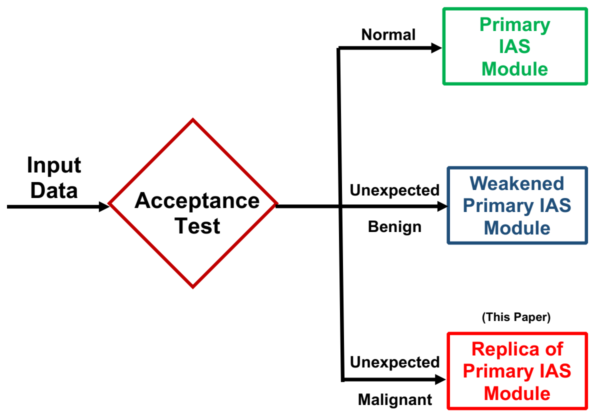

Figure 2. Graceful Degradations in IAS

Graceful degradation of IAS can be designed using replicas and acceptance tests (Figure 2). Each primary module of IAS will have alternates or replicas. In case of an unexpected output or input that was not present in the training or testing datasets and it was not factored into the learning model, the system lowers the acceptance criteria for the input/output and continue to function. In case of catastrophic failure of a primary module, the replica of the primary module takes over the functionalities and continue to process the data and operate.

In this paper, we propose a robust and fault-tolerant replication scheme—*MACROF*—using combinatorial balanced incomplete blocks. This design can aid the cognitive autonomy of IAS by autonomous updates to the replicas and replacement of primary IAS module in case of significant disruptions for the critical processes. We have identified the following research contributions from *MACROF* model,

1) *MACROF* model provides an autonomous system with the capability to gracefully degrade under uncertain circumstances by limiting the primary module's functionality instead of shutting the system down, while the replicas continue to function and move the process forward.
2) Replicas can be used for other processes while primary module is busy and replicas can update the primary module when it becomes available again. This enables the processes to progress in parallel—an intrinsic benefit of this model.
3) With frequent updates from the primary module, the scheme offers reliability with autonomous replacement of primary module and take over its functionalities while preserving the primary module's state i.e. recovery point. Thus the processes can continue to progress from the recovery point without restarting from the beginning. This can help autonomous systems to be a stateful [8].
4) Through Bayesian learning [9], the frequency of updates are changed dynamically depends on the importance of data objects for each processes.
5) Replicas in this model aid progressive enhancement of unknown data objects. Replicated modules can be used for retraining and testing the model with new datasets before introducing the unknown data objects to the primary module.
6) The MACROF model does not depend on a centralized server module thus the systems can operate coherently on their own by functioning based on their local constraints. This creates a distributed computing environment where autonomous learning can be done at individual system-level, and the governing predictive model can be updated.
7) Replicas also provide Moving Target Defense (MTD)-style [10] reflexive mechanism. If each replica has different configuration, the primary module's tasks can be shifted to replicas periodically to limit the attackers of observational space for planning attacks and thwart those attacks specifically designed for a particular observed configuration.
8) The cost of replicas can be considerably minimized in autonomous systems with large number of modules

based on the failure rate. Thus the scheme is scalable and several extensions are possible.

The rest of the paper is organized as Section II elaborates on the related work of incremental learning and combinatorial balanced incomplete blocks. Section III introduces the *MACROF* model and covers the parameter analysis of the scheme with proofs. Section IV provides experimental results and replication cost estimates. Section V discusses extensions of the design and future work. Finally, we discuss our contributions and conclude in Section VI.

II. Related Work

Our work addresses reflexivity in autonomous systems that has received attention from and is related to several topics from incremental learning to combinatorial mathematics based system designs.

A. Incremental Learning

Incremental learning involves graceful degradations and progressive enhancements of autonomous systems. When the acceptance test of an input or output of a primary module fails and the system can still function, then the system gradually reduces its functionality and operates at a lower capacity to accept the unknown data input or output result. When the system is incapable of operating with unknown data input, then it shifts its functionality to one of its replica while the primary module gets retrained and tested with new input. In progressive enhancements, the participation of unknown data objects is increased gradually while the system is retrained to accept them completely. In both of these cases, the system continues to carryout its critical functionalities without being disrupted or shutdown.

A conceptual framework for adaptive graceful degradations in autonomous vehicles is provided in [11] [12]. The authors introduce a framework that when failure occurs in a critical functionality of the vehicle such as a vision program that detects pedestrians, depends on the situation, the system adapts to either notify the driver or operate at a lower capacity depends on the environment. If an autonomous vehicle is operating in interstate roads or national highways, it can simply notify the diver about the failure and continue to operate at the same level. But if it is operating in urban areas, it will notify the driver and the vision program will be operating at a degraded level (may be with lower frame rate) on another live processing mode and at the same time slows down the vehicle. Similar studies of graceful degradations have been conducted in autonomous vehicles [13] and in Unmanned Arial Vehicles (UAVs) [14]. Similar to our research, graceful degradations have been studied for fault-tolerant robotic systems [15]. In this particular study, a failure in robotic arm is analyzed to determine whether its a local failure or a system fault. Based on the result, gracefully degrade the operation of the robotic arm. Graceful degradations are widely used in approximate computing models as well. In [16], the process of discovering hardware blocks in computing systems that display graceful degradations under voltage overs-scaling is studied. The study identifies the causes of degradations and analyzes the effects of degradations in computing systems with multimedia algorithms. The authors in the study [17] propose a graceful degradation model in Cooperative Adaptive Cruise Control (CACC) while operating under unreliable wireless communication system. When the system receives messages with packet loss, the system gets degraded to conventional Adaptive Cruise Control (ACC) to stabilize the system and avoid futher failures.

Progressive enhancements are studied as a design problem in adaptive systems. In [18], the authors introduce a framework that combines behavior models with diverse associated assumptions and risks. In this framework, when the environmental assumptions and parameter thresholds are violated then the system goes into degraded mode but after a period of degraded functionality the system starts to progressively recover to support higher level of functionality. Adaptive web designs are proposed in [19]. The guidelines for designing websites are set such that the website should progressively enhance their functionality when the browser compatibility changes. Similar enhancements for web designs have been proposed in [20] [21].

Incremental learning is an active field of interest in artificial intelligence community. Incremental learning plays a vital role in mimicking cognition of living organisms in machines. In particular, Support Vector Machines (SVMs) are used in numerous object recognition problems [22] [23]. Thus many extensions of SVMs have been proposed for incremental learning. Incremental binary SVMs were proposed in [24]. The model saves and updates Karush–Kuhn–Tucker (KKT) conditions. This approach was extended in [25] and multi-class incremental learning in object recognition was introduced. Memory-controlled online visual recognition through incremental SVMs were introduced in [26]. Since updates are very expensive in large-scale systems, the authors proposed classifiers with fast and inexpensive updates with multi-class extensions in [27] [28].

Combinatorial designs are used in several optimization and balancing problems [29] with numerous applications in computing systems. In [30], the authors use the design to extract parallelism in mobile CPUs/GPUs since the design provides a robust communication mechanism that lets the processing elements and their duplicates to interact simultaneously. A similar structure is used in [31] where model provides back up data when the primary data module is down (same replica mechanism is used in our study). The model prevents data loss by combinatorially replicating the data in computing devices. The model proposed in [32] uses combinatorial block designs to support object-oriented connects in distributed fault-tolerant systems.

In this work, we implement a similar combinatorial balanced structure to provide graceful degradations in autonomous systems. Using combinatorial optimization can provide balanced workload and reliable backups through replicas. *MACROF*'s communication links can provide dynamic changes to the frequency of updates through Bayesian inference and the updated replicas can be used for other functionalities when primary module is in good condition.

III. MACROF Autonomous System Design

A combinatorial Structure is a subset satisfying certain conditions. Each block (distributed module or a cluster) contains systems and their replicas that are distributed based on balanced incomplete block designs (BIBD) [33] in combinatorial mathematics. The systems and their replicas in the distributed blocks are strategically connected to receive updates from primary modules. Replicas can be used to aid other functionalities in the system when primary module is in working condition.

The combinatorial *MACROF* model is defined as follows: a distributed environment with a set of A systems is a collection of M R-subsets such that each system appears exactly in C subsets and each pair of systems appears precisely in O subsets. In addition, F is the frequency of updates dynamically set by Bayesian learning. The standard notation of the model can be represented as (*M, A, C, R, O, F*)-configuration or *MACROF* configuration (Figure 3).

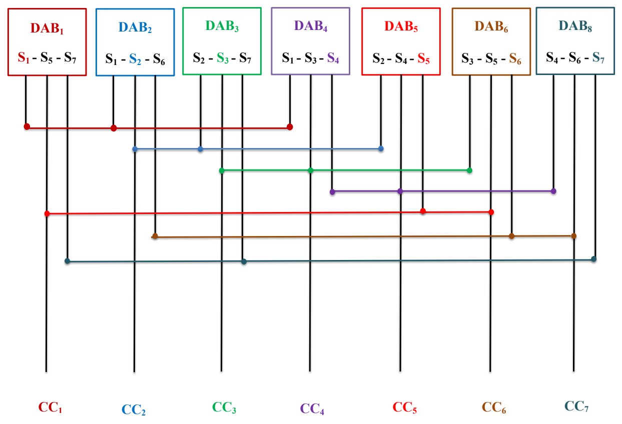

Figure 3. *MACROF* Autonomous System Design with Distributed Autonomous Blocks (DAB), Communication Channels (CC), and Systems (S)

As an example of this setting, consider an autonomous system, AS, with a set of seven primary modules (Figure 3). $AS = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ represents the design of ($M = 7$, $A = 7$, $C = 3$, $R = 3$, $O = 1$, F)-configuration. In this configuration, there are seven DABs (subsets) each with three systems: $DAB_1 = \{S_1, S_5, S_7\}$, $DAB_2 = \{S_1, S_2, S_6\}$, $DAB_3 = \{S_2, S_3, S_7\}$, $DAB_4 = \{S_1, S_3, S_4\}$, $DAB_5 = \{S_2, S_4, S_5\}$, $DAB_6 = \{S_3, S_5, S_6\}$, and $DAB_7 = \{S_4, S_6, S_7\}$.

In this setting, a set of systems are connected through communication channels to their replicas the corresponding subset in each distributed processing block. Primary module can be assigned methodically or with Round-robin (RR) scheduling algorithm. The interactions can happen locally in each DAB. Given any random pair of systems S_i and S_j, there is always a DAB_i that contains them. For example, in an IAS, if the learning model in system S_6 needs to interact with S_4 and S_7, then both of those systems can be found locally in DAB_7. This enables the system to interact fast and simultaneously the replicas of all three systems are updated. The replicas stay updated to take over the primary module at anytime. Even when some functionalities fail in primary module, the previous state of the progress is maintained at the replicas and they can update the primary module with current state instead of it having to restart from the beginning. These local interactions and simultaneous updates enable IAS to have an effective and fault-tolerant replacement model for The advantage of this combinatorial setting will be explained in parameter evaluation.

Table I
Possible combinatorial designs

M	A	C	R	O
12	9	4	3	1
13	13	4	4	1
21	21	5	5	1
26	13	6	3	1
31	31	6	6	1
35	15	9	3	1
55	55	8	8	1
73	73	9	9	1
91	91	10	10	1
132	132	12	12	1

Other than ($M = 7$, $A = 7$, $C = 3$, $R = 3$, $O = 1$, F)-configuration there are other combinatorial pairwise balanced designs are possible (Table I). An incident matrix can capture the relationship between DABs and systems in them.

$$\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \end{array}
\begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\
\begin{pmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{pmatrix} \end{array}$$

The incident matrix specifies the combinatorial design ($M = 12$, $A = 9$, $C = 4$, $R = 3$, $O = 1$, F)-configuration with 12 DABs.

The frequency F is determined based on Bayesian inference by combining prior knowledge with observed data.

In *MACROF* model the structure is known. The frequency factor can determined based on user preference set at training or dynamically based on the type of data. Given labeled data (D_i) and context (C_j), the frequency of occurrence can be calculated for data items.

$$P(C_j|D_i) = LikelihoodRatio \cdot Prior \tag{1}$$

$$LikelihoodRatio = \frac{P(D_i)}{C_j} \tag{2}$$

$$Prior = P(C_i) \tag{3}$$

Thus the frequency of the updates will be determined based on the context C_j where if it is an important (as specified by labeled data) then the frequency of updates will be changed to the predicted probability of that specific context occurring again.

$$PosteriorP(C_j|D_i) = \frac{P(D_i)}{C_j} \cdot P(C_j) \tag{4}$$

The posterior probability gets updated for every new context with interval of update. For example, if C_j has over 50% chance of occuring again in 5 minutes then the time epoch for update will be 5 minutes until the posterior probability is changed for that particular context. Thus F can be defined as a time interval,

$$F = t_{P(C_{j+1}|D_{i+1})} - t_{P(C_j|D_i)} \tag{5}$$

This Bayesian inference can be extended to unknown data items as well with a modification of Expectation-Maximization (EM) algorithm [34].

A. Evaluation of MACROF Parameters

The individual parameters in *MACROF* configuration can be defined as follows: M is number of distributed autonomous blocks with autonomous systems, A is number of systems, R is number of subset split of the systems, C is the number of times S_i is replicated, O number of times that a random pair systems S_i and S_j appear together in the system. The parameters that can be adjusted in this design are M and C.

$$C = \frac{O \cdot (A-1)}{R-1} \tag{6}$$

Theorem 1: In combinatorial design of *MACROF*, every system has C number of DABs containing a particular system S_i.

*Proof:*Let (z, γ) be a *MACRO* configuration. Suppose $z \in Z$ and C_z denote the number of DABs containing S_i. Define a set,

$$K = \{(x, Y) : x \in Z, x \neq z, Y \in \gamma, \{z, x\} \subseteq Y\} \tag{7}$$

There are $A-1$ ways to choose $x \in Z$ s.t. $x \neq z$ for each such x, s.t. there are O blocks Y s.t. $\{z, x\} \subseteq Y$. Hence,

$$|K| = O(A-1) \tag{8}$$

Then, there are C_z ways to choose a DAB Y s.t. $z \in Y$. For each choice of Y, there are $R-1$ ways to choose $x \in Y$, $x \neq z$. Hence,

$$|K| = C_z \cdot (R-1) \tag{9}$$

Combining equations (8) and (9), we get,

$$O(A-1) = C_z \cdot (R-1) \tag{10}$$

From equation (10), we can get equation (6). Thus C_z is independent of z resulting in equation (6). C is also called the *Replica Number*.
Similarly, M can be derived with,

$$M = \frac{A \cdot C}{R} = \frac{O \cdot (A^2 - A)}{R^2 - R} \tag{11}$$

Theorem 2: A combinatorial *MACROF* design has exactly M DBAs.

Proof: Let (z, γ) be a combinatorial *MACROF* configuration and $|M|$ = $|\gamma|$. Define a set,

$$K = \{(z, Y) : z \in Z, Y \in \gamma, z \in Y\} \tag{12}$$

There are A ways to choose $z \in Z$. For each such z, there are C blocks Y s.t. $z \in Y$. Hence,

$$|K| = A \cdot C \tag{13}$$

Then, there are M ways to choose a block $Y \in Y$. For each choice of Y there are R ways to choose $z \in Y$. Hence,

$$|K| = M \cdot R \tag{14}$$

Combining equation (13) and (14), we get,

$$M \cdot R = A \cdot C \tag{15}$$

It is nothing but the desired result. The parameters (except F) are not independent in *MACROF* model. They have to satisfy the following equations,

$$M \cdot R = A \cdot C \tag{16}$$

$$C \cdot (R-1) = O \cdot (A-1) \tag{17}$$

$$M \geq A \tag{18}$$

Corollary 1: If there exists a *MACROF* configuration then $O(A-1) \equiv 0$ $(mod\ R-1)$ and $O \cdot A(A-1) \equiv 0(mod\ R \cdot (R-1))$.

Proof: This is an obvious corollary. For example, a design with A = 8, R = 3, and O = 1 does not exist because $O \cdot (A-1)$ = 7 $\not\equiv 0(mod\ 2)$.

Practical implementation of balanced block designs in combinatorial mathematics do not require perfectly balanced blocks. As it can be seen from equations (6) and (11), the design constraints are directly propositional to O. For O

> 1, complex and fault-tolerant operations are possible but they may come at the cost of complexity and redundancy. The number of connections R subsets is a fixed parameter used for the construction of autonomous blocks. To reduce the complexity, R should be kept relatively small. Consider an example design with O = 1 and R = 3. Then,

$$M \approx \frac{1}{6} \cdot A^2 \tag{19}$$

$$C \approx \frac{1}{2} \cdot A \tag{20}$$

Thus a design with 20 autonomous entities would require around 60 DBAs. Here number of communication channels attached to DBAs per each system should be around 10.

IV. Experimental Results

We conducted experiments with a simulator proposed in [35] that is built with (7, 7, 3, 3, 1)-configuration with a few modifications embedded in the code. The modified simulator can be found in [36]. We set F to clock tick of the underlying operating system. The simulator is built for extracting parallelism with sequential processing with replicated data storage elements such as *MACROF* configuration. It is equipped with storage elements (such as registers) to hold data objects. A sequential processing module is also embedded into the simulator to run similar processes that will be run by *MACROF* configuration. These storage elements act as independent autonomous modules. We measure the performance of the graceful degradation structure through number of updates it requires to complete a specific process and update the replicated systems. These number of updates are compared with a sequential processing module without the combinatorial structure.

A compiler loads specified instructions for each program into both sequential processing and combinatorial processing structure. When a particular process is being run in the system, *MACROF* design enables the local elements to interact simultaneously without waiting for data from primary modules. This (a) avoids data dependence and (b) increases fault-tolerance—if data is corrupted in one storage element, its replica can replace it. Each storage element is updated with new values when the process progresses. These processes are written in assembly instructions and loaded into the simulator: sum (P1), binary search (P2), copying data (P3), print (P4), double copy (P5), moving data (P6), Fibonacci creation (P7), Fibonacci search (P8), and scalar product (P9).

To complete a particular process, depends on the complexity of the processes, the distributed combinatorial *MACROF* model performs well with considerable speed, incurring less number of updates than the sequential processing with replications (Figure 4). One of the main overhead in *MACROF* design is the replication cost when the system increases in scalability. We hypothesize that when the design is increased

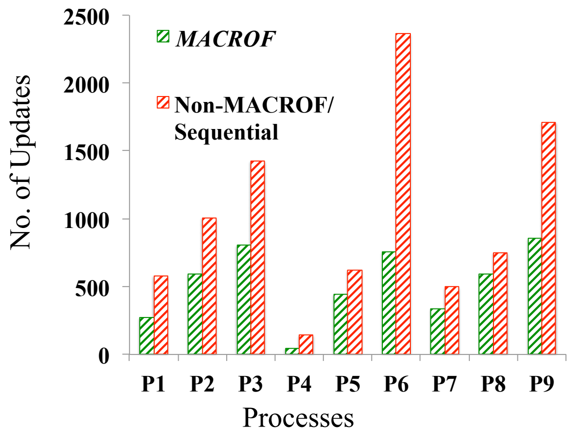

Figure 4. Updates for *MACROF* compared to a non-*MACROF* sequential processing

in its number of DABs and systems, the updates for the replicated systems can be reduce considerably. By keeping track of the failure rate of primary modules in a given time interval, the updates to replicas can be reduced. For example, if there are 700 DBAs with 300 replicas and the failure rate is 10 per day, then only 10-20 replicas may need updating where as other replicas can be updated in batches later on. The replicas that are not in use can be used for other purposes. The increase in number of interconnections may increase the complexity during the design phase but it is relatively easier to keep track once the construction is complete. Table II shows the overhead of updates incurred by non-*MACROF* processing for each process.

Table II
Updates required for Non-*MACROF* model

ID	Process	$\times MACROF$ Updates are Required
P1	FIBSEARCH	1.3
P2	DOUBLE	1.4
P3	FIBB	1.5
P4	SEARCH	1.8
P5	COPY	1.8
P6	SCALAR	2
P7	SUM	2.1
P8	PRINT	3
P9	MOVEMENT	3.1

V. Discussion and Future Work

The presented combinatorial design aids incremental learning in autonomous systems by allowing graceful degradations by the replacement of primary modules through replicas. Depends on the number of autonomous modules the cost incurred by the design may increase with the complexity of interconnections. But *MACROF* configuration provides a robust and fault-tolerant replication scheme. The Bayesian model presented can dynamically change the frequency of updates and predict failure rate. This information can be used to reduce the number of active replicas as well as their updates. This feature provides flexibility to the users to set their batch updates in their convenient intervals.

Our future research involves the following tasks:

- Integrate deep learning models to handle unknown data (data that was not present during training or testing or initial implementation). In particular, we intend to investigate Open Set Learning methods [37].
- We plan to implement Moving Target Defense (MTD)-style [38] resiliency mechanisms with *MACROF*. Each replica can be loaded with different configuration and the processes can be moved from one replica to another to thwart any attacks based on particular configuration. Even though the replicas increase the attack surface, the combinatorial structure opens up possibilities for various configurations with replicas and robust updates to them.
- Implement *MACROF* in untrusted cloud. Clients Pull from Primary server to stay in Sync. A Virtual Machine / computing system can act as both server / client. Polling algorithms [39] will be used to select a primary server. To maintain the data synchronization between all the entities, client will poll the data from the server. There are two kinds of polling: (a) lightweight poll to check if there is any update (say every 1 second, return true or false) and (b) Heavyweight poll to get the data from the server (for every minute or whenever lightweight poll returns true - whichever occurs earlier). Lightweight poll can be done using HTTP / unsecured lightweight call and Heavyweight call should be done with HTTPS / secure protocol so that updated meta data is properly authenticated. Data can be made available from server using REST protocol. All the data will be stamped with UTC timezone time stamp, so that clients can see the latest data if a conflict arises.

VI. Conclusion

In this paper, we presented *MACROF*—a combinatorial distributed design—for intelligent autonomous systems to enable incremental learning aid cognitive autonomy. The scheme offers a robust and reliable replication scheme without incurring much overhead in terms of frequency of updates. The frequency of updates to replicas from primary module is dynamically changed based on Bayesian learning model. These replicas allow the autonomous system to gracefully degrade and operate at a lower capacity in case of benign operational contexts but effectively replace primary module in case of malignant operational contexts. The model updates the IAS's predictive model with new and unknown operational contexts where it can be retrained and tested. *MACROF* can also be used for processing data in parallel with the help of replicas. The cost of replicas is considerably reduced when the system is scaled for higher number of systems with DABs. We have presented experimental results obtained through a simulator where *MACROF* performs considerably well in incurring updates overhead.

Acknowledgment

This research work is supported by NGC Research Consortium.

References

[1] NSF, "Smart and Autonomous Systems (S&AS)," 2017, Program Solicitation NSF 16-608. [Online]. Available: https://www.nsf.gov/pubs/2016/nsf16608/nsf16608.pdf

[2] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool, "Incremental learning of ncm forests for large-scale image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3654–3661.

[3] T. Yeh and T. Darrell, "Dynamic visual category learning," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[4] M. N. Giannakos, D. G. Sampson, and Ł. Kidziński, "Introduction to smart learning analytics: foundations and developments in video-based learning," *Smart Learning Environments*, vol. 3, no. 1, p. 12, 2016.

[5] T. Layh and D. Gebre-Egziabher, "Design for graceful degradation and recovery from gnss interruptions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 9, pp. 4–17, 2017.

[6] C. A. Hall, "Web presentation layer bootstrapping for accessibility and performance," in *Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A)*, ser. W4A '09. New York, NY, USA: ACM, 2009, pp. 67–74. [Online]. Available: http://doi.acm.org/10.1145/1535654.1535671

[7] A. Gustafson, *Adaptive web design: crafting rich experiences with progressive enhancement.* New Riders, 2015.

[8] H. A. Lagar-Cavilla, J. A. Whitney, A. M. Scannell, P. Patchin, S. M. Rumble, E. De Lara, M. Brudno, and M. Satyanarayanan, "Snowflock: rapid virtual machine cloning for cloud computing," in *Proceedings of the 4th ACM European conference on Computer systems.* ACM, 2009, pp. 1–12.

[9] C.-K. Wen, S. Jin, K.-K. Wong, J.-C. Chen, and P. Ting, "Channel estimation for massive mimo using gaussian-mixture bayesian learning," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1356–1368, 2015.

[10] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving target defense: creating asymmetric uncertainty for cyber threats.* Springer Science & Business Media, 2011, vol. 54.

[11] J. Kim, R. R. Rajkumar, and M. Jochim, "Towards dependable autonomous driving vehicles: a system-level approach," *ACM SIGBED Review*, vol. 10, no. 1, pp. 29–32, 2013.

[12] C. Berger and B. Rumpe, "Autonomous driving-5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system," *arXiv preprint arXiv:1409.0413*, 2014.

[13] D. B. Wilson, A. H. Göktoğan, and S. Sukkarieh, "Experimental validation of a drogue estimation algorithm for autonomous aerial refueling," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5318–5323.

[14] T. Layh and D. Gebre-Egziabher, "Design for graceful degradation and recovery from gnss interruptions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 9, pp. 4–17, 2017.

[15] C. R. Burns, V. Fuelling, G. K. Toth, and A. Patel, "Fault reaction, fault isolation, and graceful degradation in a robotic system," Sep. 20 2016, uS Patent 9,446,517.

[16] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Test Symposium (ETS), 2013 18th IEEE European*. IEEE, 2013, pp. 1–6.

[17] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cacc performance subject to unreliable wireless communication," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 1210–1216.

[18] N. D'Ippolito, V. Braberman, J. Kramer, J. Magee, D. Sykes, and S. Uchitel, "Hope for the best, prepare for the worst: multi-tier control for adaptive systems," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 688–699.

[19] A. Gustafson, *Adaptive web design: crafting rich experiences with progressive enhancement*. New Riders, 2015.

[20] H. Desruelle and F. Gielen, "Context-driven progressive enhancement of mobile web applications: A multicriteria decision-making approach," *The Computer Journal*, vol. 58, no. 8, pp. 1732–1746, 2014.

[21] L. Bassbouss, M. Tritschler, S. Steglich, K. Tanaka, and Y. Miyazaki, "Towards a multi-screen application model for the web," in *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual*. IEEE, 2013, pp. 528–533.

[22] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: fast feature extraction and svm training," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1689–1696.

[23] C. J. Veenman and D. M. Tax, "A weighted nearest mean classifier for sparse subspaces," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 1171–1176.

[24] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in neural information processing systems*, 2001, pp. 409–415.

[25] T. Yeh and T. Darrell, "Dynamic visual category learning," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[26] A. Pronobis, L. Jie, and B. Caputo, "The more you learn, the less you store: Memory-controlled incremental svm for visual place recognition," *Image and Vision Computing*, vol. 28, no. 7, pp. 1080–1097, 2010.

[27] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[28] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, no. Mar, pp. 551–585, 2006.

[29] C. J. Colbourn and J. H. Dinitz, *Handbook of combinatorial designs*. CRC press, 2006.

[30] G. Mani, S. Berkovich, and D. Liao, "Balanced block design architecture for parallel computing in mobile cpus/gpus," in *Computing for Geospatial Research and Application (COM. Geo), 2013 Fourth International Conference on*. IEEE, 2013, pp. 140–141.

[31] G. Mani, "Clone attack detection and data loss prevention in mobile ad hoc networks," *International Journal of Space-Based and Situated Computing*, vol. 5, no. 1, pp. 9–22, 2015.

[32] S. Y. Berkovich, "Multiprocessor interconnection network using pairwise balanced combinatorial designs," *Information Processing Letters*, vol. 50, no. 4, pp. 217–221, 1994.

[33] H. Hanani, "Balanced incomplete block designs and related designs," *Discrete Mathematics*, vol. 11, no. 3, pp. 255–369, 1975.

[34] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm," *IEEE transactions on medical imaging*, vol. 20, no. 1, pp. 45–57, 2001.

[35] E. Berkovich and S. Berkovich, "A combinatorial architecture for instruction-level parallelism," *Microprocessors and Microsystems*, vol. 22, no. 1, pp. 23–31, 1998.

[36] G. Mani, "MACROF Simulator," 2018, Needs Windows x86 Operating Systems to run. [Online]. Available: https://goo.gl/pgVHdk

[37] A. Bendale and T. Boult, "Towards open world recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1893–1902.

[38] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 127–132.

[39] D. Ongaro and J. K. Ousterhout, "In search of an understandable consensus algorithm." in *USENIX Annual Technical Conference*, 2014, pp. 305–319.