OL-LMS

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
embedded systems and planes and stuff

1. INTRODUCTION

Overview, stuff about platform based design...

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.

Our key objective is to design a Load Management System (LMS) for EPS that is "smarter" in terms of several criteria. First, always preserve safety. After preserving safety, maximize the quality of service. Finally, conserve fuel by using as few generators as possible.

Aircraft EPS generally have two types of loads: sheddable and non-sheddable. Examples of non-sheddable 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.

1.1 OL-LMS Goals

Talk about goals of project

2. RELATED WORK

In typical aircraft EPS, the load management systems aim to ensure safety during unexpected system behavior [citation needed]. Specifically, [these systems typically require] unsheddable loads to remain powered at all times. [or, maybe unsheddable loads can be unpowered for up to 50ms while switching generators] Therefore, if a generator fails, the EPS must shed loads and/or reallocate generators to buses such that no unsheddable load becomes unpowered. Typically, these systems rely on hard-coded priority tables for selecting loads to shed and for allocating generators to buses.

2.1 Optimizing-LMS

LMS systems that are based purely on priority tables are effective for preserving safety. However, priority table solutions tend to shed more loads than necessary, to use more generators than necessary, [and be ill-defined in their tradeoffs between generator allocation and load shedding]. Toward resolving these issues, the control systems community recently designed a model-predictive optimizing controller for aircraft EPS [Mehdi's paper]. The key insight in [Mehdi's paper] is to formulate the system requirements (e.g. the maximum power produced by each generator) as constraints in an optimization problem. Then, [Mehdi's paper] selected an objective function such that solving this optimization problem produces a power system allocation using a minimal number of generators while also minimizing the total number and duration of loads that are shed. [This optimizing controller aims to minimize three weighted parameters: load shedding preference, generator selection preference, and number of generators used.] [TODO: show a simplified version of Mehdi's math]

However, the Optimizing LMS [Mehdi's paper] 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.

3. LOW LEVEL LMS

[Talk about our LL-LMS system, its inputs/outputs, functionality, etc]

This section presents a Low-Level LMS (LL-LMS) similar to the controllers discussed in Section 2. The key goal (and perhaps the only) goal of this load management system is to preserve safety if workloads change or if generators fail. Therefore, using a pre-defined priority table of generator assignments and loads to shed, our LL-LMS sheds loads until the power requirement is within the generator's power budget.

[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

4. HIERARCHICAL LMS

Modify LL-LMS to call the Optimizing LMS periodically. LL-LMS does safety checks on results from Optimizing LMS...

5. OL-LMS

Talk about how the entire system works at a whole, how we set it up, etc.

6. EXPERIMENTAL RESULTS

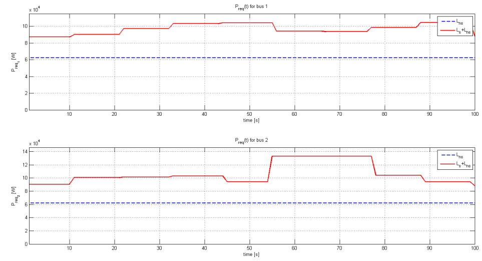

Figure 1: **Power Request** Historical data of power request on Bus 1 and Bus 2 for time 0 to 100. Notice the spikes in power usage on Bus 2 that pulls the request over 100000 Watts, well over the power rating of the generators. Load shedding is a must during those intervals.

6.1 No Failures

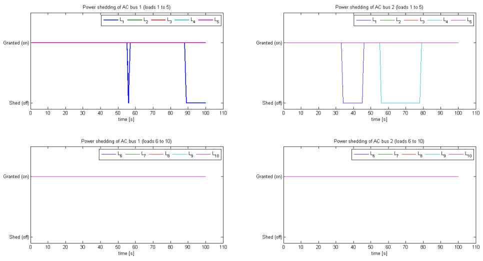

Figure 2: **HL-LMS Load Shedding** Notice the long period of load shedding around the power usage spikes.

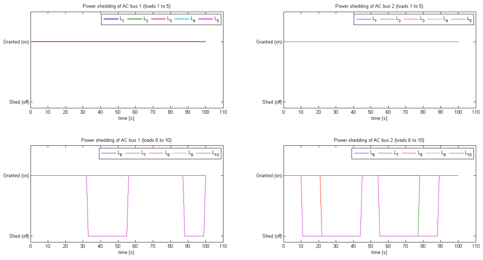

Figure 3: **LL-LMS Load Shedding** Notice that the LL-LMS sheds much more load than the HL-LMS.

For our first experiment, we simulated the case when there are no power generators failures in the system using the power usage profile shown in Figure 1. Notice that Bus 1 uses more than 100000 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 HL-LMS and the LL-LMS responds to this scenario in isolation. Our results are in Figure 2 and Figure 3. Comparing the results, we can see that the LL-LMS will shed a much larger number of loads than the the HL-LMS. In particular, the LL-LMS sheds exclusively from load 6-10 because it simply examines the priority table. The HL-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.

6.2 Generator Failure

For this experiment, we wanted to see how a combined HL-LMS and LL-LLMS system would respond to spontaneous generator failures. To this end, we simulated the case when generator 2 would fail at time 45. Obviously, if the HL-LMS were to run in isolation, it would leave the system unpowered from time 45 to 50. Figure 4 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 5 shows what would happen in a combined HL-LMS LL-LMS system. The HL-LMS produces advice to be used from time 40 to 50. Unfortunately, there is no way for this 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 at time 45 and kicks in, ignoring the unsafe advice from the HL-LMS. This behavior is observed in Figure 5 where from time 45 to 49, a large number of loads are shed. Then at time 50, the HL-LMS reoptimizes by turning on the APU. Notice the difference in the number of loads shed onward from time 50 in Figure 4 and Figure 5.

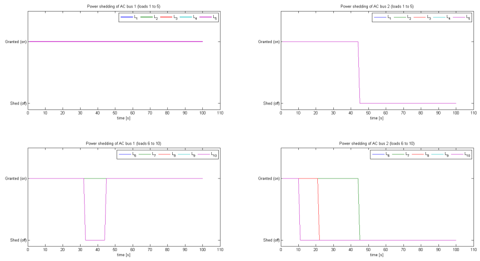

Figure 4: **LL-LMS Load Shedding – One Failure**. Notice the large amounts of load shedded in order to keep power usage below the power rating of generator 1.

6.3 Unexpected Power Spikes

The HL-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 HL-LMS to provide unsafe advice if there is a power spike. More specifically, it is possible for the HL-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

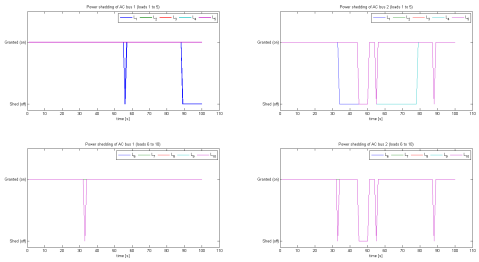

Figure 5: **OL-LMS Load Shedding – One Failure**. Notice the large number of loads dropped from time 45 to 50 before the HL-LMS can reoptimize.

as in Figure 1. However, the actual power request for this simulation is shown in Figure 6. Without the LL-LMS, the HL-LMS would use the load shedding configuration seen in Figure 2. Notice that there is a power spike from time 10 to 40 that the historical data does not account for and thus the HL-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 to keep the system in a safe configuration. Figure 7 shows that the LL-LMS will detect that the generators would be overloaded and thus shed some loads during this time interval.

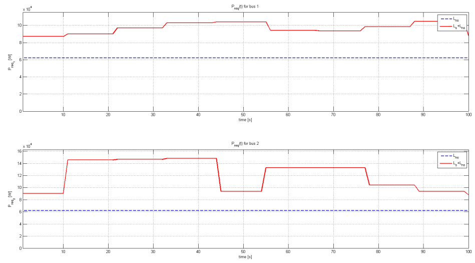

Figure 6: **Actual Power Request** Notice the difference between this power request and the power request expected by the historical data in Figure 1. In particular, there is a large power spike from time 10 to 40 that is not accounted for in the historical data.

7. RELATED WORK

Related [1].

8. CONCLUSION

Conclusion.

Acknowledgments

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

9. REFERENCES

[1] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

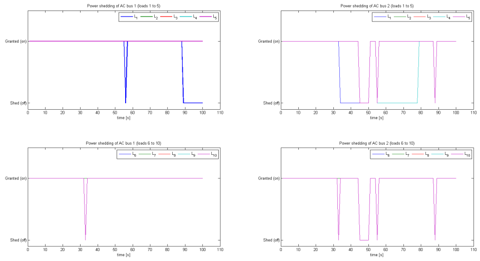

Figure 7: **OL-LMS Load Shedding – Power Spike** Whenever possible the LL-LMS will use the advice from the HL-LMS. The only time it ignores the advice from the HL-LMS is from time 10 to 40 when there is a power spike that the HL-LMS is not aware of.