Optimal Sensor Scheduling and Power Management in Sensor Networks

Ashit Talukder*, Syed-Muhammad Ali, Anand Panangadan, Chetan Jadhav, Rishi Pidva, Rajan Kumar Bhatt, L Chandramouli, and S. Monacos*
CHLA/University of Southern California
**Joint affiliation with Jet Propulsion Laboratory/NASA*
Email: talukder@usc.edu, Ph: 818 354 1000

ABSTRACT

We discuss a novel control methodology for power management in heterogeneous distributed sensor networks. Many algorithms for resource management in sensor networks require a comprehensive model of the external environment and the sensor network system, and are rule-based; this restricts their use in dynamic environments. We present an event-based control optimization formulation of the resource management problem and discuss a method to adaptively change desired system performance of the sensor network in response to events. This functionality is critical in field-deployable sensor networks where the available energy is extremely limited. This limitation disallows continuous operation as a very expensive option and necessitates system adaptation as a means to extend operational lifetime in the face of dynamic external events. We show results on synthetic sensor networks where only partially accurate information about the external world and the sensing system is available and illustrate the efficacy of the control algorithm in handling dynamic events with guaranteed minimum system lifespan via efficient usage of energy resources. We show that the control algorithm makes effective control decisions about the use of energy and storage resources with varying sensor reliabilities.

***Index Terms*—Evolutionary algorithm, control, distributed sensing, machine learning, sensor networks, pattern recognition, resource management, mathematical optimization, wireless communication.**

1 INTRODUCTION

Energy constrained mobile and static sensor networks lend themselves to a number of challenges that are normally not encountered in offline data processing or during the processing of signals that have access to lots of power, computing and transmission resources. The power usage and the communication load in a energy constrained distributed sensor network is highly dependent on the signal processing strategies used to process the observed data and on the resource management strategies used to control critical operations that consume power and use up available bandwidth. An often-ignored issue in sensor networks is the fact that the real-world being observed is dynamic in nature and varies over time. Therefore, ideal detection of dynamic external events in the real-world requires **truly adaptive** operation of multiple sensors in the sensor network.

Different sensors in a heterogeneous sensor network have varying sensitivities and specificities to various external events. A sensor(s) may have a high sensitivity and specificity for event A, but may be ill-suited to detect event B; an ECG measurement, for example, effectively detects cardiac anomalies but is not suitable for detection of diabetes, which is easy with an glucose sensor/detector. Therefore, effective management of sensors and network resources is critical for detection of a variety of events with energy constrained mobile/static heterogeneous sensor networks. Past work on bio-medical sensor networks [1], smart clothing [2, 3], target detection and surveillance [4], and environment and health monitoring [5] has focused primarily on communication issues and data logging. The inter-disciplinary nature of this problem poses several hurdles to be overcome towards the ultimate goal of autonomous extended monitoring. The primary challenges which need to be addressed in autonomous sensor

Optical Pattern Recognition XVI, edited by David P. Casasent, Tien-Hsin Chao,
Proceedings of SPIE Vol. 5816 (SPIE, Bellingham, WA, 2005)
0277-786X/05/$15 doi: 10.1117/12.607977

networks to achieve true autonomy are: i) Data fusion and distributed estimation arising from heterogeneous sensors, and ii) Resource management due to limited resources of the mobile system. [6] focuses on the issue of data fusion and distributed estimation for a heterogeneous sensor network, but do not consider the limitation on resources (power, storage and computation capabilities). In our previous work, we have discussed feature extraction, data classification and data fusion algorithms for event detection in sensor networks [7, 8].

In our current work in MUSIC (Multi-modality Sensor network for Integrated event detection, Control optimization and resource management), we concentrate on the resource management algorithms to control operation of multiple distributed sensors in an optimal and coordinated fashion. We discuss a novel resource management technique for mobile distributed sensing platforms that allows monitoring and detection of environmental events for extended periods of time via dynamic adaptation of sensor network operational parameters in response to detected events. Power is the most critical resource in a sensor network. Power is needed to operate each sensor, run the computing unit to process measured sensory data, and power is absolutely critical for wireless transmission of any raw and processed data. Lowering the operating frequency lowers power consumption in three ways: (1) it lowers energy consumption in operating each sensor, (2) it reduces the data size to be processed in-situ, and thereby saves on power used by the CPU on the sensor, and (3) it reduces the data size during transmission and therefore uses less power during communication. Therefore, our focus in this paper is on power management in the sensor network by optimal coordinated scheduling of sensor operations. Storage is another limiting factor in wearable devices. In order to ensure minimal use of storage resources and guarantee a minimum system lifetime without running out of power or storage capacity, we make real-time store/discard decisions on sensory measurements based on sensor reliability models, observed events, and current system state.

The applications that we target range from unmediated data monitoring and record keeping of the environment to the health of a patient and handicapped individuals in a hospital environment. Figure 1 shows our mobile health monitoring using body wearable medical sensors. In mobile health monitoring, the simplest solution that has been employed up until now has been to run the system at a maximum possible frequency, capture all sensory data, and transmit all raw data to a central server. However, limited power, storage and bandwidth alongwith minimum desired operational lifetime makes such a simplistic solution infeasible. For example, our experiments have indicated that continuous transmission of data reduces the lifetime of a handheld by a factor of six, compared to continuous on-board data processing. Additionally, when a connection with the central server is unavailable for long periods of time, blind data logging becomes impractical. This illustrates the usefulness of managing resources, namely power and bandwidth, in a mobile sensing system.

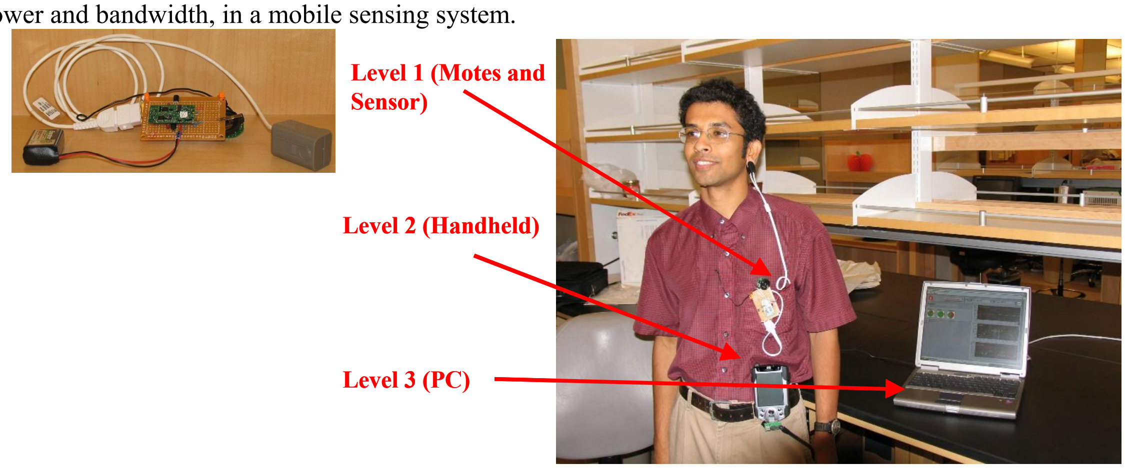


Figure 1 The CHLA/USC MUSIC autonomous health monitoring hardware

We consider sensor networks where the sensors and computational devices can be toggled between an energy conserving "idle" and energy consuming "active" state. Therefore, the operational frequency of each device can be actively controlled in real-time. From an application and end-user perspective, running the system at its maximum possible frequency is not desirable when the external environment (greenhouse, battlefield) or human (health monitoring) is in normal condition since the information collected during such times is not critical. The limited resources should be conserved and used more at time periods when events are observed through the sensors (e.g.

when the external environment or the health of patient deteriorates).

While a plethora of control optimization and resource management algorithms have been developed in the past, in diverse fields ranging from physical model-based control [9], to fuzzy control [10], reinforcement learning-based control [11], to agent-based control with roots in Artificial Intelligence [12], no single solution is suited to handle multi-parameter, multi-variable distributed systems with multiple conflicting objectives, and run with limited computational resources of a real-time system. We discuss a new multi-objective genetic algorithm (MOGA) formulation to handle the operational parameters of the distributed sensor network in response to dynamic events in the observed environment. The primary advantage of the MOGA is that it is capable of handling multiple conflicting objectives and constraints in the distributed sensing system, and a variety of variables types (integer, binary, and real-valued) that are present in our MUSIC system model. Our MOGA-based control ensures that optimal information about the external environment being observed is stored and conveyed back to the base station while maintaining an extended lifespan for the entire distributed sensor network. Novel algorithmic additions to the MOGA operations ensures an order of magnitude speedup in convergence speed without compromising accuracy.

The paper is organized as follows. In section 2, we give a brief overview of our MUSIC wireless sensor network. Our pattern recognition algorithms for event detection are left out to maintain the technical focus of this paper. The reader is encouraged to read our previous publications on pattern recognition for sensor networks [7, 8]. We give an introduction to our control optimization formulation for MUSIC resource management and our MOGA control optimization solution for optimal control in dynamic environments in Section 3. We then discuss our results for specific case-studies in Section 4.

2 MUSIC ARCHITECTURE

2.1 System Overview

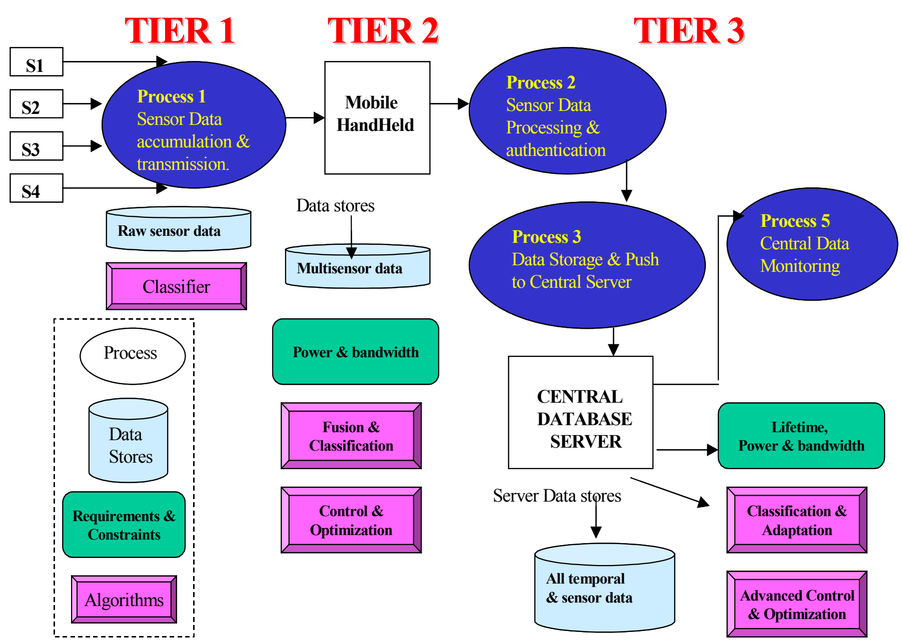


Figure 2 Mobile Distributed Sensing Concept

Figure 2 represents the key components of our MUSIC system. Our hierarchical architecture is built to handle a diverse set of heterogeneous sensors; this is in contrast to a few of the other of the sensor network architectures that use only single sensor types [4]. Our architecture is built for short-range communication between levels 1 and 2, and long-range communication between levels 2 and 3. At the lowest level in our sensor network, each (heterogeneous) sensor is connected to a wireless RF transmitter with basic computational capabilities. Event detection and data prioritization using simple pattern recognition techniques are implemented at this level. In level 2, specific data received wirelessly from several nodes in level 1 (e.g. sensors in one room for environmental monitoring) are fused using a mixture of experts classifier (whose description is provided in another paper). The control optimization algorithm at this level manages the resources for itself and all nodes under its control. Only critical data is then transmitted back to the central server and data storage point (Level 3) for off-line analysis. A brief description of each level in our intelligent sensor network is provided next.

2.1.1 Level 1

We use the UC Berkeley Mica2dot [13] with a 8 bit Atmega128L microprocessor device with inbuilt transceiver as the local data storage, data processing and transmitting device in Level 1. The primary advantage of the Mica2dot is its small form-factor (size of a US quarter and larger than a penny) and its extremely low power consumption (lasts for 4 days with 100% CPU usage, and 40 days with a 10% duty cycle). This enables its use in field-deployable sensor networks with disposable nodes with a lifetime of weeks and months (assuming a low duty cycle). The mica2dot also has 128Kb of RAM and 512 Kb of EEPROM which can be used for data storage. This allows us to build in the basic “intelligence’ in the mica2dot to command the sensor (shut down/ increase frequency, etc.), process the sensor data, classify data using simple pattern recognition methods, and prioritize the data for transmission.

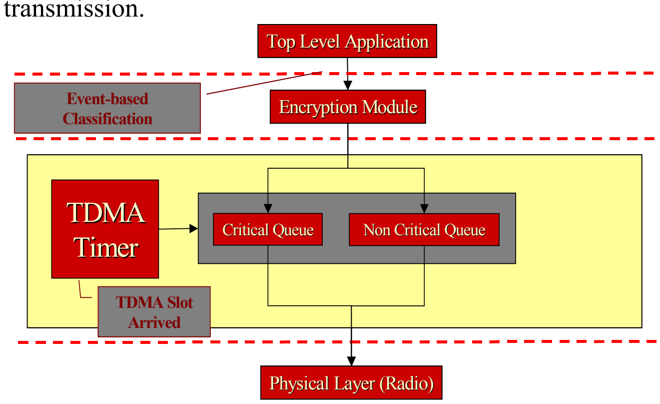


Figure 3 : Data Prioritization and Transmission at Level 1

We have implemented a dynamic TDMA scheduling mechanism for communication in MUSIC. The scheme relies on the level 2 node for initial configuration which is assumed to have information about the level 1 nodes in network. The level 2 node broadcasts a synchronization message during the system startup time which along with other information contains a fixed time slot length and schedule information for each node. After setup, each node only transmits during its own time slot. Our TDMA implementation is flexible such that one node can be assigned more than one slot on the basis of amount of data it has to send. Also it is dynamic in nature such that the length of time slots and even the transmission schedule of nodes can be changed adaptively during runtime. This is achieved through assigning command slots in each transmission schedule. During these command slots all motes are in receive mode and listen for any commands (such as increasing unit time slot length, assigning extra slots to a particular node etc.) from the level 2 node. We maintain a 20% communication duty cycle in MUSIC, thereby allowing 80% reallocation when new nodes are added, or rapid event reporting when drastic events occur.

2.1.2 Level 2

Level 2 serves as the local master and controller for a sub-group of sensors in Level 1, and communicates with other Level 2 controllers and the central server in Level 3. It has more computational resources than Level 1 nodes. The main level 2 component in our sensor network is a handheld device that acts like an interface to the 802.11b network (if present) and also a central controller for the network of level 1 sensors. The handheld in our system is an IPAQ 5450, and runs the PocketPC 2002 operating system. This handheld was selected over others because of its integrated 802.11 NIC, Biometric device for limiting unauthorized access, 64 MB of RAM for data storage, and battery expansion modules for longevity of operation. In-house experimental results have given the battery life to be in the range of 2.5-3 hours with the NIC always on, and 7.5-8 hours with sparing use of the NIC. The IPAQ also provides a 128-bit wireless encryption for secure transmission.

We use LabVIEW as the base platform for all software on the IPAQ. LabVIEW facilitates the development of serial interfaces, and seamless communication over TCP/IP. It allows interface to external code by means of which the control algorithm can be incorporated.

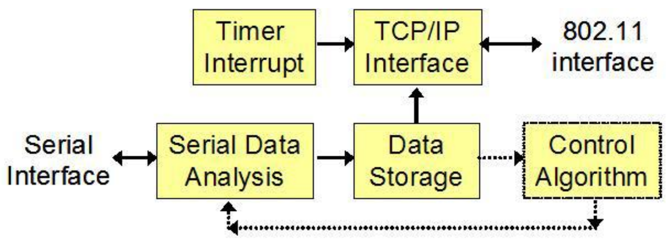


Figure 4 Functional Components of Level 2 of MUSIC

The handheld is connected to a single Mica2Dot that acts as a master mote and all the sensor motes are considered as slave motes. The master mote is connected by a serial interface to the handheld.

The handheld continuously monitors its serial port for incoming activity. It records raw data and classified data from level 1 nodes. The packets therefore contain the following: sensor ID of the slave mote that has sent the data, the criticality of the data, the time instance of data collection and raw measured data. A local copy of this record is maintained and will be provided to the control algorithm. At regular intervals the system checks for a TCP/IP connection. In the presence of wireless LAN, the local data on the handheld is transmitted to a central server.

The handheld is also responsible for sending the initialization and synchronization commands to the distributed network. On system start, an initialization is carried out specifying the initial time slots assigned to each sensor. Similarly, a re--initialization would be necessary to handle timer overflow, and would be incorporated into a handheld control. Also the validity of the data packets being sent is done by the handled at this level. The integrity test involves determination of the exact start and end of the individual packets.

Our general setup allows for peer-to-peer communication amongst groups of Level 2 devices (handhelds). This flexibility enables its use in a variety of domains, where sensing across large spatial locations is desired (1-100's of kilometers).

2.1.3 Level 3

Presently the level 3 of the system architecture is a PC server that handles data collection. The collected data is classified according to the sensor that sends it. It also is recorded into individual files for data analysis at a later stage or for getting the flow of events over a period of time.

The data arriving at the central server is tagged as a critical event or no event depending upon the pattern recognition algorithms in level 1. The percentage of data tagged as critical event is displayed as a pie-chart for every sensor. Similarly visual indicators are provided to notify the occurrence of critical events. The flow of the recorded data is also displayed against their corresponding occurrence time.

3 MULTI-OBJECTIVE GENETIC ALGORITHM FOR OPTIMAL CONTROL

As stated earlier, we will concentrate on the autonomous resource management and control aspect of the MUSIC system in this paper. The goal of control optimization is to maximize the information extracted from the sensors and the information stored while minimizing the Power and Storage costs. It turns out that the multiple objectives to be achieved often conflict and thus make the problem difficult to handle. Physical control-based methods such as PID control, and predictive control solutions typically assume that a linear model exists of the entire system, which may be hard to derive in complex distributed systems; additionally, linear non-predictive controllers fail when system conditions change rapidly. Different AI-based and machine learning techniques such as Neural Networks, Reinforcement Learning, etc. can be used to address the dynamic control and resource management issue; however, neural nets suffer from slow convergence and adaptation in dynamic environments. Additionally, they cannot explicitly handle different types of system variables and multiple functions to maximize/minimize at the same time. We first discuss the general problem formulation for MUSIC system control, followed by our MOGA algorithm for MUSIC control optimization.

3.1 Problem Formulation

A critical requirement in our distributed sensing system is the ability to detect and track important dynamic events while ensuring that certain system constraints are not compromised. This calls for an efficient dynamic control mechanism that takes into account all system resources, constraints, and measured/detected external events (derived from the sensor measurements) in order to control the system operation in real-time.

Typically, real-world control optimization problems are characterized by constraints, multiple objectives, and dynamic changes in external and internal conditions. We represent the dynamic (time-dependent) interaction between the controller, the observed process, and the controlled system as shown in Figure 1. For this paper, we consider the problem of determining the operational frequency of each sensor in the network and deciding on whether data should be stored/discarded, given predefined system parameters, current conditions of dynamic state variables, and detected events by the pattern recognition system. Together, they can be combined with the state/control/environment variables that are specific to the problem being considered to form a utility profit-loss function that should be optimized dynamically at each time step. In this paper, we investigate control optimization strategies in profit maximization. Profit maximization involves measures that maximize a desired objective while minimizing the loss incurred in using system and environmental resources. The profit is a function of variables in the controller and in the controlled system.

The objective function can be dependent on several internal system and external environmental observed variables. In MUSIC, the objective is to maximize the information extracted (measured) and conveyed back to the end-user, while minimizing the usage of resources in the system, such as power, computational resources, and storage. Therefore, we devise a profit measure for the information measured from the sensors and a cost for deriving/extracting this information about the external environment. The profit of information measured is actually a function of the event occurring at a given time; measurement of highly critical events have greater information utility than ones that are benign/normal. A larger number of measurements (higher sensor operational frequency) also increases the information utility/profit. The information utility/profit is also a function of sensor/classifier reliability, and other internal variables. The information loss or cost function expresses the cost incurred in trying to measure more information about the environment (larger sensor operational frequency incurs larger cost). We have a similar set of profit/utility and cost/loss functions for storing (or discarding) sensor measurements; both are functions of available storage, event label, sensor data size, etc. This yields a set of profit and loss functions for various system requirements. All these profit (loss) functions should be maximized (minimized) simultaneously, while satisfying all system constraints. While these multiple profit/loss functions can be combined to form a single objective function using a weighted sum of individual functions, such a simplistic solution is highly sensitive to the choice of weights which are typically chosen in an ad-hoc manner through trial and error. Instead, a more elegant and rigorous solution is to attempt to simultaneously maximize and minimize each individual function simultaneously while solving for the system constraints. We discuss our solution to this problem next.

3.2 Multi Criteria Optimization and Pareto Optimality

Let us consider a problem where m conflicting objectives exist that should be simultaneously maximized, and are denoted by F_i (i = 1,2…m). We can represent these multiple objectives as an objective function vector given as follows.

$F(x) = [F_1(x)\ F_2(x)\ F_3(x)\ \ldots\ F_m(x)]^T$ where $x = [x_1\ x_2 \ldots\ x_n]^T$

As the criteria conflict, no solution is guaranteed to be optimal for m different criteria simultaneously.

A vector x is said to be *Pareto-optimal* or *non-dominated* if and only if there exists no other solution x* such that $F_i(x^*) \geq F_i(x)$ for i = 1,2,3… n and $F_j(x) > F_j(x^*)$ for atleast one j [14].

3.3 Multi Objective Genetic Algorithm

Genetic algorithms are adaptive search techniques based on the principle of natural genetics and selection. They operate on a population of individuals (solution set), initially drawn at random. If the constraints change and make one solution infeasible, another solution existing in the population will still be feasible. Thus, this procedure provides the potential for a diversity of approaches to problem solving. They are also characterized by fast convergence to the global optima [15] if the algorithmic steps of crossover and mutation are correctly implemented.

We propose a Multi- objective Genetic algorithm (MOGA) based on the principle of Pareto-Optimality. In this approach, a population set is evolved at random and another set called the archive is filled with the non-dominated solutions from the population. One solution is chosen from both the sets, and two offspring solutions are created. The procedure then divides the objective space into a number of hyper-boxes with the value of epsilon defined for each of the objectives. Each offspring and the archive member are then assigned an identification array using the value of epsilon. The solution with the distinct value of the identification array is retained in each hyper-box thus ensuring that each box is occupied by only one solution. The archive and population sets are then updated (as discussed below in MOGA operators) and the procedure is repeated for a number of generations [16, 17]. The MOGA algorithm is illustrated in Figure 5

In each generation, the epsilon value limits the number of solutions in the archive by placing only .one solution in each hyper-box. **The number of solutions is thus bounded when the procedure is repeated for a specified number of generations, thereby drastically reducing the overall computation time**. We have noted a speedup by an order of magnitude using this bounded search over other GA algorithms. We report on the execution time for a single optimization step in the results section.

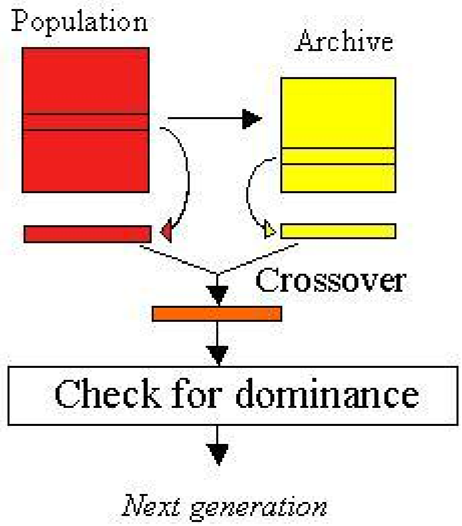


Figure 5 Illustration of multi-objective Genetic Algorithm

MOGA Operators

a) Simulated binary crossover (real variables)

In this step, two parents are selected and used to create two offsprings. Offspring solutions are symmetrically placed around parents defined by the distribution index. For two parents with variables x_i^1 and x_i^2, y_i^1 and y_i^2 offsprings are produced using a probability distribution given as follows.

$$P(\beta) = \begin{cases} 0.5(\eta+1)\beta^{\eta} & \text{if } \beta \le 1 \\ 0.5(\eta+1)/\beta^{\eta+2} & \text{otherwise} \end{cases}, \qquad \beta = \left|\frac{y_i^1 - y_i^2}{x_i^1 - x_i^2}\right| \quad (5)$$

In the first step, a random number is chosen between 0 and 1. Then the value of β is calculated using (5) and the child solutions are computed. A large value of η sets a higher probability for creating near- parent solutions [15].

b) Uniform crossover and mutation (binary variables)

In uniform crossover, the *Crossover mask* bit decides the bit selection from parents. If the mask bit is 1, the bit of parent 1 is chosen. Otherwise, the bit of parent 2 is chosen. Mutation flips a bit in the offspring with a very low probability.

4 RESULTS

We first discuss our general solution for MUSIC system control of sensor operational frequency and dynamic sensor data storage/discard decisions followed by results of the control algorithm on simulated data, where we assume we have class/event label information from a pattern recognition algorithm and a-priori information about the reliability of a sensor and its classifier. Our pattern recognition techniques in MUSIC for autonomous event detection are discussed in another publication [8] and left out of this paper due to lack space.

4.1 MUSIC Control Optimization

Consider a system with n (s = 1,2…n) heterogeneous sensors each with an associated pattern recognition classifier. We assume that the classifier associated with sensor *s* is capable of processing the raw sensor data and reporting/estimating the criticality of an event by a statistical estimate EC_s that ranges from 0 (not critical) to 1 (highly critical). An a-priori model exists for the reliability or confidence IPC_s of the classifier associated with sensor *s* in detecting events; this a-priori value ranges from 1 (highly reliably) to 0.5 (ambiguous reliability). To summarize, Event Criticality EC_s is the measure of the class label criticality i.e., benign/ dangerous nature of the event and Confidence/Reliability IPC_s is the measure of the reliability of a classified label from measurements in sensor *s* i.e., confidence of the classifier associated with sensor *s* in reporting the criticality of the event.

The other state variables in the system are power PR and Storage SS remaining on the handheld. We model the constant parameters in the system including the power used in collecting the data (PC_s), power used in processing the data (PP_s), power used in transmitting a unit byte of data (PX_s) and the size of sensor *s* data measurements in bytes (D_s). We model the degradation in sensor characteristics (electronic drift, or change in chemical/physical properties) by a sensor reliability variable I_s for the sensor s, which could be a function of time. An optimization problem can be formulated as given below:

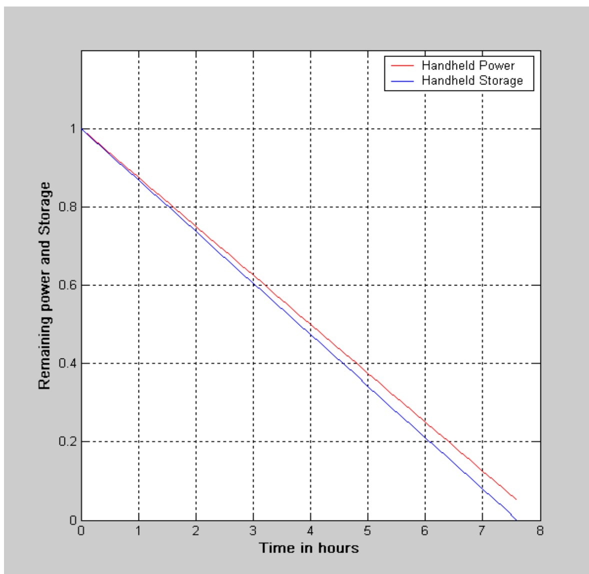


Figure 6 Inequality constraints for power (blue) and storage (red)

The objective is to determine the optimal control decisions for each time interval for the following sensor network parameters:

(i) Frequency of sensor operation, U_s . This is a real-valued control variable

(ii) Discard (1)/ Store (0) U_{ds} the sensor data measurements at a given time interval. This control variable is binary.

In our current system model, the time interval for each control step is chosen as 0.2 hours and the length of simulation is chosen to be 8 hours.

Using our profit/loss maximization formulation, we aim to maximize the profit in observing events and storing data, while reducing the loss incurred from processing and transmitting raw sensor data, and storing these measurements. The mathematical representation of this formulation is detailed below:

Objectives and Constraints

(i) Maximize Sensor information utility (*SIU*)

$SIU = f(U_s, IPC_s, EC_s, I_s)$

(ii) Maximize Storage Utility (*SSU*)

$SSU = f(U_s, U_{ds}, IPC_s, EC_s, I_s)$

(iii) Minimize Sensor Capture Power Cost (*SPC*)

$SPC = f(U_s, PR, PC_s, PX_s, PP_s)$

(iv) Minimize Sensor Storage Cost (*SSC*)

$SSC = f(U_{ds}, SS, PC_s, PX_s)$

Inequality constraints are placed on the system states. Figure 6 shows the inequality constraints for the remaining Power and Storage on the handheld. Constraints on power and storage ascertain that the sensor network functions for a minimum total time of T_{min}.

For each individual in the population, Power and Storage values are predicted based on the history of operation of the system. Inequality constraints are formulated as follows

PR_{pred} ((T_{elap} , U_s, D_s, PC_s, PX_s) < PR

SS_{pred} (T_{elap}, U_s, D_s) < SS

where T_{elap} is the time elapsed in the system.

4.2 Simulation Results

In our current simulations, we consider a single sensory system, and two test cases with varying sensor/classifier reliability models. In one case, the sensor /classifier is assumed to he highly reliable, i.e. events can be detected with high confidence (statistically speaking). In the second case, the sensor/classifier is unreliable and therefore events cannot be detected with high confidence. In both cases, we assumed a consistent model for the remaining sensor network parameters, including the power consumed by different components, in data transmission, storage, etc. The nominal values for PC_s, PX_s, PP_s (in milli-ampere hours) are shown below.

Sensor	PC_S	$\boldsymbol{PX_s}$	$\boldsymbol{PP_s}$
1	0.05mAh	0.07 mAh	0.04 mAh

Case 1: Reliable Sensor/Classifier Model

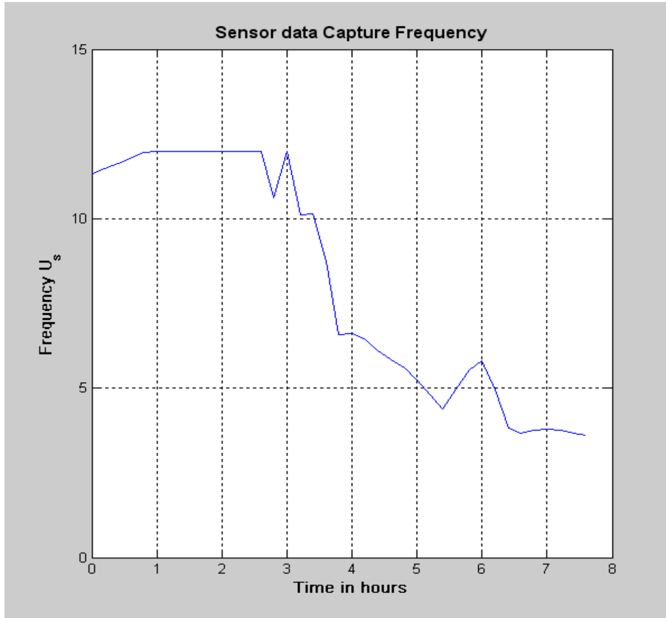
Figure 7 shows the simulation results when the sensor/classifier reliability is assumed to be high. Figure 7a shows the frequency of data capture from the sensor. Event criticality values are assumed and set high at time t = 3, 6 hours. From the figure it is clear that the sensor lowers the capture frequency U_s when it encounters no event and increases the frequency when an event occurs at the third hour of operation.

Figure 7b (blue line) shows the data store/discard decision made by the MOGA controller as a function of time and the criticality of the labeled event. the classifier sensor data is stored (0) or discarded (1) at the handheld depending on the criticality of the event (red plot in Figure 7b). Since the sensor/classifier labels are reliable, the data is discarded when there is no event, even when the available resources (power, storage) are high.

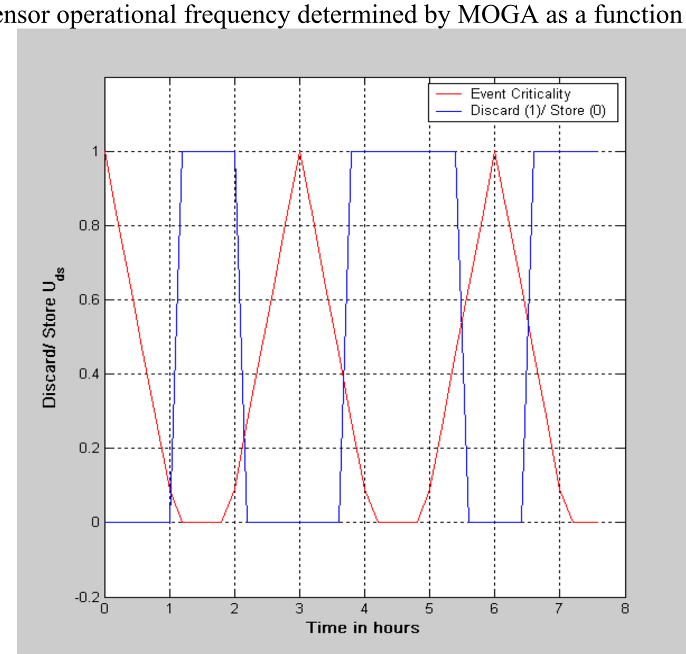
Case 2:Unreliable Sensor/Classifier Model

Figure 8 shows the simulation results for case 2. In this model, the sensor/classifier reliability is low and therefore any event is not labeled/detected with high confidence. Figure 8a shows the frequency U_s of the sensor varying with time. Figure 8b (blue plot) shows that the handheld stores the data irrespective of the criticality of the event. This is due to the fact that the sensor/classifier is unreliable and therefore its classified labels are ambiguous; measurements labeled as non-events could very well correspond to events. Therefore, in this case, the sensor measurements are stored even when the system has limited available storage space.

The run-time details of our C++ MOGA algorithm implementation are discussed below. The MOGA C++ implementation completed a simulated run of 8 hours, with control updates every 0.2 hours in 10.7 ms on a Intel® Pentium® M processor 1.3 GHz with 1.046 GB RAM. Therefore, the time taken to complete a single control decision at a single instance was 0.2675 ms. The size of the executable was 192 Kb; this will further reduce when the code is customized and ported to a real-time system. These initial results are impressive and indicate that the MOGA can be effectively used in our real-time MUSIC infrastructure with minimal algorithmic modifications.

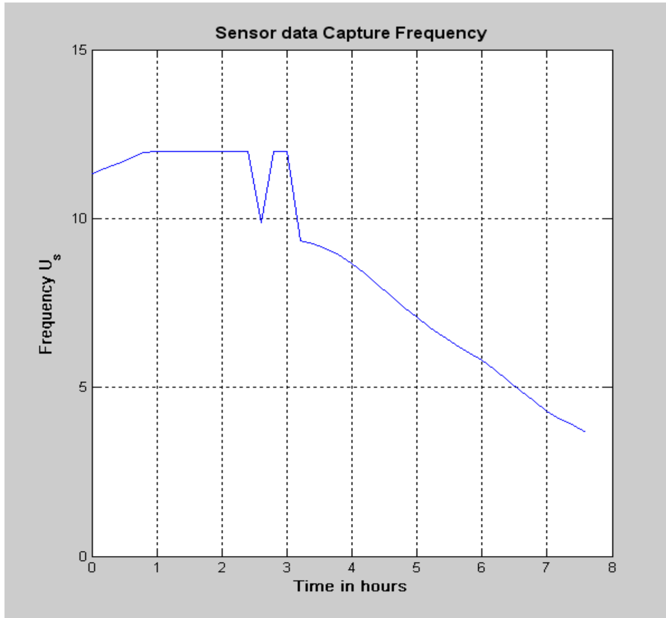


(a) Sensor operational frequency determined by MOGA as a function of time

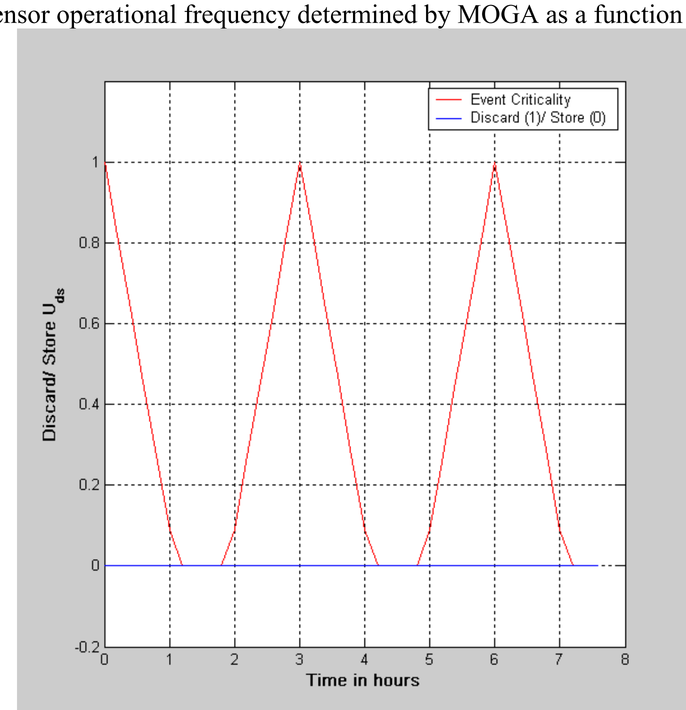


(b) Sensor data measurement store (0)/discard (1) control decisions (blue) as a function of time & critical/non-critical events (red)

Figure 7: MOGA control decisions as a function of critical/non-critical events over time when sensor is reliable (Test Case 1)



(a) Sensor operational frequency determined by MOGA as a function of time



(b) Sensor data measurement store (0)/discard (1) decisions (blue) as a function of time & critical/non-critical events (red)

Figure 8: MOGA control decisions as a function of critical/non-critical events over time when sensor is not reliable (Test Case 2)

5 CONCLUSION

We have developed a framework for resource management of a distributed sensing network that enables it to function for extended time periods beyond the active lifetime of the individual sensor nodes. Initial results are promising. Extensive tests need to be done on real-datasets, and the control optimization algorithms need to be ported to the real-time MUSIC system and tested in cohesion with our real-time pattern recognition algorithms. We would also like to incorporate data prioritization to enable transmission of critical data when bandwidth and communication time slots are limited. A distributed MOGA algorithm will enable resource management across a network with 100's to 1000's of nodes.

ACKNOWLEDGMENT

The research described in this paper was carried out by the Children's Hospital Los Angeles/Univ. of Southern California. This work has been sponsored in whole with Federal Funds from the National Institute on Alcohol Abuse and Alcoholism, National Institute of Health, Department of Health and Human Services under Contract No. N01AA33004.

REFERENCES

[1] Schwiebert, L., S. K. S. Gupta and J. Weinmann. *Research Challenges in Wireless Networks of Biomedical Sensors*. in ***International Conference on Mobile Computing and Networking***. 2001. Rome, Italy.

[2] Mann, S. *Smart Clothing: Wearable Multimedia Computing and Personal Imaging to Restore the Technological Balance between People and Their Environments*. in ***Fourth ACM International Conference on Multimedia '96***. November 1996. Boston, MA: ACM Press.

[3] Stanley-Marbell, P., D. Marculescu, R. Marculescu, and P. K. Khosla, *Modeling, Analysis, and Self-Management of Electronic Textiles.* **IEEE Transactions on Computers**, August 2003. **52**(8): p. 996-1010.

[4] Chakrabarty, K., et al., *Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks.* **IEEE Trans. on Computers**, 2002. **51**(12).

[5] Malan, D., et al. *CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care*. in ***International Workshop on Wearable and Implantable Body Sensor Networks***. 2004.

[6] Chung, T., V. Gupta, B. Hassibi, J. Burdick and R. Murray. *Scheduling for Distributed Sensor Networks with Single Sensor Measurement per Time Step*. in ***International Conference on Robotics and Automation***. April 2004. New Orleans, LA.

[7] Talukder, A., et al. *Data classification and resource management in sensor networks for remote health and environmental monitoring*. in ***IEEE Intl. Conferences on Intelligent Signal and Image Processing. (ICISIP 2004) (Recipient of the Special Mention Paper Award)***. 2004.

[8] Talukder, A., et al. *Real-time Intelligent Pattern Recognition, Resource Management and Control under Constrained Resources for Distributed Sensor Networks*. in ***Intl. Joint Conference on Neural Networks***. 2004.

[9] Martin-Sanchez, J., and J. Rodellar, *Adaptive Predictive Control : From the concepts to plant optimization*. Prentice Hall International series in Systems and Control Engineering, ed. M.J. Grimble. 1996: Prentice Hall PTR.

[10] Ribler, L.R., et al. *Autopilot: Adaptive Control of Distributed Applications*. in ***Proc. 7-th IEEE Symposium on High Performance Distributed Computing***. 1998.

[11] Kretchmar, R., et al., *Robust Reinforcement Learning Control with Static and Dynamic Stability.* **Intl. Journal of Robust and Nonlinear Control**, 2001. **11**: p. 1469-1500.

[12] Modi, J., et al. *Dynamic Distributed Resource Allocation: A distributed constraint satisfaction approach*. in ***Intelligent Agents VIII: Proc. ATAL'01***. 2001.

[13] Hill, J., R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. *System Architecture Directions for Networked Sensors*. in ***9th International Conference on Architectural Support for Programming Languages and Operating Systems***. November 2000. Cambridge, MA, USA.

[14] Corne, D., et al. *The Pareto envelope-based selection algorithm for multiobjective optimization*. in ***Sixth International Conference on Parallel Solving from Nature VI***. 2000.

[15] Ursem, R.K., et al. *Analysis and Modeling of Control Tasks in Dynamic Systems*. in ***Congress on Evolutionary Computation***. 2002.

[16] Deb, K., *Multi-objective optimization using evolutionary algorithms*. 2001, Chichester, UK: Wiley.

[17] Deb, K., et al., *A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions*, in ***KanGAL Report***. 2003. p. 2-6.