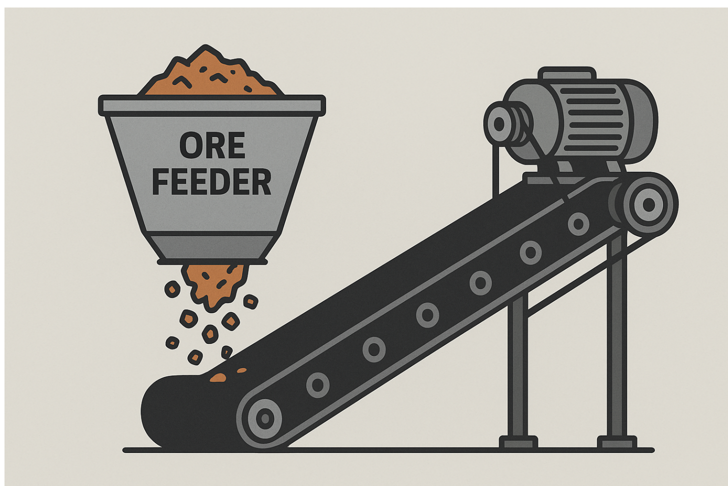
Model an AI-Driven Control and Optimization System Having Three Cascaded Levels of Control and Optimization

Rickard Blind
ricbli-7@student.ltu.se

October 2025



Executive Summary

Objective

The objective of this project is to model and design an **AI-driven control and optimization system** composed of **three cascaded levels of control and optimization**. The system aims to demonstrate how artificial intelligence can be embedded within a hierarchical control architecture to enhance system efficiency, adaptability, and decision-making capabilities.

Goals

The primary goal is to engineer a complete cyber-physical **system of systems** that integrates artificial intelligence. Each control level should interact to achieve optimized overall performance, stability, and robustness.

Solution

The proposed solution is developed using **SysML** and the open-source modeling tool **Papyrus** with the **Arrowhead Framework**. The project begins from a high-level concept and evolves toward a detailed, model-based implementation.

Project Outline

The modeling process follows an iterative refinement approach to progressively reach and improve requirements, architecture, and behavior:

- Define and refine system requirements from a vague initial concept.
- Model system functionality using **use case diagrams** and **sequence diagrams**.
- Develop the system structure using **block definition diagrams** and **internal block diagrams**.
- Integrate AI components within the control hierarchy to enable adaptive optimization across all levels.

System Concept

A control and optimization system for industrial ore handling requires several key elements: the ability to obtain information about the process environment, actuators to control material flow, and a control unit to relate sensor information to appropriate actuation. In its simplest form, this can be represented by an **ore feeder** that releases material, a **conveyor belt** that transports it, and a **motor** that drives the belt's motion. These elements together form the foundation of an automated material transport process.

As the system evolves, it can be expanded with additional sensors for measuring load, speed, vibration, or flow rate, as well as actuators for precision control. The **AI-driven optimization layer** enables the system to dynamically adjust parameters such as feeder rate or motor speed to maximize output while minimizing energy consumption.

A **scalable system design** ensures that what is initially implemented can grow both in functionality and capacity. For instance, the first iteration of the small-scale prototype, using a single feeder and conveyor module was later expanded to an entire plant material handling network. It allows for incremental expansion and technological evolution. New sensors, motors, and feeder units can be added seamlessly, while AI-modules can be integrated to enhance optimization and decision making over time.

For such distributed scalable systems to interact effectively, the components must be able to identify, communicate and coordinate with one another. This can be achieved by adopting a **Service-Oriented Architecture (SOA)** where each subsystem (feeder controller, conveyor control node, AI optimizer) exposes its services to others through standardized communication interfaces.

Finally, the system must be both **secure and resilient**. Security ensures that operational parameters such as motor speeds or material feed rates cannot be tampered with by unauthorized personel, and that control data remains confidential. Resilience implies that the system can handle partial failures gracefully. For instance, maintaining safe operation if one motor or sensor node fails, by redistributing control responsibilities dynamically. These principles are essential for developing a robust AI-driven control.

Structured List of Requirements

Top-Level Requirement

R1: AI-Driven Multi-Level Industrial Control Solution
The system shall provide a scalable and modular control solution for an industrial ore handling process, using three cascading levels of control (local PID, Local AI supervisor, and Plant AI Supervisor) to ensure efficient, adaptive, and safe operation.

Functional Requirements

- **R1.1: Local Control Layer** The system shall employ PID controllers to regulate individual subsystems such as AC induction motors, feeder tables, and conveyor belts based on sensor feedback (torque, speed, temperature, power, and weight).
- **R1.2: Sensor Integration** The system shall include sensors for torque, power, temperature, speed, weight, and ore level, providing continuous measurement data to the local controllers and AI supervisors.
- **R1.3: AI Supervision Layer** The system shall implement a local AI controller that monitors PID behavior, detects anomalies, and dynamically adjusts PID setpoints for optimal performance.
- **R1.4: Global AI Coordination Layer** The system shall include a high-level AI that oversees multiple local AI controllers, coordinating feeder and conveyor operations across local clouds to maintain consistent ore flow and system stability.
- **R1.5: Cloud Integration** The system shall be based on the Arrowhead Framework architecture, where each local control subsystem is encapsulated as a Local Cloud exposing standardized services for communication and interoperability.
- **R1.6: Data Flow Management** The system shall ensure secure and reliable data exchange between sensors, controllers, and AI agents using defined service interfaces (SpeedMeasurement_SD, PowerMeasurement_SD, HeightLevelMeasurement_SD etc.).
- **R1.7: Scalability** The architecture shall allow the addition of new sensors, actuators, and AI nodes without significant redesign, ensuring modular growth across different process areas.
- **R1.8: Safety and Fault Detection** The system shall detect abnormal operating conditions (excessive temperature, stalled motor, over-torque etc.) and initiate appropriate protective actions via the AI supervisor or PID fallback.

- **R1.9: Performance Optimization** The AI controller shall continuously optimize resource usage by adjusting motor speeds, feeder rates, and conveyor load to maximize throughput while minimizing energy consumption.
- **R1.10: Human–Machine Interface** The system may provide a dashboard for visualizing sensor data, AI status, and control actions, allowing operators to monitor and adjust process behavior.

Non-Functional Requirements

- **R2.1: Real-Time Operation** The system shall process sensor feedback and control actions with minimal latency to maintain stable PID and AI-driven regulation.
- **R2.2: Reliability** The system shall operate continuously under industrial environmental conditions (dust, vibration, and temperature) with minimal downtime.
- **R2.3: Modularity and Reusability** Each subsystem (motor, sensor, controller) shall be modeled as a reusable SysD/SysDD component within the Arrowhead architecture.
- **R2.4: Compliance** The system shall comply with industrial communication standards relevant to the Arrowhead Framework.
- **R2.5: Extensibility** The system shall support the integration of additional AI models.

System Overview Description

The proposed system is an **AI-driven multi-level control architecture** designed to manage and optimize an industrial ore handling process consisting of a feeder, conveyor belt, and AC induction motor system.
The design is implemented using the **Arrowhead Framework** and follows a service-oriented architecture (SOA) approach.

At the **lowest level (PID Control)**, individual components - the AC induction motor, feeder table, and conveyor belt are controlled and optimized through PID controllers. These controllers receive continuous sensor feedback from torque, power, temperature, speed, weight, and ore level sensors, ensuring stable real time operation and process safety.

The **second level (Local AI Supervision Layer)** introduces a cloud based AI supervisory agent that oversees the local PID controllers. This AI component continuously monitors system performance, detects anomalies (overheating, excessive torque, inconsistent ore flow etc.), and dynamically adjusts PID setpoints to maintain optimal operation under changing load conditions.

At the **top level (Global AI Coordination Layer)**, a cloud based AI agent controller that supervises multiple local clouds, each representing an Local AI Supervision subsystem composed of a feeder, conveyor, and motor unit. The global AI Accumulate sensor data, performance metrics, and reports from the local AI supervisors to gain a view of the entire process line. Based on this information, it issues control commands to optimize ore distribution, balance material flow, prevent bottlenecks, and enhance energy efficiency.
In the event of a fault or performance decline within a specific process line, the global AI dynamically reallocates operational loads across neighboring subsystems, adjusting feeder rates or conveyor speeds as necessary to maintain continuous, stable, and efficient production throughout the network.

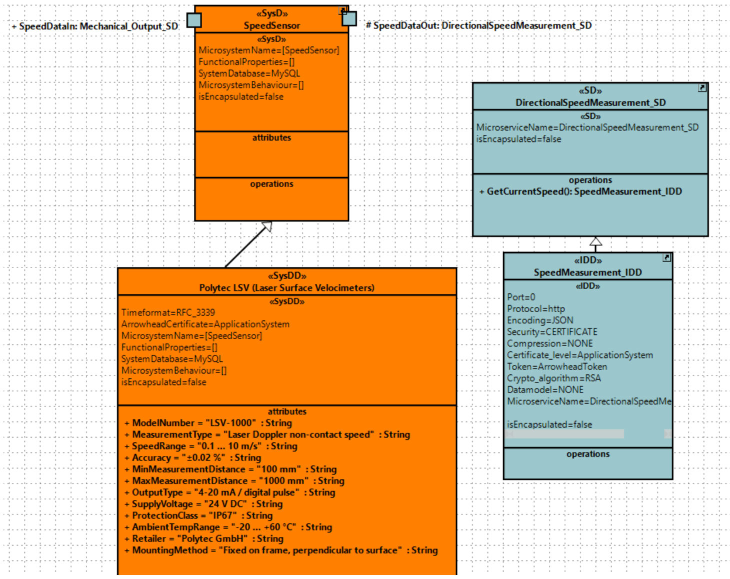
This approach ensures a scalable and self optimizing system architecture. The separation of control levels allows for clear responsibility distribution. PID controllers handles fast local responses, the local AI supervisor ensures adaptive and dynamic optimization, and the global AI maintains process wide coordination and efficiency.

Design Iterations and Development Process

Throughout the project, several design iterations have been carried out to refine the system architecture and clarify the functional relationships between components. Early stages focused on determining which sensors were necessary and how their placement would most effectively capture process relevant data such as torque, speed, temperature, weight, and material height. These iterations ensured that each sensor contributed meaningful information to the control hierarchy without introducing redundancy or unnecessary complexity.

Significant Thought was given to defining the meaning of the **three levels of cascading control**. This concept evolved into a hierarchy consisting of local PID controllers (**First level**), an intermediate AI supervision layer (**Second level**), and a global AI coordination layer(**Third level**). The iterative process helped define communication flows and necessities across these layers, providing both fast local autonomy and centralized optimization.

Another important aspect of the refinement process was the development of accurate **sequence diagrams** that correctly represent the flow of information and service interactions within the Arrowhead Framework. Multiple revisions were made to align the sequence logic with the **Service Definitions (SD)** and **Interface Design Definitions (IDD)**, ensuring that each system correctly provides or consumes the appropriate services. This iterative approach led to a logically consistent design that reflects the principles of service-oriented architecture.



System Block Types and Their Roles

In this project, each component of the system has been modeled using the Arrowhead structure within Eclipse Papyrus. This structure separates system functionality into distinct block types—**SysD**, **SysDD**, **SD**, and **IDD** which together define both the logical behavior and the implementation details of each subsystem. The figure above illustrates this structure using the **Speed Sensor** as an example.

System Design (SysD)

The **SysD** block represents the logical design of the system or subsystem. In this case, the **SpeedSensor_SysD** defines how the sensor interacts with other parts of the process. It contains the following **FullPorts**, which specify the flow of data and control signals:

- **SpeedDataIn : Mechanical_Output_SD (isConjugated = true)** – Consumes the mechanical motion output from the motor or conveyor belt. This represents the physical quantity being measured.
- **SpeedDataOut : DirectionalSpeedMeasurement_SD (isConjugated = false)** – Provides the measured speed data as a standardized service to the rest of the system (e.g., PID controller or AI supervisor).

The SysD does not specify hardware details; it only defines the system behavior and how it connects to other components in the architecture.

System Design Definition (SysDD)

The **SysDD** defines the hardware or implementation-specific details of the system. In this example, the **Polytec LSV (Laser Surface Velocimeter)** has been selected as the real-world implementation for the Speed Sensor. It contains relevant technical parameters that identify the specific sensor model and ensure reproducibility in an industrial setting.

Key attributes included:

- ModelNumber = `"LSV-1000"`
- MeasurementType = `"Laser Doppler non-contact speed"`
- SpeedRange = `"0.1 { 10 m/s"`
- Accuracy = `"±0.02%"`
- SupplyVoltage = `"24 VDC"`
- ProtectionClass = `"IP67"`
- MountingMethod = `"Fixed on frame, perpendicular to surface"`
- Retailer = `"Polytec GmbH"`

This level of detail allows the SysDD to serve as a documentation link between the digital system model and the physical components used in the final implementation. Though a simple comment section within the block would suffice.

Service Definition (SD)

The **SD** block defines the communication service provided or consumed by the system. In this example, the **DirectionalSpeedMeasurement_SD** represents the standardized service through which the speed measurement is made available. It includes the following operation:

- `GetCurrentSpeed()` – Returns the current surface velocity value from the sensor as defined in the corresponding IDD.

Interface Design Definition (IDD)

The **IDD** specifies the communication interface and technical configuration used when implementing the SD. In this example, the **SpeedMeasurement_IDD** defines details such as communication protocol, encoding, and security. These attributes ensure that data exchange between systems remains consistent and secure.

Summary

In summary, this layered structure allows clear separation between logical function (**SysD**), real-world implementation (**SysDD**), communication service definition (**SD**), and service interface configuration (**IDD**). The use of FullPorts ensures data flow—where ports on the left consume inputs from other systems, and ports on the right provide outputs or services to other subsystems. This modeling approach creates a transparent, modular, and maintainable design that can be reused for other sensor and actuator types within the larger AI-driven control architecture.

Future Work

Although the current implementation provides a solid foundation, several ideas have been though of for further development to extend the system toward a complete ore processing solution:

- **Predictive Maintenance:** Utilizing the data collected through the logging system to train AI models that forecast component wear and detect anomalies in motors, conveyors, and crushers, could reduce downtime.
- **Advanced AI Optimization:** Extend the global AI layer with reinforcement learning for dynamic energy management and throughput optimization across all subsystems.
- **Human–Machine Interaction:** Develop an operator interface for manual overrides, AI decision approval, and real-time performance visualization.
- **Data Logging and Analytics:** Centralize historical data to support long-term trend analysis, efficiency tracking, and AI model refinement.
- **Multi-Cloud Coordination:** Expand the local multi-cloud into a wide-ranging multi-cloud network, enabling collaboration between distributed processing sites.
- **Cone Crusher Integration:** Introduce a crusher subsystem for particle size reduction before separation. Integrate load, vibration, and size sensors to optimize feed rate and prevent overloading.
- **Magnetic Separation Control:** Incorporate wet and dry magnetic separators with adaptive AI control of field strength and feed rate, using moisture and density sensors for process tuning.
- **End-to-End AI Coordination:** Connect crushing, conveying, and separation into one continuous, AI-supervised pipeline that dynamically adapts to ore composition and flow variations.

These developments will transform the current prototype into a fully autonomous dynamic ore handling and separation system, capable of learning, optimizing, and coordinating complex industrial operations autonomously.