7

Leveraging Neural Engineering in the Post-Factum Analysis of Complex Systems

Jason Sherwin[1] and Dimitri Mavris[2]
[1]*Columbia University in the City of New York,*
[2]*Georgia Institute of Technology,*
USA

1. Introduction

This chapter is about the pressing problem of, and our proposed response, to data deluge in the analysis of complex systems. We begin by illustrating the problem in certain systems engineering examples, primarily focusing on aerospace-related systems but pointing out the generality of this problem in other data-intensive design problems (Section 2). Having established the need to address this problem, we then propose a solution based on current advances in the intersecting fields of neuroscience and computer engineering, increasingly being called *neural engineering* (Section 3). With a proposed solution in mind, we carry out a case study in which we utilize certain results and algorithms from neural engineering (Section 4). Though this case study gives credible results, we find that we can improve our neural-based models of complex systems data from more direct neuroscience experiments on expertise (Section 5). Finally, we draw conclusions on the current state of the art for leveraging neural engineering results and algorithms on the problem of complex systems post-factum data analysis (Section 6).

2. A problem in systems engineering: Data deluge

The need to engineer within and for both increasingly complex and sophisticated systems is continually growing. In tandem with this problem is the need to analyze ever-increasing amounts of data that describe these systems. In short, the post-factum analysis of an already-built system is a key step in the analysis and, consequently, the design processes.

For instance, within the aerospace community, this problem is not unfamiliar. In that field, the perennial aim has been to balance the various sub-disciplines (e.g., acoustics, aerodynamics, propulsion, structures) to deliver an aircraft or spacecraft that meets a set of pre-defined criteria. But with each added sub-system of an aircraft, there is an added degree of complexity that is contributed to the design process.

This phenomenon is not unique to aerospace systems design. More generally, with the explosion of remote sensing capabilities in recent years, there has been a deluge of data made available about many other complex and intricate systems. But the means to fully analyze this data and to extract a useful comprehension of its content can be a challenge.

Both of these problems – one being a subset of the other – share a common thread: there is a plethora of computation needed to arrive at a design solution. In aircraft design, there is a potential deluge of possible designs as new sub-systems are added to the analysis. Similarly, in the mining of data from complex systems, there is likewise a deluge of possible data interpretations; and no specific interpretation is more 'correct' than any other (via the 'No Free Lunch Theorem', Ho & Pepyne, 2002).

In the midst of this deluge, it is potentially easier to approach the data holistically and to provide a subjective analysis of its content. Not only does this approach allow the data's full scope to be considered, but it also allows comprehension to be communicated rapidly because of its approximate – and therefore, simpler – nature. For instance, many systems engineering techniques have been devised to simplify the potentially overwhelming aspects of a complex system's analysis. Some examples of these are the analytical hierarchy process (Saaty, 2000 and Saaty, 2008), quality function deployment (Chan & Wu, 2002 and Akao, 1990) and other quasi-quantitative methods of subjective evaluation. While these methods have proven to be rapid, their transparency is lacking due to the expert-driven nature of the processing schema. For instance, in quality function deployment (QFD), experts in a particular discipline create subjective mappings from requirements to characteristics for a given product. There is no physics-based model that determines the product's design. Rather, a graded scale is used to map design requirements to characteristics based on a subjective assessment done by an expert. Necessarily, in this and other techniques like it, there is a crucial role for an expert's input to such analysis.

There has also been an opposite response to the data deluge in system analysis: utilize the increasingly available computation power to process the excessive amounts of data. In other words, rather than resign to the need for subjective analysis (e.g., in QFD) due to the problem's complexity, the availability of greater amounts of computing power in recent years has made it possible somewhat to navigate the deluge. For example, this has been the mentality behind the approach of multi-disciplinary optimization (Vanderplaats, 2007), which is used with great success in aircraft design. In multi-disciplinary optimization (MDO), numerical optimization techniques are applied to sets of objective functions whose dependent variables must satisfy various constraints (i.e., inequality, equality and side constraints). The ultimate aim though is not to yield a design that is optimal in any one system, but rather one that is optimal with regard to all systems. Necessarily, such a process is computationally quite costly and as the number of variables grows it becomes infeasible.

Similar examples exist for the analysis of data obtained remotely. For instance, the American military increasingly relies on remote sensing for many of its activities (e.g., the MQ-1 Predator drones, National Commission, 2004). But the exponentially-increasing amounts of data leave the analysts "swimming in sensors and drowning in data" (Drew 2010). In other words, the analytic tools to comprehend the data are well behind the means to gather it.

Such an analysis problem is an inherent precursor to engineering a complex system. For instance, it exists in the case where the system has been human-constructed from many parts (e.g., an aircraft). And it also exists when the system is not human-constructed, i.e., in nature. It is this latter situation that is of the most interest to us now though because, in reality, it is a superset of the first: whether man-made or not, it is a difficult engineering analysis problem to figure out how complex systems work. In particular, although the human-constructed parts may behave in predictable ways in many situations, there are always new interactions arising

between component sub-systems that reveal a previously unknown system-level behavior. Therefore, while reductionist approaches to system construction can be successful in most cases, we have still not obtained the hoped for deterministic prediction of behavior.

Instead, it seems that a degree of uncertainty is inherent in the analysis and consequently the engineering of all systems. This does not mean that we throw the previously successful reductionist approaches to the wind though. But for the analysis and engineering of those systems for which such approaches are impossible (e.g., not accurately quantifiable, not modelled within computational constraints), the only mechanism for design choices thus far has been the aforementioned systems engineering techniques (QFD, AHP, MDO, etc.). A new approach is needed.

3. A new path for handling data deluge in analysis: Neural engineering

As a result of this problem, we suggest here to consider the breadth of results and techniques emerging from neural engineering to bolster systems analysis for engineering purposes. In particular, instead of relying on an inconsistent mapping made by human experts to design analysis (e.g., as in QFD), why not understand some cognitive elements to expertise and, in turn, apply that comprehension to both systems analysis and manipulation? Of course, these are both monumental tasks to perform, considering not only the breadth of cognitive abilities that comprise expertise but also determining how to implement them in real engineering contexts.

Despite the seemingly daunting nature of these endeavors, certain elements of expert decision-making, human situational awareness and cortical biology can inform some of the details as to how we can understand and, in turn fine tune, the ways by which we as engineers collect observations and integrate them into a cohesive analysis; such an analysis is then the foundation of ensuing engineering choices. Nowhere is this need as great as it is in the analysis of complex and large-scale systems. Therefore, a true test as to the utility of neural engineering for systems purposes would be to implement these ideas within a complex or large-scale analysis and engineering task. In this chapter, we will demonstrate a simulated use of such an application.

As a demonstration, we discuss the application of neural engineering to the analysis of Iraq's stability during 2003-2008. This application was never used in a real context, however we frame the problem within the context of its utility to a decision-maker whose actions influence the outcome of such a system. In other words, he/she must analyze and then manipulate this system. Our assumption is that the decision-maker only has access to a stream of data that measures certain conditions related to Iraq's stability. More importantly, we assume that there is no possibility of developing an analytic model to describe the time-evolution of Iraq during these years. Rather, we cast aside that futile aim and attempt to glean useful patterns directly from the data. As part of this demonstration paraphrase (seeing as the full-blown analysis comprises a Ph.D. thesis and several papers), we emphasize the importance of learning algorithms to do so. Specifically, we consider algorithms based off of some anatomical and behavioral features of the human cortex. Built-in to the rationale behind using these algorithms is the assumption that many of the cognitive faculties comprising the development and use of expertise reside in the cortex.

Building off of the hope (and shortcomings) provided by the Iraq example, we then review some of the latest developments in neural engineering that have possible applications in the

analysis of other large-scale systems. As we will see, it is important to maintain an awareness of how the biological hardware (e.g., a neuronal network) "computes" its analysis of complex, time-evolving, often self-conflicting and/or occluded data. We will offer the results of an experiment from audio cognition in which expertise of subjects is tracked directly from neural data. Finally, we will then consider how these insights would then translate back to the actual solid-state computations done in modern computers to drive systems engineering analysis.

4. Case study of neural engineering-assisted analysis: Iraq war, 2003-2008

The focus of this case study is to create a computational situational awareness (SA) usable by the Department of Defense to gauge Iraq's stability during 2003-2008. Situational awareness is a term from psychology used to describe elemental steps of an expert's mental processes (Endsley, 1995). In other words, situational awareness is the description of the cognitive processes involved in an expert's analysis of a situation. So if we can design an appropriate computational SA for the Iraq context then it is equivalent to developing a means to analyze that context for driving it to a desired state – in other words, to engineer it.

As a theoretical construct, SA was developed to analyze the decision-making processes of aircraft pilots, yet its general usage has extended into many other areas in which expertise are employed by a human controller. In general, an SA can be particular to a given scenario or context. For example, pilots have SA for flying airplanes, pianists have SA for playing music, etc. In our case study, the SA of interest applies to the stability of Iraq during the war from 2003-2008, which henceforth will be called the 'Iraq context'.

To develop this SA computationally, we implement a method that is summarized by the information flow of Fig. 1.

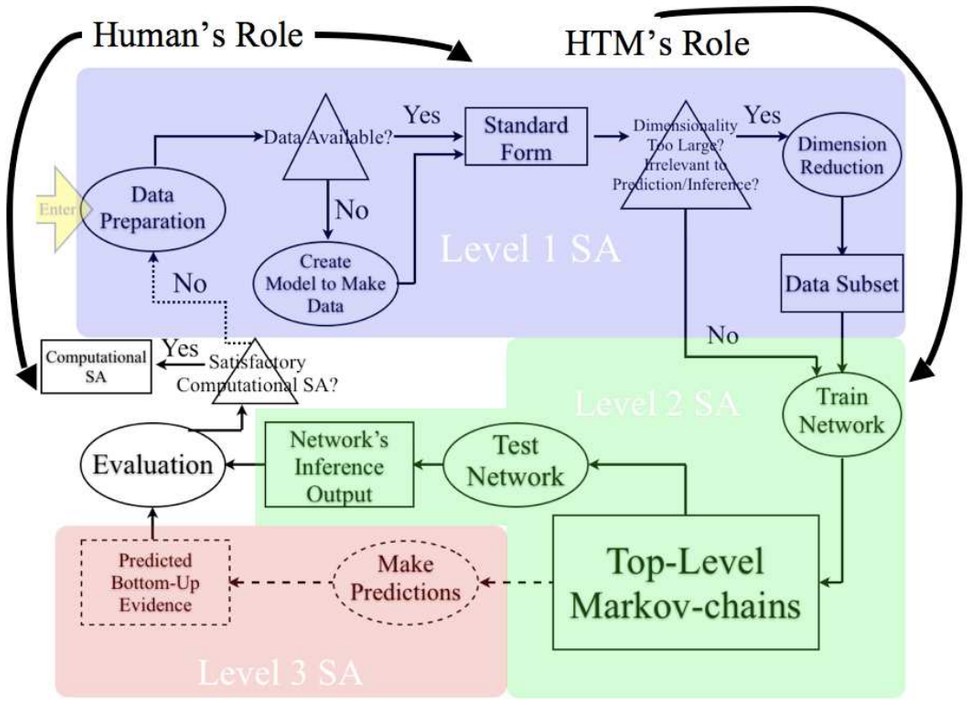


Fig. 1. Method for building/maintaining computational SA with HTM

This method maps the neurally-inspired machine learning algorithms to be used here (Hierarchical Temporal Memory, or HTM, see Section 3.2.1 for details, George & Hawkins, 2009 and George, 2008) to the three levels of SA first hypothesized by Endsley (Endsley, 1995). These three levels are Level 1 (perception of relevant elements), Level 2 (comprehension of those elements) and Level 3 (prediction). Here, we focus on Levels 1 and 2, since they are a necessary antecedent to Level 3. In particular, we present here a way to implement Levels 1 and 2 for this problem via data preprocessing and HTM training/testing.

4.1 Why use such a high-level representation of mental processes?

While this approach enhances awareness of trends in the data of the Iraq context, it also mimics the basic tenets of what constitutes SA in actual decision-makers. In particular, Endsley and others have shown that the selection of a goal is crucial to SA formation. In other words, the collection of data, its analysis and the engineering goal are all inextricable in forming SA. Here, we assume the criteria for success established by the U.S. Department of Defense: to bring Iraq to political, economic and social stability between 2003-2008 (United States House of Representatives, 2005). Consequently, we rely on data related to these aspects of the Iraq context, so that not only do we enhance a decision-maker's own SA of the Iraq context but we also create one – albeit, a rudimentary one – with a computer. By starting from such a high-level representation of expert mental processes, we can then specialize the computational tools used to find problem-relevant patterns in the data.

4.2 Deploying these processes computationally

Once the collection of data is focused onto problem-relevant elements, the analysis of that data becomes a learning problem. By conceiving of the expert's assessment of data as a learning problem (due to Endsley), we are in a position to mimic some of these processes computationally.

However, there is a question to consider before doing so: What is the right answer? In particular, no matter what machine learning approach is used, it is difficult to validate the SA learned about this context, since we do not know the right answer a priori. In other words, we do not know if our assessment of Iraq's stability is 'correct'. Although this is possible in other learning problems, such as invariant visual pattern recognition (i.e., the pattern is either object A or object B), we cannot do this here.

So to verify the accuracy of the computational SA formed in this context, another method will be introduced that has influence from system dynamics: we call it extreme-case bounding. This method has assumptions built into it that creates fictitious extreme cases of stability, either extremely unstable or extremely stable. With these fictitious bounds used for HTM network training/testing (e.g., extreme dystopia or utopia based on the data), some insight into the actual progression of events in Iraq during 2003-2008 can be obtained. Needless to say, this method is not perfect and it is somewhat arbitrary because we arbitrarily select a peg against which to measure Iraq's stability. Nevertheless, it provides an intriguing foothold in an avenue of computational SA that has thus far been difficult to probe concretely.

4.2.1 The final computational piece: A neurally-inspired machine learning algorithm

Thus far, Hierarchical Temporal Memory (HTM) algorithms have been mentioned as a way to execute the various stages of SA accounted by Endsley and have been adapted into Fig. 1. However, we have not yet discussed why these algorithms in particular are of interest. In what follows, we will argue that the hierarchical storage of spatio-temporal data and the way by which temporally adjacent data points are related to each other lend well to steps of SA laid out in Fig. 1. To make these points clear, Fig. 1 includes learning and inference steps involved in HTM training and testing as well.

An HTM attempts to mimic two crucial aspects of cortical function and anatomy. These are particularly of use for determining better ways to handle highly-varied data, so their potential utility in forming SA are apparent. First, these algorithms rely on the temporal adjacency of observed events when storing spatial patterns. The anatomical inspiration for this procedure comes from observations of cortical function. In particular, there are numerous cell groups and types in the cortex that have been identified as 'sequence detectors' (e.g., PPA, Broca's Area, FFA). Secondly, and in relation to the first aspect, the algorithms store these sequences in a hierarchical arrangement across both space and time. The result of this division of spacetime is that local regions' spatio-temporal patterns are first encoded from which more global regions' patterns are then encoded, etc. The anatomical inspiration for this compartmentalization of information comes directly from the different hierarchical functional areas observed in the human cortex (e.g., visual cortex, audio cortex, etc.).

Since an HTM is implemented on a computer, it is perhaps useful to consider a mathematical description of HTM. For starters, a trained HTM is in fact a Bayesian inference network. In particular, it is a network of nodes, each of which solving the same problem: learning spatio-temporal sequences. On a network-level, the goal behind the algorithms is to learn a schema (S) that describes data related to a given problem. That problem exists locally for each node in a vector space (v_i) and, upon grouping all nodes, exists on a global level that concerns all data related to the problem (V). Considering the local version of the problem, each vector (x_k) in v_i is an observation of an aspect of a given complex phenomenon at the k^{th} time step. The HTM node's goal then is to create q Markov-chains to which any one of the vectors in v_i can be assigned. For compression purposes, it is highly desirable that $q < k$. By collecting sets of observations in this way, each node's Markov-chain (m_q) corresponds to some spatio-temporal high-level feature of the local complex phenomenon. Consequently, the set of all M Markov-chains constitutes a schema (S) of the global phenomenon. In other words, S is a reduction of the phenomenon witnessed in each node's vector space, v_i. In particular, by using HTM, the aim is to use learning mechanisms akin to certain aspects of neural coding to develop a schema, i.e., a situational awareness of the data (V).

4.3 Implementation for the Iraq context

For the Iraq context, the phenomenon is the Iraq War during 2003-2008. The goal of the HTM then is to create a schema (S_{Iraq}) that is a suitable description of the Iraq context. This schema is based on what kind of data is used to describe the Iraq War (V_{Iraq}). Recalling that the analysis and collection of data are inextricably linked in forming SA, and due to the DoD goal of achieving political, economic and security stability, we have chosen metrics of these

aspects of Iraq stability to track during 2003-2008. In the following sub-sections, we show what specific data is used (part of forming Level 1 SA) and how that data is ‚comprehended' using a trained HTM for inference (Level 2 SA).

4.3.1 Level 1 SA: Data preparation

Entering the information flow of Fig. 1, the first task (represented by an oval) is to prepare the data. Before doing so, we must address the Boolean question (represented by a triangle) about whether data is available. For the Iraq context, we actually have data. But some effort is needed to prepare this data into standard form.

There are four issues we must confront in doing so. First, the primary source (United States Department of Defense, 2005, 2006, 2007, 2008, and O'Hanlon & Campbell, 2008) from which the data is extracted contains many blanks in the data, depending on how many metrics are used. So a set of metrics must be selected from the actual data that exhibits a minimal number of blanks.

Second, the primary source has not prepared the data in a temporally structured format suitable for HTM learning. Dileep George pioneered work on HTM algorithms and he gives guidelines for generalizing their usage in other domains. In particular, George writes, "if there is no temporal structure in the data, application of an HTM to that data need not give any generalization advantage." So the data must be arranged in this fashion if HTM is to be of use. Specifically, observations at specific time intervals should follow one another.

Third, the relative magnitudes of the chosen metrics will be necessary to consider. Consequently, a transformation of the data may be necessary before training/testing.

Fourth and finally, one of the metrics we use to describe the Iraq context is only known within given bounds at each time step. Consequently, we must select a technique to get only one value at each time step, rather than a range.

Considering all of these points, it is possible to pick a subset of metrics from the primary source that we can use to describe the Iraq context in a data-driven fashion related to our goal of tracking stability. These selected metrics are shown in Table 1 with identifying numbers next to each of them.

Metric	Units	Metric #
Iraqi_Civilian_Fatalities	persons	1
Multiple_Fatality_Bombings	persons	2
US_Troop_Fatalities	persons	3
Improvised_Explosive_Device_US_Troop_Deaths	persons	4
Car_Bomb_US_Troop_Deaths	persons	5
Mortar_and_Rocket_US_Troop_Deaths	persons	6
Rocket_Propelled_Grenade_US_Troop_Deaths	persons	7
Helicopter_Loss_US_Troop_Deaths	persons	8
Other_Hostile_Fire_US_Troop_Deaths	persons	9
Total_US_Troop_Deaths	persons	10
Attacks_on_Iraqi_Infrastructure_and_Personnel	incidents	11
Coalition_Troop_Strength	persons	12
Crude_Oil_Production	millions of barrels per day	13
Crude_Oil_Export	millions of barrels per day	14
Nationwide_Electricity	Megawatts	15
Nationwide_Unemployment_Rate	%	16

Table 1. Sixteen metrics to describe Iraq context

While it is possible to select fewer metrics, a drop off in performance was seen when this was done. We have shown this in other works (Sherwin & Mavris, 2011 and Sherwin, 2010). We believe that this occurs because the degree of stability in an operational theater already lacks a clear definition amongst stakeholders. Consequently, the more metrics that are incorporated into the analysis, the more complete the description of stability will be. Inversely, the fewer metrics that are incorporated, the more narrow the description will be. Here, we stopped at sixteen because this number approaches the upper limit of what was publically available, although more metrics may make the analysis that much more rich.

Finally, to give the HTM a baseline for stability and instability, artificial data generated from a rudimentary system dynamics was created based on the selected metrics. For instance, in this model (for stability, for instance), the number of troop deaths due to car bombs fell off to zero over time (roughly the same 60 months of time for which actual data exists). Alternatively, in this model, (for instability, e.g.), the nationwide electricity would flatten out to zero. In general, metrics associated with stable or unstable situations would monotonically be driven to an extreme maximum or minimum over the course of 60 months, starting from a real data point. In other words, we use extreme-cases to bound the reality observed in actuality – and this reality is more of a complex mix of certain features of instability and/or stability along different avenues (such as politically, socially, or economically).

4.3.2 Level 2 SA: HTM-aided comprehension of the data

With the ability to generate data for both progressively stable and unstable situations, as well as the actual time series of data on the Iraq context, it is possible to attempt HTM as an unsupervised machine learning mechanism. Recall, an HTM is a network trained is built to find a schema, S, that describes the Iraq context. This is based on each vector observed in each node's local vector space, v_i, all of which considered together constitute V. To aid the spatio-temporal grouping, these vectors are first grouped with K-means clustering before temporal adjacency is learned and grouped into the network's Marko-chains, M. These Markov-chains are then used to perform evidence-based Bayesian inference on novel data.

With HTM, the aim now is to fuse the data and to extract possibly implicit meaning from it pertinent to the Iraq context. We emphasize the unsupervised nature of the learning here because our goal is to extract implicit meaning and not to impose our possibly biased judgments. Furthermore, we attempt to extract this meaning from a system that is not ergodic (i.e., there is no end-state), not separable into components (hence, model-able) and not completely observable (i.e., uncertain data describes the system).

It has been found to be more effective to first train the HTM on the extreme-case data and then to test its inference capabilities on the actual data (Sherwin & Mavris, 2011). Therefore, we implement this approach so that the HTM can learn from the extreme-case boundaries and then use them to classify the reality in between.[1]

The evaluation of this computational SA is not entirely straightforward and so additional techniques were employed to probe the SA formed about the Iraq context. These studies will

[1] The former method has been tried and has proven unsuccessful (Sherwin, 2010).

not be reviewed in too much depth now, but a summary of them is important for our purposes.

For starters, we do not know if too many or too few metrics are being used here to describe the Iraq context. Therefore, studies were done to see what effects there are from reducing the number of metrics used to train and infer with the network. It was found that fewer metrics reduce the semantic richness of the data, thereby causing certain volatile metrics to dominate the learning and subsequent inference.

Also, we examined the degree to which information is hierarchically stored in intermediate levels of the networks. We found that, true to the promise of HTM, this was the case. Finally, we considered alternative ways of feeding the data into the networks to see what effects – if any – there are on the simulated SA.

Why would we use these techniques in particular though? For instance, what purpose could there be in probing the hierarchical storage of information in an HTM? It is necessary to recall that our purpose in using HTM for computational SA has been its declared ability to condense information into hierarchies of both space and time. For the Iraq context, we test hierarchical storage directly because it is not clear what the top-level node's output should be, as it might be for simpler recognition tasks (e.g., invariant visual pattern recognition, George & Hawkins, 2009, and George, 2008). One possible outcome of this analysis is that it might in turn help us to identify what aspects of the Iraq context are not well observed. This would then provide the beginnings of a feedback mechanism with Level 1 SA to search for more data.

In fact, in the course of this research, it was one possible feedback mechanism between Levels 1 & 2 SA that informed us to improve our extreme-case model. This resulted in the monotonic extreme-case models used below. Necessarily, this is not the only possible feedback mechanism, but as we will see, it helps to strengthen the credibility of the Level 2 SA formed here computationally. If we use data for training that becomes monotonically extreme from a realistic starting condition then we would expect an HTM network to learn to recognize clear progressions towards stability/instability.

We employ an evolutionary approach to network design here and modify a network used in an HTM demonstration example (see Numenta, 2008).[2] In order to exploit the HTM network's temporal learning algorithms, we modify the network parameters to accommodate how the metrics' values change in time. The complete network parameters employed for this computational Level 2 SA can be seen in another work (see appendix C.9 in Sherwin, 2010).[3]

From training the network, we can survey the resulting schema created by the network. As for all HTM nodes, this is described in terms of coincidence patterns (i.e., distinct spatial patterns) that form the schema's Markov-chains. Here, we find that there are sixty-one

[2] All analysis has been done with Vitamin D Toolkit 1.3.0 as a graphical user interface to NuPIC 1.6.1, which is run on Python 2.5.2. It should be noted that slightly different results are obtained if networks are created, trained and tested directly in NuPIC. See appendix D in Sherwin, 2010 for more information on this topic.

[3] Even though this work says that these parameters are for a progressively trained network, they are the same ones used for the monotonically trained one.

coincidence patterns ($C_{3,1}$) and fifty-nine Markov-chains ($G_{3,1}$) in the top-level node.[4] The coincidence patterns are the result of the K-means clustering in this top-level node, while the Markov-chains are the result of first-order transition probabilities between these coincidence patterns. This is a standard calculation for each of the nodes in an HTM network (see George & Hawkins, 2009).

After proper training, an HTM network is most valuable as an inference tool, so now we evaluate its performance. We will start plaintively by looking at inference on the training data, moving onto novel data later.

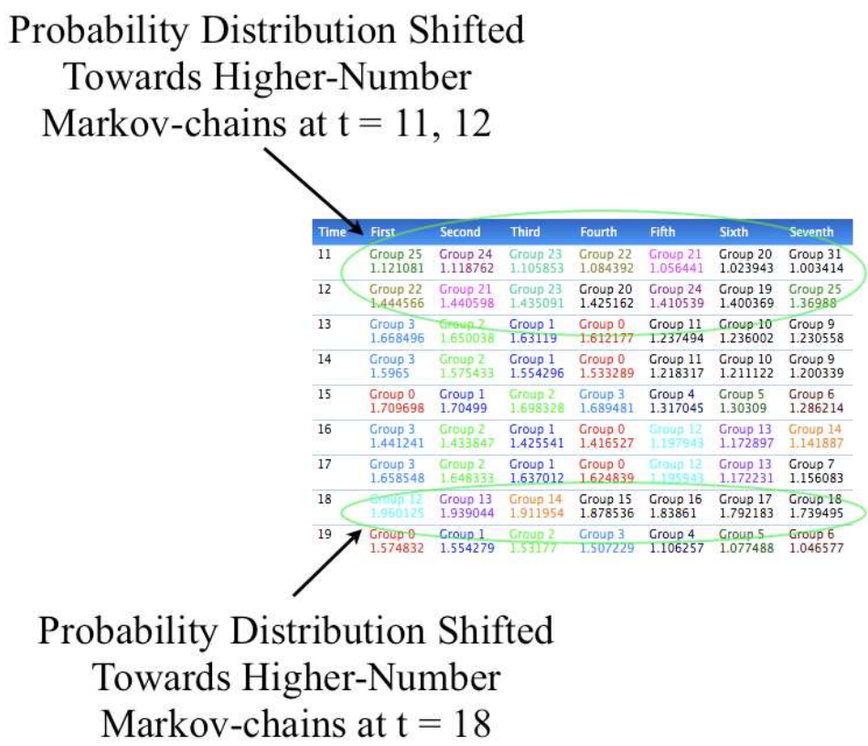
When we perform inference on the monotonic training data, we see a clear progression of Markov-chains as instability increases, but stability is still not clear. We can see this by following the probability over Markov-chains $\lambda_t(g_r)$ of the top-level node, given the bottom-up evidence ($-e_t$) at each t. In particular, we examine the maximum of this distribution ($\max[\lambda_t(g_r)]$) to see what stability state is most likely. What we find is that the progression indicated by $\max[\lambda_t(g_r)]$ is *g0*, *g1*, *g2*, ..., *g58*. Since we know the data is monotonic towards instability, we can reasonably claim that the Markov-chain labels are monotonic towards instability as well. For example, the bottom-up evidence when g45 is most likely in the top level indicates a situation that is less stable than when g5 is most likely.

Having trained the network to recognize progressions in instability, it would be useful now to test this ability on novel data. In particular, we feed into the network real data of the Iraq context. When we look for instability gradations in the actual data, we see some interesting results (Fig. 2). In Fig. 2, as in similar figures to follow, each row is a time point. The first, second, third, etc. columns indicate the groups (i.e., Markov-chains) that are most likely, second most likely, third most likely, etc., given the bottom-up evidence. The significance of this ordering of group numbers at each time point is that we can quantitatively say how unstable Iraq is at each of them. Note throughout that time points $t \in [0, 60]$ correspond to each month from May 2003 to April 2008. In particular, at $t = 11, 12$, the entire probability distribution over top-level Markov-chains shifts towards higher number Markov-chains. At $t = 11$, *g25, g24, g23* are in the top three (see Fig. 2).

At $t = 18$, the probability distribution shifts as well, indicating *g12, g13, g14* in the top three. In light of our results from inference on the monotonic-extreme-case instability data, it would seem that the Iraq context is increasingly unstable during these months. Furthermore, the actual data during these months indicates this in comparison to those months that come before them.

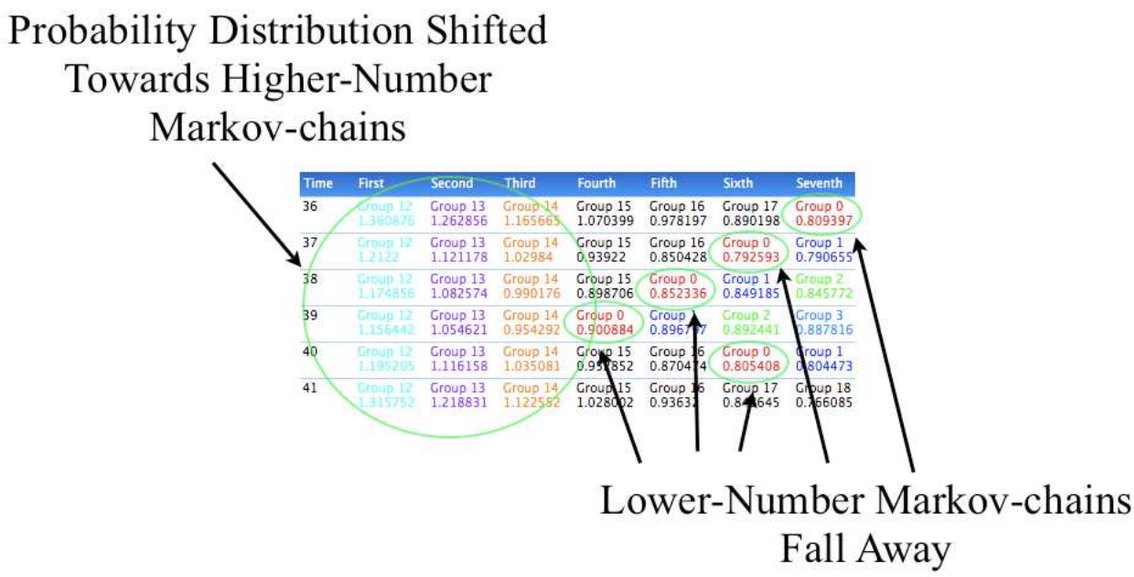
Let us expand our purview to those time points leading up to and coming out of $t \in [41,49]$, another region of heightened instability according to the network. If we consider the top seven Markov-chains of the top-level for $t \in [36,60]$ then we see something quite interesting. For $t \in [36,41]$, the distribution shifts increasingly towards *g12, g13, g14, g15, g16, g17, g18*. Also, we can see the demotion of *g0* over these time steps (Fig. 3), indicating increasing instability.

[4] Here and throughout the remainder of the paper, we follow George's notation for HTM theory (George & Hawkins, 2009 and George, 2008).



Time	First	Second	Third	Fourth	Fifth	Sixth	Seventh
11	Group 25 1.121081	Group 24 1.118762	Group 23 1.105853	Group 22 1.084392	Group 21 1.056441	Group 20 1.023943	Group 31 1.003414
12	Group 22 1.444566	Group 21 1.440598	Group 23 1.435091	Group 20 1.425162	Group 24 1.410539	Group 19 1.400369	Group 25 1.36988
13	Group 3 1.668496	Group 2 1.650038	Group 1 1.63119	Group 0 1.612177	Group 11 1.237494	Group 10 1.236002	Group 9 1.230558
14	Group 3 1.5965	Group 2 1.575433	Group 1 1.554296	Group 0 1.533289	Group 11 1.218317	Group 10 1.211122	Group 9 1.200339
15	Group 0 1.709698	Group 1 1.70499	Group 2 1.698328	Group 3 1.689481	Group 4 1.317045	Group 5 1.30309	Group 6 1.286214
16	Group 3 1.441241	Group 2 1.433847	Group 1 1.425541	Group 0 1.416527	Group 12 1.197943	Group 13 1.172897	Group 14 1.141887
17	Group 3 1.658548	Group 2 1.648333	Group 1 1.637012	Group 0 1.624839	Group 12 1.195943	Group 13 1.172231	Group 7 1.156083
18	Group 12 1.960125	Group 13 1.939044	Group 14 1.911954	Group 15 1.878536	Group 16 1.83861	Group 17 1.792183	Group 18 1.739495
19	Group 0 1.574832	Group 1 1.554279	Group 2 1.53177	Group 3 1.507229	Group 4 1.106257	Group 5 1.077488	Group 6 1.046577

Fig. 2. Instability recognition of real data



Time	First	Second	Third	Fourth	Fifth	Sixth	Seventh
36	Group 12 1.360876	Group 13 1.262856	Group 14 1.165665	Group 15 1.070399	Group 16 0.978197	Group 17 0.890198	Group 0 0.809397
37	Group 12 1.2122	Group 13 1.121178	Group 14 1.02984	Group 15 0.93922	Group 16 0.850428	Group 0 0.792593	Group 1 0.790655
38	Group 12 1.174856	Group 13 1.082574	Group 14 0.990176	Group 15 0.898706	Group 0 0.852336	Group 1 0.849185	Group 2 0.845772
39	Group 12 1.156442	Group 13 1.054621	Group 14 0.954292	Group 0 0.900884	Group 1 0.896747	Group 2 0.892441	Group 3 0.887816
40	Group 12 1.195205	Group 13 1.116158	Group 14 1.035081	Group 15 0.957852	Group 16 0.870414	Group 0 0.805408	Group 1 0.804473
41	Group 12 1.315752	Group 13 1.218831	Group 14 1.122552	Group 15 1.028002	Group 16 0.93631	Group 17 0.844645	Group 18 0.766085

Fig. 3. Probability Distribution Shifts Towards Higher-Number Markov-chains

Then for $t \in [41,49]$, these seven Markov-chains are the most likely, given the bottom-up evidence (Fig. 4).

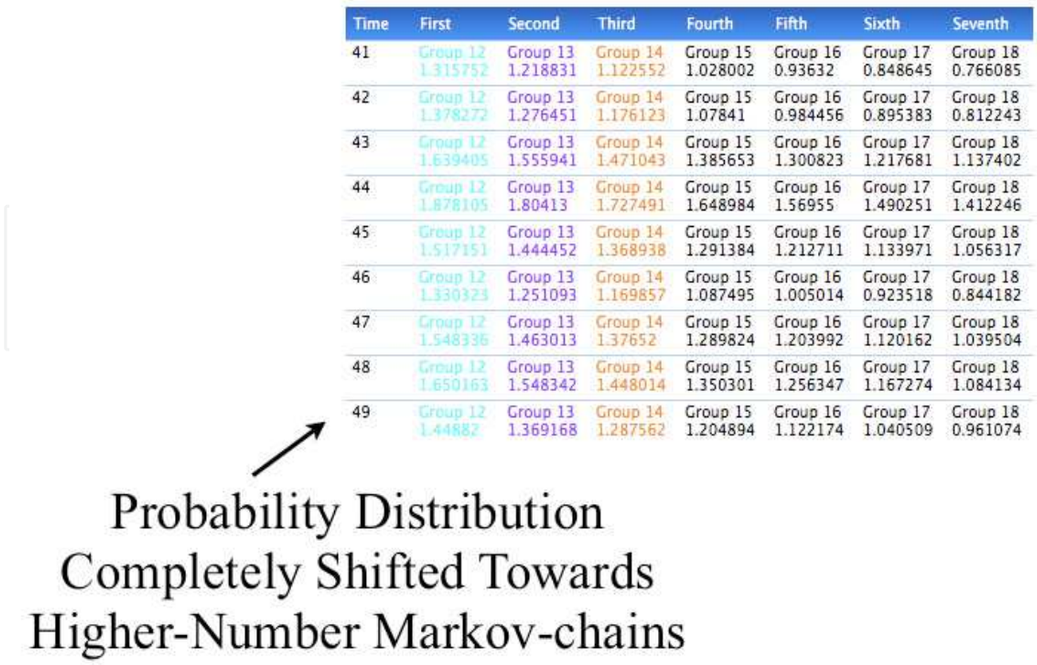


Fig. 4. Complete Shift Towards Markov-chains Indicating Instability

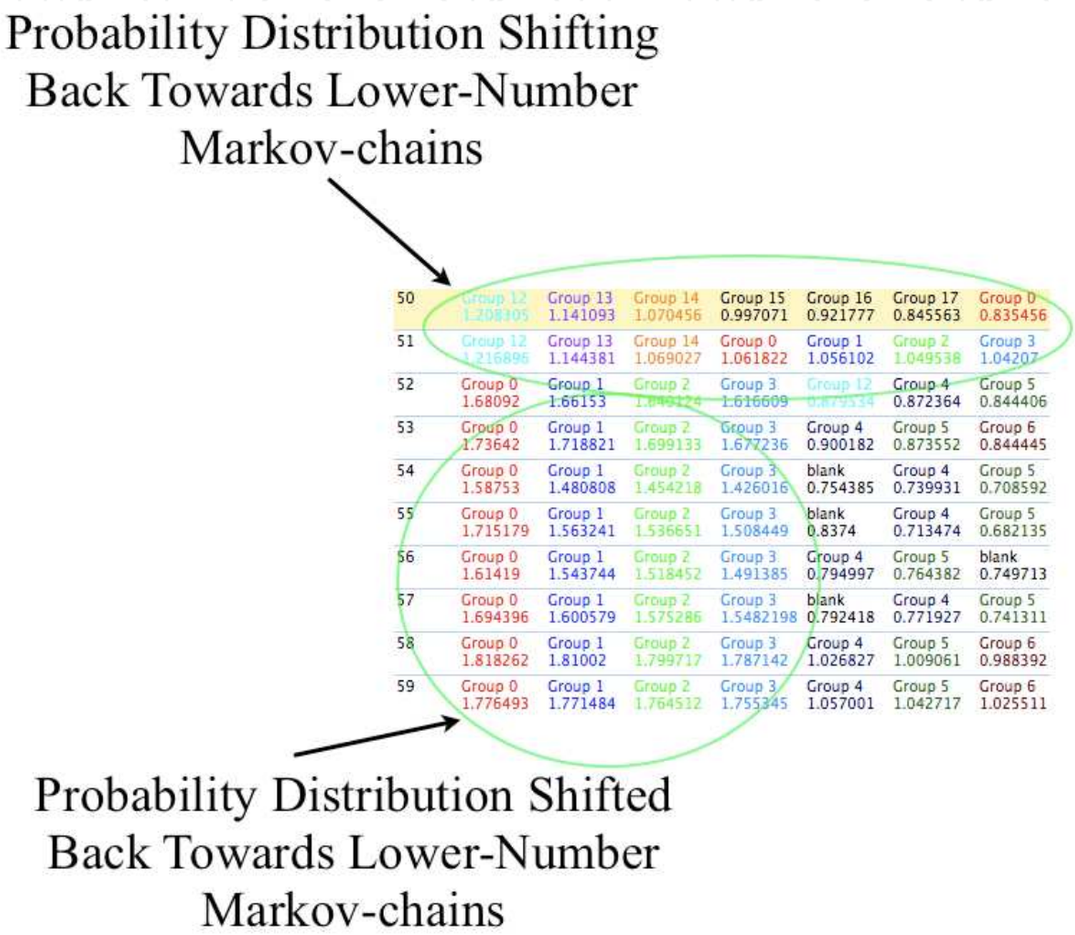
As we know from the actual data, there were peaks in violence and other attacks, sagging economic metrics, etc. during this time. But there were also metric trends that favored stability. Consequently, this conflicting data makes it difficult to characterize the stability level during this time period. But here we see the probability distribution shift for the entire time period towards these mid-grade instable states.

The situation in the Iraq context changes though with time. For $t \in [50,51]$, the probability distribution begins to shift back. Finally, for $t \in [52,60]$, *g0, g1, g2, g3* are the top four most likely Markov-chains (see Fig. 5).

Even though this does not indicate stability, it does indicate a dramatic drop in instability, according to how we trained the network. So we see here how the monotonic training data has provided a peg against which to categorize evidence that trends towards instability. But what about stability recognition?

As mentioned earlier, direct stability recognition is less clear, even with the monotonic training data. Rather, we can only infer stability recognition with the network. Why is this so? If we consider the types of metrics used here then we notice that only four of them increase with stability. So, as a more stable situation is reached, the remaining twelve metrics drop close to zero. Consequently, the bottom-up evidence does not provide enough magnitude to propagate through the network. All the change comes from the four metrics that increase with stability. In the current permutation of the data, one of them is in the receptive field of the third node in level one ($N_{1,3}$) and the other four are in the field of $N_{1,4}$. The entire left receptive field (covered by $N_{1,1}$ and $N_{1,2}$) therefore produces blank recognition. This is because there is simply not enough bottom-up evidence coming up through this side of the network. So we are not able to determine gradations of stability

because the utility function of these metrics can be assumed to be inversely proportional to stability. Consequently, as stability is reached, the magnitude of $-e_t$ goes to zero. In future implementations, it might be possible to alleviate this problem by transforming the data by an inverse or offsetting the zero. We have not done this though because we have devised an approach to recognize degrees of instability in real data, as judged against the extreme-case baseline. Furthermore, these results imply stability recognition due to the monotonic utility function of stability/instability.



50	Group 12 1.208305	Group 13 1.141093	Group 14 1.070456	Group 15 0.997071	Group 16 0.921777	Group 17 0.845563	Group 0 0.835456
51	Group 12 1.216896	Group 13 1.144381	Group 14 1.069027	Group 0 1.061822	Group 1 1.056102	Group 2 1.049538	Group 3 1.04207
52	Group 0 1.68092	Group 1 1.66153	Group 2 1.643124	Group 3 1.616609	Group 12 0.879534	Group 4 0.872364	Group 5 0.844406
53	Group 0 1.73642	Group 1 1.718821	Group 2 1.699133	Group 3 1.677236	Group 4 0.900182	Group 5 0.873552	Group 6 0.844445
54	Group 0 1.58753	Group 1 1.480808	Group 2 1.454218	Group 3 1.426016	blank 0.754385	Group 4 0.739931	Group 5 0.708592
55	Group 0 1.715179	Group 1 1.563241	Group 2 1.536651	Group 3 1.508449	blank 0.8374	Group 4 0.713474	Group 5 0.682135
56	Group 0 1.61419	Group 1 1.543744	Group 2 1.518452	Group 3 1.491385	Group 4 0.794997	Group 5 0.764382	blank 0.749713
57	Group 0 1.694396	Group 1 1.600579	Group 2 1.575286	Group 3 1.5482198	blank 0.792418	Group 4 0.771927	Group 5 0.741311
58	Group 0 1.818262	Group 1 1.81002	Group 2 1.799717	Group 3 1.787142	Group 4 1.026827	Group 5 1.009061	Group 6 0.988392
59	Group 0 1.776493	Group 1 1.771484	Group 2 1.764512	Group 3 1.755345	Group 4 1.057001	Group 5 1.042717	Group 6 1.025511

Fig. 5. Complete Shift Back Towards Markov-chains Indicating Less Instability

4.4 Consequences of the Iraq context implementation

We should be very clear at the outset: the schema formed with the HTM-based computational SA created here is not how a human brain of an expert decision-maker would function. Rather, it is an alternative analysis of the data at hand that can be used by a decision-maker in such a scenario. The key element to its utility though is the fact that the computational SA functions in analogous ways to some neuro-anatomical processes, such as coincidence detection and spatio-temporal pattern condensation into a flexible schema. In particular, the HTM-based SA learns from its experiences to infer about uncertain and novel situations. To be implemented on a computer, it does so with the aforementioned

hierarchical and temporal breakdown of its 'experiences' (in the form of spatio-temporal vectors).

Furthermore, this is not a perfect implementation of neural engineering to analyze complex systems. But as the preceding sections demonstrate, it provides a quantifiable analysis of a system that is otherwise left in the hands of subjective analyses whose justifications are missing or obfuscating. Perhaps with better insight into human expert's mental processes the results would have stronger impact in systems engineering analysis.

5. Recent neuroscientific results on expertise

It should be clear from the preceding implementation of computational SA that the following is true:

1. Situational awareness (SA) is a nebulous term used to define an equally nebulous ability of humans
2. The machine learning-based approximation of this process with HTM is imperfect
3. More details on neural markers of expertise would inform any future computerizations of human mental processes

Considering these points in succession, it is clear that a refined perspective on human-borne expertise can add tremendous value to our first attempt of forming SA computationally. In particular, we aim to highlight some recent advances in the neuroscience of expertise that can ultimately be of use in how we analyze and design complex systems in engineering.

5.1 What happens when you poke an expert's brain?

The most insightful way to examine an expert's brain is to subject him/her to stimuli that violate their expertise-borne predictions. This experimental paradigm has seen tremendous success in tracking the neural markers of unexpected stimuli (most notably in analysis of the P300, a positivity that emerges around 300ms after a repeatedly unexpected stimulus is observed). By tracking neural signatures like the P300 and others, it is possible to see how a human brain – experienced in a certain stimulus domain – responds to errant stimuli.

Although research in many modalities have been done (e.g., functional magnetic resonance imaging (fMRI) and magnetoencephalography (MEG)), we focus here on electroencephalography (EEG) measurements of neural data. We do this for two reasons: 1) single-trial classification of neural data from EEG is generally more robust than it is from fMRI (and not very developed for MEG), 2) EEG neural data has a much higher temporal resolution than fMRI (and slightly higher than MEG), making it an ideal candidate for more immediate integration into systems engineering problems.

5.1.2 A simple experiment in error-detection

To illustrate the kinds of neural processes observable with EEG systems, we will summarize some experimental work on expectation violation. While this experiment may seem removed in the specific sense from analyzing expertise for the purposes of systems engineering analysis, the abstract concept at the heart of this experiment could not be more on target. In particular, subjects are asked to listen to an audio stimulus with which they are

quite familiar. In this case, American popular songs, such as „Eye of the Tiger“ or „Sweet Home Alabama,“ are used because the subjects all have strong prior expectations for the course of these songs once they start listening to them. In other words, they are experts at how these songs unfold. In this experiment, we analyze the subject's expertise about these audio stimuli.

The aim of this experiment is to alter the song in such a way that the following occurrences are true: 1) a subject with normal hearing should be able to discern the alteration, 2) the subject should be able to reorient his/her expectations after the alteration has taken place. The latter condition is a requirement if multiple alterations are to be performed within one hearing of the song. To balance these two requirements, it was chosen that the song's key should be altered either up or down by a semi-tone at various points in the recording.

After analyzing the neural data with learning algorithms based on logistic regression classification (Parra et al., 2002 and Parra et al., 2005), it is found that we can distinguish from neural data alone when the subject perceived an alteration and when he/she did not, regardless of where they were in the song. In other words, we are able to distinguish times in the experiment when the subject's expertise (and associated predictions) were at a conflict with the data (i.e., the audio stimuli) in front of him/her. An example of this phenomenon can be seen in Fig. 6.

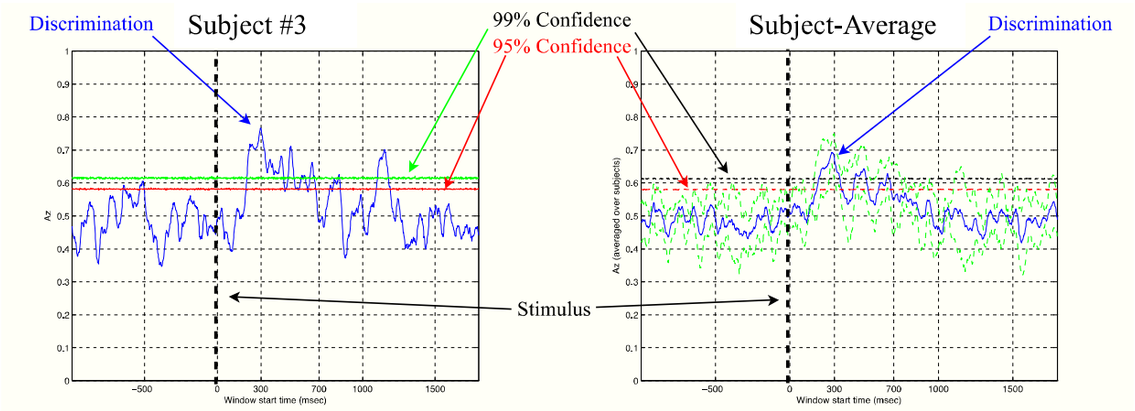


Fig. 6. Individual subject and subject-average discriminations of expertise violation

What this plot shows is the classification performance in blue (measured by Az, or the area under the receiver-operator characteristic curve, see Green & Swets, 1966) vs. the time when this classification is performed relative to when the alteration happened (in the case of the alteration) or did not happen (in the case of the corresponding condition in a control listening). The alteration is shown in the dotted-black line. The green and red solid lines in the left figure indicate the 99% and 95% confidence lines. In the right figure, the subject-averaged significance lines are differentiated from the individual lines (because they are computed differently) by being dashed. The red-dashed line is the 95% and the black-dashed line is the 99% significance line. Finally, the dashed-green line is the 5%-standard deviation from the averaging. As the average plot shows, similar plots exist for other subjects in the experiment than the one shown on the left.

This experiment shows that there is a measurable neural process happening when an expert witnesses something unexpected. In particular, just as with HTM, the key component to deciphering the incoming phenomenon (here, it is audio, V_{Audio}) is the subject's schema (S_{Audio}) and how it predicts the evolution of V_{Audio} in time. In other words, we are playing with the subjects' Level 3 SA (i.e., prediction). If this result is to be used in the analysis of complex systems then it must involve a dynamic system. Luckily, this challenge in complex system analysis is actually a boon for an expert's assessment of a phenomenon, V.

5.2 What does this mean for systems engineering?

As described earlier, systems engineering is a complex process that is largely driven by expert insight into a range of problems. But we see both from the computational SA demonstration and one result (of many related others) from neuroscience that a new way to think about expertise is developing. Furthermore, both results depend on a temporal evolution of data as a means by which SA is created, either computationally or by a human.

It may be possible to integrate this understanding technologically with the needs of systems engineering today. For instance, why not handle the data deluge with an approach that couples the speedy perception of cortical circuitry to the vast computational power of modern computers? This is already being done in remote sensing in the form of cortically-coupled computer vision (Sajda, 2010). Similar versions of this problem exist in aircraft analysis and design, as described earlier. As we become more familiar with the insights on expertise provided by neuroscience, it could only be a matter of time before neural engineering is a needed integrative enabler of analyzing and manipulating increasingly complex and dynamic systems.

6. Conclusion

The systems engineering task is only getting more difficult. As systems become more complex and demands for reliability increase, there is a growing need – already being met in certain ways – to build appropriate analytic tools to engineer within this context. But systems engineers are increasingly required to analyze an already existing system before the design process begins. In other words, post-factum analysis is a key aspect to the systems engineering process. Consequently, subject matter expertise becomes a key enabler for the design process. However, the elusive nature of expertise remains an intractable aspect to that process. Here, we have shown a computational approach built on insights into expertise and how it was used in a decision-making context. In other words, we showed how the analysis of the provided data enabled justifiable conclusions on an otherwise unpredictable and already established complex system (e.g., the Iraq War, 2003-2008). However, we noticed some shortcomings in this approach and so turned our focus to more basic neuroscience questions about what neural processes occur as expertise is used. In particular, we discussed one recent experiment as an example of current work going in the neural markers of expertise. Since such markers are stimulus driven, in particular, the expert reacts to data being either in line with expectations or not, we forecasted a potential role for neural engineering in the analysis, and consequent design, of complex systems. Not only does this need exist but such an approach would also complement the current techniques used in this endeavor.

7. Acknowledgment

We want to thank the Aerospace Systems Design Laboratory at Georgia Institute of Technology for supporting this research.

8. References

Akao, Y. (1990). *Quality Function Deployment: Integrating Customer Requirements Into Product Design,* Productivity Press, ISBN 1-56327-313-6, New York, USA

Chan, L. & Wu, M. (2002). Quality function deployment: A literature review. *European Journal of Operational Research,* Vol.143, No.3, (December 2002), pp. 463-497

Drew, C. (2010). Military is awash in data from drones, In: *New York Times,* 10.01.2010, Available from http://www.nytimes.com/2010/01/11/business/11drone.html?pagewanted=all

Endsley, M. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors,* Vol.37, No.1, (March 1995), pp. 32-64

George, D. & Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology,* Vol.5, No.10, (January 2009), pp. 1-26.

George, D. (2008). *How the Brain Might Work: A Hierarchical and Temporal Model for Learning and Recognition,* Ph.D. Dissertation, Stanford University, Palo Alto, USA

Green, D. & Swets, J. (1966). *Signal Detection Theory and Psychophysics,* Wiley & Sons, Inc., ISBN 0-932146-23-6, New York, USA

Ho, Y. & Pepyne, D. L. (2002). Simple explanation of the no-free-lunch-theorem and its implications. *Journal of Optimization Theory,* Vol.115, No.3, (December 2002), pp. 549-570

National Commission on Terrorist Attacks upon the United States (2004). *The 9/11 Commission Report,* Norton and Co., ISBN 0-393-32671-3, New York, USA

Numenta, Inc. (2008). Getting Started with NuPIC, Numenta, Inc., Available from http://www.numenta.com/archives/education/nupic_gettingstarted.pdf, pp. 34-35

O'Hanlon, M. E., & Campbell, J. H. (2008). *Iraq Index: Tracking Variable of Reconstruction & Security in Post-Saddam Iraq,* The Brookings Institution, Washington, DC, USA

Parra, L. et al. (2002). Linear Spatial Integration for Single Trial Detection in Encephalography. *NeuroImage,* Vol.17, (December 2002), pp. 223-230

Parra, L. et al. (2005). Recipes for the linear analysis of EEG. *NeuroImage,* Vol.28, No.2, (May 2005), pp. 326-341

Saaty, T. L. (2000). *Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process,* RWS Publications, ISBN 0-9620317-6-3, Pittsburgh, USA

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences,* Vol.1, No.1, (2002), pp. 83-98

Sajda, P. et al. (2010). In a blink of an eye and a switch of a transistor: Cortically-coupled computer vision, *Proceedings of the IEEE*, vol. 98, no. 3, March, 2010

Sherwin, J. S. & Mavris, D. N. (2011). A computational approach to situational awareness: A follow-up on the details. *Journal of Battlefield Technology,* Vol.11, No.1, (March 2011), pp. 1-11

Sherwin, J. S. (2010). *A Computational Approach to Achieve Situational Awareness from Limited Observations of a Complex System,* Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, USA

United States Department of Defense (2005). Report to Congress: Measuring Stability and Security in Iraq, July 2005
United States Department of Defense (2005). Report to Congress: Measuring Stability and Security in Iraq, October 2005
United States Department of Defense (2005). Report to Congress: Measuring Stability and Security in Iraq, May 2006
United States Department of Defense (2006). Report to Congress: Measuring Stability and Security in Iraq, February 2006
United States Department of Defense (2006). Report to Congress: Measuring Stability and Security in Iraq, August 2006
United States Department of Defense (2006). Report to Congress: Measuring Stability and Security in Iraq, November 2006
United States Department of Defense (2007). Report to Congress: Measuring Stability and Security in Iraq, March 2007
United States Department of Defense (2007). Report to Congress: Measuring Stability and Security in Iraq, June 2007
United States Department of Defense (2007). Report to Congress: Measuring Stability and Security in Iraq, September 2007
United States Department of Defense (2007). Report to Congress: Measuring Stability and Security in Iraq, December 2007
United States Department of Defense (2008). Report to Congress: Measuring Stability and Security in Iraq, March 2008
United States Department of Defense (2008). Report to Congress: Measuring Stability and Security in Iraq, June 2008
United States House of Representatives (2005). Making Emergency Supplemental Appropriations for the Fiscal Year Ending September 30, 2005, and for Other Purposes, *Conference Report 109-72*, May 2005
Vanderplaats, G. N. (2007). *Multidiscipline Design Optimization,* Vanderplaats R&D, Inc., ISBN 0-944956-04-1, New York, USA

Systems Engineering - Practice and Theory
Edited by Prof. Boris Cogan

ISBN 978-953-51-0322-6
Hard cover, 354 pages
Publisher InTech
Published online 16, March, 2012
Published in print edition March, 2012

The book "Systems Engineering: Practice and Theory" is a collection of articles written by developers and researches from all around the globe. Mostly they present methodologies for separate Systems Engineering processes; others consider issues of adjacent knowledge areas and sub-areas that significantly contribute to systems development, operation, and maintenance. Case studies include aircraft, spacecrafts, and space systems development, post-analysis of data collected during operation of large systems etc. Important issues related to "bottlenecks" of Systems Engineering, such as complexity, reliability, and safety of different kinds of systems, creation, operation and maintenance of services, system-human communication, and management tasks done during system projects are addressed in the collection. This book is for people who are interested in the modern state of the Systems Engineering knowledge area and for systems engineers involved in different activities of the area. Some articles may be a valuable source for university lecturers and students; most of case studies can be directly used in Systems Engineering courses as illustrative materials.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen