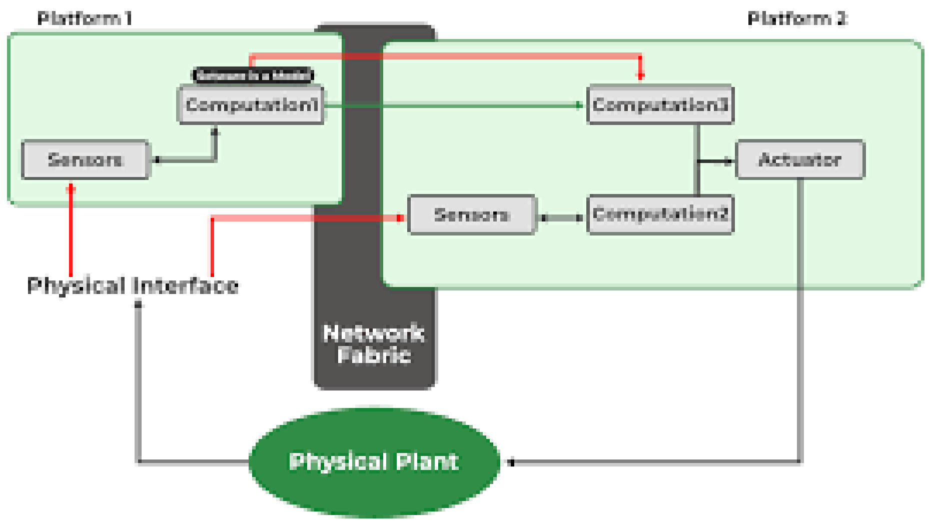
Principles of Automated Control Design

Cyber Physical Systems

Introduction

- Automated control design for Cyber-Physical Systems (CPS) integrates physical processes with computational and communication elements to achieve desired behavior through feedback mechanisms.
- The key challenge in CPS is ensuring the correct and safe interaction between the physical components (e.g., sensors, actuators) and the cyber elements (algorithms, controllers, communication networks).

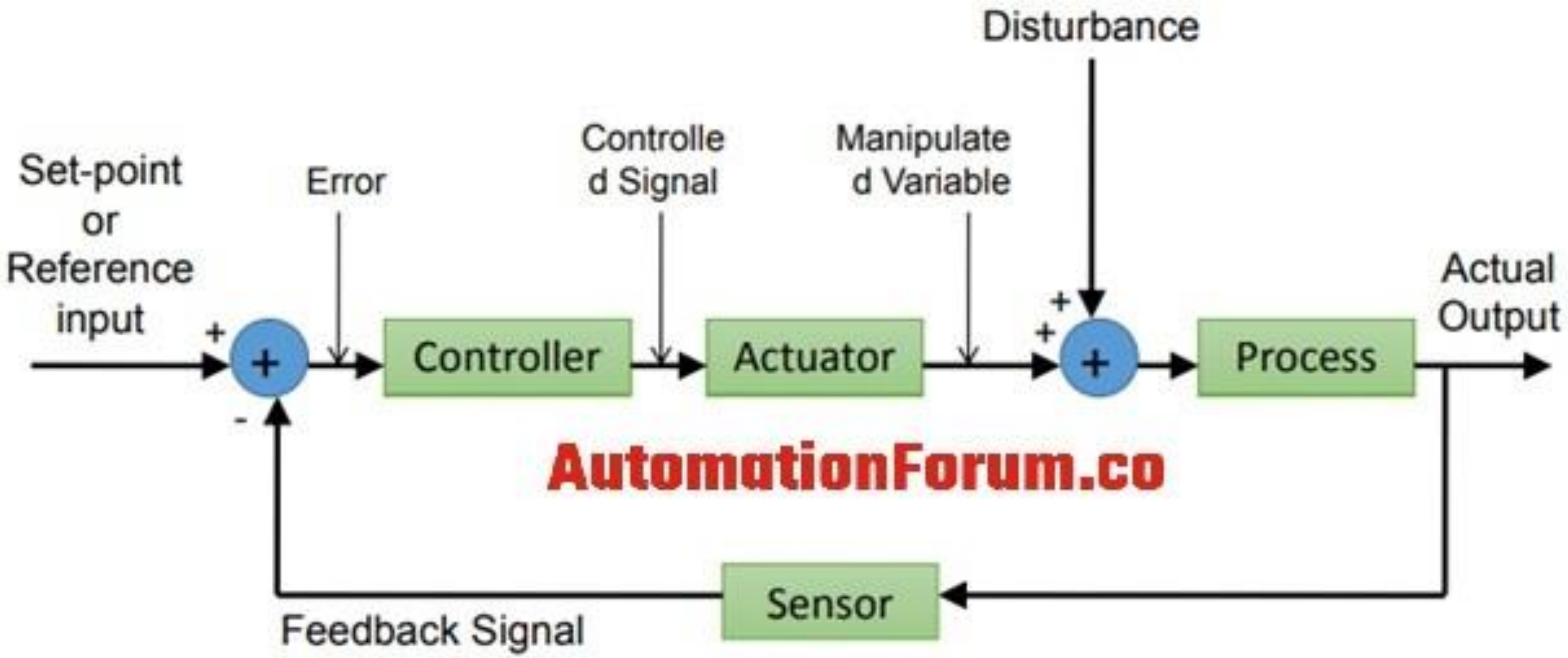


Modeling of CPS

- Physical Modeling: The physical system (mechanical, electrical, biological, etc.) is represented by mathematical models, such as differential equations or state-space representations. These models describe the dynamics and behaviour of the physical part of the system.

- Cyber Modeling: The cyber components include software algorithms, computation models, and network dynamics. They interact with the physical model through sensors and actuators.

- Hybrid Systems: CPS typically combines continuous (physical dynamics) and discrete (computation, decision-making) elements, often modelled as hybrid systems. This mix of discrete-event and continuous-time dynamics is a fundamental characteristic of CPS.



Disturbance
Set-point
or
Reference
input
Error
Controlle
d Signal
Manipulate
d Variable
Actual
Output
+
-
+
+
+
+
+
Controller
Actuator
Process
AutomationForum.co
Sensor
Feedback Signal

Feedback Control System

- Closed-Loop Control: CPS relies on feedback control to continuously monitor the state of the physical system (via sensors) and adjust control actions (via actuators) to maintain desired performance.
- State Estimation: The system controller often uses estimators (e.g., Kalman filters) to infer the system state from noisy sensor measurements.
- Control Objectives: The control algorithms are designed to maintain stability, optimize performance (e.g., energy efficiency, speed), and ensure safety under uncertainties (like disturbances or noise).

Automation and Autonomy

- Self-Adaptation: CPS can incorporate adaptive control techniques, allowing the system to learn and optimize its behavior in real-time based on observed changes in the environment or system dynamics.

- Autonomous Decision Making: The system can make decisions based on pre-programmed logic (rule-based) or through advanced methods such as machine learning, where decisions are based on historical data and predictions.

Real-Time and Communication Constraints

- Time-Sensitive Control: CPS often operates under real-time constraints, meaning control decisions must be made within strict time bounds to ensure proper system behaviour. This requires careful design of communication protocols, scheduling, and resource management.
- Networked Control Systems: In CPS, the communication network (e.g., wireless, wired) linking sensors, controllers, and actuators introduces latency, packet loss, and bandwidth limitations, which must be considered in the control design.

Formal Methods and Verification

- Safety and Correctness: CPS often operates in critical domains (e.g., autonomous vehicles, medical devices) where safety is paramount. Formal verification techniques (e.g., model checking, theorem proving) are used to ensure that the system meets safety and functional specifications.
- Robustness: The control design must be robust to uncertainties, including model inaccuracies, environmental variations, and cyber-attacks. Techniques like robust control or resilient design ensure that the system continues to operate safely under such conditions.

Optimization-Based Control

- Model Predictive Control (MPC): MPC is commonly used in CPS for making control decisions based on optimizing future behavior over a prediction horizon. The system repeatedly solves an optimization problem at each time step, adjusting the control inputs based on the predicted future states.
- Resource-Efficient Control: CPS often operates under resource constraints (e.g., energy, computation), requiring optimization techniques to efficiently allocate resources while maintaining control performance.

Human-Centric Considerations

- Human-in-the-Loop (HITL): Some CPS applications require human interaction or supervision, making it necessary to design control systems that account for human factors such as decision-making time, attention, and possible errors.

- Human-Machine Interaction: The design of CPS must also consider how humans interact with the system, both in terms of interface design and ensuring that control systems can override or complement human actions for safety and efficiency.

Security and Privacy

- Cybersecurity: CPS is vulnerable to cyber-attacks (e.g., data tampering, denial of service) due to its reliance on communication networks and software systems. Control designs must incorporate security measures to detect and mitigate such threats.

- Privacy-Preserving Control: In some CPS applications (e.g., smart grids, healthcare systems), sensitive data is involved. Control designs may need to ensure that private data is not compromised, while still enabling efficient control decisions.

Tools and Methods Used in CPS Control Design

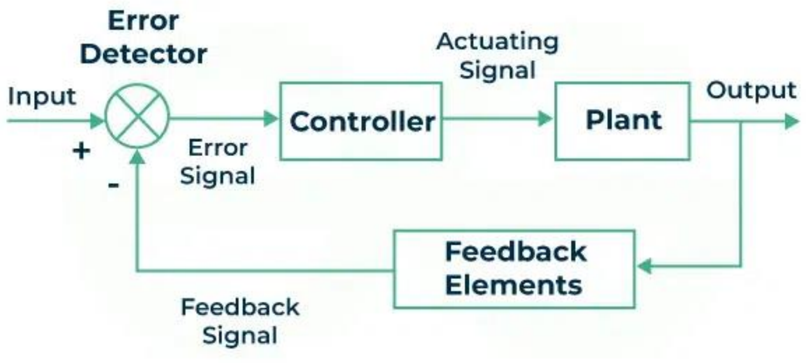
- Simulation and Emulation: Simulation tools (e.g., MATLAB/Simulink, Modelica) are often used for testing and verifying the performance of CPS control designs before implementation.
- Co-Simulation: Tools that allow the simulation of both the physical and cyber elements simultaneously (co-simulation) are crucial in verifying the integration between control algorithms and physical models.
- Control Design Tools: Control theory, optimization solvers, and machine learning algorithms are widely used for automated control design. Tools such as reinforcement learning and neural networks may also be applied for systems requiring adaptive or intelligent control.

Summary

- The principles of automated control design for CPS focus on the tight integration of computational algorithms with physical systems, ensuring real-time, autonomous, secure, and reliable performance.
- This involves feedback control, modeling of hybrid systems, real-time computation, and addressing challenges like communication delays, safety, and cyber-security.

Basic Control Theory for Cyber Physical Systems

- Basic control theory for Cyber-Physical Systems (CPS) involves the study and application of feedback loops to manage and regulate the behavior of physical systems through computational elements.
- The goal is to ensure the system responds in a desired manner, even in the presence of disturbances or uncertainties.



Components of Control Systems

System Representation and Modeling

- At the core of control theory is the mathematical modeling of the system being controlled.
- It is often represented by state-space models, differential equations, or transfer functions.
- In CPS, the physical part (e.g., a car, robot, or power grid) is often modeled using continuous dynamics, while the cyber part (e.g., sensors, actuators, controllers) is represented using discrete models.

State-Space Representation

- In state-space models, the system is described by a set of state variables, capturing the current condition of the system.

$\dot{x}(t)$=$Ax(t)$+$Bu(t)$($State\ Equation$)

$y(t)$=$Cx(t)$+$Du(t)$($Output\ Equation$)

where:$x(t)$ represents the state of the system.

$u(t)$ is the control input.

$y(t)$ is the output.

A, B, C, D are matrices that define the system's behaviour.

Transfer Function

- The transfer function represents the relationship between the input and the output in the frequency domain and is often used in linear time-invariant (LTI) systems.

$$G(s) = \frac{Y(s)}{U(s)}$$

where $G(s)$ is the transfer function, $Y(s)$ is the Laplace transform of the output, and $U(s)$ is the Laplace transform of the input.

Feedback Control

- Closed-Loop Control: In a closed-loop control system, sensors continuously monitor the output of the system and send that information to a controller. The controller compares the actual output to the desired output (or reference) and computes corrective actions to minimize the error.

- Error = Reference Value – Measured Output
- The controller’s goal is to minimize this error over time.

- Open-Loop Control: In contrast, an open-loop system does not use feedback. It simply applies a predefined control input, but it may not respond to changes in the system’s behavior or external disturbances.

Types of Controller

- **Proportional (P) Controller**: This is the simplest controller where the control signal is proportional to the error:

$$u(t) = K_p \cdot e(t)$$

where $e(t)$ is the error, and K_p is the proportional gain.

- **Proportional-Integral (PI) Controller**: The PI controller adds an integral term to eliminate steady-state error. The control signal is based on both the magnitude of the error and the accumulated error over time:

$$u(t) = K_p \cdot e(t) + K_i \int e(t)dt$$

- **Proportional-Derivative (PD) Controller**: The PD controller introduces a derivative term to predict future error based on the rate of change of the error:

$$u(t) = K_p \cdot e(t) + K_d \frac{de(t)}{dt}$$

- **Proportional-Integral-Derivative (PID) Controller**: This combines all three terms (proportional, integral, and derivative) to provide robust control:

$$u(t) = K_p \cdot e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

PID controllers are widely used in CPS due to their simplicity and effectiveness in a variety of applications.

Stability

- Lyapunov Stability: A system is Lyapunov stable if, after a disturbance, it returns to equilibrium over time. This can be analyzed using Lyapunov functions.

- BIBO Stability: A system is Bounded-Input, Bounded-Output (BIBO) stable if a bounded input always produces a bounded output.

- Routh-Hurwitz and Nyquist Criteria: These are frequency-domain techniques used to determine system stability based on its transfer function.

Real Time Control

- CPS operates in real-time environments where control actions must be computed and executed within strict time limits. Delays in sensor measurements, communication networks, or computational load can degrade control performance or even destabilize the system.

- Sampled Data Control: In CPS, digital computers sample the continuous signals from the physical world at discrete intervals. The controller must account for the sample time and delays in the control loop.
- Dead-Time Compensation: Techniques like Smith Predictors or Kalman filters are used to predict future states and compensate for time delays in control actions.

Hybrid Systems

- CPS often combines both continuous-time dynamics (from the physical world) and discrete-time dynamics (from the computational world).
- Hybrid systems consist of a mix of continuous variables (representing physical quantities like speed or temperature) and discrete events (like switching between control modes).
- State Machine Models: These are commonly used to model the discrete switching logic.
- Piecewise-Linear Control: This approach is used when different control laws are applied depending on the mode or state of the system.

Summary

- Basic control theory for CPS revolves around understanding how to model, analyze, and design control systems that ensure desired performance, stability, and robustness in real-time, often under uncertainty.
- This involves selecting appropriate feedback control strategies, analyzing system stability, and accounting for real-world constraints such as delays, disturbances, and network issues.
- These control systems are crucial for making CPS behave predictably and efficiently across a wide range of applications.

Dynamical System and Stability for Cyber Physical Systems

- A cyber-physical system (CPS) is an integration of computation with physical processes.
- These systems involve a tight interaction between physical components and cyber components (computation and control).
- Examples include autonomous vehicles, smart grids, and industrial automation systems.
- The dynamical system aspect of a CPS refers to the time-dependent evolution of the system's state, while stability refers to the system's ability to maintain or return to a desired state in the presence of disturbances.

Dynamical Systems in CPS

- 1. Physical Dynamics: These are often modeled using differential equations or discrete-time systems, representing how the physical components of the system (e.g., the movement of a robot or the flow of electricity in a smart grid) evolve over time.

- 2. Cyber Dynamics: These involve algorithms, software, and control policies. They interact with the physical dynamics by processing sensor data, making decisions, and sending commands to actuators.

- 3. Hybrid Systems: CPS are often modeled as hybrid systems, where both continuous and discrete changes occur. For example, the physical system might follow continuous dynamics, but the control algorithm might switch modes based on thresholds (discrete events).

Stability in CPS

1. Lyapunov Stability:

- A system is Lyapunov stable if small perturbations or changes in the system’s initial state result in small deviations in its future states.
- Essentially, small changes should not cause the system to spiral out of control.

Asymptotic Stability: The system not only resists large deviations but also tends to return to a desired state over time after being disturbed.

2. Exponential Stability:

- This is a stronger form of stability, where the system returns to its equilibrium state at an exponential rate. The speed of convergence can be controlled by the parameters of the system.

3. Input-to-State Stability (ISS):
- This concept refers to the ability of a system to maintain bounded behavior even when external inputs (e.g., disturbances or sensor noise) are applied. The system's state remains bounded if the inputs remain bounded.

4. Robust Stability:
- This type of stability ensures that a CPS can withstand uncertainties or variations in its model (e.g., modeling errors or external disturbances) while still maintaining stable behavior.

5. Stability in Hybrid Systems:
- For hybrid systems (systems that involve both continuous and discrete dynamics), stability needs to be defined in a way that accounts for the interactions between discrete transitions (cyber aspects) and continuous evolution (physical aspects). Common approaches include analyzing the switching conditions and ensuring that the system does not oscillate indefinitely between modes.

Analyzing stability in CPS

- 1. Lyapunov Functions: To prove stability, Lyapunov functions are often used, which are scalar functions that decrease over time along the trajectories of the system. If such a function can be found, the system is considered stable.
- Lyapunov functions, titled after Aleksandr Lyapunov, are scalar functions that can be used to verify the stability of equilibrium of an ordinary differential equation in the concept of ordinary differential equations (ODEs).
- 2. Linearization and Eigenvalue Analysis: For linear systems, eigenvalues of the system matrix are used to determine stability. For non-linear systems, local linearization around equilibrium points can be used to analyze local stability.

3. Reachability Analysis: In some CPS, stability analysis can be performed by determining the set of states that the system can reach from a given set of initial conditions. This is important for ensuring that the system does not evolve into unsafe or unstable states.

4. Control Theoretic Approaches: Controllers like PID, state feedback, and model predictive control (MPC) are often designed to ensure that the system remains stable. For CPS, this often involves designing the cyber component (control algorithm) to ensure the stability of the overall system.

- Safety: Stability ensures that the CPS can operate safely even in the presence of disturbances or uncertainties. For example, an unstable autonomous vehicle could lead to erratic driving behavior.

- Performance: A stable system performs predictably and efficiently, ensuring that it meets performance criteria (e.g., a power grid delivering consistent energy supply).

- Reliability: Stability contributes to the reliability of CPS, ensuring that the system can continue functioning as expected over time

CLF and MLF for Cyber Physical Systems

- In Cyber-Physical Systems (CPS), CFLs (Control Flow Logic) and MLFs (Model Logic Flow) are key concepts related to how the system's behavior is managed, especially with respect to control, decision-making, and the interaction between the cyber (computation) and physical (actuation) components.
- Although CFLs and MLFs are not universally standardized terminologies, they are associated with the broader concepts of controlling the logic and model behaviors in CPS.

Control Flow Logic (CFL)

- CFL refers to the sequence or process by which the computational or cyber part of a CPS executes decisions, manages data flow, and directs the control of physical processes.
- The CFL governs the system's decision-making process and ensures that commands are executed in the correct order and at the correct time.

Key Aspects of CFLs

1. Sequential Control: The order in which control commands are executed is critical to the behavior of a CPS. For example, in an autonomous vehicle, CFL ensures that the decision to stop occurs before the application of brakes.

2. Event-Driven Control: CPS often rely on event-based triggers (e.g., sensor readings, thresholds) to initiate certain control actions. CFL manages these events and determines when and how control signals should be sent to the physical components.

3. Scheduling: The timing of control actions is essential in CPS, especially when there are real-time requirements. CFL manages the scheduling of tasks to ensure that the physical system reacts to inputs within specified deadlines.

4. Concurrency: Many CPS must manage multiple tasks concurrently (e.g., controlling multiple subsystems). CFL coordinates these tasks, ensuring that resources are shared properly and no conflicts arise between different control actions.

5. Safety and Fault Tolerance: CFL incorporates logic to detect and handle faults or anomalies in the system. This might include redundancy, fallback strategies, or triggering safety protocols when the system behaves unexpectedly.

Examples of CFLs in CPS

- State Machines: CFL can be implemented using state machines that transition between different states based on inputs from the physical environment. For example, a drone may have states such as "takeoff," "hover," "move," and "land."
- Real-Time Scheduling Algorithms: In real-time systems, CFL ensures that tasks are scheduled and executed according to priority or deadlines (e.g., rate monotonic scheduling).
- Event-Triggered Logic: In networked CPS, event-triggered controllers decide when to send control signals based on sensor events rather than periodically, thus optimizing resource usage.

Model Logic Flow (MLF)

- Model Logic Flow (MLF) refers to the flow of logic within the models that describe the system’s physical and cyber dynamics.
- In CPS, the physical processes are often modeled using mathematical equations (differential equations, hybrid automata, etc.) that describe how the system evolves over time.
- The MLF governs how these models are integrated with control logic to ensure the correct and efficient interaction between the cyber and physical layers.

Key aspects of MLF

1. Hybrid Models: CPS are frequently modeled as hybrid systems, with both continuous (physical) and discrete (cyber) dynamics. The MLF coordinates the transition between different modes in the system (e.g., switching between operating modes of an autonomous system).

2. Model Integration: CPS are composed of various subsystems, each with its own model. The MLF coordinates how these models interact with each other and how they influence the overall system's control. For example, in a smart grid, the power distribution model may interact with a consumption prediction model.

3. Data Flow and Feedback: MLF handles the flow of data between the physical system (sensors) and the cyber controller. It ensures that data from the physical world is fed into the models in real-time and that decisions made by the control system are based on accurate, up-to-date model outputs.

4. Predictive Modelling: In many CPS, the models are used to predict future states of the system. For example, Model Predictive Control (MPC) uses a model to predict future behavior and optimizes control actions accordingly. MLF manages how these predictions are integrated into the control flow.

5. Model Refinement and Adaptation: As the CPS operates, the models governing its behaviour may need to be refined or adapted based on real-time data. MLF handles this process, ensuring that the models remain accurate and aligned with the physical system's current state.

Examples of MLF

- Model Predictive Control (MPC): MLF governs how the model of the physical system is used to predict future behavior and optimize control decisions. For example, in a robotic arm, MLF ensures that the physical dynamics model is used to compute the optimal movement trajectory.
- State Estimation Models: MLF is used to estimate the state of the physical system based on noisy sensor data. For instance, Kalman Filters are often used to provide accurate estimates of system states, which the controller uses to make decisions.
- Fault Detection and Isolation (FDI): MLF integrates models that can detect deviations from normal behavior and isolate the root cause of faults. This is common in CPS like industrial plants and autonomous systems.

CFL and MLF Interaction in CPS

- The interaction between CFL and MLF is crucial for the proper functioning of CPS.
- While CFL governs the execution of control actions, MLF ensures that the models describing the physical system’s behavior are accurate and integrated into the control loop.
- Control Decisions Based on Model Outputs
- Model Updates Based on Control Actions
- Event-Driven Control with Predictive Models

Importance of CFL and MLFs in CPS

- Safety and reliability
- Efficiency
- Handling complex systems
- In cyber-physical systems, Control Flow Logic (CFL) and Model Logic Flow (MLF) represent the coordination between control actions and system models.
- CFL handles the execution of control tasks, ensuring correct sequencing, timing, and decision-making.
- MLF governs how system models are integrated into the control process, handling predictions, data flow, and real-time system state updates.
- The effective interaction between CFL and MLF is essential for achieving safety, efficiency, and robustness in CPS applications.

Stability under slow switching for CPS

- In Cyber-Physical Systems (CPS), many systems are modeled as hybrid systems that consist of both continuous dynamics (described by differential equations) and discrete events (such as switching between different modes or controllers).
- In such systems, stability under slow switching refers to conditions under which the system remains stable when switching between different modes of operation happens infrequently enough.
- This concept is particularly important in switched systems and mode-based control systems, where the system can switch between different subsystems or controllers, each with its own set of dynamics.
- The challenge is to ensure that the overall system remains stable despite these switches, especially when the individual subsystems are stable.

Performance under Packet drop and Noise for CPS

- In Cyber-Physical Systems (CPS), performance can be significantly affected by packet drops and noise, particularly in systems where the control and communication processes are interconnected via a network.
- This situation is common in applications such as industrial automation, autonomous vehicles, and smart grids, where sensors, controllers, and actuators communicate over wireless or wired networks.
- Understanding how packet drops and noise affect the system's performance and designing strategies to mitigate these effects are crucial for ensuring reliable and safe operations.

Packet drops in CPS

- Sensor to controller packet drops
- Controller to actuator packet drops
- Controller performance degradation
- Stability issues
- Increased latency
- Error propagation

Mitigation strategies for packet drops

- Redundancy: Duplicate packets using multiple channels
- Packet drop compensation: actions based on drops
- State estimation
- Delay compensation
- Various protocols

Noises in CPS

- In CPS, noise refers to unwanted disturbances or variations in signals, which can affect both sensor measurements and communication channels.
- Noise can degrade system performance by introducing errors into the data used for decision-making, leading to incorrect control actions.
- Sensor Noise
- Actuator Noise
- Communication Noise

Impact of Noise

- Reduced control accuracy
- State estimation error
- Instability
- Delayed convergence

Mitigation strategies of noise

- Noise filtering
- Robust control design
- Adaptive control
- Error correction techniques
- Sensor fusion

Examples of Packet drop and Noise in CPS

- Autonomous vehicles
- Smart grids
- Industrial automation
- Agriculture automation