Issue of Complexity in Data Fusion Systems.

S.A. Quadri, Ph.D.*and Othman Sidek, Ph.D.

Collaborative Microelectronic Design Excellence Centre (CEDEC),
Universiti Sains Malaysia, Engineering Campus, Nibong Tebal, Pulau Pinang, Malaysia 14300.

E-mail: reachquadri@yahoo.com*

ABSTRACT

Data fusion deals with the synergistic combination of information made available by different measurement sensors, information sources, and decision makers. The process of data acquisition, processing, extracting and inferring decision is a complex task in data fusion system. Although a rich legacy in data fusion technology exists, ranging from various data fusion models to taxonomies of algorithms. The selection of a processing architecture and specific algorithms is still a challenging system-engineering problem. The lack of common engineering standards, well-documented performance evaluation, and architecture paradigms are major rudiments to pioneer notion of complexity in data fusion system. The paper discusses issue of complexity in data fusion system and suggests an engineering approach to reduce the complexity.

(Keywords: data fusion, algorithms, complexity)

INTRODUCTION

Data fusion is a multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats and their significance [1].

Data fusion deals with the synergistic combination of information made available by different measurement sensors, information sources, and decision makers, Thus, sensor fusion is concerned with distributed detection, sensor registration, data association, state estimation, target identification, decision fusion, user interface and database management. It uses many techniques such as least square method, Bayesian method, Dempster- Shafer, Fuzzy logic and neural network etc [2].

The concept of data fusion initially occurred in multi-sensor processing. In the last 10 years or so, data fusion has been used by researchers in the information retrieval area to combine multiple document lists for the same information need. One particular situation is that we use several different information retrieval systems (or several different settings/retrieval strategies in the same system) to retrieve the same collection of documents, then merging these results into a single list for higher effectiveness [3]. The feasibility of this solution mainly depends on whether we can obtain improvement on effectiveness by data fusion.

Implementation of data fusion system is a complex task. Several of the most critical issues related to the implementation are requirement analysis, sensor selection, architecture selection, algorithm selection, software implementation, and testing and evaluation. D.L. Hall has discussed in detail about these issues in [4]. A number of data fusion frameworks have been developed to serve the purpose. On the surface, the notion of data fusion may appear to be straightforward but the design and real time implementation of fusion systems is an extremely complex task. Modeling, processing, fusion and interpretation of diverse sensor data for knowledge assimilation and inference are challenging problems [5]. These problems become even more difficult when the available data is incomplete, inconsistent or imprecise.

In order to satisfy more and more demanding requirements of application, the algorithms are designed and developed. The choice of the most appropriate algorithm depends on the complexity of the target problem, obviously the more complex the problem is, the algorithm also becomes more complex. There is no perfect algorithm that is optimal under all conditions [6].

In spite of the difficulties, research and development effort is being carried out vigorously because of the potential for significantly superior system performance. A group of many incomplete sets of data from many sensors may be fused and lead to useful and unambiguous declarations. This effect obtained from the fusion is called synergy. Synergy increases the complexity of data fusion system. [4, 7, 8].

Today complexity has become an important and, at the same time, one of the most popular notions in science and society. It is a frequent word in present days' scientific literature, in various fields and with diverse meanings, appearing in some contexts as a precise concept, while being a vague idea in other texts. The reason is that people study and create more and more complex systems. This is especially true for such fields as information technology and software development.

As it is written in [9], recently Bill Gates, in a Microsoft internal memorandum, implicitly admitted of past complexity sins and introduced plans to redirect development efforts toward providing better systems. Paul Horn, the IBM Vice President of Research, confessed the complexity sins of the computer industry and proposed an ambitious program for cleansing the sins [10].

Complexity has proved to be an elusive concept. Different researchers in different fields are bringing new philosophical and theoretical tools to deal with complex phenomena in complex systems. In order to reduce the complexity we need a developed theory emphasizing complexity that should explicitly explains why and how complexity emerges and how to solve problems that involve very complex systems. This is especially true for data acquisition and processing system.

The paper discusses issue of complexity in general and finally suggests an engineering approach to reduce complexity in data fusion system.

COMPLEX SYSTEM AND COMPLEXITY

A complex system is any system featuring a large number of interacting components (agents, processes, etc.) whose aggregate activity is nonlinear (not derivable from the summations of the activity of individual components) and typically exhibits hierarchical self-organization under selective pressures. This definition applies to systems from a wide array of scientific disciplines. Indeed, the sciences of complexity are necessarily based on interdisciplinary research [11].

Complexity is emerging as a post-Newtonian paradigm for approaching from a unifying point of view a large body of phenomena occurring in systems constituted by several subunits, at the crossroads of physical, engineering, environmental, life, and human sciences. For a long time the idea prevailed that, the perception of systems of this kind as complex arises from incomplete information, in connection with the presence of a large number of variables and parameters masking the underlying regularities.

Over the years, experimental data and theoretical breakthroughs challenging this view have become available, showing that complexity is on the contrary rooted into the fundamental laws of physics. This realization opens the way to a systematic study of complexity, which constitutes today a highly interdisciplinary, fast growing branch of science, drawing on the cross-fertilization of concepts and tools from nonlinear dynamics, statistical physics, probability and information theories, data analysis, and numerical simulation [12]. Complex systems constitute a privileged interface between mathematical and physical sciences on the one side, and social and economic sciences on the other.

The complexity of a system R is the amount of resources necessary for (used by) a process P that involves R. There are different kinds of system involvement in a process. P may be a process in the system R. For example, R is a computer, P is an electrical process in R, and the resource is energy. In cognitive processes, complexity is closely related to information representing specific kind of information measures. If it is impossible to solve a problem with given resources, we assume that it has infinite complexity with respect to this resource.

The halting problem, being restricted to recursive algorithms, is an example of a problem with infinite complexity since we know that it has no solution. In general, complexity is a relative characteristic, which depends on considered processes and related resources. For instance, there are systems that are simple for usage but complex for study. There are computations that demand little memory (one resource) but take a lot of time (another resource) to finish [13].

The complex systems modeling research area is concerned with basic and applied research on simulations of complex systems and development of applications to understand and control such systems. By complex system we refer to any system featuring a large number of interacting components (agents, processes, etc.) whose aggregate activity is nonlinear (not derivable from the summations of the activity of individual components) and typically exhibits hierarchical self-organization. The research area's focus is on network and multi-agent modeling, as well as computational biology [11].

Computational complexity theory is a branch of the theory of computation in theoretical computer science and mathematics that focuses on classifying computational problems according to their inherent difficulty. In this context, a computational problem is understood to be a task that is in principle amenable to being solved by a computer (which basically means that the problem can be stated by a set of mathematical instructions). Informally, a computational problem consists of problem instances and solutions to these problem instances. For example, primality testing is the problem of determining whether a given number is prime or not. The instances of this problem are natural numbers, and the solution to an instance is yes or no based on whether the number is prime or not [14].

A problem is regarded as inherently difficult if solving the problem requires a large amount of resources, whatever the algorithm used for solving it. The theory formalizes this intuition, by introducing mathematical models of computation to study these problems and quantifying the amount of resources needed to solve them, such as time and storage. Other complexity measures are also used, such as the amount of communication (used in communication complexity), the number of gates in a circuit (used in circuit complexity) and the number of processors (used in parallel computing). One of the roles of computational complexity theory is to determine the practical limits on what computers can and cannot do.

Closely related fields in the theoretical computer science are analysis of algorithms and computability theory. A key distinction between analysis of algorithms and computational complexity theory is that, the former is devoted to analyzing the amount of resources needed by a particular algorithm to solve a problem, whereas the latter asks a more general question about all possible algorithms that could be used to solve the same problem. More precisely, it tries to classify problems that can or cannot be solved with appropriately restricted resources. In turn, imposing restrictions on the available resources is what distinguishes computational complexity from computability theory; the latter theory asks what kind of problems can be solved in principle algorithmically [14].

Best, Worst, and Average Case Complexity

Visualization of the quick sort algorithm that has average case performance O (n log n). The best, worst and average case complexity refer to three different ways of measuring the time complexity (or any other complexity measure) of different inputs of the same size. Since some inputs of size n may be faster to solve than others, we define the following complexities:

- Best-case complexity: This is the complexity of solving the problem for the best input of size n.
- Worst-case complexity: This is the complexity of solving the problem for the worst input of size n.
- Average-case complexity: This is the complexity of solving the problem on an average. This complexity is only defined with respect to a probability distribution over the inputs. For instance, if all inputs of the same size are assumed to be equally likely to appear, the average case complexity can be defined with respect to the uniform distribution over all inputs of size n.

For example, consider the deterministic sorting algorithm quick sort. This solves the problem of sorting a list of integers that is given as the input. The best-case scenario is when the input is already sorted, and the algorithm takes time O (n log n) for such inputs. The worst-case is when the input is sorted in reverse order, and the algorithm takes time O (n2) for this case. If we assume that all possible permutations of the input list are equally likely, the average time taken for sorting is O (n log n)[15].

Theories of Complexity (TOC)

During the last two decades or so, a new field of interdisciplinary research, often referred to as the “science of complexity,” emerged from the interplay of physics, mathematics, biology, economy, engineering, and computer science. Its mission is to overcome the simplifications and idealizations that have lead to unrealistic models in these sciences [16].

One of the most important methods of the science of complexity is the use of a particular kind of computational model, so-called agent-based models (ABM) [17, 18]. Examples include rather abstract models such as Holland’s ECHO [19, 20], artificial stock markets [21], simulations of social systems such as Sugarscape [22], realistic models of social insects [23, 24] or accurate implementations of real road traffic systems such as Transims [25], to name but a few. These and other empirical successes have served as motivation for reflection and debate on the possibilities of developing a body of scientific theory in the extension of these new lines of research. The theoretical issues are, however, far from being resolved. For instance, there is still no generally accepted definition of complexity, despite a vast number of proposed ansatzes [26, 27].

Several authors (particularly Holland [19, 28], but also see Casti [29], Fontana and Ballati [30]) have expressed the feeling that such a TOC would be necessary in order to make the science of complexity more coherent, general and precise; indeed, the search for universal and unifying theories is something of an ideal in most scientific disciplines.

Thus, both with regard to the internal theoretical foundation of the science of complexity and its external use in neighboring sciences, it is important to assess to what extent complex adaptive systems (CAS) CAS can serve as a general approach to complexity.

The three central aims of TOC are [31]:

● Prediction of the future behavior of a system given a set of observational data about it (predictive component).

● Theoretical understanding and/or description of a system (explanatory component).

● Provision of guidelines and control mechanisms for the intervention and manipulation of systems (control component).

Ideally, a scientific theory would explain, predict and facilitate control at the same time. The degree of complexity involved is usually beyond the reach of the conventional methods of physics, but ABMs and other approaches to complex systems, such as neural networks, genetic algorithms, etc. have proven to be powerful methods in this context (Table 1[31]).

Table 1: Degrees of Complexity and Models.

Degrees of Complexity and Models			
Areas of applications	**Examples**	**Features**	**Models**
Policy related issues; when the overall impact of intervention into the world is to be estimated, but also reconstructions of evolutionary development. Others	Lake Victoria	CAS + radical openness + contextuality	??
Real systems, as long as radical openness and contextuality are reducible.	Evolutionary systems, road traffic systems, business simulations, but also the ecological system of Lake Victoria	CAS paradigm (non linearity, adaptive agents, internal in homogeneity, net-like causal structure	ABMs, neural networks, evolutionary computing, etc.
Laboratory systems and limited applicability to real systems	Physics, chemistry, engineering, classical economic theories	Platonist/Galileian paradigm	Linear differential equations, analytic mathematical models

ENGINEERING APPROACH TO REDUCE COMPLEXITY IN DATA FUSION SYSTEM

Literature has plenty of descriptions of data fusion systems which fuse and process target data from various sensors. However, most of those systems are applied to simple idealized and stand alone simulated platforms with simplistic targets and sensors data. Real military platforms have complex and uncommon software and hardware architectures, many interface protocols, and operate in real environments where the noise, the bias or any other source of inconsistency are not always due to target data but also inherent to the design of the system [32]. Most monitoring and fault diagnosis algorithms used in data fusion system are computationally complex, but their results are often needed in real-time [33].

For data fusion in networks with a very large number of sensors, the computational time of the fusion algorithm is strongly dependent on the number of sensors. Hong et al., [34] adapted the spatial decomposition approach [35]. The region of interest will be divided into consecutive small regions. The fusion time for each sub network can be reduced close to n-fold depending on the parallelization overhead due to the n fold reduction of the number of sensors on an n-node cluster. Each slave processor works as a data processing unit collecting and processing data only from the sensors in the corresponding sub region. A master process controls and synchronizes the network target detecting process. At the end of the data fusion process, all the slave processes send their local network state estimate to the master process. The master process collects the observations from each sub region and performs the data association with Nearest Neighbor Data Association (NNDA) [36].

Considering multi sensor data fusion algorithms in aircraft navigation system, in which increase in number of sensors leads to complexity in filtering algorithm, J.A. David [37] has given a detailed description of various algorithms developed in accordance with developing filter architecture. The centralized filter computation can be time-consuming as a result of the large number of states in the dynamic model of the filter. Accordingly, the centralized filter is not necessarily an appropriate methodology in the development of fault tolerant multi sensor navigation systems [38-40]. To overcome the limitations of the centralized filter with respect to complexity it is suggested to opt for other filter architectures namely, Cascaded Filter Architecture (CFA), Federated Filter Architecture (FFA) and Distributed Filter Architectures (DFA). It is asserted that proper selection of filter reduces much of complexity in the system.
In case of data fusion technique for tracking non-maneuvering and maneuvering targets with mobile sensors application, Joy Iong-Zong Chen [41] has proposed an algorithm using gating technique which reduces complexities as compared with many traditional algorithms [42-46] proposed earlier. The gating technique is applied to solve the problem of MSDFT (mobile-sensor data fusion tracking) for targets; the simple approach is implemented with an adaptive Kalman filter consisting of a data association technique denoted as CML (conditional maximum likelihood).

An exciting and substantial area of research in the computer field is the computational complexity of an algorithm. Chakraborty and Choudhury [47] have presented a comparison of mathematical method of analysis for finding out the complexity of an algorithm with the statistical counterpart. Developing an algorithm with the least asymptotic execution time is one aspect while another offshoot is the minimum number of operations required to compute a given function.

The customary practice is to express the order of complexity of an algorithm by simply computing the minimum number of operations required (and likewise, the average complexity by the number of operations required on an average) and this is expressed in terms of the input parameter(s). In case the operations are of the same type, there is no problem. In case they are different, one can argue that simply counting the total number of operations will not suffice unless each specific operation is weighed against the corresponding execution time it consumes. In such a situation, letting ti as the execution time for the i^{th} operation, the point of interest is observing $T = \sum iti$ (summed over all operation) by varying the input parameter(s) and then set a functional relationship between T (i.e., total execution time) and the input parameter(s) using standard statistical methods.

Thus it seems proper selection/development of model and algorithm for data fusion system are prominent factor to reduce complexity. The load on CPU, memory allocation and storage, inter process communication and finally computation time is all dependent on algorithmic complexity.

The algorithm should use simpler communication interfaces and abstraction to enhance the throughput and reduce the delay. Algorithm should allow real time operation without complex global state maintenance. The necessity of adding extra hard ware (processors and memory) to improve computation efficiency can be bypassed by a better engineering approach applied to data fusion algorithms which will explicitly manage with the limited resources of the system.

Modeling the Problem and Algorithm Design

Developing or selecting an appropriate model can be crucial for the overall success of the data fusion system. Modeling of the problem is done after thorough understanding of requirement analysis. The requirement analysis must consider the effects of the observing environment, the end user, the platform (on which the sensors are located), communication constraints, and computing limitations. Key issues include the observing and decision timeline and required level of specificity and accuracy.

The model defines relationships between the sources of data and the types of processing that might be carried out to extract the maximum possible information from it. In essence, modeling is postulating assumptions how real world behaves. For complex applications, this modeling phase is a non-trivial and highly demanding task. Operations research (OR) and mathematical programming (MP) disciplines can be used to model the problem. The problem of selecting an appropriate type of sensor for required specific application is an important issue related to overall performance of data fusion. Optimal number of sensors can lead to reduced maintenance costs and the creation of compact online databases for future use.

A key component of the system development process is to understand how the sensors perform, both individually and in concert, to contribute to inferences sought by the data fusion system. The system designer should be able to develop or utilize high fidelity models that predict how sensors will perform in realistic environments. These models should include the effects of the target, the signal propagation environment, the location of the sensor antenna on an observing platform, and the internal sensor processing. Architecture selection is basically a system engineering problem. The system designer at this stage should explicitly consider alternative architectures for the fusion system to be designed. Each alternative can be modeled to evaluate the system effectiveness and the demands on system resources such as computing and communications.

Perhaps the most controversial issue in data fusion is the selection of algorithms/techniques. There are many algorithms that can be applied to the different process within the data fusion process. As commented by D.L. Hall and A.K. Garga [6], there is no such thing as a magic or golden data fusion algorithm. There is no perfect algorithm that is optimal under all conditions. While this statement may seem obvious, there continue to be controversies in the data fusion literature, about which algorithm is best, optimal, or robust.

One of the obvious observations that one can arrive at after looking at the techniques used for fusion is that all of them are known techniques and none of them has been developed by the sensor data fusion research community. Obviously, using old techniques is not a crime. The choice of a set of algorithms for a fusion system must be based on a system's engineering approach. The designer must have a clear understanding of the algorithms (including the underlying assumptions, required a priori data, etc.) the processing constraints of the fusion system, and the limitations in the observing environment. In some cases sophisticated algorithms may be mathematically appealing. But cannot be effectively used because the requisite *a priori* data is not available.

For many applications, multiple techniques are required for the fusion process. Techniques for fusion are drawn from disciplines such as signal and image processing (for characterization and processing of single sensor data), statistical estimation and pattern recognition, and decision-level processing methods from the domain of artificial intelligence. The selection of a processing architecture and specific algorithms is a systems engineering problem, dependent upon a number of factors such as specification application, types of sensors, computing resources available, communication bandwidth available, and many other factors.

The algorithm development is based on successful construction of equivalent platform and global models. The algorithm should be designed such that to achieve maximum time and space efficiency exploiting algorithm engineering guidelines. The algorithm should use simpler communication interfaces. Algorithm should allow real time operation without complex global state maintenance. Simplicity of an algorithm has positive impact on its applicability. The aspects of simplicity, scalability and robustness should be considered with prominence. An efficient algorithm developed will manage the data from various sensors in a robust and logical manner. Implementation of a data fusion system generally involves development of an extensive set of software. It is the lowest level. It concerns coding the outcome of the algorithm design phase in the chosen programming language. Preserving correctness in the implementation phase by program testing, debugging, checking, and verification are important tasks.

CONCLUSION

Data fusion is a challenging, mathematically complex, inter-disciplinary research problem. Significant advances have been made by domain researchers in signal processing, artificial intelligence, and data mining to develop methods, and algorithms, for multi-sensor data fusion. However, significant challenges remain; one such task is to reduce complexity in data fusion system without forfeiting the performance. Complexity has proved to be an elusive concept. Different researchers in different fields are bringing new philosophical and theoretical tools to deal with complex phenomena in complex systems. Currently, the interactions between different communities, namely mathematics, computer science, and physics are still weak. The intellectual challenge motivating is the limited scope of available approaches to study complex system behavior. Construction of theories for complex system is a big venture. There exists a need for new and complementary approaches to study complexity with universality approach.

REFERENCES

1. US Department of Defense. 1991. "Data Fusion Lexicon". Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3 (F.E. White, Code 4202, NOSC) San Diego, CA.

2. Kokar, M. and K. Kim. 1993. "Review of Multisensor Data Fusion Architectures and Techniques". Proceeding of the 1993 International Symposium on Intelligent Control: Chicago, IL.

3. Wu, S. and S. McClean. 2006. "Performance Prediction of Data Fusion for Information Retrieval". *Information Processing and Management Journal*. 42(4):899-915.

4. Hall, D.L. 2000. "Implementation of Data Fusion System". *Proceedings of NATO Advance Research Workshop on Multisensor Data Fusion*. Pitlochry, Perthshire, Scotland, June 25-July 7, 2000.

5. Varshney, P.K. 2000. "Multisensor Data Fusion". *Lecture Notes in Computer Science*. 1821/2000:103-186, DOI: 10.1007/3-540-45049-1_1.

6. Hall, D.L. and A.K. Garga. 1999. "Pitfalls in Data Fusion". In: *Proceedings of Fusion '99*. Sunnyvale, CA.

7. Llinas, J. and D.L. Hall. 1998. "An Introduction to Multi-Sensor Data Fusion". *Proceedings of IEEE International Symposium on Circuits and Systems*. 6:537-540.

8. Luo, R. and M. Kay. 1998. "Multisensor Integration and Fusion: Issues and Approaches". *SPIE Sensor Fusion*. 931:42-49.

9. Lawson. H.W. 2002. "Rebirth of Computer Industry". *Communication ACM*. 45:25–29.

10. Horn, P. 2001. "Announcement of the Autonomous Computing Program". www.research.ibm.com/autonomic

11. Rocha, L.M. 1999. *BITS: Computer and Communications News*. Computing, Information, and Communications Division. Los Alamos National Laboratory. November 1999. http://informatics.indiana.edu/rocha/complex/csm.html#N_1_

12. Website: http://www.scholarpedia.org/article/Complex_systems

13. Burgin, M. 2004. "Algorithmic Complexity of Recursive and Inductive Algorithms". *Theoretical Computer Science*. 317(1/3):31-60.

14. Website: http://en.wikipedia.org/wiki/Computational_complexity_theory

15. Szirmay-Kalos. L. and G. Marton. 2000. "Worst-Case Versus Average Case Complexity of Ray-Shooting". *Budapest University of Technology and Economics Report*. www.fsz.bme.hu/~szirmay/ray6.pdf

16. Bak, P. 1997. *How Nature Works*. Oxford University Press: New York, NY.

17. Epstein, J. 1999. “Agent-Based Computational Models and Generative Social Science”. *Complexity*. 4(5):41–60.

18. Casti, J. 1997. *Would-Be Worlds*. John Wiley & Sons: New York, NY. ISBN 0-471-19693-2.

19. Holland, J. 1995. *Hidden Order*. Addison-Wesley: Reading, MA. ISBN 978-0738201429.

20. Hraber. P, Jones. T, and Forrest, S. 1997. “The Ecology of ECHO”. *Artificial Life*, 3:165–190.

21. LeBaron, A.W., B. Palmer, and R. Taylor. 1997. “Asset Pricing under Endogenous Expectations in an Artificial Stock Market”. In: *Economy as an Evolving Complex System II, vol. XXVII of Studies in the Sciences of Complexity*. A.W. Durlaf and S. Lane (eds.). Santa Fe Institute, Addison-Wesley: Reading, MA.

22. Epstein, J. and R. Axtell. 1998. *Growing Artificial Societies*. MIT Press: Cambridge, MA. ISBN 978-0-262-55025-3.

23. Bonabeau, E., G. Theraulaz, and M. Dorigo. 1997. “Self-Organization in Social Insects”. April 1997. Santa Fe Institute Working Paper 97-04-032.

24. Bonabeau, E and G. Theraulaz. 1999. “Swarm Intelligence: From Natural to Artificial Systems”. *SFI Studies in the Sciences of Complexity*. Oxford University Press: Oxford, UK.

25. Website: http://transims.tsasa.lanl.gov/ accessed 2002.

26. Rescher, N. 1998. “Complexity: A Philosophical Overview”. *NJ Transaction*. New Brunswick, NJ.

27. Edmonds, B. 1999. “Syntactic Measures of Complexity”. Ph.D. Thesis, University of Manchester. Available at http://www.cpm.mmu.ac.uk/bruce/thesis.

28. Holland, J H. 1998. *Emergence: From Chaos To Order*. Oxford University Press: Oxford, UK. ISBN-10: 9780738201429.

29. Casti, J. 1999. “The Computer as a Laboratory”. *Complexity*. 4(5):12–14.

30. Fontana, W. and Ballati, S. 1999. “Complexity—Why the Sudden Fuss”. *Complexity*. 4(3):14–15.

31. Chu, C., R. Strand, and R. Fjelland. 2003. “Theories of Complexity”. *Complexity*. 8(3):19-25.

32. Simard, M.A. and J.C. Eloi Bossé. 1996. “Data Fusion of Multiple Sensors Attribute Information for Target Identity Estimation Using a Dempster-Shafer Evidential Combination Algorithm”. *Proceedings of SPIE, the International Society for Optical Engineering*. 2759:577-588.

33. SengNg, Y. and R. Srinivasan. 2010. “Multi-Agent Based Collaborative Fault Detection and Identification in Chemical Processes”. *Engineering Applications of Artificial Intelligence*. 23:934–949.

34. Lin, H., J. Rushing, S. Graves, and E. Criswell. 2007. “A Data Fusion Algorithm for Large Heterogeneous Sensor Networks”. *Proceedings of International Conference on Wireless Algorithms, Systems and Applications*. Chicago, IL.

35. Pedersen, C. 2001. “Architecture and Performance of a Parallel Data-Fusion Multi-Target Tracking Code for Real-Time Applications”. *Software Tech News*, 4(1), High Performance Computing, www.softwaretechnews.com/stn4-1/datafusion.html

36. Klein, L.A. 2004. *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making*. SPIE Press: Bellingham, WA.

37. Allerton, D.J. and H. Jia. 2005. “A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems”. *The Journal of Navigation*. 58:405–417.

38. Lawrence, P.J. and M.P. Berarducci. 1994. “Comparison of Federated and Centralized Kalman Filters with Fault Detection Considerations”. *Proceedings of IEEE PLANS*. 703–710.

39. Hashemipour, H.R., S. Roy, and A.J. Laub.1988. “Decentralized Structure for Parallel Kalman Filtering”. *IEEE Transactions on Automatic Control*. 33(1):88–94.

40. Roy. S., R.H. Hashemi, and A.J. Laub. 1991. “Square Root Parallel Kalman Filtering using Reduced Order Local Filter”. *IEEE Transactions on Aerospace and Electronic Systems*. 21(2):276–289.

41. Joy Iong-Zong Chen. 2009. “An Algorithm of Mobile Sensors Data Fusion Tracking for Wireless Sensor Networks”. *Wireless Personal Communications* www.springerlink.com/content/274rt781753356xm/fulltext.pdf

42. Chung, Y.N. and Chen. J I -Z. 1997. “Applying both Kinematic and Attribute Information for a Target Tracking Algorithm”. *Journal of Control Systems and Technology*. 5(3):203–209.

43. Rezaei, S. and R. Sengupta. 2007. “Kalman Filter-Based Integration of DGPS and Vehicle Sensors for Localization”. *IEEE Transactions on Control Systems Technology*. 15(6):1080–1088.

44. Honarbacht, A., P. Rauschert, and A. Kummert. 2004. "Data Fusion in Vireless Sensor Networks: A New Application for Kalman Filtering". *Proceedings of IEEE conference, TENCON 2004*. Thailand. 511–514.

45. Li, S., D. Zhang, Z. Yang, and N. Chang. 2006. "A Novel Algorithm for Node Localization and Motion Analysis in Wireless Sensor Networks". *Proceeding of the 2006 IEEE International Conference on Auto Science and Engineering*. Shanghai, China. 574–579.

46. Wang, X., D. Wang, D.P. Agrawal, and A. Mishra. 2007. "On Data Fusion and Lifetime Constrains in Wireless Sensor Networks". *IEEE Proceedings in ICC 2007*. Glasgow, Scotland. 3942–3947.

47. Chakraborty, S. and P. Choudhury. 2000. "Statistical Analysis of an Algorithm's Complexity". *Applied Mathematics Letters*.13:121-126.

ABOUT THE AUTHORS

S.A. Quadri, is a Ph.D. student in CEDEC, USM, Malaysia; he graduated B.E. (Electronic and Communication) from Karnataka University Dharwad, completed M.Tech. in Computer Science and Engineering from V.T.U. Belgaum. His research domain is Multi sensor data fusion in various engineering applications.

Associate Professor Dr. Othman Sidek, holds a Ph.D. (Info. Sys. Eng.) from Bradford University in England. He has served Universiti Sains Malaysia since 1984 as a Lecturer, Researcher, and Administrator. Prior to joining USM, he was a trainee engineer with Multinational Companies and continued to take very active role in both academic and university-industry research collaborations. His current research areas mainly focus on Micro-Electro Mechanical Systems (MEMS) and Wireless Sensor Network, Embedded System/SOC and VLSI/ASIC Design. All these are the focused areas of CEDEC. He is also the Founder and Director of the Collaborative Micro-electronic Design Excellence Centre (CEDEC).

SUGGESTED CITATION

Quardi, S.A. and O. Sidek. 2012. "Issue of Complexity in Data Fusion Systems". *Pacific Journal of Science and Technology*. 13(1):252-260.