COMPUTATIONAL INTELLIGENCE AND MACHINE LEARNING
e-ISSN: 2582-7464

Computational Intelligence and Machine Learning
Volume 6, Issue 2, October 2025

Integration of Large Language Models and Artificial Intelligence in Aeronautical Systems Engineering

Shaul Eliahou-Niv

Aeronautical Engineering, Technion, Haifa, Israel
Corresponding Author Email: seliahou@technion.ac.il

Abstract

In recent years, artificial intelligence systems have become an integral part of modern life, permeating both personal domains and a wide range of professional and technological fields. These technologies are increasingly being adopted across disciplines such as engineering, medicine, transportation, and defence, expanding the scope of automation and contributing to significant improvements in efficiency, precision, and performance. We are now entering an era in which artificial intelligence systems can perform tasks that were once considered uniquely human, including complex data analysis, decision-making, and even the generation of new knowledge. To fully harness the potential of artificial intelligence, organizations and individual users must undertake substantial adjustments to their operational models and workflows, integrating these technologies in a systematic, controlled, and meaningful manner. This evolution raises a fundamental question: to what extent can artificial intelligence systems be trusted to support decision-making processes, and what level of human intervention remains necessary to ensure quality, ethical integrity, and professional accountability? Moreover, it is essential to establish robust evaluation frameworks, including quantitative metrics and benchmarking methodologies, that enable objective comparisons and provide clear, evidence-based assessments of artificial intelligence performance. This study explores these issues within the context of systems engineering, with a particular focus on the field of aeronautics. The research aims to delineate the boundaries of autonomous artificial intelligence capabilities and to identify the conditions required for optimal integration between human judgment and advanced technological systems.

Keywords

Artificial intelligence, benchmark, decision-making, systems engineering.

INTRODUCTION

In the domain of engineering design, and particularly in systems engineering, Large Language Models (LLMs) are emerging as powerful tools for significantly enhancing the design process [1]. Models such as GPT-4 [2] offer substantial potential, especially in the early stages of design and conceptualization, due to their ability to rapidly search information repositories and compare a wide range of reported solutions. These capabilities are particularly relevant to the initial phases of engineering design, which typically involve multiple iterations, data synthesis, and evaluation of existing solutions found in literature or embedded in the model's training data. LLMs have proven effective in supporting engineers by proposing ideas and concepts, and potentially even applying methodologies and workflows within systems engineering processes, thereby accelerating and improving the overall design effort [3].

Nonetheless, the potential of LLMs may extend far beyond these capabilities. Open questions remain regarding their ability to support decision-making processes, exercise judgment, and account for constraints and ecosystem-level considerations. In the development of complex systems—such as those in transportation and aeronautics—systems engineers are often required to make decisions under constraints and apply human judgment beyond optimal decision-making within technical boundaries. These challenges are especially evident in critical stages such as translating customer needs into engineering requirements, selecting among alternatives, and managing risks [4].

A key challenge in systems engineering is the comparison of different technological solutions, such as hydraulic versus electric steering systems [5]. Complex systems are characterized by high investment and stringent requirements for reliability and safety, necessitating quantitative analysis of new concepts to justify further development. Such analysis is typically conducted through computer-based simulations, which require precise definition of engineering concepts. In recent years, increasing efforts have been made to integrate AI-based tools into these processes, aiming to enhance simulation accuracy, automate model calibration, and support decision-making under uncertainty. [6]. Disciplines such as systems engineering and knowledge-based engineering have been developed to address these challenges, though they often demand significant human resources [7]. In many cases, the development of new product generations in systems engineering does not begin from a blank slate but rather builds upon existing system architectures. This evolutionary approach enables organizations to leverage proven design foundations while integrating emerging technologies to enhance functionality, performance, and adaptability. Such integration often involves reconfiguring system components, updating interfaces, and ensuring compatibility across legacy and novel subsystems. This strategy not only reduces development time and cost but also mitigates risks associated

with entirely new designs. Recent studies in systems engineering emphasize the importance of architectural reuse and modularity as key enablers of innovation and scalability in complex systems [8].

Textual representations of system architectures—such as mappings between system functions and product elements—make this domain a promising candidate for LLM applications [9]. Recent studies suggest that integrating LLM's with Model-Based Systems Engineering (MBSE) frameworks, such as Object-Process Methodology (OPM), may enhance the accuracy and reliability of engineering design [10].

This paper presents an initial investigation into the potential of LLM's to augment early-stage design of complex systems and to evaluate their decision-making capabilities. Through a comparative case study, we aim to answer the following questions: What are the challenges and opportunities of using LLM's to solve complex engineering problems? And can LLM's effectively support decision-making processes? We examine whether performance improvements in LLM's also lead to enhanced decision-making capabilities.

MOTIVATION

The primary objective of this study was to evaluate the effectiveness of a Large Language Model (LLM) in addressing a structured engineering problem through the application of systems engineering methodologies within the field of aeronautics. A broader aim of the research was to compare the performance of human participants with that of the LLM in executing a complex engineering task, and to identify methods and strategies for enhancing the model's capabilities.

The engineering challenge selected for this investigation was adapted from a project assigned to undergraduate students enrolled in a systems engineering course at the Faculty of Aerospace Engineering, Technion – Israel Institute of Technology. As part of the coursework requirements, students were provided with a set of customer-defined project specifications and were tasked with applying systems engineering principles to develop a feasible solution. Their work was assessed through a series of executive design reviews, simulating professional design and evaluation processes commonly employed in industry.

Methodology

The experimental framework was structured to systematically evaluate the performance of an LLM in addressing a complex engineering problem using systems engineering methodologies. The methodology comprised the following stages:

Problem Definition

A clearly defined engineering challenge was selected, derived from a real academic assignment, to ensure relevance and alignment with established systems engineering practices.

Customer Requirements

A comprehensive set of customer requirements was provided, serving as the foundational input for the engineering process and guiding the development of the proposed solution.

Application of Systems Engineering Methodologies

Participants were required to apply a broad suite of systems engineering tools and techniques, including:

- Requirements Analysis: Decomposition of customer needs into detailed, actionable specifications.
- Functional Analysis: Identification and definition of system-level functions necessary to meet the requirements.
- Conceptual Design: Generation of multiple design concepts consistent with the defined requirements.
- Trade Studies: Comparative evaluation of alternative designs based on performance metrics, cost considerations, and associated risks.
- Risk Management: Identification of potential risks and formulation of mitigation strategies.
- Market Analysis: Survey of existing technologies to identify capability gaps and innovation opportunities.
- Alternative Selection: Structured selection of the most suitable design concept based on multi-criteria evaluation.
- System Architecture: Definition of system components, their interfaces, and interactions.

Executive Reviews

Progress was assessed at key milestones through formal design reviews, emulating industry-standard evaluation processes:

- Preliminary Design Review (PDR): Assessment of initial design concepts and their alignment with customer requirements.
- Critical Design Review (CDR): Evaluation of the detailed design and its readiness for implementation.

Final Design Proposal

Based on feedback received during the executive reviews, participants refined their solutions and submitted a final design proposal. This included comprehensive documentation of the design process and a justification for the selected solution.

Evaluation of LLM Performance

The LLM was evaluated based on its ability to contribute meaningfully to each stage of the systems engineering process. This included its effectiveness in generating design concepts, conducting trade studies, managing risks, performing market analysis, selecting alternatives, and supporting executive reviews.

Preliminary Findings

Initial results revealed that the LLM encountered difficulties in adhering to a structured systems engineering framework when attempting to address the problem in a

holistic manner. The generated outputs lacked coherence and substantive engineering value. However, when the problem was decomposed into discrete, well-defined tasks and the LLM was guided to address each stage independently, its performance improved significantly. Under these conditions, the model demonstrated satisfactory proficiency in handling individual components of the systems engineering process.

To improve the model's performance in addressing complex engineering problems, a customized Retrieval-Augmented Generation (RAG) infrastructure was developed. This framework was specifically tailored to support the LLM within the context of systems engineering, enabling more accurate and context-aware responses by integrating relevant domain-specific knowledge throughout the problem-solving process.

These results served as a foundation for the subsequent phases of the study, guiding further refinement of the methodology and informing the evaluation of the model's capabilities under enhanced conditions.

COMPERISON OF HUMAN AND MACHINE PERFORMANCE

This section presents a systematic comparison between the performance of the Large Language Model (LLM) and that of human participants across the distinct phases of the systems engineering process. The evaluation focuses on the model's ability to replicate structured engineering reasoning and decision-making, benchmarked against human execution of equivalent tasks.

Requirements Analysis

A project was defined for the LLM with client requirements like those defined for the student group that served as the control group. The objective of this project was to design a drone that would function as a hovering platform for a public address system, intended for advertising and delivering messages to audiences during football matches or musical performances held in open-air stadiums.

The following table presents the results of the engineering requirements derivation phase, based on the initial customer requirements.

Table 1. Conversion of Customer Requirements into Engineering Requirements as Generated by the AI Model

Client Requirements	Engineering Requirements
Project budget is 15K$	Development, production, and assembly cost of the prototype will not exceed $15,000.
Project duration for prototype completion is 6 months from the start date	A functional prototype will be completed within 6 months from the project start date.
The drone will be capable of maneuvering in an urban environment.	The drone will be designed with a navigation and stabilization system that allows maneuvering between buildings of varying heights, including Radar/LiDAR sensors for collision avoidance, Accurate GPS system with Visual SLAM backup.
Operate within a radius of 2 kilometers around its operator	The drone will be designed to operate within a 2 km radius from the operator, including: Stable two-way communication (e.g., RF or LTE), Automatic return system in case of signal loss.
The drone will be equipped with a system comprising multiple loudspeakers and amplifiers, seamlessly integrated into a public address infrastructure, thereby facilitating the transmission of audio messages in accordance with the operator's directives.	The drone will carry a speaker system with a total weight of up to X grams (to be determined based on component selection), with: Remote control over broadcast content, Sufficient sound volume for clear hearing from a height of 15 meters.
The drone will be able to maneuver at a height of 15 meters so that it can be heard by the surrounding people	The drone will be designed to operate at a height of 15 meters above ground level, maintaining stability and full functionality of the public address system.
The drone will be designed to sustain uninterrupted flight operations for up to two hours without necessitating a landing.	The drone will be able to operate continuously for at least two hours without the need for landing or recharging, including: Suitable battery capacity, Efficient energy management system.
Other requirements should be according to engineering discretion	Total weight of the drone will not exceed regulatory limits (e.g., 2.5 kg according to local regulations), Resistance to light to moderate wind (up to 20 km/h), Simple maintenance and quick component replacement, Compliance with civil aviation regulations (CAA/FAA/UAV local regulations).

As shown in table 1, the model produced a wide range of engineering requirements derived from the customer inputs. However, these outputs were generated without consideration for schedule or cost constraints. This behavior likely reflects

the model's capacity for extensive information retrieval, coupled with its limited ability to apply contextual judgment or prioritize constraints.

Alternative Selection

During the alternative selection phase, the LLM model was instructed to identify four viable design alternatives that fulfill the specified engineering requirements, and to present a comparative analysis highlighting the respective strengths and weaknesses of each proposed solution.

The LLM model exhibited strong search capabilities, identifying multiple potential solutions. Nevertheless, it failed to align its proposals with the defined project requirements, frequently suggesting alternatives that exceeded the specified operational or technical needs. This indicates a tendency to optimize for completeness or generality rather than constraint-driven feasibility.

The four most relevant alternatives identified by the LLM are detailed in Table 2 below

Table 2. The four most relevant alternatives identified by the model

Configuration	Quadrotor Light	Hexacopter Professional	Foldable Drone	Professional DJI Matrice
Structure	Quadrotor (4 motors)	Hexacopter (6 motors)	Foldable Quadrotor	Commercial Platform (DJI Matrice 30 or similar)
Battery	Li-Ion 6S 10,000mAh	Li-Po 6S 16,000mAh	Li-Ion 8,000mAh + small solar panels	Dual smart batteries
Sensors	GPS, LiDAR, IMU	GPS RTK, Visual recognition camera, LiDAR	GPS, Camera, Ultrasonic sensors	GPS RTK, Thermal/Optical cameras, LiDAR
PA System	Bluetooth speaker, 300g	Professional PA system with remote control	Lightweight speaker with app control	Dedicated PA module from DJI or third party
Range	Up to 2 km	2.5 km	Up to 2 km (LTE)	Up to 7 km (limited to 2 km as per requirement)
Flight Time	Approx. 2 hours	90–120 minutes	Up to 2 hours (depending on sunlight)	40–55 minutes (requires battery replacement)
Advantages	Low cost, Easy maintenance, Suitable for urban environment	High stability, good payload capacity, Wind resistance	High mobility, Stable communication, Energy saving	High reliability, Full integration, technical support
Disadvantages	Sensitive to wind, Limited payload capacity	More expensive, more complex maintenance	Dependence on sunlight, less stable in wind	Very expensive, Requires advanced licensing

Risk Analysis

The risk management section is described in the following paragraph and summarized in Table 3.

A review of Table 3 indicates that a significant portion of the risks identified by the model stem from overestimation during the formulation of engineering requirements and the selection of design alternatives. These elements were generated without adequate consideration of budgetary constraints, scheduling limitations, or performance boundaries. A deeper analysis suggests that this outcome is primarily due to the model's limited ability to make informed and context-sensitive decisions. Moreover, this limitation becomes increasingly pronounced as the range of available options and the volume of retrieved information expands, leading to a higher likelihood of misalignment with project constraints.

Table 3. The risk management table proposed by the model

#	Risk Area	Risk	Impact	Mitigation
1	Project Budget	Exceeding the defined budget of $15,000	Additional costs may cause project delays or termination	Close monitoring of expenses, using cheaper components, emergency budget
2	Project Duration	Not meeting the 6-month timeline for prototype completion	Project delays may result in not meeting client requirements	Detailed scheduling, tracking progress, backup plan for delays

#	Risk Area	Risk	Impact	Mitigation
3	Urban Manoeuvring	Navigation/stabilization failure may cause collisions	Damage to drone / buildings, safety risks	Comprehensive testing, LiDAR sensors, accurate GPS with Visual SLAM backup
4	Operational Range	Failure in two-way communication	Loss of drone, safety risks	Stable RF/LTE communication, automatic return system
5	Public Address System	System failure may result in unmet client requirements	Not meeting project requirements	Comprehensive testing, reliable components
6	Flight Time	Energy management failure may shorten flight time	Not meeting client requirements	Suitable batteries, efficient energy management
7	Drone Weight	Exceeding allowed weight may violate regulations	Not meeting civil aviation regulations	Careful weight planning, lightweight components
8	Wind Resistance	Instability in wind may cause loss of control	Damage to drone, safety risks	Testing in wind conditions, suitable motors and sensors
9	Maintenance	Complex maintenance may cause delays	Project delays, additional costs	Maintenance planning, easy-to-replace components
10	Regulatory Compliance	Not meeting aviation regulations may lead to disapproval	Project disapproval, delays	Testing for compliance, collaboration with aviation authorities

Project Uncertainty Assessment

The following table presents the estimated uncertainty ranges for key parameters in the drone develop Estimated Overall Project Uncertainty: ~27.5%ment project. Each entry includes the parameter name, the associated uncertainty range, and relevant notes.

Table 4. The estimated uncertainty ranges for key parameters

#	Parameter	Uncertainty Range (%)	Notes
1	Project Budget	20%–30%	Risk of exceeding $15,000 due to technical challenges or scope changes
2	Project Duration	25%–35%	Potential delays during development and testing
3	Navigation & Stabilization System	30%–40%	Complexity in urban manoeuvring and sensor integration
4	Stable Two-Way Communication	20%–30%	Signal interference in urban environments
5	Public Address System	25%–35%	Achieving sufficient sound output from 15 meters altitude
6	Maneuvering at 15 Meters Height	20%–30%	Stability under wind and altitude constraints
7	Two-Hour Operation Time	25%–35%	Energy management and endurance limitations
8	Regulatory Compliance	15%–25%	Uncertainty due to evolving UAV regulations

Program to Reduce Uncertainty and Cost Impact Summary

As part of the risk mitigation and uncertainty reduction strategy proposed by the LLM, the integration of commercially available navigation and public address systems was recommended. These off-the-shelf solutions are typically well-established and have undergone extensive validation, thereby minimizing technological risk. Their implementation enables accelerated development timelines and increases the likelihood of meeting strict project deadlines. Additionally, the use of pre-engineered systems reduces the need for internal research and development, resulting in significant cost savings. Such systems generally adhere to regulatory standards, thereby reducing compliance-related uncertainties. Moreover, they enhance communication reliability and maneuverability, particularly at operational altitudes of approximately 15 meters. The deployment of pre-tested components contributes to overall system robustness and minimizes the necessity for post-deployment modifications. A revised assessment of project

uncertainty, conducted in accordance with the mitigation strategy proposed by the model, reveals a moderate reduction in risk levels. Specifically, the uncertainty associated with the navigation and stabilization system decreased from an initial range of 30%–40% to 10%–20%, while the uncertainty related to the public address system declined from 25%–35% to 10%–20%. As a result, the overall project uncertainty was recalculated to approximately 23.1%. However, this reduction in uncertainty is relatively marginal when considered in the context of the financial costs associated with implementing the proposed mitigation measures.

Table 5. Program to Reduce Uncertainty and Cost

Aspect	Impact
Technological Uncertainty	Reduced due to proven systems
Development Time	Shortened by using ready-made systems
Development Costs	Lower due to reduced R&D and integration
Regulatory Compliance	Improved standard-compliant systems
Communication & Stability Risks	Reduced with reliable systems at 15m height
System Quality	Higher due to pre-tested components
Navigation System Uncertainty	Reduced to 10%-20% (from 30%-40%)
Public Address System Uncertainty	Reduced to 10%-20% (from 25%-35%)
Overall Project Uncertainty	Reduced to approx. 23.1%
Cost of Purchasing	Potentially higher, but immediate and reliable
Savings on Development	Significant reduction in internal costs
Time Savings	Economic advantage due to faster delivery
Risk Reduction	Lower costs from fewer unexpected issues

Decision Making - Decision Tree

The LLM model was tasked with deciding regarding the most suitable alternative among those presented in Table 2. While the model was unable to perform decision-making in a structured manner and lacked familiarity with methodologies such as decision trees, as commonly applied in systems engineering, it nevertheless attempted to select an option by identifying the alternative most frequently associated with various attributes.

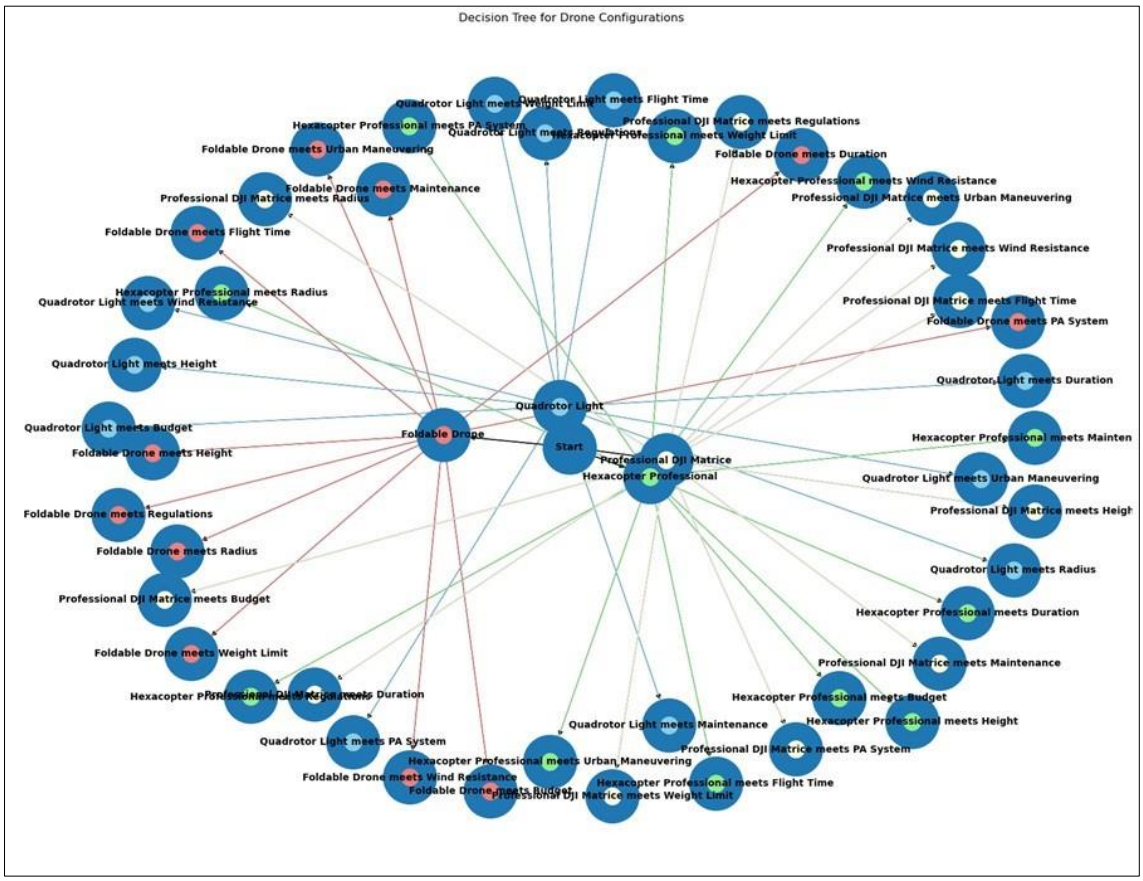

Figure 1. Decision making process suggested by the LLM

Clearly, this approach is neither accurate nor methodologically sound. Decisions are made through a form of parameter-based voting, without any weighting or probabilistic consideration. The decision made by the model relies solely on a voting mechanism involving parameters associated with each alternative, assigning equal weight to all parameters. Although this method is objectively flawed in terms of outcome validity, it appears that the model genuinely applied a consistent internal logic in its decision-making process.

Project's Milestones

The LLM model was tasked with identifying the project's key milestones and mapping them along a timeline. Although the model was able to define the project's milestones, it failed to implement parallel processes within the Gantt chart it generated. As a result, the transition from one milestone to the next was strictly sequential, with no ability to recognize processes that could be executed concurrently as illustrated in figure 2 below.

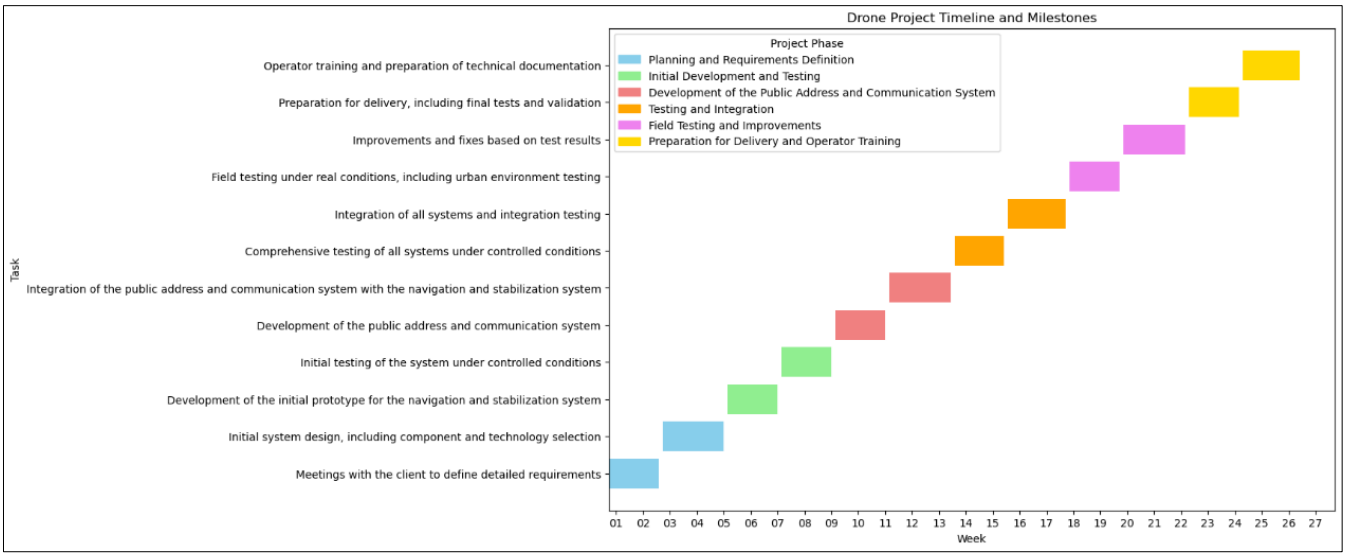

Figure 2. Project's key milestones identified by the LLM

PERT Analysis

Surprisingly, the model was able to perform a Program Evaluation and Review Technique (PERT) analysis, identify the critical path, and even present it visually in a diagram.

In systems engineering, PERT is a statistical project management method used for analyzing and planning tasks within a project, particularly when there is uncertainty in task durations.

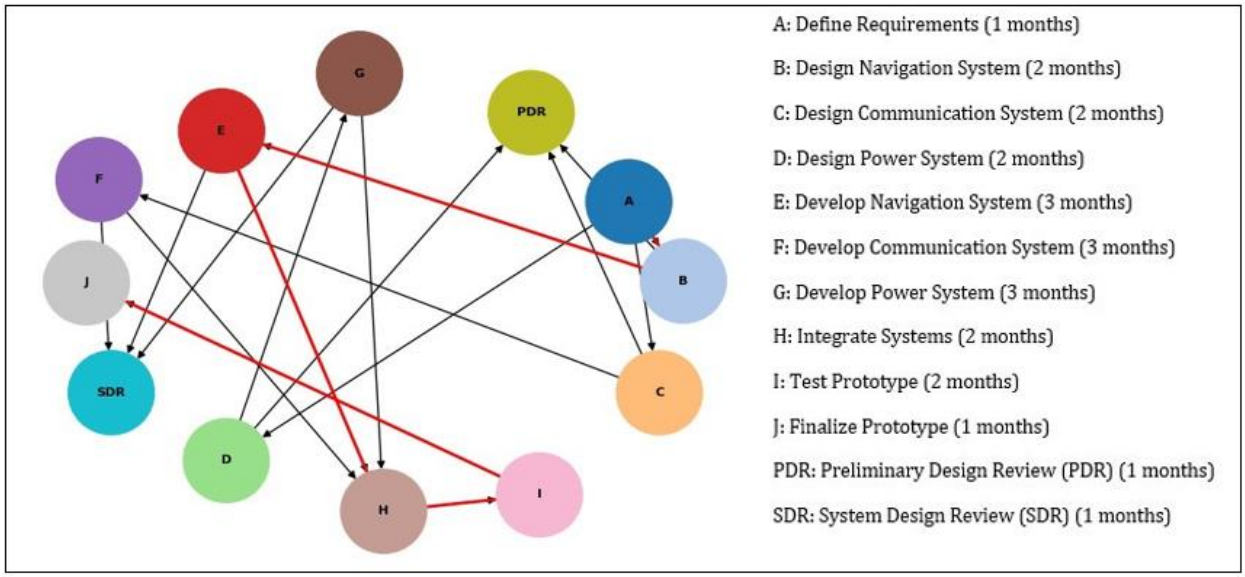

Figure 3. Program Evaluation and Review Technique analysis proposed by the LLM

The critical path is highlighted in red on the PERT diagram presented in Fig. 3. The critical path represents the sequence of tasks that determine the minimum project duration. Any delay in these tasks will directly impact on the overall project completion time.

The critical path that was identified for this project includes the following sequential phases: Define Requirements → Design Navigation System → Develop Navigation System → Integrate Systems → Test Prototype → Finalize Prototype. The total duration of this critical path is estimated at 11 months.

According to the project requirements, the prototype should be completed within 6 months. Therefore, the current project schedule does not meet the required timeline. It is necessary to re-evaluate and optimize the project plan to meet the deadline.

The task causing the most delay in the critical path is 'Develop Navigation System', with an estimated duration of three months. This task has a significant impact on the overall project timeline and should be carefully evaluated for potential optimization strategies. This placed the model in a dilemma: it had to choose between purchasing a navigation system—thereby meeting the required project timeline but exceeding the budget—or developing the system in-house, which would keep the project within budget but result in a schedule overrun. It is important to note that the navigation system was introduced into the project by the model itself during the engineering requirements definition phase, although it is not necessarily essential for meeting the original customer requirements.

If the decision is made to purchase a navigation system instead of developing one in-house, it could result in saving two months of design time and three months of development time.

INTERNALLY DEVELOPED AND VALIDATED APPROACHES FOR IMPROVING LLM EFFICIENCY AND ASSESSING DECISION-MAKING IMPACT

To overcome the limitations observed in large language models (LLMs) and to leverage their rapid retrieval capabilities across diverse data repositories including those utilized during pretraining, we developed and implemented a Retrieval-Augmented Generation (RAG) framework. This architecture enables dynamic contextual enrichment by incorporating external, domain-specific datasets, thereby addressing inherent knowledge gaps in the base model. The conceptual foundation and methodological implementation of this approach are detailed in our previous publication: *'Leveraging the Performance of Large Language Models in Systems Engineering Applications', Journal of Information Systems Engineering and Management (JISEM), Vol. 10 No. 27s, 2025.*

This framework enhances an existing LLM by retrieving relevant documents from a curated knowledge dataset composed of the following systems engineering documents and textbooks:

- USA Department of Defense: Systems Engineering Guidebook [10].
- MITRE: Systems Engineering Guide [11].
- DAU: Systems Engineering Fundamentals [12].
- INCOSE: SE Handbook, Version 2a [13].
- Systems Engineering Body of Knowledge (SEBoK) [14].

In our previous studies, we evaluated similar RAG-based frameworks using a streamlined version of the benchmark—typically by sampling questions at fixed intervals, thereby assessing approximately one-tenth of the full dataset. In contrast, the present study extends the evaluation to encompass the entire benchmark question set, offering a more comprehensive assessment of model performance.

Within the RAG framework, each benchmark query is augmented by appending a set of k relevant documents, retrieved from the data corpus via a k-Nearest Neighbors (k-NN) search based on the original query string. This augmentation aims to enhance the model's ability to generate accurate responses by enriching the input context with semantically relevant information.

In this work, we systematically investigate the impact of varying the parameter k on model performance. While a lower k is preferable from a computational efficiency standpoint—resulting in shorter input sequences, fewer tokens, and reduced inference cost—larger values of k often lead to improved accuracy due to the inclusion of more contextual information. This trade-off between computational efficiency and response quality is critical for practical implementations of RAG-based systems. The findings presented herein are intended to guide optimal parameter selection strategies for future applications.

In this study, we implemented the Retrieval-Augmented Generation (RAG) framework on four open-source language models to evaluate its effectiveness across different model architectures and sizes. The selected models include:

google/gemma-2-2b-it
openchat/openchat_3.5
microsoft/Orca-2-7b
CohereLabs/c4ai-command-r7b-12-2024

Among these, google/gemma-2-2b-it is a 2-billion-parameter model, whereas the other three models each contain 7 billion parameters. Following the initial implementation, our analysis primarily focused on the performance of google/gemma-2-2b-it, due to its compact architecture and promising results when integrated with the RAG framework. This comparative setup enabled us to explore the trade-offs between model size, computational efficiency, and the benefits of retrieval-based augmentation in generating accurate and contextually enriched responses.

MODEL SELECTION AND INTEGRATION WITH RAG FRAMEWORK

For extensive profiling of the infrastructure under investigation, we selected the Gemma 2 2B, a lightweight, open-source, decoder-only large language model developed by Google AI as part of the broader Gemma family. With 2 billion parameters, Gemma 2 2B is optimized for deployment in environments with limited computational resources, such as laptops, desktops, or private cloud infrastructure.

Despite its relatively small size, the model demonstrated impressive capabilities in learning from retrieved data. As expected, the foundational model exhibited lower baseline performance compared to larger models. However, its accuracy significantly improved when queries were enhanced using a Retrieval-Augmented Generation (RAG) framework. This integration yielded notable performance gains, particularly in systems engineering tasks.

The enhancement is primarily attributed to the model's ability to leverage contextually relevant documents retrieved from external corpora. These documents enrich the input context and reduce reliance on the model's internal knowledge. Given the complexity and multi-domain nature of many systems engineering queries, a single document often fails to provide comprehensive coverage. The RAG mechanism addresses this limitation by retrieving multiple documents that collectively span the necessary domains, thereby enabling the model to generate more accurate and complete responses.

Furthermore, Gemma 2 2B's compact architecture facilitates efficient deployment in resource-constrained environments, making it an ideal candidate for scalable and cost-effective RAG-based solutions. The observed reduction in hallucinations and increase in contextual precision further underscore the value of this integration in real-world engineering applications.

To evaluate performance, we executed the benchmark on our custom RAG infrastructure, comprising 1,144 systems engineering questions, distributed across systems engineering topics. The questions taken from Hagging Face SysEngBench https://huggingface.co/datasets/rabell/SysEngBench. The distribution of question categories is illustrated in the accompanying diagram.

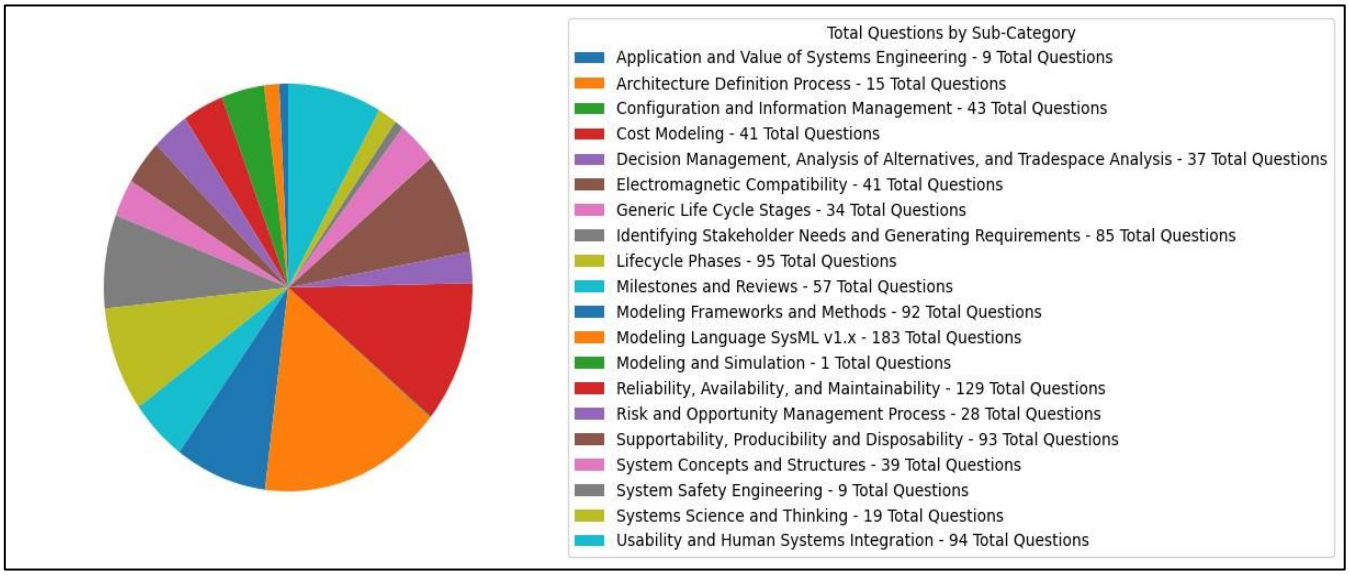

Figure 4. The distribution of the benchmark question categories

RESULTS

Selecting an appropriate value for *k* in the k-Nearest Neighbors (k-NN) algorithm is critical for achieving optimal model performance. A small *k* (e.g., 1–3) tends to make the model highly sensitive to noise and outliers, which can lead to overfitting. In contrast, a larger *k* smooths predictions by incorporating a broader set of neighbors, thereby reducing variance but potentially introducing bias and obscuring local patterns.

To identify the optimal balance between bias and variance, practitioners commonly evaluate model performance across a range of *k* values using cross-validation techniques. Additionally, the use of weighted k-NN—where closer neighbors exert greater influence can improve accuracy without necessitating a large *k*. Ultimately, the choice of *k* should be guided by empirical testing and informed by the specific characteristics of the dataset, including class distribution and dimensionality.

In our study, we assessed the performance of the k-NN algorithm using three distinct values of *k*. For each configuration, we measured classification accuracy, as summarized in Table 5. This comparative analysis enabled us to observe the influence of *k* on predictive performance. The results underscore the sensitivity of model accuracy to the choice of *k*, highlighting the importance of empirical validation in parameter selection.

	No RAG	top_k = 1	top_k = 10	top_k = 20
Total Accuracy (%)	67.05	83.13	84.97	85.66

Table 6: Impact of K Value on the Classification of the LLM Accuracy

Comparison between the best result (top_k=20) and the No RAG case, analyzed by the questions' categories in the benchmark (including only the top 15 categories), is provided in table 5.

Gemma 2 2B demonstrated significant improvement, with accuracy increasing from 63.75% to 75.0% corresponding to a reduction of 31.03% in the error rate. The best results were achieved using LLM as a judge and retrieving only the top 1 relevant document. On the custom corpus, Gemma 2 reached an accuracy of 80%, showcasing its ability to handle concise and highly relevant information effectively.

Figures 5 and 6 below illustrate, respectively, the improvement in accuracy and confidence score of the Gemma 2 2b model. These figures present the enhancements achieved across the 15 leading categories in the benchmark. As shown, the model's accuracy improved by an average of 27.77%, rising from 67.05% without the RAG infrastructure to 85.66% with RAG integration. Similarly, the confidence score increased by an average of 46.66%, from 58.17% without RAG to 85.32% with RAG. These findings underscore the significant contribution of RAG-based augmentation to both predictive accuracy and model confidence across a wide range of systems engineering domains.

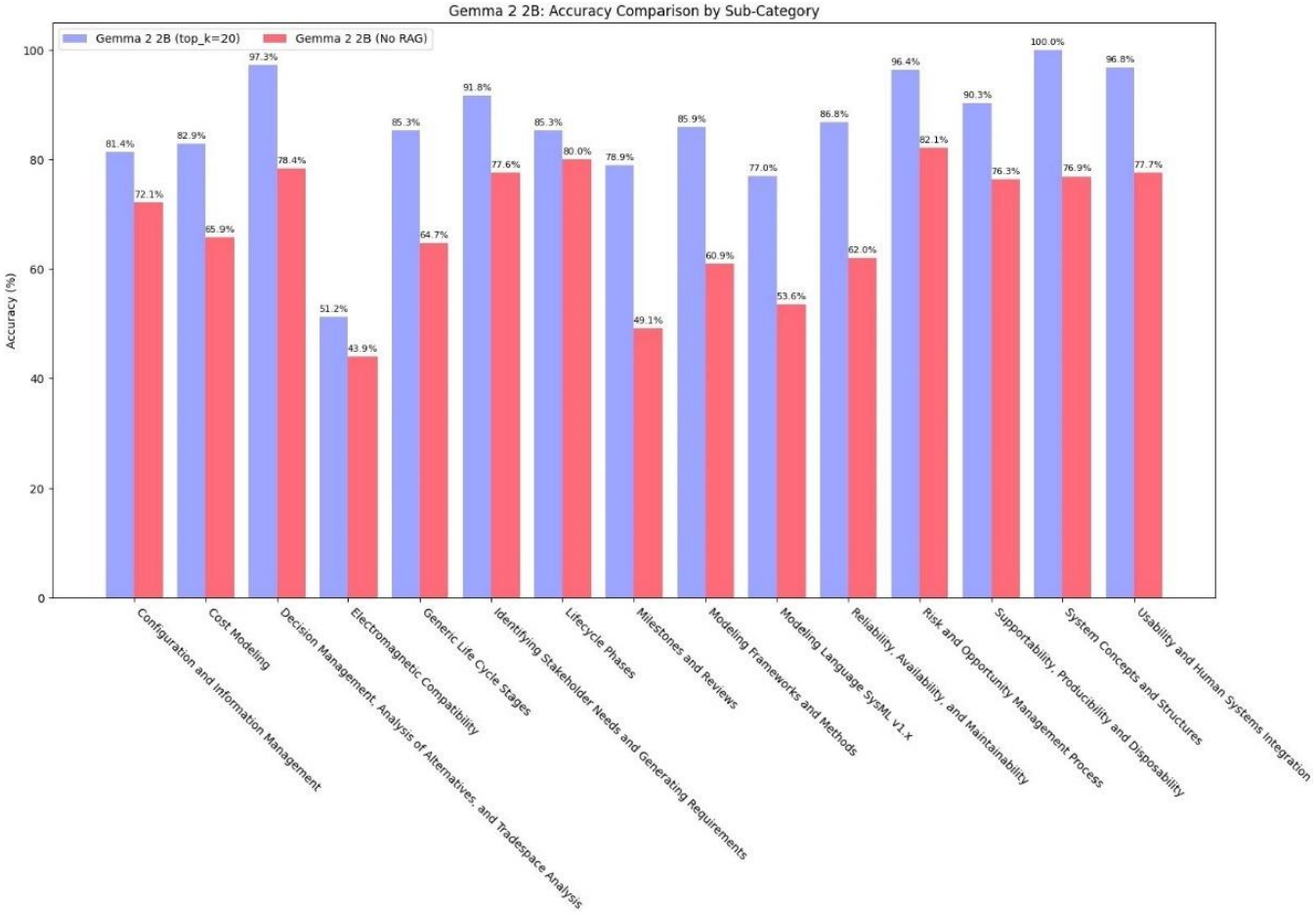

Fig. 5. The Accuracy improvement of Gemma 2 2B by using the RAG infrastructure

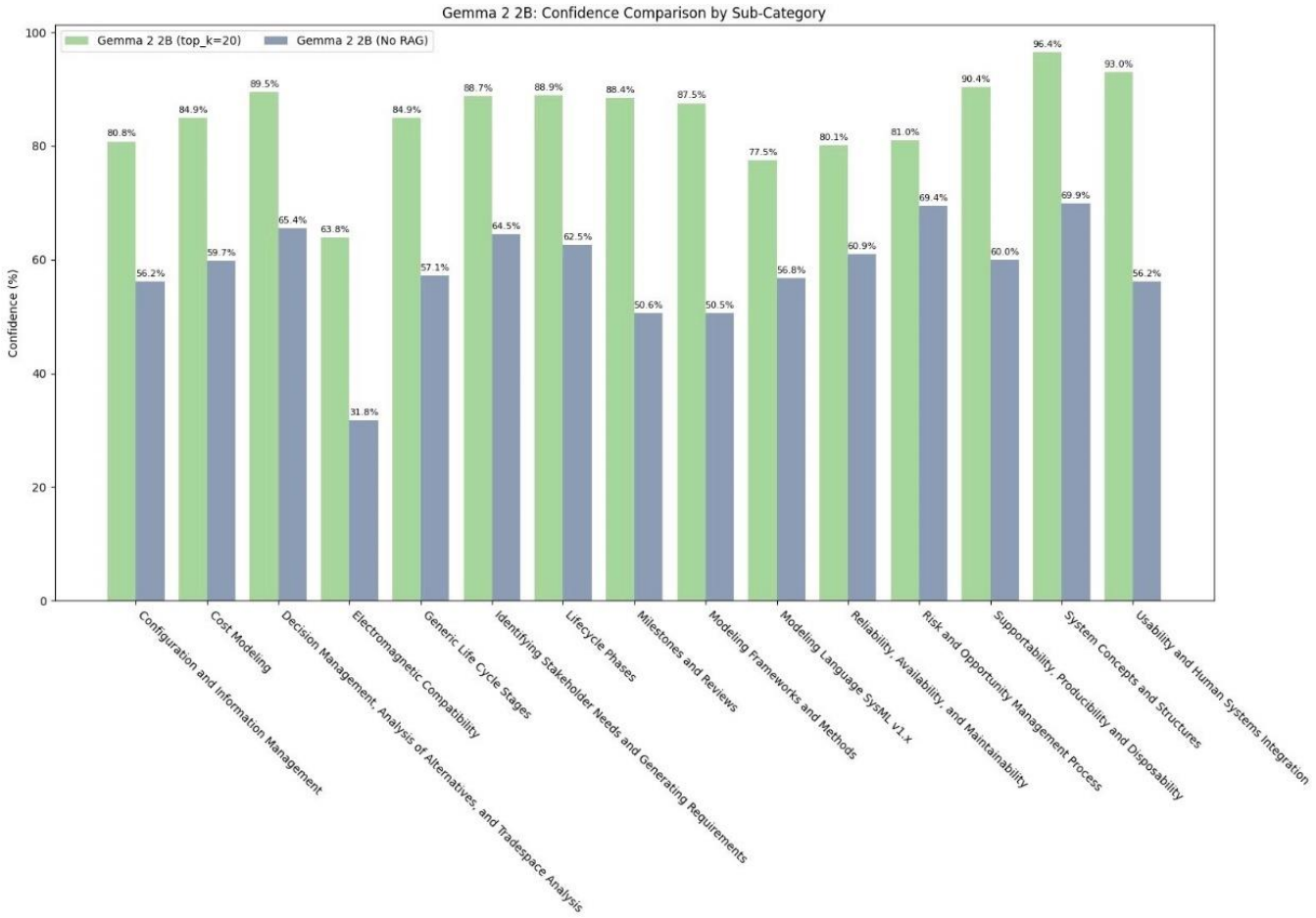

Fig. 6. The Confidence Score improvement of Gemma 2 2B by using the RAG infrastructure

DISCUSSION

Influence of the *top-k* Parameter on Retrieval Quality

- The observed improvement in response accuracy with increasing values of the *top_k* parameter can be attributed to the broader retrieval of potentially relevant documents. This expansion enhances the probability of including at least one contextually appropriate source within the Retrieval-Augmented Generation (RAG) pipeline. As *top-k* increases, the model accesses a more diverse set of information during generation, thereby improving its ability to produce accurate and context-aware outputs.
- In the context of the systems engineering benchmark, many queries span multiple domains and require multifaceted understanding. A single document may only address a subset of the query. Consequently, higher *top-k* values increase the likelihood of retrieving a more comprehensive set of documents that collectively cover a wider range of relevant domains, enabling more complete and precise responses. However, the relatively modest performance gains observed at higher *top_k* values suggest that the top-ranked documents already contain most of the critical information required for accurate answer generation.

Performance of the RAG-Based Infrastructure

The RAG-based infrastructure developed in this study demonstrated consistent improvements across all benchmark sections, with particularly notable gains in areas where baseline models had previously underperformed. These findings support the hypothesis that, through strategic integration of RAG frameworks, small language models can achieve performance levels comparable to those of significantly larger models.

Despite these improvements, attempts to replicate the systems engineering process—outlined in the initial sections of this paper—did not yield meaningful enhancements in model performance within decision-making and judgment-oriented tasks. This indicates that while RAG improves factual and contextual accuracy, it may be insufficient for tasks requiring deeper reasoning or domain-specific expertise.

Execution of Core Systems Engineering Activities

While the model could generate relevant content and support specific subtasks, successful completion of the overall engineering workflow consistently required continuous human oversight and intervention. Even under guided conditions, the model exhibited insufficient domain-specific knowledge and lacked the contextual reasoning necessary to navigate complex, constraint-driven decision-

making processes. These limitations underscore the current gap between generative language models and the nuanced demands of structured engineering methodologies.

Role of Human Judgment in Systems Engineering

Systems engineering is a multidisciplinary field that demands not only profound technical expertise but also the capacity to make complex decisions while balancing a wide range of, often conflicting, considerations. Human judgment plays a pivotal role in this process, encompassing several key dimensions:

- **Contextual Awareness**: Understanding the broader organizational landscape—including strategic goals, business constraints, and regulatory requirements—is essential. These factors are typically assessed through the lens of human experience and domain-specific insight.
- **Risk and Impact Evaluation**: Decision-making frequently involves selecting among alternative technological solutions, each with distinct trade-offs in terms of cost, development time, reliability, and maintainability. Optimal choices are rarely purely technical; they require informed judgment grounded in experience, intuition, and a nuanced understanding of long-term implications.
- **Stakeholder Coordination**: Systems engineering involves collaboration among diverse stakeholders, such as software and hardware engineers, product managers, clients, and regulatory bodies. Human judgment is crucial for mediating between differing perspectives and guiding the development of a coherent and balanced system architecture.
- **Decision-Making Under Uncertainty**: In many cases, complete information is unavailable, and uncertainty must be managed. Systems engineers must apply judgment to select viable solutions under current conditions while maintaining adaptability for future changes.

CONCLUSIONS

Large language models (LLM's) demonstrate significant capabilities in data analysis, solution generation, simulation execution, and rapid information retrieval. Our study shows that their performance, particularly in terms of accuracy and confidence, can be substantially enhanced through the development of tailored infrastructures. Specifically, we demonstrated that small models can be elevated to performance levels comparable to those of larger models by leveraging domain-specific corpora and customized datasets.

Despite these advancements, LLM's remain inherently dependent on human guidance for interpreting context, prioritizing objectives, and making strategic decisions. This dependency was exemplified in our application of a systems engineering process to an aeronautics project, where human expertise played a critical role in steering the model's outputs toward meaningful and actionable insights. Therefore, effective systems engineering is best achieved through a synergistic collaboration between human experts and decision-support technologies. The integration of LLM's within such frameworks holds great promise for enhancing productivity, precision, and scalability in complex engineering domains.

As a point of interest, this section was inspired by the model's own response to the question of why human guidance is indispensable in systems engineering.

Acknowledgement

The author would like to thank Mr. Ofir Ben-Avi, an undergraduate student from the Computer Department at Technion, for his assistance with the software issues.

This research was made possible thanks to the generous support of the Bernard M. Gordon Center for Systems Engineering at the Technion.

REFERENCES

[1] Bordas, P. Le Masson, M. Thomas, and B. Weil, "What is generative in generative artificial intelligence? A design-based perspective," Research in Engineering Design, vol. 35, no. 3, pp. 427–443, 2024. https://link.springer.com/article/10.1007/s00163-024-00441-x

[2] J. A. Baktash and M. Dawodi, "GPT-4: A Review on Advancements and Opportunities in Natural Language Processing," arXiv preprint arXiv:2305.03195, May 2023. https://www.opastpublishers.com/open-access-articles/gpt4-a-review-on-advancements-and-opportunities-in-natural-language-processing-6660.html

[3] Z. B. Akhtar, "Unveiling the evolution of generative AI (GAI): A comprehensive and investigative analysis toward LLM models (2021–2024) and beyond," Journal of Electrical Systems and Information Technology, vol. 11, Article 22, 2024. https://doi.org/10.1186/s43067-024-00145-1

[4] E. S. Ortigossa, T. Goncalves, and L. G. Nonato, "EXplainable Artificial Intelligence (XAI)—From Theory to Methods and Applications," IEEE Access, Jun. 2024. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10549884

[5] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, Analysis of Electric Machinery and Drive Systems, 2nd ed. Wiley-IEEE Press, 2002. https://ieeexplore.ieee.org/servlet/opac?bknumber=5265638

[6] Nature Editorial Team "AI for Science 2025," Nature, 2025. https://www.nature.com/articles/d42473-025-00107-9

[7] C. Ebert and F. Kirschke-Biller, "Agile Systems Engineering," IEEE Software, vol. 38, no. 4, pp. 7–15, Jul.–Aug. 2021, doi: 10.1109/MS.2021.3071806. https://ieeexplore.ieee.org/

[8] Kumar and R. Sharma, "Software Architecture Meets LLMs: A Systematic Literature Review," arXiv preprint arXiv: 2505.16697, 2024. https://arxiv.org/abs/2505.16697

[9] R. M. García Alarcia, P. Russo, A. Renga, and A. Golkar, "Bringing Systems Engineering Models to Large Language Models: An Integration of OPM with an LLM for Design Assistants," Proc. 12th Int. Conf. on Model-Based Software and Systems Engineering, pp. 334–345, 2024. [Online]. Available: https://www.scitepress.org/PublishedPapers/2024/126219/

[10] USA Department of Defense. "Systems Engineering Guidebook." https://ac.cto.mil/wp-content/uploads/2022/02/Systems-Eng-Guidebook_Feb2022-Cleared-slp.pdf

[11] MITRE Corporation. "Systems Engineering Guide." https://www.mitre.org/sites/default/files/publications/se-guide-book

-interactive.pdf
[12] Defense Acquisition University Press. "Systems Engineering Fundamentals." January 2001 https://apps.dtic.mil/sti/tr/pdf/ADA606327.pdf
[13] INCOSE. "Systems Engineering Handbook", Version 2a June 2004. https://productrealize.ir/library/INCOSE-Systems-Engineering-Handbook-A-%E2%80%9CWhat-To%E2%80%9D-Guide-For-.pdf
[14] SEBok Editorial Board "Guide to the Systems Engineering Body of Knowledge" 27 May 2025 https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)