MSE 402 - Final Design Review

Ben Fogerty - 20940401

Chintan Desai - 20938455

Shailleze Mahenrenathan - 20951692

Sher Verma - 20953080

Sneh Shah - 20951119

Group 18

Word Count: 4339

Abstract - *This report presents a battery-health decision-support system for electric aircraft. The system converts noisy battery-management-system and aircraft telemetry into event-level battery records, estimates a condition-aware latent state of health (SOH), forecasts near-term degradation, and translates these outputs into operator-facing advisory guidance. Results show that the latent SOH pipeline substantially reduces implausible raw SOH variation and enables useful short-horizon forecasting for maintenance and flight-planning support. The final design is compatible with existing telemetry interfaces and demonstrates a practical framework for battery-health monitoring, degradation forecasting, and conservative operational decision support in electric aviation.*

Keywords: electric aircraft, battery state of health, telemetry analytics, degradation forecasting, decision-support systems

1. PROBLEM ANALYSIS

A central challenge in electric aircraft operations is that battery health matters enormously, but is difficult to measure reliably in day-to-day use. As batteries age, they lose usable capacity and their performance becomes less predictable, affecting endurance (the amount of time the aircraft can remain in flight on its available battery energy), turnaround time (the time needed to recharge, prepare, and return the aircraft to service between flights), and battery replacement timing. Therefore, operators need accurate State of Health (SOH) information to plan flights and manage batteries effectively. The problem is that battery health cannot be measured directly in flight in the same way fuel can be measured in a conventional aircraft. Instead, it has to be inferred from onboard telemetry and Battery Management System (BMS) outputs such as voltage, current, temperature, state of charge (SOC), charge throughput, and calculated capacity. These outputs are useful, but they are also influenced by operating conditions, sensor noise, estimator recalibration, and BMS logic (Synopsys, n.d.). As a result, changes in reported SOH do not always reflect true degradation; sometimes they reflect the conditions under which the battery is being observed. This is why battery health estimation grows from a monitoring problem to a difficult inference problem.

This issue has become more important as electric aircraft have moved beyond demonstration flights and into practical use cases such as pilot training and short-duration operations. In these settings, the battery is one of the most expensive and operationally limiting parts of the aircraft. Unlike fuel, battery capability changes over time, so the same aircraft does not always deliver the same endurance or performance from one month to the next. That matters because electric flight is governed by energy margins (extra battery energy kept in reserve for safety) and reserve requirements (minimum battery charge that must remain available at the end of a flight for safety), which means operators need confidence not only in the battery's present charge, but also in its underlying condition (Pipistrel, n.d.). If that condition cannot be estimated reliably, flight planning becomes more conservative, and long-term operations become harder to manage.

The causes of the problem exist at both the physical and measurement levels. Physically, lithium-ion batteries degrade through electrochemical aging processes accelerated by repeated charge-discharge cycles, thermal stress, high current demand, and time spent at high states of charge (Rahman, 2024). These stressors are common in aviation, where batteries experience repeated flights, charging between flights, and demanding phases such as takeoff and climb. At the measurement level, the challenge is harder because SOH is a latent property; it must be inferred from battery behavior rather than measured directly. That means a reported SOH value can shift because of temperature, load, or charging conditions, even when the battery's actual condition has changed only gradually. This gap between real degradation and observed telemetry is what makes the problem hard and shapes the design challenge; any useful solution needs to separate genuine aging from context-driven noise.

The stakeholders most affected by this problem are aircraft owners, flight schools, and operators responsible for scheduling and using the electric aircraft. Uncertain battery health makes it harder for them to decide how many flights an aircraft can support in a day, whether turnaround times are realistic, and when battery performance has declined enough to justify replacement (Pipistrel, n.d.). This uncertainty also has a direct cost impact. Replacing a battery too early wastes potential useful life, while replacing it too late can reduce reliability and restrict operations. More broadly, poor visibility into battery health lowers scheduling confidence, reduces aircraft availability, and makes fleet planning more conservative.

The impact of the problem is significant in three main ways. First, uncertain SOH can lead to premature battery replacement, increasing costs and reducing the value extracted from each pack. For the Pipistrel VELIS Electro, battery replacement costs are substantial: a 2026 industry report states that the current cost of a battery pack is US$29,050, with some operators reporting even higher total replacement costs once shipping is included (Lightstone, 2026). Across a fleet of aircraft, improper battery usage or overly conservative replacement decisions can therefore become very costly. Second, uncertain SOH creates uncertainty in endurance and reserve planning, making it harder to determine whether a flight can be completed with adequate energy margins. This is especially important in the VELIS Electro because the pilot's operating handbook states that available battery energy depends on both SOC and SOH, and that reduced SOH lowers energy storage capability and therefore reduces endurance (Pipistrel, n.d.). Third, uncertain SOH complicates long-term asset management by obscuring degradation trends and making abnormal behavior harder to identify. Together, these effects reduce utilization and make electric aircraft less attractive as dependable operational assets.

Several approaches are already used to manage this problem, but each only goes part of the way. The most common is reliance on manufacturer BMS estimates and warning indicators for routine monitoring. Operators also use conservative practices such as limiting charge windows, restricting mission profiles, and following fixed maintenance or replacement schedules. Some supplement these methods with manual telemetry review or periodic capacity checks. In research settings, more advanced approaches such as data-driven prognostics, machine learning, and remaining-useful-life models have also been explored. These methods all offer some value, but they also have clear limits. BMS estimates are often the best available onboard indicators, yet they can still be unstable because they depend on the operating context. Conservative policies reduce risk, but they also reduce utilization and may cause batteries to be retired earlier than necessary. Manual analysis is difficult to scale, and predictive models depend on high-quality data and careful handling of noisy measurements. Overall, current efforts only partially solve the problem. Reliable battery-health estimation remains a critical need if electric aircraft are to become practical, cost-effective, and scalable in regular operation.

2. REQUIREMENT ANALYSIS

The final design was defined as a battery-health decision-support system for electric aircraft, not as a telemetry viewer and not as a certified dispatch optimizer. The system must convert noisy BMS and aircraft telemetry into decisions that are useful for maintenance staff, operators, and student users. This requirement set is based on the actual capstone operating context: two demo aircraft, four batteries, and 1,204 event-level records (1,106 for plane 166 and 98 for plane 192). Because raw SOH is unstable and the dataset is small and imbalanced, the design is expected to do three things well: estimate battery health more credibly than the raw BMS signal, forecast degradation over short operational horizons, and convert those outputs into conservative advisory guidance for maintenance and flight planning.

The highest-priority requirements, therefore, focus on latent-state estimation and verification discipline. On February 24, 2026, the team formally dropped the assumption that raw BMS SOH could be used as direct ground truth and replaced it with a requirement for a causal, latent SOH estimate. On the same date, the project scope was narrowed from full dispatch optimization to an advisory planner that ranks lower-stress operating windows and flags infeasible ones. This trade-off intentionally prioritizes credible estimation and conservative operational advice over unsupported claims of exact battery-life prediction or certified dispatch optimization. The appendix provides the full hierarchical requirements table and change history; Table 1 summarizes only the key report-level requirements.

Please refer to Appendix A for a full requirements table, and Appendix B for a detailed requirements evolution log.

Table 1. Summary of Key Requirements

ID	Pri	Requirement summary	Specification target	Verification
FR1	P1	Build a reliable event-level telemetry pipeline	Standardize the full capstone dataset into event records with plane ID, battery ID, timestamp, event type, SOC, current, voltage, temperature, and duration features; preserve support for 2 aircraft and 2 batteries per aircraft	Dataset inspection, schema validation, and successful snapshot export

FR2	P1	Produce a credible latent SOH estimate from noisy telemetry	Output both retrospective latent_soh_smooth_pct and causal latent_soh_filter_pct; for plane 166, reduce implausible upward SOH jumps from 25-29 points raw to below 1 point after smoothing	Raw vs. latent diagnostics, jump analysis, residual, and spike diagnostics
FR3	P1	Forecast near-term battery health and replacement outlook	Support 1-, 5-, 10-, 15-, and 20-flight forecasts and estimate replacement date/days/cycles using a 40% SOH threshold; maintain plane-166 validation MAE <= 0.35 SOH points across supported horizons	Chronological backtesting, holdout-plane reporting, and saved forecast metrics
FR4	P1	Support maintenance and operational planning	Convert SOH into advisory outputs that respect a 30% reserve SOC, reject infeasible charge/flight windows, and rank lower-wear operating days and charge windows	Planner scenario tests, recommendation snapshot review, reserve-feasibility checks
FR5	P2	Communicate results clearly to operators and students	Provide dashboard/API views for health, 30/90-day trend, forecast, recommendations, and glossary-backed explanations; refresh live health at 45 s intervals	Frontend build/lint, API schema checks, Playwright smoke tests
NFR 1	P2	Ensure traceability, reproducibility, and credible validation	Use chronological splits, keep plane 192 as holdout where applicable, and reproduce demo outputs from committed scripts and snapshot artifacts	Pipeline review, regression checks, reproducible export verification

3. SOLUTION DEVELOPMENT

3.1 Conceptual Design Evolution

The final solution was developed through several rounds of design refinement rather than being chosen at the outset. A key challenge in this project was that battery health could not be measured directly with certainty from the available aircraft telemetry, so the team first had to determine what battery-health signal could realistically support forecasting and operational planning.

The simplest concept was to use the aircraft-reported battery-management-system (BMS) state of health (SOH) directly as ground truth. This option was attractive because it was already available at every event and would have enabled a straightforward supervised learning pipeline. However, verification of this idea showed that the reported SOH signal was not reliable enough to serve as the main health label. Exploratory analysis revealed implausibly large upward jumps, strong dependence on event type, and especially unstable behavior around charging events, which is likely due to the battery systems recalibrating. On plane 166, the largest event-to-event upward jumps in the raw SOH signal were 29.0 and 25.0 SOH percentage points across the two batteries. In practical terms, that would mean the battery appeared to regain roughly a quarter to a third of its total health in a single event, which is not physically realistic for real battery aging. Since actual battery degradation should change gradually rather than suddenly recover, this behavior invalidated raw BMS SOH as a trustworthy production target.

We then explored other independent SOH estimation approaches based on methods reported in the battery-health literature, particularly research on electric vehicles and other lithium-ion battery systems. One of these was a charge-event physics-based estimator using coulomb counting and partial-charge capacity reconstruction (Gao et al., 2024). This approach was appealing because it was more physically interpretable and less dependent on the black-box BMS estimator. In validation, it did recover a meaningful degradation trend, which made it useful as an independent reference. However, it remained noisy and incomplete, especially when clean, well-covered charge events were not available. We also explored ICA-style charge-shape analysis (Saini & Renganathan, 2025), which was diagnostically useful but depended on telemetry that was not consistently rich or clean enough to make it robust as a production estimator. The engineering trade-off was therefore clear: these approaches improved interpretability and cross-checking, but they did not provide enough consistency across the full dataset to act as the core health signal.

Based on these results, the team selected a condition-aware latent SOH estimation (Gao et al., 2024) approach as the final estimator. This method extends Kalman filtering by incorporating operational context, such as temperature, load, and charge state, into the observation and transition models(*Kalman filter - an overview | sciencedirect topics*). In doing so, it estimates battery health as a hidden state while accounting for the fact that measurement reliability and system behavior can vary across operating conditions. This directly addressed the core requirement of handling noisy, uncertain observations while still preserving a physically plausible degradation trend. The validation evidence for this decision was strong. On plane 166, smoothing reduced the cumulative event-to-event fluctuation in the SOH signal from 484.0 and 469.0 SOH percentage points to 46.89 and 45.48 across the two batteries, while the largest single event-to-event upward jumps fell from 29.0 and 25.0 SOH percentage points to approximately 0.70. This made the condition-aware latent approach the most credible foundation for downstream forecasting and decision support.

3.2 Final System Architecture

The final design is a layered battery-health decision-support system rather than a single model. It contains four connected parts. First, the telemetry layer ingests and standardizes aircraft data into a structured event-based battery dataset. Second, the SOH estimation layer converts noisy observations into a more trustworthy latent health estimation. Third, the forecasting layer predicts future health and battery-life outlook. Fourth, the frontend planning layer translates those outputs into operator-facing recommendations.

This architecture was selected because it separates the problem into manageable engineering components. Battery condition must first be estimated credibly before it can be forecasted, and forecasts must then be translated into practical planning outputs before they are useful to a user. Structuring the system this way also improves traceability, since each stage has a clear role and can be validated separately. In the report, this section should be supported by a system diagram that shows the end-to-end flow from raw telemetry through battery-health estimation and forecasting to recommendation outputs.

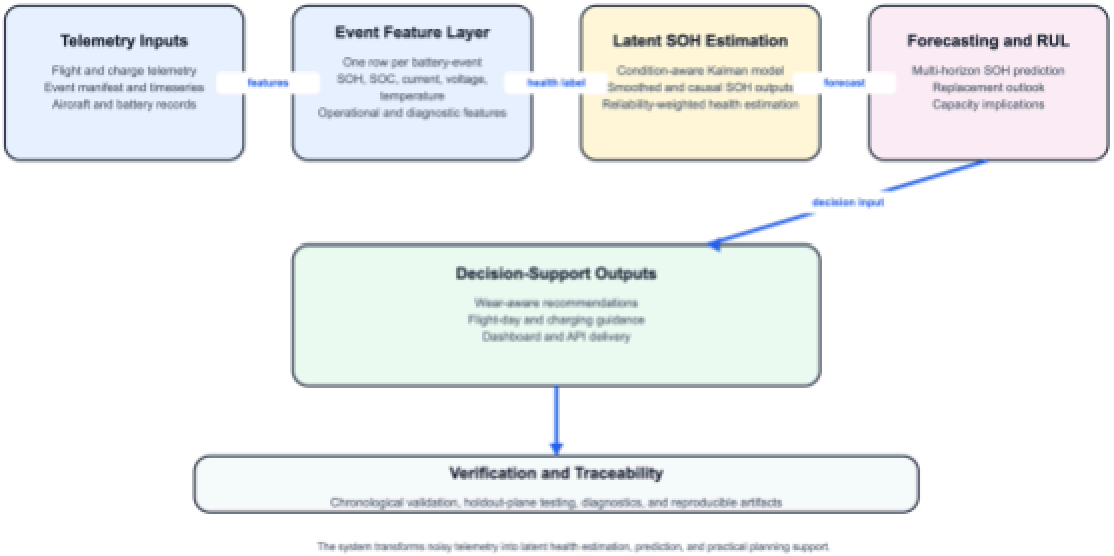

Figure 1: The workflow converts aircraft telemetry into event-level features, latent SOH estimates, multi-horizon forecasts, and decision-support outputs, with verification and traceability applied across all stages.

3.3 Latent Kalman SOH Estimation

The core technical decision in the final design was the condition-aware latent SOH estimator. The goal of this component was to recover a believable battery-health trend from noisy and inconsistent observations. Rather than treating SOH as a directly trusted measured value, the method estimates it as a hidden state using Kalman filtering and smoothing, while incorporating operational context, such as temperature, load, and charge state, into the observation and transition models (Gao et al., 2024). This makes the estimator more stable under varying conditions and better suited to real aircraft operations, where measurement quality and battery behavior are not constant from event to event.

This choice involved the trade-off of introducing more modeling complexity than directly using raw SOH or applying a simpler curve-fitting method, but that added complexity was justified because it produced a signal that was far more usable for the rest of the system. Verification of this design decision came from comparing the estimated latent trajectory against the raw signal and against the independent SOH methods explored earlier. As seen in Figure 1, the latent trajectory removed unrealistic jump behavior while still preserving the same overall direction of degradation. For engineering purposes, this made it a much stronger production label than either the raw BMS SOH or the other estimation methods on their own.

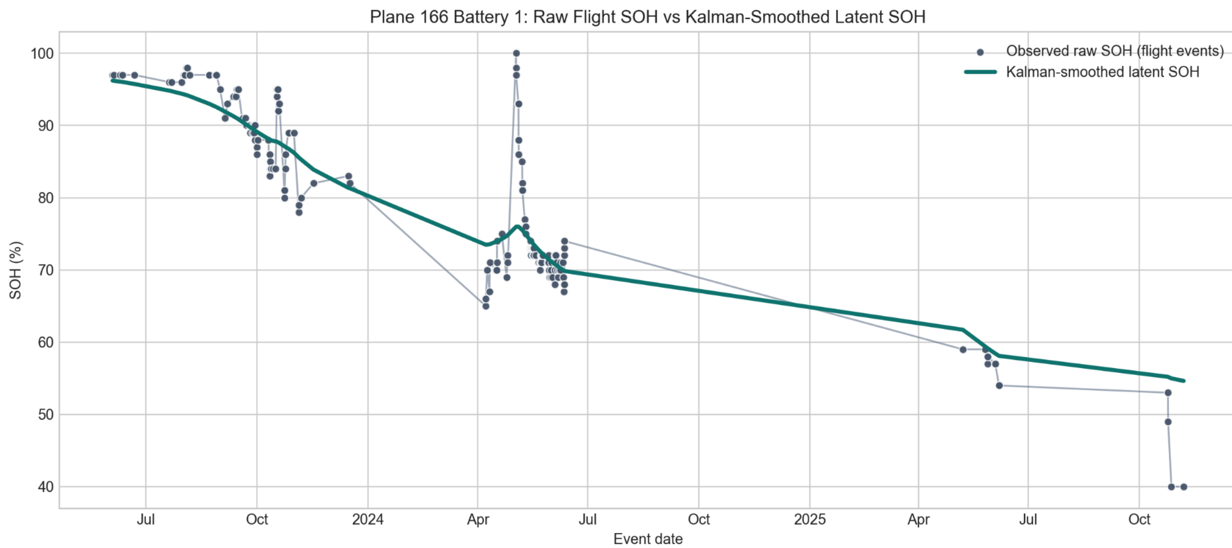

Figure 2: The raw flight event SOH observations compared to the Kalman Smoothed Latent Observations

3.4 Forecasting and Long-Term Life Prediction

Once a credible SOH signal had been established, the next design challenge was forecasting. We evaluated multiple model families for event-based SOH prediction, including linear models, tree-based models, sequence models, and physics-inspired neural models. Sequence models were selected for most committed horizons because they best captured event order and degradation history, making them the strongest option for aircraft-specific forecasting.

Validation through chronological backtesting showed that these models performed well at short horizons, but degraded as the forecast window expanded. The committed benchmark backtest mean level MAE increased from 0.192 at 1 flight to 0.664 at 5 flights, then to 1.140 at 10 flights, and plateaued around 1.165 to 1.187 at 15 to 20 flights. In other words, the 20-flight error was about 6.17x larger than the 1-flight error. This showed that aircraft-specific forecasting was credible for near-term operational prediction, but much less reliable for longer-range extrapolation.

This limitation was not only a modeling issue, but also a data-coverage issue. Our aircraft dataset had not yet reached the true end of life, so it did not contain full beginning-to-end battery degradation trajectories. As a result, long-range forecasts based only on in-domain aircraft history were underconstrained. To address this, we introduced a second forecasting component: a long-term degradation backbone built from an external dataset of other eVTOL batteries with much more complete lifecycle coverage (Bills et al., 2023). This external dataset was used to fit a normalized degradation curve representing the general shape of battery-health decline over life, and our aircraft batteries were then calibrated onto that backbone using their observed SOH trajectory.

The backbone approach was also validated against the aircraft data. For plane 166, the fitted backbone MAE was 1.810 SOH points for battery 1 and 1.629 for battery 2, with an average fitted MAE of 1.719. When the observed aircraft trajectory was compared against the calibrated backbone across all 376 committed flight rows, the overall match MAE was 1.815 SOH points, and the RMSE was 3.001. This showed that the external backbone captured the aircraft's long-run degradation shape reasonably well, even though the in-house fleet had not yet aged to failure. The trade-off was therefore intentional: aircraft-specific sequence forecasts were used for near-term operational prediction, while the external backbone was used for longer-term degradation interpretation and remaining-life reasoning.

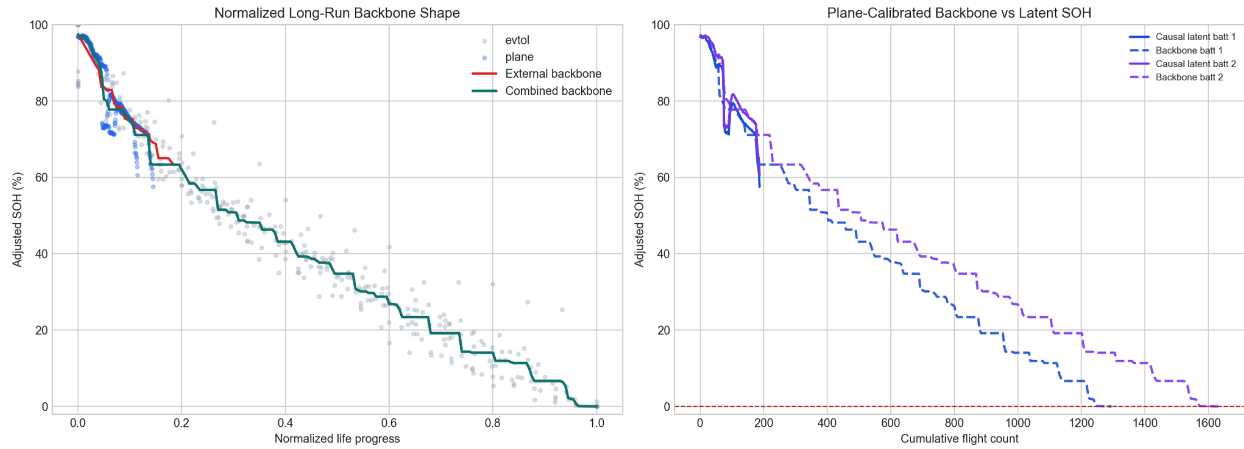

Figure 3 (Left) shows the normalized long-run degradation backbone, where the external backbone is learned from external eVTOL battery test data, and the combined backbone refines that shape using Plane 166's calibrated latent SOH trajectory. Figure 3 (Right) shows the resulting plane-calibrated backbone trajectories against the latent SOH traces for each battery.

Forecast outputs were then translated into more actionable operational metrics. Rather than stopping at predicted SOH alone, the system connected health forecasting to battery-life outlook and aircraft capability. This step was important because users need to understand not only whether battery health is changing, but what that change means for planning, utilization, and replacement timing.

3.5 Frontend and Operational Planning Design

For the frontend and planning layer, we initially considered formulating the problem as a full optimization model. Under that approach, the system would take aircraft health, charging constraints, weather, reserve requirements, and mission demand, then compute an optimal dispatch or operating schedule that maximizes utilization while minimizing degradation. This option was attractive because it would have produced mathematically optimal recommendations rather than advisory guidance.

However, this was not selected as the final implementation for several engineering reasons. First, the available data were not rich enough to support a fully specified optimization model with confidence. We did not have complete maintenance rules, charger availability data, turnaround logistics, or enough fleet-scale operational examples to calibrate a realistic objective. Second, the health forecasts themselves, while strong at short horizons, still carry uncertainty, especially for long-range extrapolation. Embedding uncertain forecasts into a hard optimizer would risk producing brittle recommendations with a false sense of precision. Third, a full optimization framework would have required stronger assumptions about future demand, weather certainty, and operational priorities than the project could defensibly guarantee.

For these reasons, the final frontend was implemented instead as a battery-life-aware advisory recommendation system. It simulates expected wear, reserve feasibility, charging feasibility, and weather impact, then ranks dates and mission windows as recommended, watch, or infeasible. This was a more credible engineering choice because it preserved realism, remained explainable to users, and avoided overstating the certainty of the underlying models. It also aligned better with the available data and with the intended scope of the capstone.

3.5 Practical Application, Deployment Limits, and Integration

A key design decision was to keep the system realistic for practical use. The planning layer was therefore implemented as an advisory recommendation system, not a certified dispatch or scheduling optimizer. It evaluates projected battery wear, reserve feasibility, charging feasibility, and weather-related stress, then ranks candidate dates or mission windows as recommended, watch, or infeasible. This scope was intentional: it provides useful operational guidance without overstating forecast certainty or relying on unsupported assumptions.

The solution was also designed to fit existing workflows. It uses processed telemetry exports and event-based battery records derived from data already collected by the aircraft, so it does not require new onboard sensors or hardware changes. The architecture separates data processing, battery-health estimation, forecasting, and frontend recommendation logic into modular components, which supports easier integration, testing, and maintenance.

At the same time, the team recognized clear deployment limits. Although the models were validated on the available fleet data, broader deployment would require testing across more aircraft, longer battery life histories, and more operating conditions. The recommendation outputs should therefore be treated as decision support, not automated control, since the project did not include full maintenance rules, charger scheduling, turnaround logistics, or certified dispatch constraints. A production deployment would require stronger validation, tighter system integration, and clear governance around how recommendations are reviewed and used.

4. SOLUTION EVALUATION

The final solution was evaluated against the principal project requirements: reliable battery-health estimation, useful forward prediction, and feasibility of integration into an operator-facing decision-support workflow. The results show that the system met the central objective of the project by converting noisy aircraft telemetry into a battery-health estimate and advisory outputs that are more credible and operationally useful than the original BMS-reported SOH alone. However, the evaluation also identified clear limitations, particularly in long-horizon prediction and broader deployment readiness.

The strongest verification result was obtained in the latent SOH estimation layer. The most important technical requirement was that the final design produce a health signal more trustworthy than the raw SOH stream. This was verified by comparing observed SOH with the latent SOH outputs produced by the condition-aware state-estimation pipeline. For plane 166, the raw SOH signal exhibited implausibly large jumps and excessive short-term variation, especially around charge events. In the final latent output, this instability was substantially reduced. Raw total variation per battery was approximately 484 and 469 SOH points, while smoothed total variation decreased to about 46.9 and 45.5. The maximum upward jump was similarly reduced from 29 and 25 SOH points to about 0.70 and 0.73. These results demonstrate that the estimator did not merely smooth the signal visually; it removed physically unrealistic behavior while preserving the long-term degradation trend expected from real battery aging.

Additional diagnostics supported the same conclusion. Spike analysis showed that the largest raw SOH jumps were concentrated around charging conditions rather than normal degradation. For plane 166, spike events were approximately 71-73% charge events, compared with about 53-54% for non-spike events. This indicates that much of the instability in the raw SOH signal was driven by operating context and estimator behavior rather than by true battery recovery. This directly validates the central design decision to treat BMS SOH as a noisy observation of the underlying battery condition rather than as ground truth.

The forecasting layer was evaluated using chronological walk-forward benchmarking and final-test results. The clearest finding was that predictive performance was strongest at short horizons, which is also where the system is most valuable for operational planning. The best committed models achieved a backtest mean level MAE of approximately 0.192 SOH points at the 1-flight horizon and 0.664 at the 5-flight horizon. At longer horizons, error increased to about 1.140, 1.165, and 1.187 at 10, 15, and 20 flights, respectively. This indicates that the forecasting requirement was satisfied convincingly for short-term prediction, but only partially for longer-range forecasting. The system is therefore well supported as a near-term advisory forecasting tool, but not yet as a precise long-term life predictor.

The requirement to translate battery-health estimates into operationally relevant outputs was addressed through the circuit-capacity, SOC-burn, and planning layers. The SOC-rate model achieved a test MAE of approximately 0.192 SOC-%/min, RMSE of 0.263, and test R^2 of 0.555, with out-of-distribution R^2 of about 0.541. These results indicate that the system can represent mission-related battery burden with moderate credibility. The planning layer was verified behaviorally: longer missions increased expected wear, harsher weather worsened projected wear, and restrictive SOC or reserve settings produced infeasible operating days. These outcomes show that the recommendation system responds in a manner consistent with engineering expectations, although it remains an advisory planner rather than a certified optimization system.

Feasibility and compatibility with existing interfaces were also assessed. A significant strength of the final solution is that it operates on existing aircraft telemetry and BMS outputs rather than requiring new onboard instrumentation. The system uses processed event data containing aircraft ID, battery ID, timestamps, SOH, SOC, current, voltage, temperature, and event summaries, making it compatible with current telemetry workflows. The software architecture is also modular, with clear interfaces between telemetry processing, latent SOH estimation, forecasting, planning, and frontend presentation.

Important deployment limits remain. The dataset is small, particularly for cross-aircraft validation: of 1,204 total event rows, 1,106 are from plane 166 and only 98 from plane 192. This limits confidence in fleet-wide generalization. In addition, long-range RUL and replacement predictions remain less certain than short-horizon forecasts and should therefore be interpreted as advisory rather than authoritative maintenance guidance. The current system is best characterized as demonstration-ready rather than production-hardened.

Overall, the evaluation shows that the final solution satisfies the core project requirements. It produces a substantially more credible SOH estimate than the raw BMS signal, supports useful short-horizon forecasting, translates battery condition into practical operational guidance, and integrates with existing telemetry and software interfaces. Its principal limitations are in long-horizon life prediction, fleet-scale validation, and production deployment readiness.

Acknowledgements

The authors would like to sincerely thank our faculty advisors, Dr. Mehrdad Pirnia, Dr. Ada Hurst, and Dr. Chris Rennick, for their guidance, feedback, and support throughout this project. We also gratefully acknowledge Pipistrel Velis Electro as the industry partner whose aircraft platform, operational context, and data environment helped shape the direction and practical relevance of this work. In addition, we thank the University of Waterloo and the Department of Management Sciences and Engineering for providing the academic environment and opportunity to complete this project. The team also acknowledges the use of Generative AI tools as supporting resources during research, ideation, literature exploration, and documentation. These tools were used to support exploration and communication, but all engineering decisions, implementation, validation, and final conclusions remained the responsibility of the team.

References

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering, 82*(1), 35–45. https://doi.org/10.1115/1.3662552

Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA Journal, 3*(8), 1445–1450. https://doi.org/10.2514/3.3166

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Hastie, T., & Tibshirani, R. (1986). Generalized additive models. *Statistical Science, 1*(3), 297–310. https://doi.org/10.1214/ss/1177013604

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). https://doi.org/10.3115/v1/D14-1179

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics, 378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Saini, U., & Renganathan, S. (2025). Incremental capacity analysis of battery under dynamic load conditions. *MethodsX*, *14*, 103331. https://doi.org/10.1016/j.mex.2025.103331

Gao, R., Zhang, Y., & Lyu, Z. (2024). A SOH estimation method of lithium-ion batteries based on partial charging data. *Journal of Energy Storage, 103*(Part A), 114309. https://doi.org/10.1016/j.est.2024.114309

Kalman filter - an overview | sciencedirect topics. (n.d.). https://www.sciencedirect.com/topics/social-sciences/kalman-filter

Gao, R., Zhang, Y., & Lyu, Z. (2024). A SOH estimation method of lithium-ion batteries based on partial charging data. Journal of Energy Storage, 103(Part A), 114309. https://doi.org/10.1016/j.est.2024.114309

Bills, A., Viswanathan, V., Sripad, S., Frank, E., Charles, D., & Fredericks, W. L. (2023). eVTOL battery dataset. Carnegie Mellon University. https://kilthub.cmu.edu/articles/dataset/eVTOL_Battery_Dataset/14226830

Synopsys. (n.d.). What is a battery management system (BMS)? https://www.synopsys.com/glossary/what-is-a-battery-management-system.html

Lightstone, P. (2026, January 8). *Electric aircraft economics*. Wings Magazine. https://www.wingsmagazine.com/electric-aircraft-economics/

Pipistrel. (n.d.). *VELIS Electro pilot's operating handbook*. https://www.manualslib.com/manual/2129606/Pipistrel-Velis-Electro.html

Rahman, T. (2024). *Exploring lithium-ion battery degradation: A concise review of critical factors*. **Batteries, 10**(7), 220. https://doi.org/10.3390/batteries10070220

Appendix

Appendix A. Full Requirements Table

ID	Pri	Requirement description	Specification target	Verification method	Change history
FR1.1	P1	Ingest and standardize telemetry into event-level records	Each record shall include plane_id, battery_id, event_datetime, event_type, SOC information, current, voltage, temperature, duration, and latent-SOH-ready features	Event-table inspection and schema validation	Added 2026-02-12
FR1.2	P1	Preserve aircraft- and battery-level traceability	The dataset shall support at least 2 aircraft and 2 batteries per aircraft, with all outputs traceable by plane and battery ID	Key review in event tables, APIs, and snapshots	Added 2026-02-12
FR1.3	P2	Export processed outputs for the frontend/API layer	Generate typed JSON artifacts for plane health, trend, flights, KPIs, glossary, and monthly recommendations from committed scripts	Snapshot build success and contract validation	Revised 2026-02-16 from static/demo-only outputs to snapshot-backed integration
FR2.1	P1	Estimate latent SOH from noisy BMS observations	Produce latent_soh_smooth_pct and latent_soh_filter_pct; on plane 166, reduce maximum upward SOH jumps to < 1.0 point/event after smoothing and reduce total variation by > 80% versus raw SOH	smoother_summary.json, raw-vs-smoothed plots, spike review	Revised 2026-02-18; raw BMS SOH dropped as sufficient ground truth
FR2.2	P1	Provide a forecasting-safe health label	100% of forecasting targets shall be derived from the causal filter series rather than the retrospective smoother	Feature-pipeline audit and leakage checks	Formalized 2026-02-28

FR2.3	P2	Quantify observation confidence and noise conditions	Store measurement uncertainty plus spike/residual diagnostics for each battery and each processed plane	Diagnostic CSV/plot review	Added 2026-02-28
FR3.1	P1	Support multi-horizon SOH forecasting	Generate future SOH predictions at 1-, 5-, 10-, 15-, and 20-flight horizons for each battery track with sufficient history	Saved benchmark outputs and forecast tables	Added 2026-03-01
FR3.2	P1	Meet a useful short-horizon accuracy level without overstating certainty	On plane 166 validation, level MAE shall be <= 0.35 SOH points at every supported horizon; plane 192 results shall be reported separately as holdout evidence, not used for model selection	best_models_by_horizon.csv, walk-forward validation review, holdout-plane metrics	Revised 2026-03-24 to emphasize advisory use and honest reporting on sparse holdout data
FR3.3	P1	Estimate replacement outlook	Output predicted replacement date, remaining days, and remaining cycles using a 40% SOH replacement threshold	KPI snapshot review and rollout checks	Added 2026-03-01
FR3.4	P2	Extend forecasts with a long-run degradation backbone	Use a calibrated backbone trend only for long-run life projection beyond the well-supported short-horizon region	Backbone calibration plots and rollout review	Added 2026-03-1
FR4.1	P1	Translate SOH into usable capacity and reserve implications	Use the SOH-to-circuit model over SOH grid [0, 20, 40, 60, 80, 100], and SOC-per-circuit values [20, 16, 13, 12, 10, 9], with reserve SOC set to 30%	Circuit-model output review and scenario checks	Added 2026-03-1

FR4.2	P1	Provide wear-aware planning recommendations	Rank candidate operating days/windows by projected wear while rejecting options that violate reserve SOC or available charge windows	Planner scenario tests and recommendation ranking review	Revised 2026-03-24 from optimizer language to advisory ranking language
FR4.3	P2	Recommend health-preserving charging behavior	Provide recommended charge windows and default target SOC guidance that minimizes high-SOC dwell; current recommendation artifacts shall support 80% target-SOC planning and flag avoidable > 95% overnight storage	Recommendation snapshot review and manual scenario inspection	Added 2026-03-2
FR4.4	P2	Include weather/thermal context in operational advice	Recommendation outputs shall include day-level score components for weather, thermal stress, wear/stress, and charging	Snapshot inspection and frontend/API review	Added 2026-03-4
FR5.1	P2	Present results in a user-facing dashboard	The dashboard/API shall expose current health, 30-day trend, 90-day trend, forecast, replacement outlook, flights, and recommendations for each plane	Frontend route review, schema checks, smoke tests	Added 2026-03-01
FR5.2	P3	Improve explainability for non-specialist users	Provide glossary-backed explanations plus at least one educational interaction that links operating choices to SOH impact, range, or RUL	Manual UI review	Added 2026-03-04
FR5.3	P3	Keep the interface operationally responsive for demo use	Live health data shall refresh on a 45 s interval and use typed contracts to avoid silent UI/data mismatches	Frontend inspection and contract checks	Added 2026-03-04

NFR1.1	P1	Prevent data leakage and preserve evaluation credibility	Use chronological train/validation/test splits for plane 166 (70/15/15) and preserve plane 192 as holdout evaluation where applicable	Split review and saved metric tables	Added 2026-03-04
NFR1.2	P2	Ensure planner behavior is physically plausible	Harsher or longer missions and weaker charge timing choices shall never be ranked as lower-wear than milder feasible scenarios under the same conditions	Scenario-based regression tests and manual review	Added 2026-03-04
NFR1.3	P2	Keep outputs reproducible from committed artifacts	Frontend and report/demo outputs shall be rebuildable from committed scripts, saved model artifacts, and snapshot exports rather than ad hoc notebook state	Repo review and export-path verification	Formalized 2026-03-07
NFR1.4	P3	Maintain demonstration compatibility	The system should build and run through the documented frontend deployment path with validated schemas and smoke coverage for key routes	Build/lint checks and Playwright smoke tests	Added 2026-03-07

Appendix B Requirements Evolution Log

The dated requirements log begins in February 2026 because the team pivoted capstone direction during the term after the original e-Zinc battery company collaboration became unavailable. That pivot changed the problem definition, stakeholders, data sources, and verification approach, so the log intentionally starts at the point where the final electric-aircraft battery-health project was defined. Requirements from the pre-pivot concept were not carried forward because they were no longer relevant to the final design evaluated in this report.

Date	IDs affected	Change	Reason/impact
2026-02-12	FR1.1, FR1.2	Initial telemetry ingestion and traceability requirements added.	Established the base scope around event-level monitoring for the two demo aircraft and their batteries.
2026-02-12	FR5.1	Dashboard/API communication requirement added.	Stakeholders needed interpretable outputs for operators and reviewers, not only offline modeling artifacts.

2026-02-18	FR5.2	Explainability requirement added as a lower-priority feature.	Student and non-specialist users needed plain-language interpretation of SOH, trend, and operating choices.
2026-02-18	FR2.2, FR2.3, FR3.1, FR3.3, NFR1.1	Forecasting, replacement-outlook, diagnostics, and leakage-control requirements added.	The event-based ML pipeline had matured enough to support predictive requirements and formal validation rules.
2026-02-19	FR1.3, FR4.3, FR4.4, FR5.3, NFR1.2	Snapshot-export, charging-guidance, weather-context, and demo-readiness requirements added or revised.	Frontend integration required reproducible API-ready outputs and operational recommendations beyond model scores alone.
2026-02-20	FR2.1	Raw BMS SOH as the primary target was explicitly dropped; latent SOH became the canonical health estimate.	Spike diagnostics showed charge-related artifacts and estimator instability, so raw SOH was not credible enough as ground truth.
2026-02-22	FR4.2	Full dispatch optimization requirement dropped; replaced with advisory feasible-window ranking.	Available data and the capstone scope supported conservative battery-aware recommendations, not certified dispatch decisions.
2026-03-01	FR3.2, FR3.4	Forecasting requirements were revised to emphasize short-horizon advisory use and add a long-run backbone trend.	In-domain aircraft data were strongest for near-term prediction but insufficient by themselves for late-life extrapolation.

Appendix C. GitHub Link: https://github.com/benFogerty/CapstoneEPlane

Appendix D. Flight Simulator Demo:
https://drive.google.com/file/d/1GJzkYWT9mOi2u9pxvQJTTjl_-yyXwsSU/view

The flight simulator demo was created for our symposium to teach students and faculty members how flight telemetry is collected per second for all metrics and variable states. It was an extremely interactive and fun way to engage the audience while demonstrating the critical first step in our process: collecting a vast amount of accurate flight data.

<u>Appendix E:</u> Run Website Locally on your machine.

Due to the high machine learning computing, all hosting services were charging a large premium to run the website at a decent speed. As a result, please run the website / Application that the team has developed locally on your machine using the following steps. Please refer to the GitHub link ReadMe for further details. Alternatively, please refer to Appendix F for a detailed demo walkthrough of the app

1. Python environment
   python3 -m venv .venv
   source .venv/bin/activate
   pip install -r requirements.txt
2. Frontend dependencies
   cd frontend
   npm install
3. Build frontend snapshots
   cd frontend
   npm run snapshots
4. Run the app locally
   cd frontend
   npm run dev

The production container path is defined by Dockerfile and render.yaml.

Appendix F: Demo Walkthrough of the Website / Application.

This flight simulator was done for our symposium to teach students and faculty members how the flight telemetry is collected per second for all of our metrics and variable states. It was an extremely interactive and fun way to engage the audience while also showing them the first step in our process, which is collecting the right flight data and a ton of it! It provides a frontend to the team's excellent work and backend calculations.

https://drive.google.com/file/d/1vHKA7hXMG4eVkNDw84qfD23jJ8VqL73j/view?usp=sharing

Appendix G:

Generative AI (GenAI) was used by the team as a supporting tool during research, ideation, and documentation, but not as a substitute for engineering judgment, validation, or implementation. Its role was primarily to accelerate exploration of solution options, support background research, and improve communication of technical work already grounded in the repository, experiments, and design decisions made by the team. One of the main uses of GenAI was methodology ideation during the early and middle stages of the project. Because the project involved noisy battery telemetry, uncertain state-of-health (SOH) labels, and multiple possible modeling directions, GenAI was used to brainstorm candidate machine-learning and estimation approaches that might be appropriate for the problem. This included exploring the suitability of state-space methods, Kalman filtering and smoothing, supervised regression models, multi-horizon forecasting approaches, and decision-support interfaces. These interactions helped the team compare tradeoffs between approaches such as direct raw-SOH prediction, latent-state estimation, event-based modeling, and practical recommendation systems. However, all final methodological choices were made by the team based on repository outputs, diagnostics, and engineering evaluation rather than on GenAI suggestions alone. GenAI was also used for research support and literature discovery. In particular, it helped the team identify useful search directions, keywords, and candidate research topics related to electric-aircraft batteries, battery prognostics, state-of-health estimation, remaining useful life prediction, Kalman-based filtering, and degradation modeling. This was useful for narrowing the literature search space and finding papers more efficiently. The team still reviewed papers manually and selected references based on relevance to the project's technical problem and dataset. In other words, GenAI supported the search process, but did not replace critical reading or source evaluation.

The team did not use GenAI as a black-box generator of final engineering decisions or unverified technical claims. All code, model outputs, visualizations, and system behavior included in the final project were checked against the repository and, where applicable, validated using diagnostics, backtesting, plots, and implementation review. GenAI therefore played a supplementary role: it accelerated ideation, exploration, literature search, explanation, and communication, while the core engineering work, implementation, validation, and final design decisions remained the responsibility of the team.

Example Prompt Types Used

Methodology ideation: "Given a noisy battery SOH signal from aircraft telemetry, what modeling approaches could be used to estimate a hidden health state before forecasting future degradation?" *Forecasting strategy exploration:* "What are the tradeoffs between predicting next-event battery SOH change and predicting multi-horizon degradation over the next 5 to 20 flights?" *Literature search assistance:* "Suggest search terms and paper topics related to battery state-of-health estimation, Kalman filtering for battery systems, and remaining useful life prediction for lithium-based batteries."*Technical explanation support:* "Explain the difference between a Kalman filter and a Kalman smoother in a way that is clear for an engineering capstone report."