This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

An LLM-enabled Multi-Agent Autonomous Mechatronics Design Framework

Zeyu Wang[1], Frank P.-W. Lo[1], Qian Chen[1], Yongqi Zhang[1], Chen Lin[1], Xu Chen[1], Zhenhua Yu[2], Alexander J. Thompson[1], Eric M. Yeatman[1*], Benny P. L. Lo[1*]
[1]Imperial College London, London, UK
[2]Univeristy of Aberdeen, Aberdeen, UK
e.yeatman@imperial.ac.uk, benny.lo@imperial.ac.uk

Abstract

Existing LLM-enabled multi-agent frameworks are predominantly limited to digital or simulated environments and confined to narrowly focused knowledge domain, constraining their applicability to complex engineering tasks that require the design of physical embodiment, cross-disciplinary integration, and constraint-aware reasoning. This work proposes a multi-agent autonomous mechatronics design framework, integrating expertise across mechanical design, optimization, electronics, and software engineering to autonomously generate functional prototypes with minimal direct human design input. Operating primarily through a language-driven workflow, the framework incorporates structured human feedback to ensure robust performance under real-world constraints. To validate its capabilities, the framework is applied to a real-world challenge involving autonomous water-quality monitoring and sampling, where traditional methods are labor-intensive and ecologically disruptive. Leveraging the proposed system, a fully functional autonomous vessel was developed with optimized propulsion, cost-effective electronics, and advanced control. The design process was carried out by specialized agents, including a high-level planning agent responsible for problem abstraction and dedicated agents for structural, electronics, control, and software development. This approach demonstrates the potential of LLM-based multi-agent systems to automate real-world engineering workflows and reduce reliance on extensive domain expertise.

1. Introduction

In the last few years, large language models (LLMs) [1–3] have made rapid progress across a wide range of cognitive tasks, including natural language understanding [4], code generation [5, 6], symbolic reasoning [7, 8], and scientific problem-solving [9, 10]. Powered by transformer architectures [11, 12] and trained on massive dataset, models such as GPT-4 [13], Claude [14], DeepSeek [15], and PaLM [16] exhibit strong performance in chain-of-thought reasoning [17, 18], few-shot learning [19, 20], and even multimodal understanding when extended to vision-language settings [21, 22]. These capabilities have enabled LLMs to perform not only linguistic tasks, but also to engage in procedural synthesis [23], and structured decision-making [24, 25], laying the foundation for their integration into agent-based systems capable of autonomous planning and tool use.

The rise of agent-based systems [26] marks a critical milestone in artificial intelligence, enabling entities to autonomously perceive, reason, and act within specific environments [27]. LLM-driven agents [28] further enhance these capabilities through sophisticated linguistic comprehension and generation, proving effective in diverse applications such as text summarization [29], software debugging [30, 31], documentation automation [32, 33], customer support [34, 35], mathematical theorem synthesis [36, 37], virtual environment navigation [38], and structured data querying [39, 40]. In industry, LLM agents have streamlined narrowly defined workflows, including report generation [41–43] and basic data analytics [44, 45], significantly improving operational efficiency.

However, current LLM-based agent implementations remain primarily confined to digital or simulated environments, thus limiting their practical application in complex engineering tasks which require the design of physical embodiment, cross-domain integration, and constraint-aware reasoning. Recent attempts to bridge this gap have emerged, demonstrating initial integration of LLM agents with physical experimentation in domains such as autonomous chemical synthesis [46], materials design [47–49], drug discovery [50], and adaptive multi-agent manufacturing systems [51]. Despite these advancements, little attention has been given to LLM-enabled multi-agent frameworks targeting autonomous mechatronics design, a field inherently requiring multidisciplinary expertise across mechanical engineering, electronics, control systems, and software development. The inherent complexity of mechatronics de-

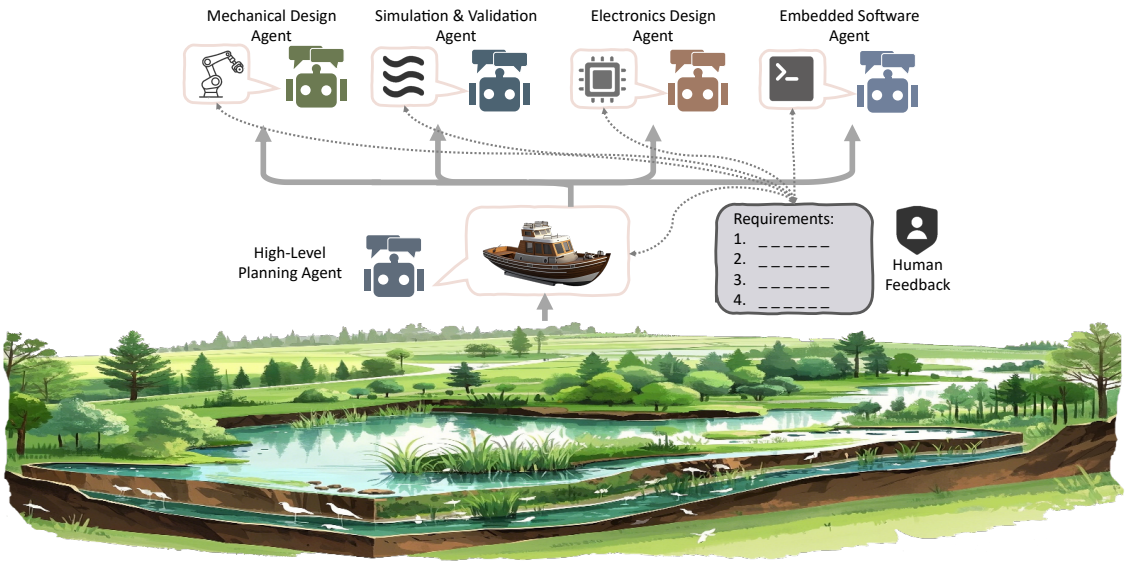


Figure 1. Conceptual architecture of the autonomous mechatronics design framework, illustrating multi-agent collaboration for system development with human-in-the-loop feedback.

sign presents significant barriers to expertise and innovation, yet the cross-domain knowledge synthesis capabilities of LLMs offer emerging potential to support and accelerate certain aspects of mechatronics development. Leveraging LLMs could democratize access to mechatronics design, streamline the development process, and foster innovation by reducing the dependency on domain experts. Addressing this research gap, we propose an LLM-enabled multi-agent framework for autonomous mechatronics system design, specifically targeting complex, interdisciplinary engineering challenges involving real-world physical embodiment. To illustrate the framework effectiveness, we apply it to an autonomous water-quality monitoring applications for wetlands [52, 53]—an ecologically critical yet traditionally labor-intensive and expensive task. Leveraging specialized collaborative agents guided by structured human feedback, our approach significantly reduces complexity and streamline the processes. The resulting autonomous vessel demonstrates practical environmental monitoring potential, validating the proposed method and underscoring the potential of LLM-based agents in reshaping mechatronics design paradigms. This scalable framework could broadly facilitate future advances in areas such as biorobotics for precision medicine [54, 55], autonomous surgical robotics [56, 57], and AI-enabled multimodal health monitoring [58], significantly lowering barriers and expanding participation in interdisciplinary engineering innovation.

2. Intelligence Levels Definition

We proposed Intelligence Levels for Autonomous Mechatronics Design (AMD Levels) as followings:

- **AMD Level 0 – Manual Design:** Entirely human-driven; all tasks and integration are performed manually.
- **AMD Level 1 – Assisted Automation:** Limited AI support (e.g., retrieval, basic code); decision-making remains fully human-controlled.
- **AMD Level 2 – Semi-Autonomous Design:** The system autonomously executes domain-specific tasks; humans manage integration and validation.
- **AMD Level 3 – Highly Autonomous Design:** Cross-domain coordination is automated; human involvement is restricted to high-level supervision and final evaluation.
- **AMD Level 4 – Fully Autonomous Design:** End-to-end automation. The system independently handles planning, integration, optimization, and validation, with human input limited to objectives setting and outcome assessment.

3. Methodology

3.1. Framework design

The proposed employs a hierarchical architecture governed by a High-Level Planning Agent, which decomposes complex challenges into modular tasks for domain-specific agents. Each agent autonomously generates, refines, and validates designs within its specialized discipline, while structured human input ensures feasibility and compliance with practical constraints.

Taking the autonomous water-quality monitoring and sampling system as an example, the High-Level Planning Agent initially analyzes functional requirements—such as mobility, environmental adaptability, and energy efficiency—to define the optimal configuration: a compact,

dual-propeller vessel. Subsequently, the Mechanical Design Agent optimizes the propeller and hull structures, validated by the Simulation & Validation Agent using computational fluid dynamics (CFD) and Finite Elements Analysis (FEA) for hydrodynamics and structural integrity. The Electronics Design Agent synthesizes propulsion and control circuits, leveraging available hardware to simplify integration. The Embedded Software Agent then generates motor-control and communication firmware, for optimal control of the hardware components. Meanwhile, human feedback is introduced as an external input, guiding the agents through ambiguous design choices and refining their outputs where limitations arise. This multi-agent approach integrates high-level objectives with low-level implementation, significantly reduce the iterative design process and dependency on domain experts, providing a scalable methodology for autonomous mechatronics design.

3.2. High-Level Planning Agent

The High-Level Planning Agent translates system-level requirements into actionable design objectives, guiding downstream engineering tasks. Leveraging LLM-based decision-making, it integrates functional goals, constraints, and human feedback into a unified design strategy. Mathematically, the agent's function can be expressed as:

$$P = f(\boldsymbol{F}, \boldsymbol{C}, \boldsymbol{H}) \tag{1}$$

where $\boldsymbol{F}$ denotes the functional requirements (e.g., mobility, energy efficiency, environmental adaptability), $\boldsymbol{C}$ represents system constraints, and $\boldsymbol{H}$ refers to human feedback. By incorporating human feedback [59] directly into this loop, the agent ensures real-world insights continuously shape the design path, enabling robust, adaptable, and manufacturable mechatronic solutions.

3.3. Mechanical Design Agent

Mechanical design integrates structural mechanics [60], fluid dynamics [61], and manufacturability considerations [62] to ensure functional performance and production feasibility. Traditionally, engineers employ parametric modeling [63], finite element analysis [64], and iterative refinement, balancing constraints, such as material properties, hydrodynamic efficiency [65], and structural integrity. This requires substantial computational expertise, geometric reasoning, and practical fabrication experiences and knowledge. However, current LLMs face intrinsic limitations in mechanical design due to insufficient spatial reasoning and geometric cognition [66]. Unlike natural language processing or code generation, mechanical design inherently relies on real-time spatial visualization and parametric interaction—capabilities central to traditional CAD tools (e.g., SolidWorks [67], Fusion 360 [68], COMSOL [69]) yet largely absent in existing LLMs. Additionally, mechanical design demands intuitive understanding of manufacturability constraints and material properties, which LLMs struggle to infer from text-based training alone.

To address these challenges, we propose a hybrid approach combining LLM-driven geometry generation and targeted human-in-the-loop feedback. This structured methodology forms the basis of our Mechanical Design Agent, enabling autonomous, and iterative optimization of mechanical structures yielding functional and effective mechanical designs. Specifically, the Mechanical Design Agent autonomously interprets high-level functional requirements to generate and refine mechanical components. For example, in designing the autonomous water-quality monitoring system, the agent iteratively optimizes the propeller and hull structures, systematically improving their stability, hydrodynamic efficiency, and manufacturability. This design workflow involves three structured stages: parametric geometric modeling, airfoil section generation, and transformation of 2D airfoil profiles into finalized 3D blade geometries.

Specifically, an LLM-driven mapping function formalizes this process:

$$\mathcal{L} \colon \mathcal{P} \to \mathcal{C} \tag{2}$$

where $\mathcal{P}$ denotes the set of design parameters (e.g., blade root and tip chords, pitch angles, airfoil thickness, rake/skew angles) and $\mathcal{C}$ represents the generated CAD code. Iterative refinement is achieved via:

$$\mathcal{P}_{n+1} = \mathcal{F}(\mathcal{P}_n, R_n) \tag{3}$$

with R_n denoting simulation feedback at iteration n.

a) Parametric Geometric Modeling

The blade is defined along its span $z \in [0, L]$ by introducing the normalized coordinate:

$$r = \frac{z}{L}, \quad r \in [0, 1] \tag{4}$$

Two primary functions govern the blade geometry: Chord Distribution $C(z)$ is defined as:

$$C(z) = C_r + (C_t - C_r)r \tag{5}$$

where C_r and C_t are the chord lengths at the root and tip, respectively. More complex variations (e.g., incorporating a Gaussian bulge) are expressed as:

$$C(z) = [C_r + (C_t - C_r)r]\left[1 + \beta \exp\left(-\gamma(r - r_0)^2\right)\right] \tag{6}$$

with β, γ, and r_0 as design parameters. Pitch Distribution $\alpha(z)$ is defined as:

$$\alpha(z) = \alpha_r + (\alpha_t - \alpha_r)r \tag{7}$$

where α_r and α_t are the pitch angles (in radians) at the root and tip.

b) Airfoil Generation

The agent generates a 2D airfoil profile at each spanwise station using a parameterized thickness distribution function $f(x)$, defined over the chord-wise coordinate $x \in [0, 1]$. In general, the upper and lower surfaces are given by:

$$\{(x - x_0, +f(x))\} \quad \text{and} \quad \{(x - x_0, -f(x))\} \tag{8}$$

c) Transformation of 2D Airfoil Sections into a 3D Blade

Each generated 2D airfoil section is transformed into its 3D location by applying:

Scaling: The normalized airfoil coordinates (x, y) are scaled by the local chord $C(z)$:

$$X = C(z)x, \quad Y = C(z)S(z)y \tag{9}$$

where $S(z)$ is an optional thickness scaling factor.

Rotation (Twist): The scaled coordinates are rotated by the local pitch angle $\alpha(z)$:

$$X_r = X \cos \alpha(z) - Y \sin \alpha(z) \tag{10}$$

$$Y_r = X \sin \alpha(z) + Y \cos \alpha(z) \tag{11}$$

Translation (Rake and Skew): To capture planform effects, a translation is applied:

$$S_{\text{rake}}(z) = -z \tan(\theta_{\text{rake}}) \tag{12}$$

$$S_{\text{skew}}(z) = rL \tan(\theta_{\text{skew}}) \tag{13}$$

The final coordinates become:

$$X_f = X_r + S_{\text{skew}}(z) \tag{14}$$

$$Y_f = Y_r + S_{\text{rake}}(z) \tag{15}$$

3.4. Simulation & Validation Agent

Structural validation is essential for mechatronic systems, ensuring robustness and reliability under operational conditions. Traditionally, engineers employ FEA and computational methods to assess structural integrity, deformation, and stress distribution, guiding material selection, geometric optimization, and load-bearing capacity. Integrating LLMs into simulation-driven workflows could accelerate the in structural validation tasks such as model configuration and boundary condition definition. However, current LLMs struggle with interpreting complex multiphysics phenomena, non-linear material behaviors, and ensuring numerical convergence, thus necessitating a hybrid approach that combines AI-driven methodologies with empirical validation.

Within the proposed framework, the Simulation & Validation Agent autonomously validates mechanical structures generated by the Mechanical Design Agent. Specifically, it automatically configures simulations, defines materials and boundary conditions, and performs physics-based analyses using finite element methods. Results from these simulations guide iterative design refinement to enhance structural performance and manufacturability. For validating the autonomous boat's propulsion system, the agent employs CFD and structural FEA, evaluating fluid-structure interactions, mechanical stresses under operational loads, and potential fatigue points.

Initially, to establish baseline safety margins, the agent performed a fully coupled fluid–structure interaction (FSI) simulation under laminar flow conditions as a baseline analysis, offering insights into stress distribution and potential failure regions under extreme passive exposure. Specifically, the agent sets up a FSI interface [70] to transfer pressure and shear to the blades, followed by a stationary stress analysis computing von Mises stress:

$$\sigma_{vM} = \sqrt{0.5\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]} \tag{16}$$

with σ_1, σ_2, and σ_3 being the principal stresses.

Subsequently, to model the propeller's high-speed rotation and fluid interaction, the agent uses rotating machinery analysis in COMSOL, solving the Reynolds-Averaged Navier–Stokes (RANS) equations [71] with the Shear Stress Transport (SST) turbulence model [72]. A rotating reference frame is applied to the propeller domain, with the surrounding fluid kept stationary to ensure proper interface conditions. The momentum equation in the rotating frame is given by:

$$\rho(\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot \left[\mu_{\text{eff}}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)\right] + F_{\text{rot}} \tag{17}$$

where ρ is the fluid density, $\mathbf{u}$ is the velocity field, p is the pressure, and $\mu_{\text{eff}} = \mu_t + \mu$ is the effective viscosity accounting for both molecular (μ) and turbulent (μ_t) contributions. The term F_{rot} encompasses Coriolis and centrifugal forces. By blending k–ω and k–ϵ formulations [73, 74], the SST model ensures accurate near-wall treatment and robust free-stream predictions.

3.5. Electronics Design Agent

Electrical engineering design involves circuit analysis, electronics design, signal processing, control systems, and embedded computing to develop robust electronic systems. Traditional workflows include schematic design, component selection, PCB layout [75], and system-level testing, which require expertise in electromagnetic compatibility (EMC), thermal management, and manufacturability. Recent advancements in LLMs offer transformative potential by assisting in tasks, such as automated component selection [76], schematic generation, and design optimization. Trained on extensive technical documentation, these models can efficiently parse datasheets, recommend suitable

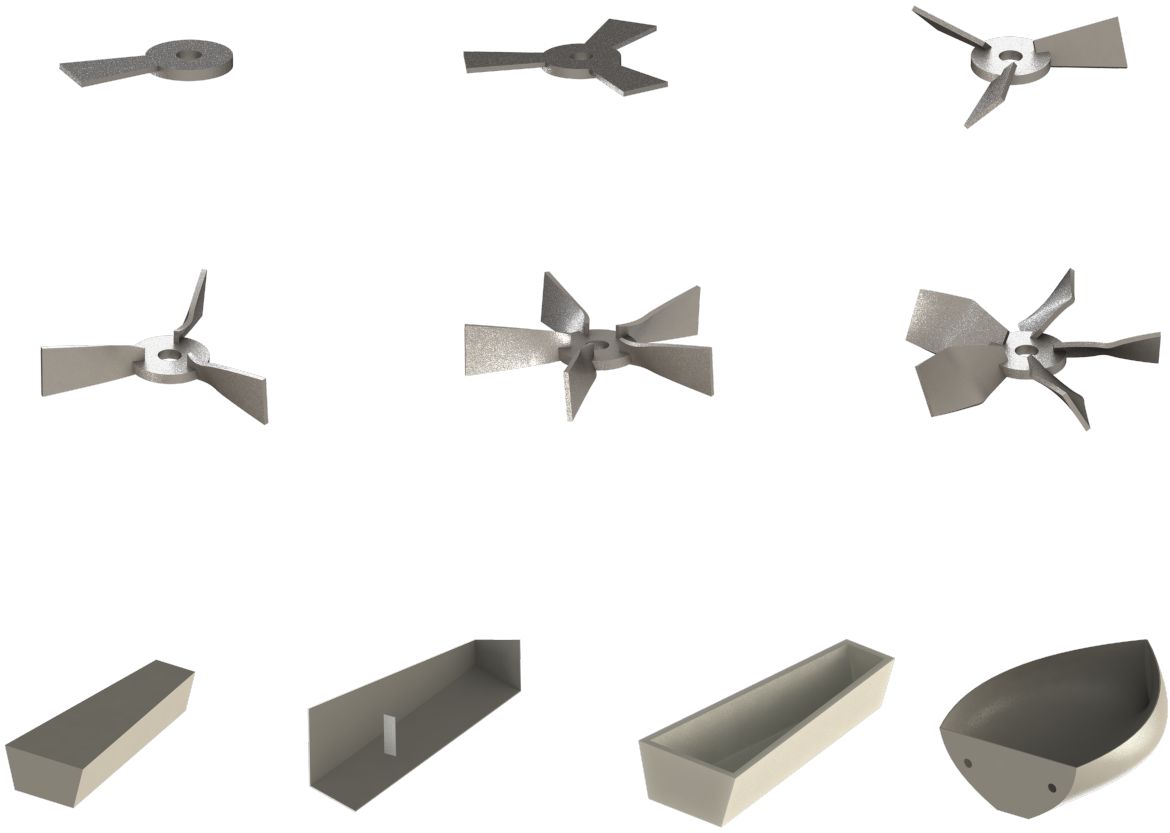


Figure 2. Iterative design process of propellers and hulls generated by the Mechanical Design Agent.

components, and streamline procurement processes, significantly reducing design cycles.

However, LLMs currently face limitations in real-time design constraints, and robustness under practical constraints. To overcome these limitations, the Electronics Design Agent is designed to employ a hybrid workflow that integrates LLM-based reasoning, structured human feedback, and existing hardware utilization. Rather than designing circuits from scratch, the agent first evaluates available hardware resources for potential reuse, optimizing resource efficiency and accelerating development.

3.6. Embedded Software Agent

Embedded systems serve as computational backbones in mechatronics design, translating high-level objectives into precise control commands [77]. Effective embedded software design requires expertise in real-time control algorithms, communication protocols, firmware development, and sensor-driven actuation. Recent advances in LLMs offer opportunities for automating embedded software tasks, including firmware generation, adaptive control logic optimization, and efficient resource allocation. Trained on extensive programming datasets, LLMs can autonomously generate structured code and adapt existing control strategies. However, current LLM-driven code still faces limitations in real-time execution efficiency and direct peripheral interaction, thus necessitating empirical validation.

Within our framework, the Embedded Software Agent translates high-level specifications into executable firmware tailored specifically to hardware architectures defined by the Electronics Design Agent. This inter-agent coordination ensures seamless integration between hardware and software. Additionally, the agent implements adaptive motor-control strategies essential for tasks such as differential speed control in rudderless propulsion systems, dynamically adjusting motor actuation to achieve reach the targets precisely under varying operational conditions.

4. Experimental Results

4.1. Mechanical Design Agent

Guided by the autonomous boat objective established by the High-Level Planning Agent, the Mechanical Design Agent iteratively optimized the propeller and hull structures, as shown in Fig. 2. The propeller design process, illustrated in the top two rows of Fig. 2, began with a three-blade

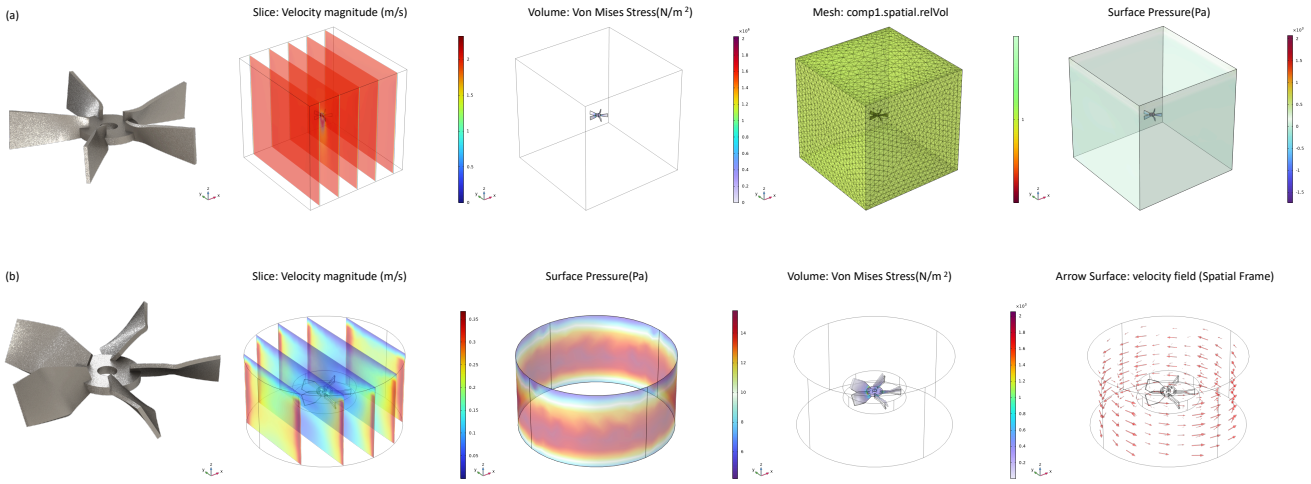


Figure 3. CFD and structural analysis of the optimized propeller design performed by the Simulation & Validation Agent. (a) Structural stress distribution under laminar flow conditions (baseline analysis). (b) Results obtained from rotating turbulent flow analysis using the Shear Stress Transport (SST) model.Results demonstrate stable hydrodynamic performance, reasonable structural stress levels, and highlight areas for further refinement.

configuration featuring a blade length of 26 mm and a hub diameter of 20 mm. Early iterations exhibited poor hydrodynamic performance. Through successive refinements in blade curvature, twist angle, and hub-to-blade ratio, the agent improved hydrodynamic efficiency by enhancing the lift-to-drag ratio and mitigating flow separation. The final geometry balances structural rigidity with stable thrust generation.

Retrieving the detailed implementation of agent, the agent tailored chord and pitch distributions to maximize mid-span loading using piecewise interpolation, as shown below:

$$C(z) = \begin{cases} C_r + (C_m - C_r)(2r), & 0 \le r \le 0.5 \\ C_m + (C_t - C_m)(2r+1), & 0.5 \le r \le 1 \end{cases} \tag{18}$$

with C_m as the maximum chord at mid-span. The pitch distribution is similarly interpolated:

$$\alpha(z) = \begin{cases} \alpha_r + (\alpha_m - \alpha_r)(2r), & 0 \le r \le 0.5 \\ \alpha_m + (\alpha_t - \alpha_m)(2r+1), & 0.5 \le r \le 1 \end{cases} \tag{19}$$

Generic airfoil sections were generated and transformed into a finalized three-dimensional blade geometry optimized for thrust.

The bottom row of Fig. 2 illustrates the hull design evolution. Initial outputs consisted of solid rectangular forms that failed to satisfy buoyancy and weight constraints. Later iterations introduced hollowed structures, though early designs lacked viable compartmentalization. Through progressive refinement, the agent incorporated internal reinforcements, strategic cutouts, and streamlined contours, resulting in a curved bow design that reduces drag, improves maneuverability, and accommodates onboard components. These results demonstrate the Mechanical Design Agent's capacity for autonomous structural refinement under evolving constraints, producing mechanically feasible and performance-optimized components suitable for downstream validation

4.2. Simulation & Validation Agent

To evaluate the propeller design generated by the Mechanical Design Agent, the Simulation & Validation Agent conducted a two-stage CFD analysis, as shown in Fig. 3. For the initial baseline analysis, the agent autonomously performed a laminar-flow fluid–structure interaction simulation using, as an example, the fifth design iteration shown in Fig. 2. The resulting velocity magnitude slices and surface pressure distributions confirmed smooth flow behavior, minimal separation, and coherent pressure fields, validating early-stage design stability. Given the high Reynolds number regime of the final propeller configuration, the agent transitioned to a turbulence-resolving analysis, employing the SST model for enhanced fidelity in near-wall flow characterization, again exemplified by the final iteration shown in Fig. 2. Unlike the baseline scenario, however, the agent could not autonomously configure this turbulent-flow analysis, requiring substantial human feedback and expert assistance to iteratively adjust solver settings, boundary conditions, and spatial discretization until convergence. Further detailed validation of these results remains necessary to fully ensure accuracy and reliability.

Despite demonstrating physics module inference, the Simulation & Validation Agent remains limited in execu-

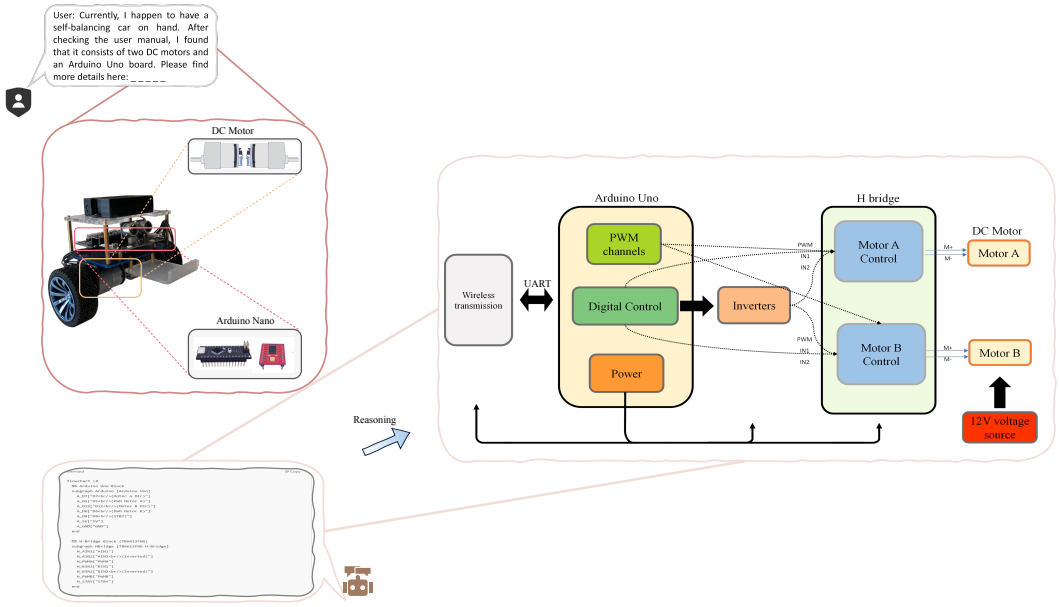


Figure 4. Electronics Design Agent workflow based on user feedback. The agent analyzes the existing balance car system and reconstructs the core control architecture, including Arduino-based PWM control and H-bridge motor driving

tion. Configuration inconsistencies—partly due to software interface variability—undermine its reliability across solver settings. Defining boundary conditions and spatial domain partitioning remains particularly challenging, constrained by the agent's lack of geometric intuition and visuospatial inference capability. These limitations are pronounced in transient, multiphysics simulations, where complex temporal and inter-domain dependencies often lead to convergence failure. Addressing these challenges will require integration of structured expert priors, few-shot learning from validated workflows, and physics-informed guidance to systematically elevate robustness and generalizability.

4.3. Electronics Design Agent

To evaluate the agent's adaptive design capability, an Arduino Nano-based self-balancing car was introduced as external input. The agent received specifications of the system, including motor drivers and power modules, and assessed whether its components could be disassembled and reconfigured into a functional propulsion system for the autonomous boat.

As shown in Fig. 4, the Electronics Design Agent synthesized a system architecture that repurposes the self-balancing car hardware for dual-motor actuation. Leveraging circuit knowledge and structured human input, the agent analyzed the feasibility of reusing the motor control, power distribution, and signal processing elements to construct an efficient and compact drive system.

The resulting architecture comprises two subsystems: the Arduino Nano control unit and an H-bridge motor driver. Pulse-width modulation (PWM) signals from the control unit are processed via logic circuitry and inverters to regulate motor speed and direction. The H-bridge enables bidirectional current flow for two independently controlled DC motors, each powered by a 12V source. Differential motor speed control provides rudderless steering, enabling smooth turning and precise navigation—crucial for autonomous vessels operations. Wireless transmission further supports remote operation in real-world environments.

4.4. Embedded Software Agent

Fig. 5 presents the embedded firmware architecture structured by the agent. The system initializes by configuring GPIOs and peripherals, then continuously monitors incoming serial data. Upon receiving a complete command, it parses the motion type—such as forward, backward, left, right, or stop—and generates corresponding PWM signals to control Motor A and Motor B via an H-bridge driver. For turning maneuvers, the agent applies differential velocity control, adjusting motor speeds to modulate turning radius without the need for a physical rudder.

Real-world execution is assessed using logic analyzer traces, as shown in Fig. 5, which capture IN1, IN2, and PWM signals for both motors under various commands. The observed duty cycles—39.2% (forward), 31.4% (backward), 58.8% (left), and 15.7% (right)—confirm precise motor actuation aligned with the embedded control logic. Directional transitions are accurately executed through appropriate phase shifting. A minor deviation was observed in the first PWM pulse during turning, attributed to initializa-

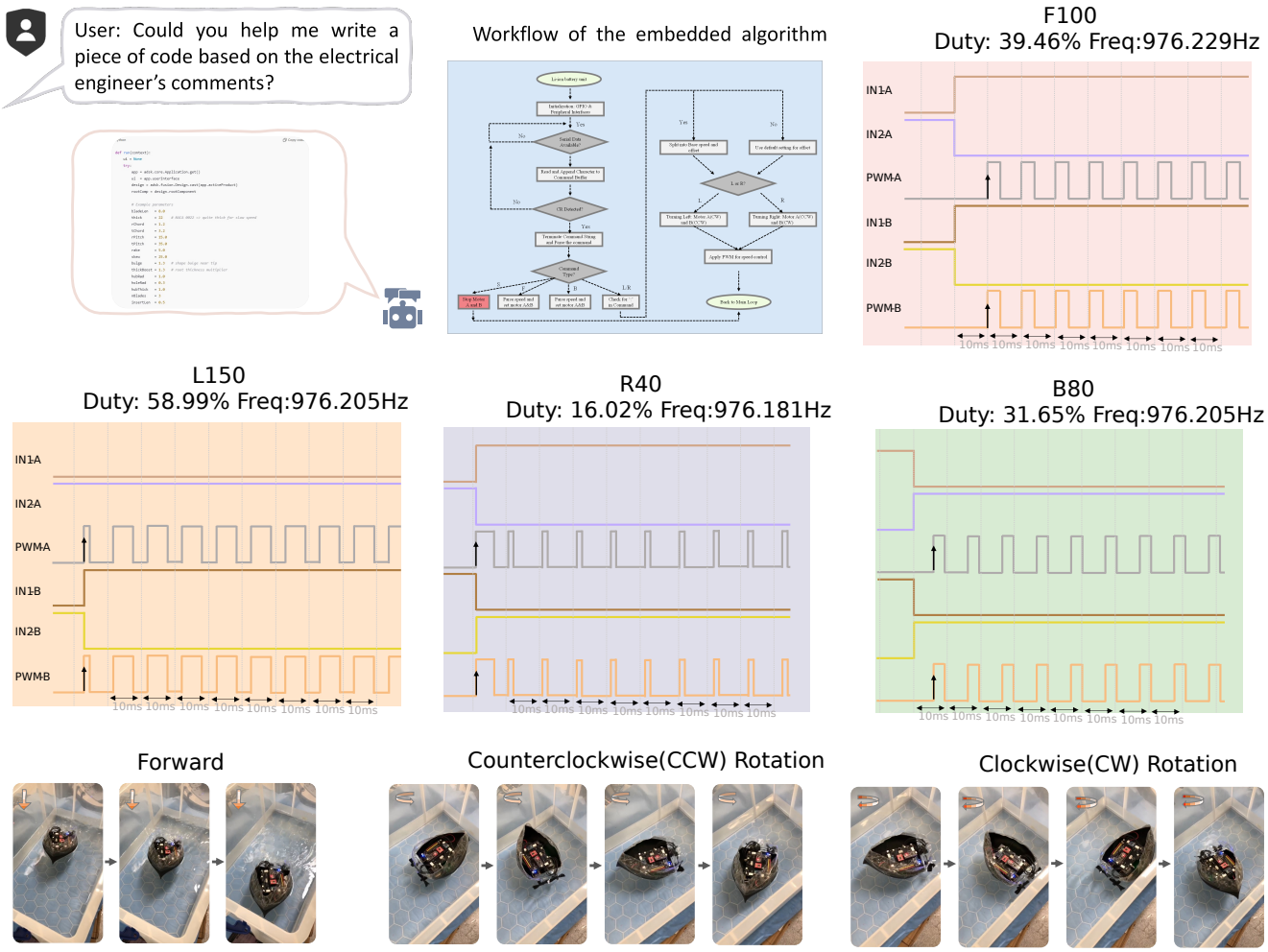


Figure 5. Embedded control logic and human-verified signal outputs. The agent generates control logic based on the Electronics Design Agent's architecture, while a logic analyzer is used to capture and verify PWM signals under different scenarios.

tion latency from transient register updates. This effect is brief and would not affect the system-level control performance, affirming the agent's ability to autonomously generate robust, real-time motor control code suitable for autonomous navigation.

5. Discussion

The proposed framework is designed to assist the design of functioning and effective mechatronic systems with structured human feedback. Human guidance proved crucial, particularly for defining modeling sequences in mechanical design tasks. Embedded Software agents demonstrated higher autonomy given clear architectural specifications provided by the Electronics agents, while simple simulations can be autonomously executed via the COMSOL–MATLAB interface, more complex setups—especially involving domain definition and intricate boundary conditions in multiphysics modules such as transient fluid simulations—remain highly challenging, suggesting the potential benefits of few-shot learning and structured prior injection. Future work will address hardware optimization, such as waterproofing electronic components, and leverage multimodal LLM capabilities, particularly vision-based functionalities, to enhance autonomous control logic.

6. Conclusion

This work introduces an LLM-enabled multi-agent framework that advances autonomous mechatronics design across disciplines boundaries. By integrating language-driven intelligence with structured human feedback, it redefines engineering workflows, lowers expertise barriers, and signals a broader shift toward scalable, democratized innovation in physical-system design.

References

[1] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024. 1

[2] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

[3] Dorottya Demszky, Diyi Yang, David S Yeager, Christopher J Bryan, Margarett Clapper, Susannah Chandhok, Johannes C Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, et al. Using large language models in psychology. *Nature Reviews Psychology*, 2(11):688–701, 2023. 1

[4] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023. 1

[5] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. *Advances in Neural Information Processing Systems*, 36:21558–21572, December 2023. 1

[6] Using an LLM to Help With Code Understanding | Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. https://dl.acm.org/doi/10.1145/3597503.3639187. 1

[7] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models, October 2024. 1

[8] Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning, October 2023. 1

[9] Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models, June 2024. 1

[10] Qiang et al. Zhang. Scientific Large Language Models: A Survey on Biological & Chemical Domains. *ACM Comput. Surv.*, 57(6):161:1–161:38, February 2025. ISSN 0360-0300. doi: 10.1145/3715318. 1

[11] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in Vision: A Survey. *ACM Comput. Surv.*, 54(10s):200:1–200:41, September 2022. ISSN 0360-0300. doi: 10.1145/3505244. 1

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1

[13] OpenAI et al. GPT-4 Technical Report, March 2024. 1

[14] Aman Priyanshu, Yash Maurya, and Zuofei Hong. AI Governance and Accountability: An Analysis of Anthropic's Claude, May 2024. 1

[15] DeepSeek-AI et al. DeepSeek-V3 Technical Report, February 2025. 1

[16] Rohan et al. Anil. PaLM 2 Technical Report, September 2023. 1

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837, December 2022. 1

[18] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs. *Advances in Neural Information Processing Systems*, 37:333–356, December 2024. 1

[19] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 1

[20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. volume 33, pages 1877–1901, 2020. 1

[21] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models, October 2023. 1

[22] Jing Yu Koh, Daniel Fried, and Russ R. Salakhutdinov. Generating Images with Multimodal Language Models. *Advances in Neural Information Processing Systems*, 36: 21487–21506, December 2023. 1

[23] Yixuan Li, Julian Parsert, and Elizabeth Polgreen. Guiding Enumerative Program Synthesis with Large Language Models. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification*, pages 280–301, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-65630-9. doi: 10.1007/978-3-031-65630-9_15. 1

[24] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, Jacob Andreas, Igor Mordatch, Antonio Torralba, and Yuke Zhu. Pre-Trained Language Models for Interactive Decision-Making. *Advances in Neural Information Processing Systems*, 35:31199–31212, December 2022. 1

[25] Paul Hager, Friederike Jungmann, Robbie Holland, Kunal Bhagat, Inga Hubrecht, Manuel Knauer, Jakob Vielhauer, Marcus Makowski, Rickmer Braren, Georgios Kaissis, and Daniel Rueckert. Evaluation and mitigation of the limitations of large language models in clinical decision-making. *Nature Medicine*, 30(9):2613–2622, September 2024. ISSN 1546-170X. doi: 10.1038/s41591-024-03097-1. 1

[26] Mark d'Inverno, Michael M Luck, M D'Inverno, and M Luck. *Understanding Agent Systems*. Springer, 2004. 1

[27] Randall D. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1):173–215, January 1995. ISSN 0004-3702. doi: 10.1016/0004-3702(94)00005-L. 1

[28] Jianing Qiu, Kyle Lam, Guohao Li, Amish Acharya, Tien Yin Wong, Ara Darzi, Wu Yuan, and Eric J Topol. Llm-based agentic systems in medicine and healthcare. *Nature Machine Intelligence*, 6(12):1418–1420, 2024. 1

[29] Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerová, et al. Adapted large language models can outperform medical experts in clinical text summarization. *Nature medicine*, 30(4):1134–1142, 2024. 1

[30] Runchu Tian, Yining Ye, Yujia Qin, Xin Cong, Yankai Lin, Yinxu Pan, Yesai Wu, Haotian Hui, Weichuan Liu, Zhiyuan Liu, and Maosong Sun. DebugBench: Evaluating Debugging Capability of Large Language Models, June 2024. 1

[31] Weiqing Yang, Hanbin Wang, Zhenghao Liu, Xinze Li, Yukun Yan, Shuo Wang, Yu Gu, Minghe Yu, Zhiyuan Liu, and Ge Yu. COAST: Enhancing the Code Debugging Ability of LLMs through Communicative Agent Based Data Synthesis, February 2025. 1

[32] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. mPLUG-DocOwl: Modularized Multimodal Large Language Model for Document Understanding, July 2023. 1

[33] Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arık. Chain of Agents: Large Language Models Collaborating on Long-Context Tasks. *Advances in Neural Information Processing Systems*, 37: 132208–132237, December 2024. 1

[34] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. Generative Feature Language Models for Mining Implicit Features from Customer Reviews. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 929–938, New York, NY, USA, October 2016. Association for Computing Machinery. ISBN 978-1-4503-4073-1. doi: 10.1145/2983323.2983729. 1

[35] Sudhakar Reddy Peddinti, Subba Rao Katragadda, Brij Kishore Pandey, and Ajay Tanikonda. Utilizing Large Language Models for Advanced Service Management: Potential Applications and Operational Challenges, March 2023. 1

[36] Chao Zhang, Xuechen Liu, Katherine Ziska, Soobin Jeon, Chi-Lin Yu, and Ying Xu. Mathemyths: Leveraging Large Language Models to Teach Mathematical Language through Child-AI Co-Creative Storytelling. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, pages 1–23, New York, NY, USA, May 2024. Association for Computing Machinery. ISBN 979-8-4007-0330-0. doi: 10.1145/3613904.3642647. 1

[37] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large Language Models for Mathematical Reasoning: Progresses and Challenges, September 2024. 1

[38] Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit Reasoning in Vision-and-Language Navigation with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):7641–7649, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i7.28597. 1

[39] Haojie Pan, Zepeng Zhai, Hao Yuan, Yaojia Lv, Ruiji Fu, Ming Liu, Zhongyuan Wang, and Bing Qin. KwaiAgents: Generalized Information-seeking Agent System with Large Language Models, January 2024. 1

[40] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large Language Models for Information Retrieval: A Survey, September 2024. 1

[41] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. RecMind: Large Language Model Powered Agent For Recommendation, March 2024. 1

[42] Fang Zeng, Zhiliang Lyu, Quanzheng Li, and Xiang Li. Enhancing LLMs for Impression Generation in Radiology Reports through a Multi-Agent System, December 2024.

[43] Shijie Han, Haoqiang Kang, Bo Jin, Xiao-Yang Liu, and Steve Y Yang. Xbrl agent: Leveraging large language models for financial report analysis. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 856–864, 2024. 1

[44] Hongyang Yang, Boyu Zhang, Neng Wang, Cheng Guo, Xiaoli Zhang, Likun Lin, Junlin Wang, Tianyu Zhou, Mao Guan, Runjia Zhang, and Christina Dan Wang. FinRobot: An Open-Source AI Agent Platform for Financial Applications using Large Language Models, May 2024. 1

[45] Tushar Chugh, Kanishka Tyagi, Rolly Seth, and Pranesh Srinivasan. Intelligent agents driven data analytics using Large Language Models. In *2023 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD)*, pages 152–157, November 2023. doi: 10.1109/ICoABCD59879.2023.10390973. 1

[46] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00832-8. 1

[47] Alireza Ghafarollahi and Markus J. Buehler. ProtAgents: Protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery*, 3(7):1389–1409, 2024. doi: 10.1039/D4DD00013G. 1

[48] Mehrad Ansari and Seyed Mohamad Moosavi. Agent-based learning of materials datasets from the scientific literature. *Digital Discovery*, 3(12):2607–2617, 2024. doi: 10.1039/D4DD00252K.

[49] Mayk Caldas Ramos, Christopher J. Collison, and Andrew D. White. A review of large language models and autonomous agents in chemistry. *Chemical Science*, 16(6): 2514–2572, 2025. doi: 10.1039/D4SC03921A. 1

[50] Yoshitaka Inoue, Tianci Song, and Tianfan Fu. DrugAgent: Explainable Drug Repurposing Agent with Large Language Model-based Reasoning, September 2024. 1

[51] Yuchen Xia, Manthan Shenoy, Nasser Jazdi, and Michael Weyrich. Towards autonomous system: flexible modular production system enhanced with large language model agents. In *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2023. 1

[52] Umair Ahmed, Rafia Mumtaz, Hirra Anwar, Sadaf Mumtaz, and Ali Mustafa Qamar. Water quality monitoring: From conventional to emerging technologies. *Water Supply*, 20(1): 28–45, October 2019. ISSN 1606-9749. doi: 10.2166/ws.2019.144. 2

[53] Alisa L. Gallant. The Challenges of Remote Monitoring of Wetlands. *Remote Sensing*, 7(8):10938–10950, August 2015. ISSN 2072-4292. doi: 10.3390/rs70810938. 2

[54] Zeyu Wang, Frank P-W Lo, Yunran Huang, Junhong Chen, James Calo, Wei Chen, and Benny Lo. Tactile perception: a biomimetic whisker-based method for clinical gastrointestinal diseases screening. *npj Robotics*, 1(1):3, 2023. 2

[55] Zeyu Wang, Frank P-W Lo, Junhong Chen, James Calo, Benny PL Lo, Alex J Thompson, and Eric M Yeatman. An ai-driven bionic whisker system assisting for clinical gastrointestinal disease screening. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024. 2

[56] Ruiyang Zhang, Junhong Chen, Zeyu Wang, Ziqi Yang, Yunxiao Ren, Peilun Shi, James Calo, Kyle Lam, Sanjay Purkayastha, and Benny Lo. A step towards conditional autonomy-robotic appendectomy. *IEEE Robotics and Automation Letters*, 8(5):2429–2436, 2023. 2

[57] Junhong Chen, Zeyu Wang, Ruiqi Zhu, Ruiyang Zhang, Weibang Bai, and Benny Lo. Path generation with reinforcement learning for surgical robot control. In *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4. IEEE, 2022. 2

[58] Frank P-W Lo, Jianing Qiu, Zeyu Wang, Junhong Chen, Bo Xiao, Wu Yuan, Stamatia Giannarou, Gary Frost, and Benny Lo. Dietary assessment with multimodal chatgpt: a systematic analysis. *IEEE Journal of Biomedical and Health Informatics*, 2024. 2

[59] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 3

[60] Keith D Hjelmstad. *Fundamentals of Structural Mechanics*. Springer Science & Business Media, 2007. 3

[61] George Keith Batchelor. *An Introduction to Fluid Dynamics*. Cambridge university press, 2000. 3

[62] Satyandra K. Gupta, William C. Regli, Diganta Das, and Dana S. Nau. Automated manufacturability analysis: A survey. *Research in Engineering Design*, 9(3):168–190, September 1997. ISSN 1435-6066. doi: 10.1007/BF01596601. 3

[63] Gabriella Bolzon and Vladimir Buljak. An effective computational tool for parametric studies and identification problems in materials mechanics. *Computational Mechanics*, 48(6):675–687, December 2011. ISSN 1432-0924. doi: 10.1007/s00466-011-0611-8. 3

[64] Nam-Ho Kim, Bhavani V Sankar, and Ashok V Kumar. *Introduction to Finite Element Analysis and Design*. John Wiley & Sons, 2018. 3

[65] Subrahmanyan Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Courier Corporation, 2013. 3

[66] Yutaro Yamada, Yihan Bao, Andrew K. Lampinen, Jungo Kasai, and Ilker Yildirim. Evaluating Spatial Understanding of Large Language Models, April 2024. 3

[67] SOLIDWORKS. https://www.3ds.com/products/solidworks, January 2023. 3

[68] Autodesk Fusion | 3D CAD, CAM, CAE, & PCB Cloud-Based Software | Autodesk. https://www.autodesk.com/ca-en/products/fusion-360/overview. 3

[69] COMSOL: Multiphysics Software for Optimizing Designs. https://www.comsol.com/. 3

[70] Earl H. Dowell and Kenneth C. Hall. MODELING OF FLUID-STRUCTURE INTERACTION. *Annual Review of Fluid Mechanics*, 33(Volume 33, 2001):445–490, January 2001. ISSN 0066-4189, 1545-4479. doi: 10.1146/annurev.fluid.33.1.445. 4

[71] Giancarlo Alfonsi. Reynolds-Averaged Navier–Stokes Equations for Turbulence Modeling. *Applied Mechanics Reviews*, 62(040802), June 2009. ISSN 0003-6900. doi: 10.1115/1.3124648. 4

[72] Florian R. Menter. Review of the shear-stress transport turbulence model experience from an industrial perspective. *International Journal of Computational Fluid Dynamics*, January 2009. doi: 10.1080/10618560902773387. 4

[73] MM Rahman, V Vuorinen, J Taghinia, and M Larmi. Wall-distance-free formulation for sst k-ω model. *European Journal of Mechanics-B/Fluids*, 75:71–82, 2019. 4

[74] A. Parente, C. Gorlé, J. van Beeck, and C. Benocci. Improved k–ε model and wall function formulation for the RANS simulation of ABL flows. *Journal of Wind Engineering and Industrial Aerodynamics*, 99(4):267–278, April 2011. ISSN 0167-6105. doi: 10.1016/j.jweia.2010.12.017. 4

[75] Tan Yan and Martin D. F. Wong. Recent research development in PCB layout. In *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 398–403, November 2010. doi: 10.1109/ICCAD.2010.5654190. 4

[76] Zoltán Tafferner and Attila Géczy. Comparing large language model artificial intelligence tools in aid of electrical engineering. In *2024 IEEE 30th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 9–14, October 2024. doi: 10.1109/SIITME63973.2024.10814860. 4

[77] Steve Heath. *Embedded Systems Design*. Elsevier, 2002. 5