

Hierarchical Control for Aircraft Electronic Power Systems

Forrest N. Iandola
University of California, Berkeley, USA
forresti@eecs.berkeley.edu

Huy Vo
University of California, Berkeley, USA
huytbvo@eecs.berkeley.edu

ABSTRACT

Best abstract ever

Keywords

avionics, model-predictive control, optimization, linear programming

1. INTRODUCTION

Aircraft Electrical Power Systems (EPS) are systems that provide power to critical loads (landing gear, navigation systems, etc.) as well as non critical loads (cabin lights, coffee machine, etc). Above all else, an aircraft EPS controller must ensure that power is constantly provided to all critical parts of the system. Since EPS are evolving into increasingly complex systems, EPS controllers must become “smarter”. We address the challenge of designing such a controller in this project. More specifically, we contribute a Low Level Load Management System that closely monitors system status and ensures safety. We combine this controller with an Optimizing Load Management System [1] to obtain a Hierarchical Load Management System that is both safe and optimal.

The rest of the paper is organized as follows. In Sections 2 and 3, we give an overview of Aircraft EPS. This is followed by a detailed discussion of an Optimizing Load Management System (Section 4) and a Low Level Load Management System (Section 5). After presenting these two controllers, we then show how to combine the two into a Hierarchical Load Management System. Finally, in Sections 7-9, we show that our hierarchical controller achieves safety where the Optimizing Load Management System would have been unable to and achieves optimality where the Low Level Load Management System cannot.

2. AIRCRAFT EPS OVERVIEW

Intro to aircraft EPS... explain the buses and generators. Possibly show the diagram from the Honeywell patent. Possibly show our “cartoon” diagram of an EPS system.

2.1 Load Shedding

Aircraft EPS generally have two types of loads: sheddable and nonsheddable. Examples of nonsheddable loads include navigation systems, electronic actuators, and other mission-critical devices. Sheddable loads include the cabin lights, kitchen, and electrical outlets for passengers’ laptop computers. If an EPS power bus requires more power than the

generators can provide, the LMS is responsible for disabling (shedding) some of the sheddable loads.

3. RELATED WORK

In typical aircraft EPS, the load management systems aim to ensure safety during unexpected system behavior [citation needed]. Specifically, these systems require nonsheddable loads to remain powered at all times.¹ Therefore, if a generator fails, the EPS must shed loads and/or reallocate generators to buses such that no nonsheddable load becomes unpowered. Typically, these systems rely on hard-coded priority tables for selecting loads to shed and for allocating generators to buses.

4. OPTIMIZING LMS

LMS systems that are based purely on priority tables are very simplistic, allowing them to frequently sample the system. These LMS systems are thus more responsive to failures in the system. However, pure priority table solutions tend to shed more loads than necessary, to use more generators than necessary. Toward resolving these issues, the control systems community recently designed a model-predictive Optimizing LMS for aircraft EPS [1]. The key insight in [1] is to formulate the system requirements (e.g. the maximum power produced by each generator) as constraints in an linear optimization problem. The Optimizing LMS minimizes an objective function comprised of three weighted vectors: load shedding preference, generator selection preference, and number of generators used. The following expression gives a high-level view of the objective function in [1], where the weights can correspond with priorities in priority tables and h is a time horizon:

$$\int_t^{t+h} \min \left(\binom{\text{sheddable}}{\text{loads}} * \text{weights}_1 + \binom{\text{generator}}{\text{selection}} * \text{weights}_2 + \binom{\# \text{ of generators}}{\text{used}} \right)$$

Notice the integral from time t to time $t + \text{horizon}$. If the horizon=1, then we simply optimize the objective function for one system timestep. However, if the horizon=10, then we do one call to the CPLEX MILP solver to compute the optimal system configurations for 10 timesteps. Batching several timesteps into one MILP call can reduce the overall computation time: using a 6-core Intel 3930K processor, optimizing 1 timestep takes an average of 350ms, while optimizing 10 timesteps takes an average of 500ms (so, 50ms per timestep in the batched configuration).

¹More precisely, nonsheddable loads must not be more unpowered for more than one cycle of the LMS. In practice, 50ms is a typical LMS sampling frequency.

Table 1: **Load Priority Table Example**, NS = **non-sheddable**, S = **sheddable** Loads lower in the list have lower priority while loads higher up in the list have higher priority. LL-LMS will shed load starting at the lower end of the list.

NS loads (W)	S loads (W)	Shed priority
1000	1000	1
1000	5000	2
500	2000	3
5000	2000	4
1000	1000	5
1000	5000	6
2000	1000	7
5000	2000	8
500	2000	9
2000	1000	10

However, our *receding horizon* approach of doing a batched optimization at every 10 timesteps is not foolproof. This is because, at time t , we can't perfectly predict the power loads or generator failures at timesteps in the range $[t + 1, t + horizon]$. Therefore, instead of using live data from the load sensors, the Optimizing LMS uses historical data, averaged from previous flights that flew the same route using the same model of aircraft.

However, the Optimizing LMS [1] is not without shortcomings. Chiefly, the Optimizing LMS is computationally expensive (0.5s for 10 timesteps of optimization in CPLEX on an Intel 3930K processor). [Discuss receding horizon and the need for historical data] Since the Optimizing LMS relies on historical data, it may produce unsafe allocations that use a broken generator or operate above the generator power caps. Thus, it is potentially unsafe to use Optimizing LMS to control the electronics in a real airplane. However, as we will see later in the paper, the Optimizing LMS can be effective if it is coupled with a fail-safe Low Level LMS.

5. LOW LEVEL LMS

5.1 Overview

As the name suggest, the LL-LMS is implemented as the last level of controller in the system. Above all else, the LL-LMS must guarantee safe operation conditions². Briefly, the LL-LMS uses various sensors to frequently sample the state of the system. It then uses its sensor readings along with its priority tables, such as the one in Table 1, to produce a safe system configuration.

5.2 Implementation

Every cycle, the LL-LMS reads its sensors to collect data on generator status and power requests on all the nonsheddable and sheddable loads. The LL-LMS first filters out unhealthy generator tables and then uses its generator priority table to determine which of the remaining healthy generators it should use to power the system. Once it has determined

²For the purposes of this paper, we only consider the following safety conditions: no load remains unpowered for more than 50ms, loads are not connected to unpowered generators, and no generator is connected to loads whose power request exceeds the power rating of the generator

what generator to use, the LL-LMS will then determine how much load it has to shed. It first sums up power requests of all the loads (sheddable and nonsheddable) on each bus. It then determines if the generator for each of the buses are able to power all of the loads. If the power request is more than a generator can provide, the LL-LMS will start shedding sheddable loads starting at the load with the lowest priority. This simple generator selection and load shedding scheme ensure that the LL-LMS can quickly arrive at a safe system configuration.

[TODO: 4-line pseudo-code for how we apply priority tables]

Other details: Low-level controller samples system at every timestep (e.g. 50ms of system time) Ensure that no non-sheddable load is unpowered for more than 50ms

6. HIERARCHICAL LMS

6.1 Overview

Recall that the goal of our project is not to design an LMS that is safe OR optimal. Rather, we want to build an LMS that is both safe AND optimal. To this end, we propose a hierarchical LMS that achieves the best of both worlds by combining an optimizing LMS with a low level LMS. The OL-LMS periodically, though at a slow rate, samples the status of the system and produces a system configuration that is optimal. More specifically, the OL-LMS checks the status of the generators and solves the optimization problem detailed in Section 4. The LL-LMS collects the advice from the O-LMS and determines whether or not the advice is safe. If the advice is safe, the LL-LMS will actuate the system based on the O-LMS's advice. Otherwise, the LL-LMS will fall back to its priority table. Because failures are rare, the LL-LMS will spend most of the time actuating the system based on the O-LMS's advice. This hierarchical LMS (H-LMS) is optimal because the O-LMS is optimal and is safe because the LL-LMS is safe.

6.2 O-LMS Modifications

Instead of having the O-LMS system read the sensors directly as it would in a system that relies solely on an O-LMS, it will instead receive its system status information from the LL-LMS. This communication strategy saves on communication links between sensors and the O-LMS and reuses the communication link between the LL-LMS and the O-LMS. We also use the O-LMS to perform live optimizations of the system. That is, instead of solving one large offline optimization problem use a receding horizon that covers the entire flight, we have the O-LMS periodically solve many optimization problems using a much smaller receding horizon. Our implementation makes the O-LMS system more resilient to generator failures. Previously, the O-LMS would be completely oblivious to generators that fail mid-flight. With this modification, the O-LMS will eventually see that there is a generator failure and reoptimize.

6.3 LL-LMS Modifications

The LL-LMS will now receive advice from the HL-LMS in addition to its other functionalities detailed in Section 5. The LL-LMS will then check to see that the HL-LMS's advice is safe. Specifically, the LL-LMS will check that the HL-LMS did not recommend the use of a failed generator

and that the HL-LMS has shed enough loads so that no generator is overloaded. In the case that the HL-LMS recommended the use of a failed generator, the LL-LMS will use the next best generator according to its priority table and perform the appropriate load shedding. In the case that HL-LMS advice would cause a generator overload, the LL-LMS will shed additional loads so that the power request on the bus is under the corresponding generator's power rating.

6.4 H-LMS Implementation

Figure 1 shows the block diagram of our hierarchical controller. The different clocks fed into the two controllers indicate that the two controllers operate at different frequencies. More specifically, the LL-LMS operates at a high frequency, fast enough to ensure that the system is not in an unsafe configuration for more than 50 ms. On the other hand, the HL-LMS runs at a much slower rate due to the complexity of the optimization problem. It produces a set of advice (generator usage recommendations as well as load shedding configurations) that can the LL-LMS can use until the next O-LMS cycle. Notice that there is an arrow going from the LL-LMS to the H-LMS because the LL-LMS is responsible for providing the H-LMS with information of the status of the generators. The actual generator usage and load shedding that the LL-LMS will actuate on will be either the advice provided by the H-LMS if it is safe or the configuration determined by the LL-LMS's priority table if the O-LMS's advice is unsafe.

7. NUMERICAL EXAMPLES

8. EXPERIMENTAL DESIGN

Parameters:

generator 1: 100,000 W

generator 2: 100,000 W

APU: 104,000 W

[TODO: show priority tables; numerical example]

9. EXPERIMENTAL RESULTS

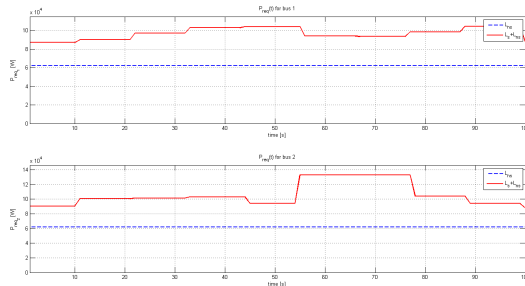


Figure 2: **Power Request** Historical data of power request on Bus 1 and Bus 2 for time 0 to 100. Notice the spikes in power usage on Bus 2 that pulls the request over 100,000 Watts, well over the power rating of the generators. Load shedding is a must during those intervals.

9.1 No Failures

For our first experiment, we simulated the case when there are no power generator failures in the system using the power

usage profile shown in Figure 2. In other words, the real power usage profile (Figure 2) is given to the Optimizing LMS and LL-LMS without any perturbations. Notice that Bus 1 requests more than 100,000 Watts between time 30 and 60 and again around time 90. Bus 2, on the other hand, has a massive power usage between time 50 and 80. We wanted to see how the Optimizing LMS and the LL-LMS responds to this scenario in isolation. Comparing the results, we can see that the LL-LMS will shed a much larger number of loads (Figure 3) than the Optimizing LMS (Figure 4). In particular, the LL-LMS sheds exclusively from loads 6-10 due to the configuration of the priority table. However, the Optimizing LMS however is much smarter. It is able to minimize the number of loads shed by shedding loads much higher up in the priority table.

9.2 Generator Failure

Now, we investigate how our Hierarchical LMS system responds to spontaneous generator failures. To this end, we simulated the case when generator 2 would fail at time 45. Obviously, if the Optimizing LMS were to run in isolation, it would leave the system unpowered from time 45 to 50. Figure 5 shows what would happen if the LL-LMS were to run in isolation. Starting at time 45, the LL-LMS sees that generator 2 has failed and it has switched to using generator 1 to power both buses according to its priority table. In order to stay under generator 1's power rating, the LL-LMS sheds a large number of loads on the second bus. Figure 6 shows what would happen in a combined Optimizing LMS LL-LMS system. The Optimizing LMS produces advice to be used from time 40 to 50. Unfortunately, there is no way for the Optimizing LMS's advice to be aware of the failure at time 45. So, while the advice from time 40 to 44 is optimal and safe, the advice from 45 to 49 is unsafe. The LL-LMS is able to see this failure at time 45 and kicks in, ignoring the unsafe advice from the Optimizing LMS. This behavior is observed in Figure 6 where from time 45 to 49, a large number of loads are shed. Then at time 50, the Optimizing LMS reoptimizes by turning on the APU. Notice the difference in the number of loads shed onward from time 50 in Figure 5 and Figure 6.

9.3 Unexpected Power Spikes

The Optimizing LMS performs a live optimization using historical data for load power requests from previous flights. Due to the nature of the optimization problem, it is impossible to program the optimization problem with live data. As a result, it is possible for the Optimizing LMS to provide unsafe advice if there is a power spike. More specifically, it is possible for the Optimizing LMS to generate a load shedding configuration that could have one of the power generators driving a higher power request than its rating. We want to simulate such a failure case and show that the LL-LMS performs the required adjustment to keep the system in a safe state. For this experiment, we use the same historical power request as in Figure 2. However, the actual power request for this simulation is shown in Figure 7. Without the LL-LMS, the Optimizing LMS would use the load shedding configuration seen in Figure 3. Notice that there is a power spike from time 10 to 40 that the historical data does not account for and thus the Optimizing LMS is unaware of. As a result, its load shedding configuration would overload the power generator. Because we have a LL-LMS, we are able

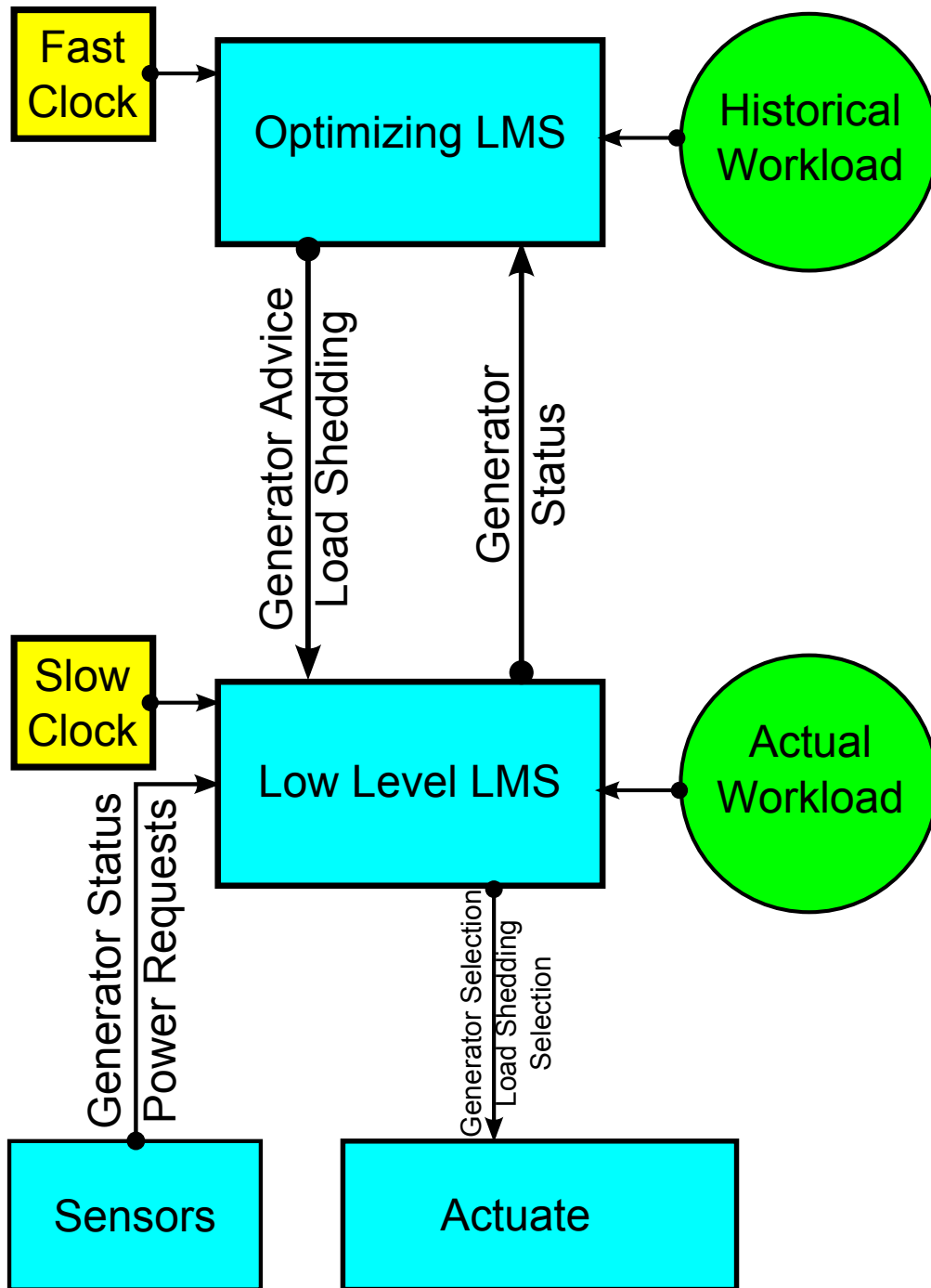


Figure 1: **H-LMS Block Diagram** The O-LMS and the LL-LMS runs at different rates. The O-LMS no longer communicates directly with the EPS sensors and actuators. The LL-LMS is the last level of controller and communicates directly with the EPS sensors and actuators. The LL-LMS periodically sends generator status to the O-LMS which the O-LMS uses in its next iteration. The O-LMS produces advice that it sends to the LL-LMS. Whether or not the O-LMS's advice is ever actuated on is left to the discretion of the LL-LMS.

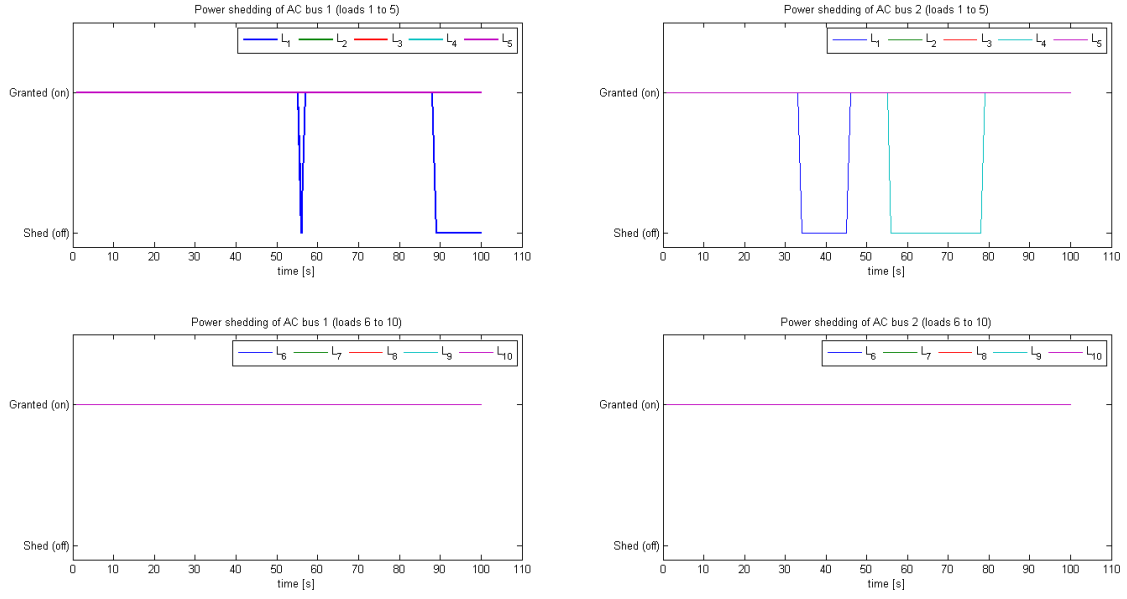


Figure 3: **Optimizing LMS Load Shedding** Notice the long period of load shedding around the power usage spikes.

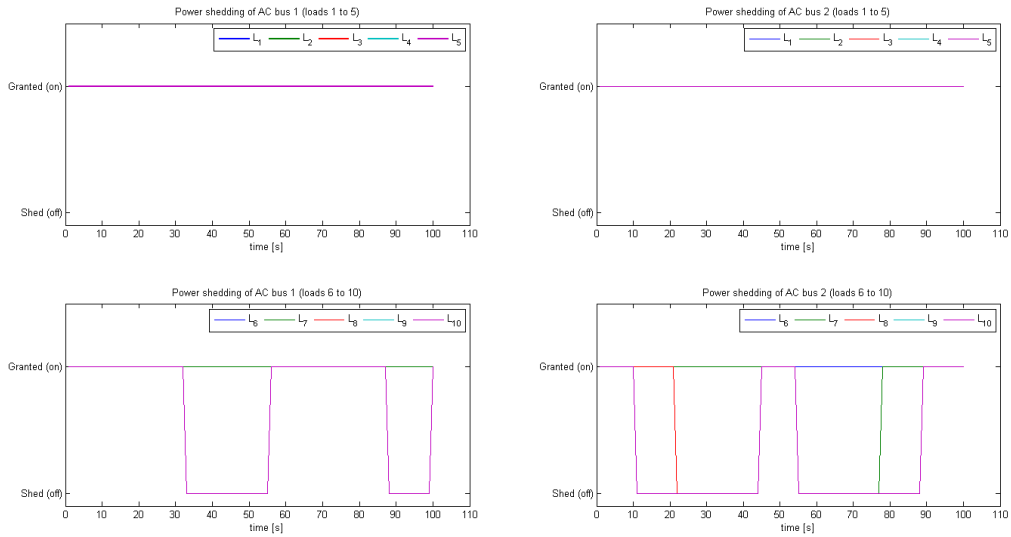


Figure 4: **LL-LMS Load Shedding** Notice that the LL-LMS sheds much more load than the Optimizing LMS.

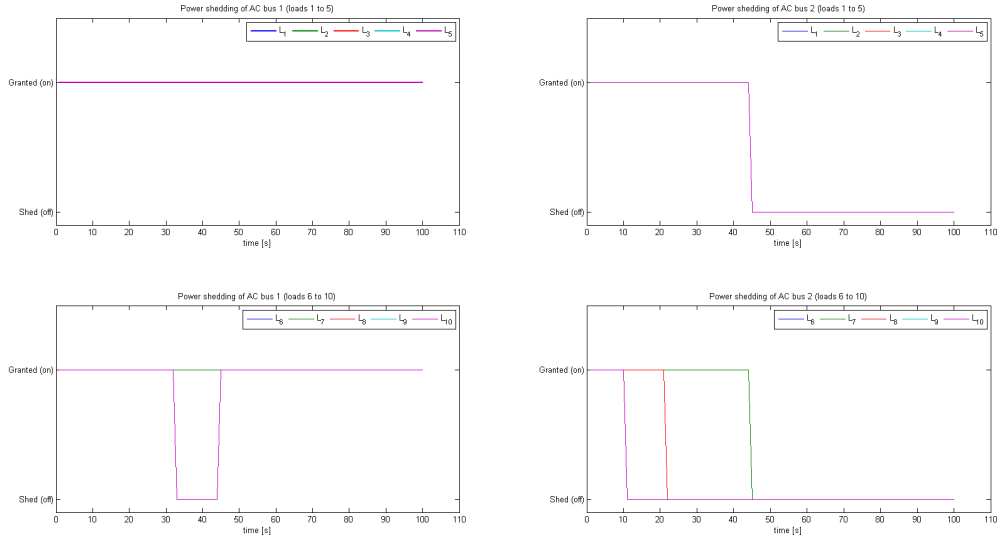


Figure 5: **LL-LMS Load Shedding – One Failure**. Notice the large amount of loads shedded in order to keep power usage below the power rating of generator 1.

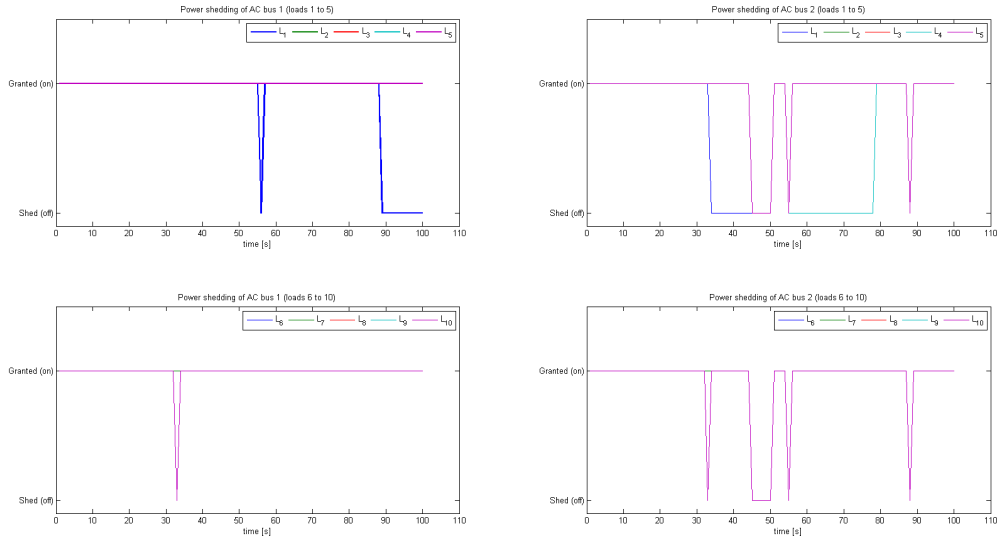


Figure 6: **Hierarchical LMS Load Shedding – One Failure**. Notice the large number of loads dropped from time 45 to 50 before the Optimizing LMS can reoptimize.

to keep the system in a safe configuration. Figure 8 shows that the LL-LMS will detect that the generators would be overloaded and thus shed some loads during this time interval.

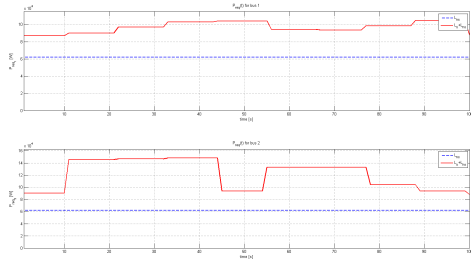


Figure 7: **Actual Power Request** Notice the difference between this power request and the power request expected by the historical data in Figure 2. In particular, there is a large power spike from time 10 to 40 that is not accounted for in the historical data.

10. CONCLUSION

Conclusion.

Acknowledgments

This work was supported by the US Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

11. REFERENCES

- [1] M. Maasoumy, P. Nuzzo, M. Kamgarpour, C. Tomlin, A. Sangiovanni-Vincentelli, and M. Mozumdar. OLMS: Optimal load management system for aircraft electric power distribution. Technical report, University of California, Berkeley, 2012.

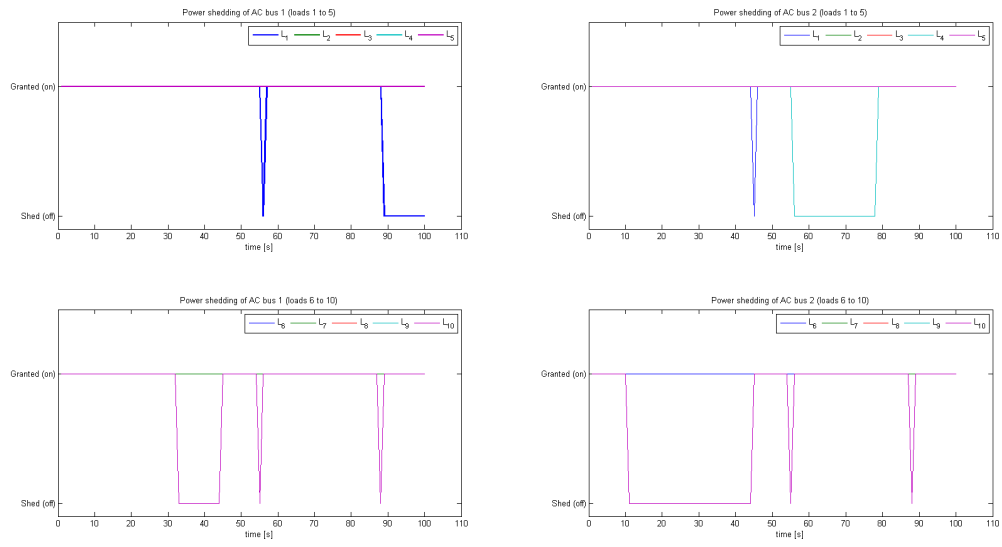


Figure 8: **Hierarchical LMS Load Shedding – Power Spike** Whenever possible the LL-LMS will use the advice from the Optimizing LMS. The only time it ignores the advice from the Optimizaing LMS is from time 10 to 40 when there is a power spike that the Optimizing LMS is not aware of.