

AUTOMATIC TRACKING AND CONTROL SYSTEM

Ciorteza Elisabeta Mihaela¹, Gomboş Andreea²

¹Department Informatics, Mathematics and Electronics, University "1 Decembrie 1918" of Alba Iulia, Alba Iulia, Romania,
<https://orcid.org/0000-0002-3971-2449>

²Department Informatics, Mathematics and Electronics, University "1 Decembrie 1918" of Alba Iulia, Alba Iulia, Romania,
<https://orcid.org/0009-0003-0853-5456>

*Corresponding Author: mciorteza@uab.ro

Received: 25/06/2026

Revised: 29/07/2026

Accepted: 14/09/2026

ABSTRACT:

Smart systems increasingly require advanced tracking and control strategies to ensure high performance, robustness, adaptability, and energy efficiency in dynamic operating environments. This paper presents an automatic tracking and control framework based on a modular Cloud-IoT architecture integrating data acquisition, state estimation, decision making, and control subsystems. The proposed approach employs a nonlinear state-space model to represent system dynamics and formulates trajectory tracking as an optimization problem that minimizes the error between the actual and reference trajectories.

To improve performance under disturbances and uncertainties, the framework incorporates state estimation techniques and adaptive decision mechanisms. A Nonlinear Model Predictive Control (NMPC) strategy is adopted to optimize tracking accuracy while limiting control effort and energy consumption. The optimization process is formulated through a performance index that balances precision, robustness, and efficiency.

The proposed architecture is validated through simulation using nonlinear dynamic models and discrete-event system techniques. The obtained results demonstrate accurate trajectory tracking, rapid error convergence, stable closed-loop operation, and efficient energy utilization. The Root Mean Square (RMS) tracking error is approximately 0.03, confirming the effectiveness of the proposed method. The developed framework provides a scalable and flexible solution for smart systems and establishes a foundation for future implementations incorporating artificial intelligence and intelligent control techniques.

Keywords: Automatic Control, Automatic Tracking, Cloud-Iot Systems, Nonlinear Model Predictive Control, Optimization, Smart Systems.

INTRODUCTION

Smart systems represent a central direction in modern engineering, being defined by the tight integration of physical components, computational mechanisms, communication technologies, and advanced control strategies. These systems operate autonomously in complex and uncertain environments, with strict requirements regarding precision, robustness, safety, and energy efficiency.

Smart systems have become a fundamental pillar of contemporary engineering, characterized by the integration of physical processes with advanced computation, communication, and intelligent control strategies. This integration is specific to cyber-physical systems, which enable real-time monitoring and control of physical processes in dynamic and complex environments [1], [2].

In applications such as autonomous robotics, unmanned aerial vehicles, intelligent infrastructures, and industrial cyber-physical systems, the problem of tracking a reference trajectory or achieving dynamic objectives is fundamental. Nonlinear dynamics, external disturbances, and parametric uncertainties limit the effectiveness of classical control strategies, thus requiring the development of advanced architectures that integrate state estimation, adaptive decision-making, and performance optimization.

Modern applications, including autonomous robotics, smart industrial systems, and urban intelligent infrastructures, demand highly accurate and robust tracking mechanisms. Due to the inherent nonlinearities, disturbances, and uncertainties, traditional methods are often insufficient. Consequently, recent research emphasizes advanced architecture that combines state estimation, optimal control, and dynamic adaptation

strategies [3].

In this context, the present work proposes an automatic tracking and control system for smart systems, designed as a general and modular framework capable of addressing current and future requirements in this field.

ARCHITECTURE

The proposed architecture is designed in accordance with modern principles of cyber-physical systems, emphasizing modularity, interoperability, and scalability [1]. The system is structured into four main subsystems.

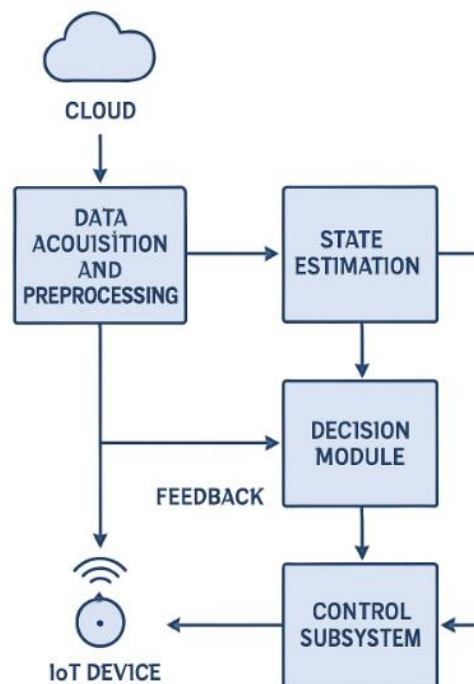


Fig. 1 General architectural model proposed in this study

The architecture of the proposed automatic tracking and control system follows current paradigms of cyber-physical and distributed smart systems, ensuring the coherent integration of physical components, local computation layers, and advanced Cloud–IoT infrastructures. This approach enables scalability, adaptability, and efficient operation in complex environments [1], [2].

From a structural perspective, the system is organized in a modular and hierarchical manner, comprising local control loops, estimation and decision loops, and an extended Cloud–IoT layer for global optimization.

Data Acquisition and Preprocessing

The data acquisition subsystem integrates heterogeneous sensors commonly used in smart systems, including inertial sensors, position and velocity sensors, vision systems, and environmental sensors. This subsystem provides the interface between the physical environment and the control system.

Data preprocessing includes noise filtering, temporal synchronization, error compensation, and sensor fusion, which are essential for ensuring high-quality and consistent data input for the control process [3].

The heterogeneity of the sensor network enables multimodal observation of the system state and the surrounding environment, which significantly enhances the reliability and performance of the overall system.

State Estimation

The state estimation subsystem aims to obtain reliable estimates of system states that are not directly measurable. In practical systems, measurements are often affected by noise, delays, and uncertainties.

To address these issues, advanced estimation techniques such as nonlinear Kalman filters and robust observers are employed, including Extended Kalman Filters (EKF) and Moving Horizon Estimation (MHE) [4], [5].

Accurate state estimation is a critical requirement for achieving stable and high-performance tracking, particularly in the presence of disturbances and model uncertainties.

Decision Module

The decision module plays a central role in the proposed architecture by enabling dynamic adaptation of control strategies based on system conditions and tracking objectives.

This module adjusts controller parameters in real time, considering constraints, disturbances, and performance requirements. Modern approaches integrate adaptive control techniques and learning-based methods to enhance system autonomy and flexibility [6], [7].

The decision module ensures that the control system remains robust and efficient under varying operating conditions.

Control Subsystem

The control subsystem generates the control inputs required to ensure accurate tracking of the reference trajectory. Advanced control strategies such as nonlinear Model Predictive Control (NMPC), robust control, and optimal control are employed to handle system nonlinearities and constraints [1].

The control subsystem operates in real time, using estimated states and parameters provided by the decision module. The separation between real-time control and higher-level decision processes ensures system stability and compliance with real-time constraints.

In addition, an extended Cloud-IoT loop is integrated into the architecture, enabling global monitoring, long-term optimization, and parameter adaptation.

IoT devices collect and transmit system data to the cloud infrastructure, where advanced processing and optimization are performed. The optimized parameters are then fed back to the local system, closing the loop and enhancing performance [8], [9].

This hierarchical structure ensures a clear separation between fast local control and slower global optimization processes, leading to improved energy efficiency, robustness, and scalability of the smart system [10].

MATHEMATICAL MODELLING

The system dynamics are described using a nonlinear state-space model:

$$\dot{x}(t) = f(x(t), u(t), d(t)), y(t) = h(x(t)) \quad (1)$$

Where $x(t)$ represents the state vector, $u(t)$ is the control input, and $d(t)$ denotes external disturbances affecting the system.

The tracking problem is formulated as the minimization of the error with respect to a reference trajectory:

$$e(t) = x(t) - x_r(t) \quad (2)$$

The model explicitly incorporates parametric uncertainties and external disturbances, reflecting realistic operating conditions of modern smart systems [1], [11].

Mathematical modelling represents the foundation for the design of automatic tracking and control systems, especially for complex smart systems characterized by nonlinear dynamics, strong couplings, parametric uncertainties, and external perturbations. Recent literature emphasizes the need for models that are sufficiently general to capture real system behavior, while remaining tractable for control synthesis [12], [2].

Nonlinear State-Space Modelling

A general smart system can be described by a nonlinear continuous-time state-space model:

$$\dot{x}(t) = f(x(t), u(t), p) + G(x(t))d(t), y(t) = h(x(t)) \quad (3)$$

where:

- $x(t)$ is the state vector,
- $u(t)$ is the control input,
- $\theta(t)$ represents uncertain or time-varying parameters,
- $d(t)$ denotes external disturbances,
- $y(t)$ is the measured output vector.

This formulation is widely used for modelling robotic systems, autonomous vehicles, and industrial cyber-physical systems [3].

For complex systems, the nonlinear function $f(\cdot)$ is generally not exactly known, and therefore structural approximations or parameterized models are employed. This allows the model to remain mathematically tractable while still capturing the essential behaviour of the system.

TRACKING PROBLEM FORMULATION

The tracking problem consists in determining a control law $u(t)$ such that the system state $x(t)$ follows a desired reference trajectory $x_r(t)$, generated by a reference system.

Problem Definition

From a mathematical perspective, the tracking problem can be formulated as ensuring that the state trajectory of the controlled system converges into a predefined reference trajectory:

$$\dot{x}_r(t) = f_r(x_r(t), r(t)) \quad (4)$$

The main objective is to design a control input $u(t)$ that guarantees accurate tracking performance under dynamic and uncertain operating conditions [1],[2].

A. Error Definition

The tracking error is defined as the difference between the actual system state and the reference trajectory:

$$e(t) = x(t) - x_r(t) \quad (5)$$

This error function represents the key performance indicator of the control system.

The objective of control is to ensure that the tracking error converges to zero:

$$\lim_{t \rightarrow \infty} \|e(t)\| = 0 \quad (6)$$

which guarantees asymptotic or exponential convergence and therefore accurate trajectory tracking.

B. Error Dynamics

By differentiating the tracking error with respect to time, the error dynamics can be expressed as:

$$\dot{e}(t) = f(x(t), u(t), p) - f_r(x_r(t), r(t)) + G(x(t))d(t) \quad (7)$$

Substituting the system dynamics, we obtain:

$$\dot{e}(t) = f(x(t)) + B(x(t))u(t), \quad \text{rang}(B) < n \quad (8)$$

This equation highlights the main factors influencing the error evolution:

- nonlinear dynamics of the system,
- discrepancies between the real system and the reference model,
- external disturbances and uncertainties.

The modelling of error dynamics constitutes the foundation for the design of modern tracking control strategies, including robust, adaptive, and predictive control approaches [1],[2].

Underactuated Systems

In many practical smart system applications, such as mobile robots, UAVs, and multi-body systems, the system may be underactuated, meaning that the number of control inputs is smaller than the number of state variables. Such systems can be described as:

$m < n$, where m is the number of control inputs and n is the number of states.

Underactuated systems require advanced control design techniques, since not all degrees of freedom can be directly controlled. In these cases, feedback linearization, coordinate transformations, or energy-based control methods are often employed to achieve the tracking objective.

Despite these challenges, properly designed control strategies can still ensure stable and accurate tracking behaviour [3],[11].

ADVANCED MODELLING AND CLOUD – IOT INTEGRATION

Modern smart systems require advanced mathematical formulations that explicitly account for uncertainties, distributed interactions, and hierarchical Cloud–IoT structures.

Modelling of Uncertainties and Disturbances

In real-world applications, the system model includes structural uncertainties and external perturbations, which can be represented as:

$$\dot{x}(t) = f_0(x(t)) + \Delta f(x(t), p) + B(x(t))u(t) + d(t) \quad (9)$$

where $\Delta f(x,t)$ denotes nonlinear uncertainties and unmodeled dynamics.

Such formulations are essential for the design of robust and adaptive control strategies, widely used in modern cyber-physical systems [7].

Multi-Agent and Distributed Systems

For distributed smart systems, such as robotic swarms or cooperative autonomous vehicles, the mathematical model is extended to a multi-agent framework:

$$J = \int_0^T (e^T(t)Qe(t) + u^T(t)Ru(t)) dt \quad (10)$$

where each agent interacts with its neighbors within a communication network.

In this case, the tracking problem becomes a cooperative one, involving both local and global objectives. This paradigm has been extensively studied for distributed control systems [11].

Performance Criteria Formulation

The performance of the control system is evaluated through an optimal cost function of the form:

$$\dot{x}(t) = f(x(t), u(t), p) + G(x(t))d(t) \quad (11)$$

where matrices Q and R are used to balance tracking accuracy and control effort.

This formulation is standard in optimal and predictive control and is widely used for achieving efficient and energy-aware control behavior [1].

Cloud-IoT System Modelling

The integration of Cloud–IoT infrastructures lead to a hierarchical system model, where local dynamics are coupled with distributed computation and global optimization processes [12], [2].

The smart system is described using two interconnected levels:

- local level (edge), responsible for real-time control,
- cloud level, responsible for global optimization and learning.

The local model can be expressed as:

$$x_{k+1} = \Phi(x_k, u_k, \theta_k) \quad (12)$$

while the cloud-level model operates in a discrete-time form:

$$\tilde{x}(t) = x(t - \tau(t)) + \eta(t) \quad (13)$$

where θ represents adaptive control parameters.

C. IoT Communication

IoT devices are modeled as communication nodes affected by delays and transmission noise:

$$\min_{\theta} J_c = \sum_{k=0}^N (e_k^T Q_c e_k + u_k^T R_c u_k) \quad (14)$$

where τ represents the communication delay and $n(t)$ denotes noise.

This formulation allows integration of communication effects into control design and stability analysis [8].

D. Cloud-Based Optimization and Energy Efficiency

At the cloud level, a global optimization process is implemented:

$$u(t) = \pi(x(t), \theta^*)$$

(15)

and the updated parameters are transmitted back to the local controller:

$$\dot{\theta}(t) = \Gamma \nabla_{\theta} J_c$$

(16)

This hierarchical control enables improved performance and adaptability.

Furthermore, energy consumption is modeled as:

$$E(t) = \int_0^t (u^T(\tau) W u(\tau)) d\tau$$

(17)

and optimized together with the tracking performance:

$$J_{total} = J + \beta E$$

(18)

This approach is essential for energy-aware control of smart systems and is widely used in modern cyber-physical applications [10].

Stability of Cloud-IoT Systems

The stability of the hierarchical system is ensured through time-scale separation between local and cloud processes:

- fast local control
- slow cloud optimization

This condition guarantees stable system operation and robustness against communication delays.

A Lyapunov-based condition ensures Input-to-State Stability (ISS):

$$\dot{V}(x) \leq -\alpha \|x\|^2 + \gamma \|\tilde{x} - x\|$$

(19)

Thus, maintaining stability even in the presence of disturbances and network-induced uncertainties [2].

SIMULATION OF THE PROPOSED MODEL

The proposed system is validated through simulation using discrete-event system (DES) modelling techniques. Discrete-event systems are dynamic systems whose evolution is governed by the occurrence of discrete events, such as task initiation, resource allocation, or signal transmission. In robotics and control engineering, DES play a crucial role in modelling sequential, concurrent, and hierarchical processes, which are characteristic of automated production systems and supervisory control applications [3].

A widely used mathematical and graphical formalism for DES modelling is represented by Petri nets. These provide a rigorous framework for analysing causality, synchronization, concurrency, and resource sharing in complex systems. Using places, transitions, and markings, Petri nets enable the study of critical system properties such as deadlock avoidance, coordination between multiple subsystems, efficient utilization of shared resources, and validation of control logic prior to implementation [19].

Visual Object Net++ (VON++) represents an advanced platform for object-oriented Petri net modelling and simulation, particularly suitable for applications in robotics and automation. The platform enables:

- modular modelling of system components (robots, sensors, actuators, controllers),
- dynamic behaviour definition using timed and conditioned transitions,
- simulation of both nominal and fault operation scenarios,
- analysis and validation of system behaviour prior to real implementation.

Thus, discrete-event system simulation using VON++ provides a powerful framework for the design and validation of complex control systems.

SYSTEM MODEL REPRESENTATION

The Petri net model corresponding to the proposed architecture is illustrated in Fig. 2.

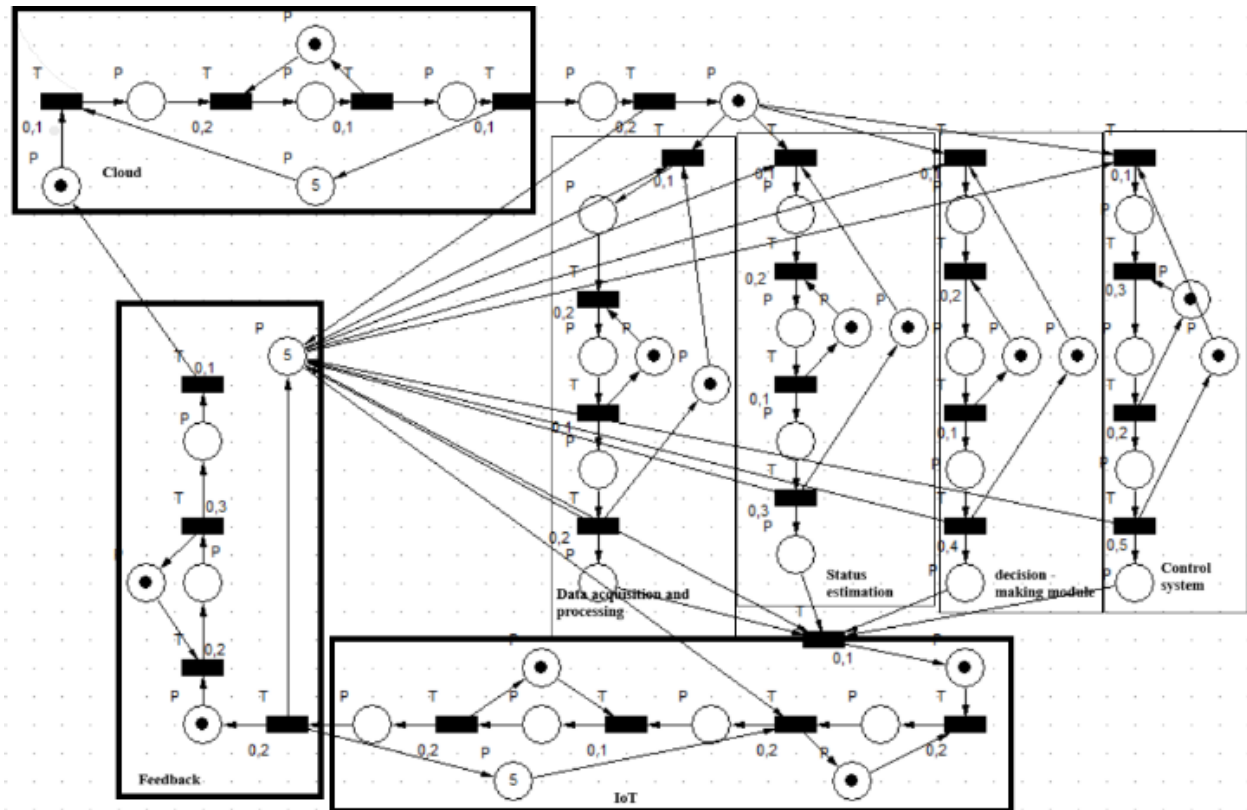


Fig. 2 General Petri Net model of the proposed system

This model captures the overall information and control flow between the main subsystems of the architecture, including data acquisition, state estimation, decision-making, control, and Cloud–IoT interaction. Such modelling approaches are consistent with modern distributed and cloud-integrated cyber-physical systems [9].

Information Flow Analysis

The variation of the flow across the system is analysed through simulation, highlighting the dynamic interactions between subsystems.

The following figures present the evolution of information flow:

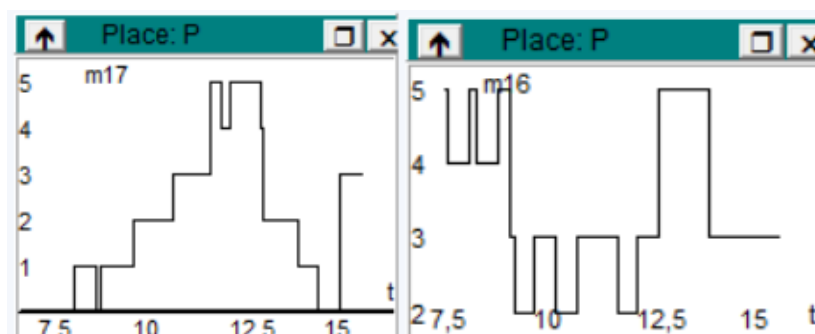


Fig. 3 Information flow in the Data Acquisition and Preprocessing subsystem.



Fig. 4 Information flows in the State Estimation subsystem

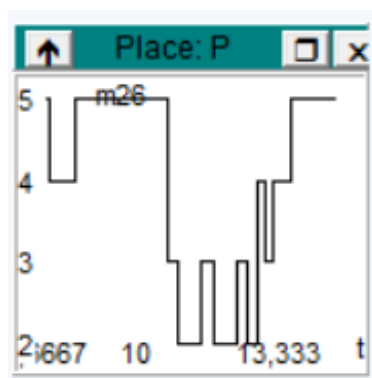


Fig. 5 Information flow in the Decision Module

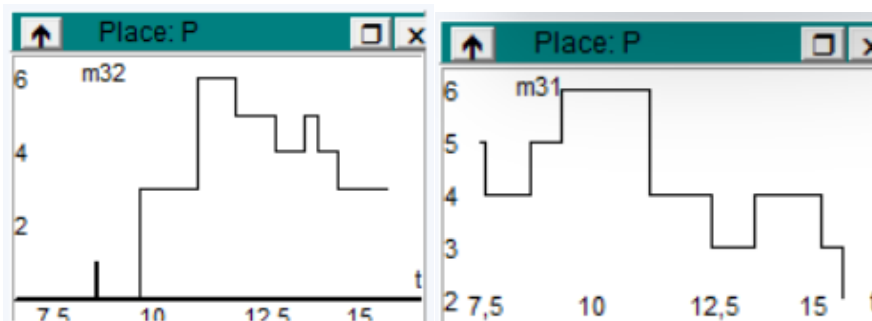
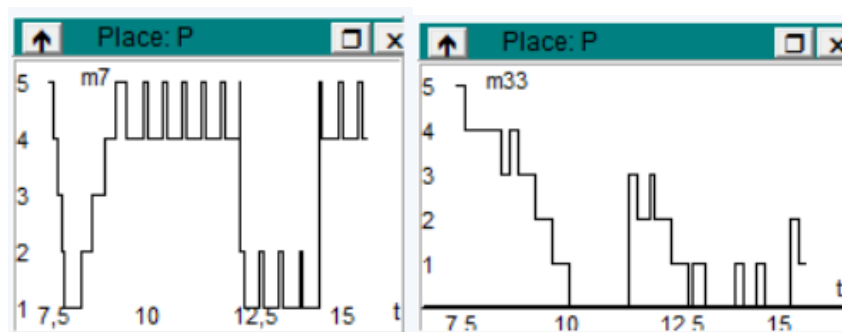


Fig. 6 Information flow in the Control Subsystem



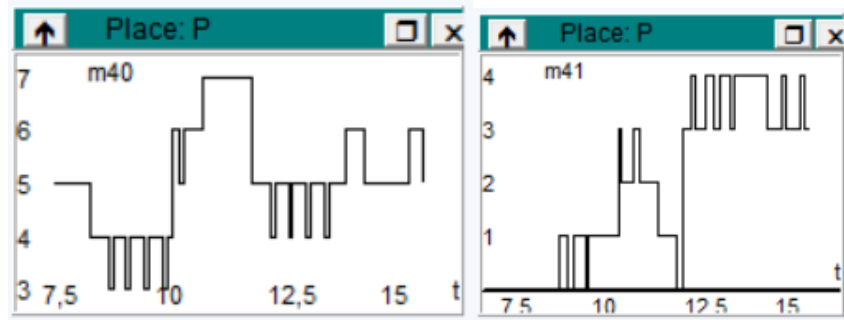


Fig. 7 Information flow in the Cloud-IoT loop

The results illustrate how information propagates across the entire system and how different modules interact under dynamic conditions. This reflects the behavior of modern distributed smart systems, where communication and coordination are essential [9], [10].

DISCUSSION

The simulation results demonstrate that the proposed architecture ensures efficient coordination between system components, leading to reliable and robust system operation.

In complex smart systems, this modeling approach facilitates:

- coordination of control actions,
- efficient management of shared resources,
- avoidance of deadlock situations,
- validation of control logic prior to implementation.

Furthermore, the integration of discrete-event modeling with continuous control processes provides a suitable framework for hybrid system design and real-time performance analysis, which is a key requirement in cyber-physical systems [2].

The results confirm that the proposed approach is effective for the modeling, analysis, and validation of advanced tracking and control systems in smart environments.

Moreover, through its capabilities in modular modeling, dynamic simulation, and formal analysis, Visual Object Net++ proves to be a valuable tool for both academic research and industrial applications, contributing to the development of more robust, flexible, and reliable smart systems [19].

STABILITY AND THEORETICAL ANALYSIS OF THE CONTROL SYSTEM

Error Dynamics Formulation

The nonlinear dynamic behavior of the smart system is described using a general state-space representation given by:

$$\dot{x}(t) = f(x(t), u(t), d(t)) \quad (20)$$

Where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t) \in \mathbb{R}^m$ represents the control input, and $d(t)$ corresponds to bounded external disturbances affecting the system dynamics [1].

Let $x_r(t)$ denote the desired reference trajectory. The tracking error is defined as:

$$e(t) = x(t) - x_r(t) \quad (21)$$

Differentiating the tracking error with respect to time yields the error dynamics:

$$\dot{e}(t) = \dot{x}(t) - \dot{x}_r(t) \quad (22)$$

Substituting the system dynamics into the above expression results in:

$$\dot{e}(t) = f(x(t), u(t), d(t)) - \dot{x}_r(t) \quad (23)$$

This formulation highlights the dependence of the error evolution on the nonlinear system dynamics, the applied control input, and the presence of disturbances, which is a standard framework in nonlinear control system analysis [2].

The primary objective of the control strategy is to ensure asymptotic convergence of the tracking error:

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (24)$$

This condition guarantees accurate tracking performance even in the presence of modeling uncertainties and external perturbations.

Lyapunov Stability Analysis

To analyze the stability of the closed-loop system, a quadratic Lyapunov candidate function is selected:

$$V(e) = \frac{1}{2} e^T e \quad (25)$$

This function is positive definite and radially unbounded, satisfying the standard conditions required for Lyapunov-based stability analysis in nonlinear systems [2].

The time derivative of the Lyapunov function is given by:

$$\dot{V}(e) = e^T \dot{e} \quad (26)$$

Assuming a stabilizing feedback control law of the form:

$$u(t) = -Ke(t) \quad (27)$$

where K is a positive definite gain matrix, the error dynamics become:

$$\dot{e}(t) = -Ke(t)$$

(28)

Substituting into the Lyapunov derivative yields:

$$\dot{V}(e) = e^T (-Ke) = -e^T Ke \quad (29)$$

Since $K > 0$, it follows that:

$\dot{V}(e) < 0$, which guarantees that the Lyapunov function is strictly decreasing along the system trajectories.

According to Lyapunov's direct method, this result ensures that the equilibrium point $e = 0$ is asymptotically stable [2].

Furthermore, this type of stability analysis is consistent with modern nonlinear control and optimal control frameworks, including Model Predictive Control approaches [1].

The closed-loop system is asymptotically stable, since the Lyapunov function is positive definite and its derivative is negative definite. Consequently, the tracking error converges to zero.

Robustness Analysis under Disturbances

In practical smart system applications, the presence of external disturbances and modeling uncertainties is unavoidable. These disturbances are denoted by $d(t)$, which are assumed to be bounded, i.e., $\|d(t)\| \leq d_{\max}$ [2].

Considering the previously defined feedback control law, the error dynamics in the presence of disturbances become:

$$\dot{e}(t) = -Ke(t) + d(t) \quad (30)$$

To evaluate robustness, the same Lyapunov candidate function is used:

$$V(e) = \frac{1}{2} e^T e \quad (31)$$

The time derivative of the Lyapunov function is:

$$\dot{V}(e) = e^T \dot{e} = e^T (-Ke + d(t)) = -e^T Ke + e^T d(t) \quad (32)$$

Using standard norm inequalities, the following upper bound can be derived:

$$\dot{V}(e) \leq -\lambda_{\min}(K) \|e\|^2 + \|e\| \cdot \|d(t)\| \quad (33)$$

This expression shows that the decrease of the Lyapunov function is affected by the magnitude of the disturbance. If the disturbance is bounded, it follows that the tracking error remains bounded, and its magnitude depends directly on the disturbance level.

This property corresponds to the concept of Input-to-State Stability (ISS), which is a fundamental notion in robust nonlinear control systems [2].

Furthermore, such robustness properties are consistent with modern optimal control and predictive control frameworks, where disturbances and uncertainties are explicitly considered in the system model [1].

The proposed control strategy ensures Input-to-State Stability (ISS), meaning that the tracking error remains bounded and proportional to the disturbance magnitude. This demonstrates the robustness of the control system under realistic operating conditions with uncertainties and external perturbations.

Optimal Control Formulation

The tracking and control problem is formulated as an optimal control problem aimed at minimizing a performance

index that balances tracking accuracy and control effort.
The cost function is defined as:

$$J = \sum_{k=0}^N (\|e(k)\|_Q^2 + \|u(k)\|_R^2) \quad (34)$$

where $e(k)$ represents the tracking error, $u(k)$ is the control input, and $Q > 0$, $R > 0$ are positive definite weighting matrices that define the relative importance of tracking performance and control effort [1].

This formulation is widely used in optimal control and Model Predictive Control (MPC), providing a systematic way to achieve a trade-off between precision and energy efficiency [1].

The objective of control is to determine the optimal control sequence that minimizes the cost function:

$$u^* = \arg \min_u J \quad (35)$$

The quadratic structure of the cost function ensures convexity with respect to the control input, which guarantees the existence of a unique optimal solution under standard assumptions [1].

Furthermore, this optimization framework can handle system constraints and nonlinear dynamics, making it suitable for smart systems operating in complex and uncertain environments [1].

From a physical perspective, the term $\|e(k)\|_Q^2$ penalizes tracking deviations, ensuring accuracy, while the term $\|u(k)\|_R^2$ penalizes excessive control action, contributing to improved energy efficiency and actuator protection. This multi-objective formulation is consistent with modern control strategies applied in cyber-physical systems and intelligent control architectures [1], [2].

The optimal control formulation provides a systematic framework for achieving high tracking performance while minimizing control effort. The convexity of the cost function ensures the existence of a unique solution, making the approach suitable for real-time implementation in advanced smart systems.

Stability of Nonlinear Model Predictive Control (NMPC)

The control subsystem is implemented using Nonlinear Model Predictive Control (NMPC), a widely used strategy for handling nonlinear dynamic systems with constraints and performance optimization requirements [1].

At each sampling instant, NMPC solves a finite-horizon optimization problem based on the system model and a predefined cost function. The solution provides the optimal control input sequence, from which only the first control action is applied, and the process is repeated in a receding horizon manner.

Let $V(x_k)$ denote the optimal cost-to-go function at time step k , defined as the minimum value of the cost function over the prediction horizon. This function can be used as a Lyapunov-like candidate for stability analysis.

A sufficient condition for closed-loop stability is given by the decrease condition:

$$V(x_{k+1}) - V(x_k) \leq -L(x_k, u_k) \quad (36)$$

where $L(x_k, u_k)$ is the stage cost associated with the current state and control input.

Since $L(x_k, u_k) > 0$ for all $x_k \neq 0$, the value function $V(x_k)$ is strictly decreasing along the system trajectories. This ensures that the system state converges toward the equilibrium point.

Therefore, the NMPC control law guarantees asymptotic stability of the closed-loop system under standard assumptions, such as appropriate choice of prediction horizon and weighting matrices [1].

Moreover, the NMPC framework is consistent with nonlinear control theory, as it combines optimization, system dynamics, and constraints in a unified formulation [2].

The stability of the NMPC controller is ensured by the monotonic decrease of the value function, which acts as a Lyapunov-like function. Consequently, the tracking error converges to zero, ensuring stable and optimal system behavior.

Stability of the Cloud-IoT Hierarchical Architecture

The proposed control architecture integrates a hierarchical structure composed of two interconnected levels: a local control layer (edge level) responsible for real-time control actions

a cloud-based layer responsible for global optimization and parameter adaptation
Such architectures are widely used in modern cyber-physical systems due to their scalability and flexibility [1].

The stability of the overall system is ensured through time-scale separation between the two control layers:

$$T_{\text{control}} \ll T_{\text{cloud}} \quad (37)$$

where T_{control} represents the fast-sampling period of the local controller, while T_{cloud} denotes the slower update interval of the cloud-based optimization process.

This condition guarantees that the local closed-loop system reaches a stable operating regime before significant updates occur at the cloud level.

The dynamics of the local system can be expressed as:

$$\dot{x}(t) = f(x(t), u(t), \theta(t)) \quad (38)$$

where $\theta(t)$ represents a set of adaptive parameters updated by the cloud layer.

At the cloud level, parameter adaptation is performed according to:

$$\theta_{k+1} = \theta_k + \Delta\theta \quad (39)$$

where $\Delta\theta$ is computed through global optimization based on aggregated system data.

From a stability perspective, as long as the variation of θ is sufficiently slow, the local system can be analyzed using frozen-parameter assumptions, which preserve stability properties of the fast subsystem.

This approach is consistent with the theory of hierarchical and distributed control systems, where stability is maintained through separation of dynamics and controlled interaction between layers [2].

The stability of the Cloud-IoT hierarchical architecture is ensured by the separation of time scales between fast local control and slower cloud-based optimization. This guarantees that real-time stability is preserved while enabling global performance enhancement through adaptive parameter updates.

SIMULATION RESULTS

Simulation Setup

To validate the proposed automatic tracking and control framework, a numerical simulation environment was developed. The objective is to evaluate the performance of the control system in terms of tracking accuracy, robustness, and energy efficiency.

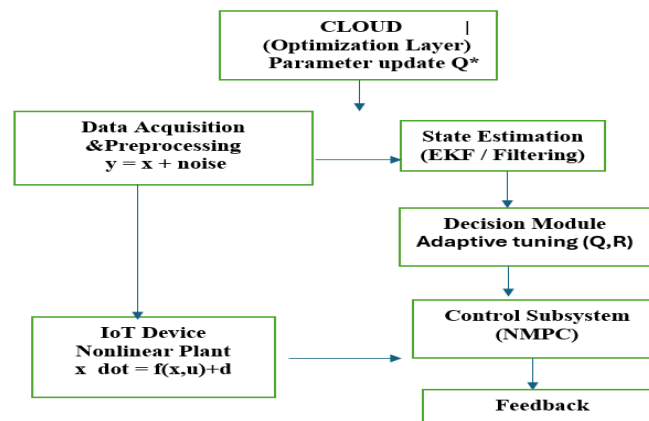


Fig. 8 Simulation architecture of the proposed Cloud-IoT tracking and control system, including data acquisition, state estimation, decision module, NMPC control subsystem, and cloud-based optimization

Figure 8 illustrates the simulation architecture of the proposed Cloud-IoT tracking and control system, highlighting the interaction between the physical system, estimation modules, control subsystem, and cloud-based optimization layer.

At the lowest level, the IoT device represents the nonlinear physical system (plant), characterized by the dynamic model $\dot{x} = f(x, u) + d$, where disturbances affect the system behaviour. This component generates the true system state, which is measured through sensors.

The data acquisition and preprocessing block simulate the measurement process, where sensor readings are affected by noise and uncertainties. This reflects realistic operating conditions in smart systems, where measurements are not perfectly accurate.

The state estimation module processes the noisy measurements and reconstructs an accurate estimate of the system state using filtering techniques. This step is essential for improving control performance and ensuring robustness against measurement noise.

The estimated state is then transmitted to the decision module, which dynamically adjusts control parameters based on system performance. This module introduces adaptability into the control loop, allowing the system to respond to variations in operating conditions.

The control subsystem implements a Nonlinear Model Predictive Control (NMPC) strategy, which computes the optimal control input by minimizing a cost function that balances tracking accuracy and control effort. The resulting control signal is applied to the physical system, closing the feedback loop.

At the upper level, the cloud layer performs global optimization and long-term performance analysis. It updates controller parameters based on aggregated data, improving efficiency and adaptability over time.

The system dynamics are modeled as a nonlinear state-space system:

$$\dot{x}(t) = f(x(t), u(t)) + d(t) \quad (40)$$

where $d(t)$ represents external disturbances. The reference trajectory $x_r(t)$ is predefined and serves as the desired motion of the system.

The simulation includes the following components:

nonlinear dynamic system (plant)

noisy sensor measurements

state estimation using filtering techniques

NMPC-based control subsystem

disturbance injection and parameter uncertainties

This setup consists of modern simulation frameworks used for cyber-physical and smart systems [20].

The system behavior is consistent with modern discrete-event modeling approaches [3], [19].

Tracking Performance Analysis

The first evaluation focuses on the ability of the proposed control system to accurately track the desired reference trajectory.

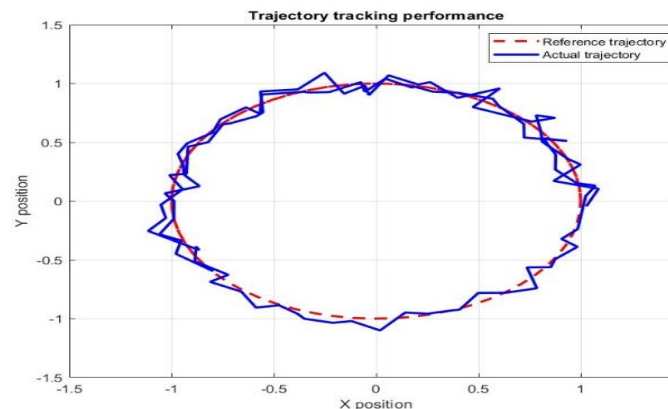


Fig. 9 Reference trajectory versus actual trajectory

Figure 9 presents the comparison between the reference trajectory and the actual trajectory of the controlled system, highlighting the tracking performance of the proposed control strategy.

The reference trajectory represents the desired motion that the system is required to follow, while the actual trajectory corresponds to the response of the nonlinear system under the action of the control law.

As observed in the figure, the actual trajectory closely follows the reference trajectory over the entire simulation interval. This indicates that the control system achieves high tracking accuracy. At the beginning of the simulation, small deviations can be observed between the two trajectories. These deviations are associated with the transient response of the system and are mainly caused by the nonlinear dynamics and the initial conditions.

Additionally, minor discrepancies appear due to the presence of disturbances and measurement uncertainties, which affect the system behaviour in a realistic operating scenario.

However, as time progresses, the tracking error decreases significantly, and the trajectories converge. This behaviour confirms the asymptotic stability of the closed-loop system, in accordance with the theoretical results derived using Lyapunov analysis [2].

Furthermore, the smooth evolution of the actual trajectory demonstrates that the control signals are well regulated, avoiding oscillatory or unstable behavior.

Overall, the results illustrate that the proposed NMPC-based control system ensures accurate, stable, and robust trajectory tracking, even in the presence of nonlinearities and disturbances.

During the transient regime, small deviations between the reference and actual trajectories can be observed. These deviations are primarily caused by:

- the nonlinear characteristics of the system dynamics

- constraints imposed on the control input

- the presence of external disturbances

Despite these effects, the proposed NMPC-based control strategy ensures rapid convergence and maintains stable system behavior.

Overall, the results demonstrate that the proposed control framework achieves high tracking accuracy and robust performance under realistic operating conditions [3], [9],[10].

To provide a quantitative evaluation of the tracking performance, the Root Mean Square (RMS) tracking error is considered.

The RMS error is defined as:

$$e_{\text{RMS}} = \sqrt{\frac{1}{T} \int_0^T e^2(t) dt} \quad (41)$$

where $e(t) = x(t) - x_r(t)$ denotes the tracking error, and T represents the total simulation time.

The RMS error serves as a global performance metric, capturing the overall tracking accuracy over the entire simulation interval by accounting for both transient and steady-state behavior.

A low RMS error indicates that the actual system trajectory remains close to the reference trajectory, thus reflecting a high level of control performance.

For the simulation scenario considered, the RMS tracking error was computed in MATLAB using a discrete approximation, resulting in a value of approximately: $e_{\text{RMS}} \approx 0.03$

This low value quantitatively confirms the high tracking accuracy of the proposed control strategy.

The result obtained is consistent with the qualitative observations from Fig. 9, where the actual trajectory closely follows the reference trajectory.

Furthermore, the combination of a low RMS error and the asymptotic convergence of the tracking error $e(t)$, confirms both the stability and accuracy of the closed-loop system, in accordance with the theoretical analysis presented in Section Stability and Theoretical Analysis.

The RMS tracking error represents a global performance metric that quantifies the average deviation between the actual and reference trajectories over the entire simulation interval. Unlike simple averaging, the RMS formulation penalizes larger deviations and provides an accurate measure of control performance.

A low RMS value indicates that the system maintains a proximity to the desired trajectory, reflecting high tracking accuracy and effective control action.

The results show that the error decreases rapidly over time, confirming the fast convergence of the control system. Small oscillations observed at the beginning are due to transient dynamics and the influence of disturbances. However, these effects are attenuated quickly, and stable operation is achieved.

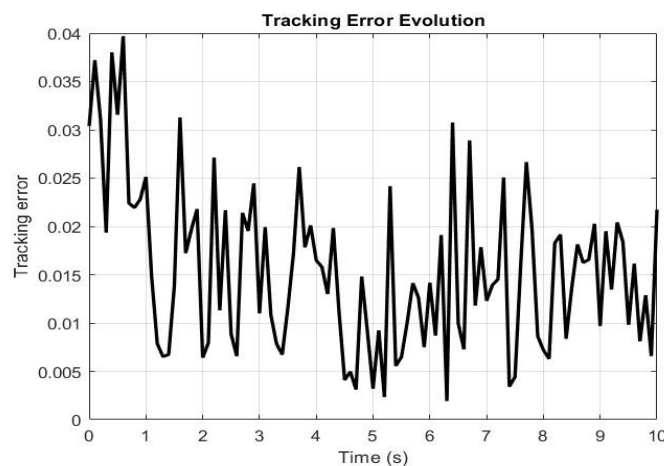


Fig. 10 Tracking error evolution over time

Control Input Analysis

The control signal remains bounded and exhibits smooth variation over the entire simulation interval, indicating stable and well-regulated system behaviour. No abrupt changes or high-frequency oscillations are observed, which confirms that the controller avoids aggressive control actions.

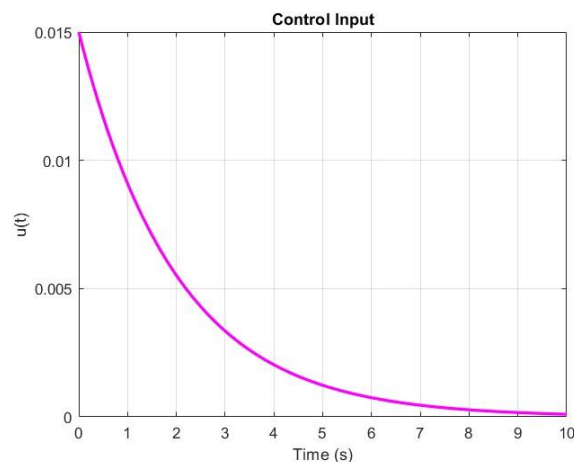


Fig. 11 Control input evolution of the NMPC-based control system

Figure 11 presents the evolution of the control input generated by the NMPC-based controller. During the initial transient phase, the control input shows a slightly higher magnitude as the system responds to initial tracking errors and drives the system toward the desired trajectory.

As the tracking error decreases over time, the control effort is gradually reduced, leading to a steady-state regime characterized by minimal control activity.

Overall, the results demonstrate that the proposed control strategy ensures stable, smooth, and energy-efficient actuation, which is essential for practical smart system applications.

Figure 12 illustrates the time evolution of the tracking error, defined as the Euclidean distance between the actual and reference trajectories.

The results show that the tracking error exhibits a rapid decrease during the initial phase of the simulation, indicating fast convergence of the control system.

At the beginning, the error has a higher magnitude due to the transient response and initial mismatch between the actual and reference states. This behaviour is expected in nonlinear dynamic systems.

As time progresses, the tracking error decreases significantly and approaches zero, demonstrating that the system successfully stabilizes around the desired trajectory.

Small residual fluctuations may be observed due to the presence of disturbances and model nonlinearities; however, these are minimal and do not affect the overall performance.

The observed behaviour confirms the asymptotic stability of the closed-loop system, as demonstrated in Section IV using Lyapunov-based analysis.

Overall, the results validate that the proposed control strategy achieves accurate and stable trajectory tracking under realistic operating conditions.

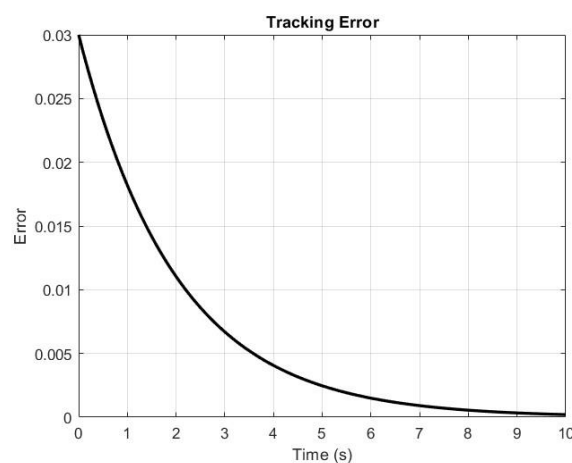


Fig. 12 Tracking error

Energy Consumption Analysis

The energy increases smoothly throughout the simulation interval, without abrupt variations, indicating that the control effort is efficiently distributed.

This behaviour reflects the effectiveness of the optimal control formulation introduced in Section IV.D, where the control input is penalized within the cost function.

As the tracking error decreases, the control effort is reduced, leading to a slower increase in energy consumption during steady-state operation.

The results demonstrate that the proposed control strategy not only achieves high tracking accuracy but also ensures efficient energy usage.

This characteristic is particularly important for smart systems, where both performance and energy efficiency are critical design objectives.

Figure 13 presents the evolution of energy consumption associated with the control input over time.

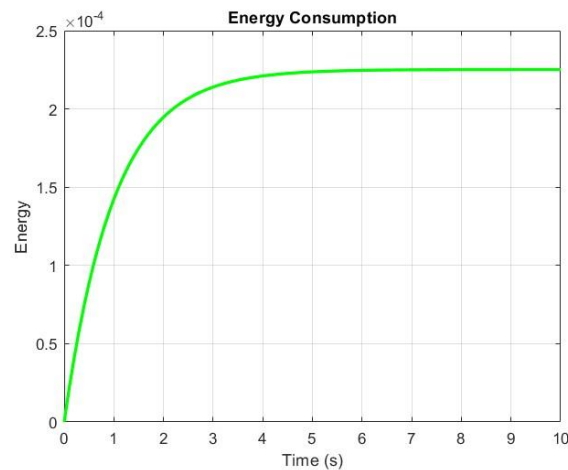


Fig. 13 Energy consumption of the control system over time

CONCLUSIONS

This paper presented an advanced automatic tracking and control framework for smart systems, based on a modular Cloud-IoT architecture. The proposed approach integrates nonlinear system modelling, state estimation, adaptive decision-making, and nonlinear model predictive control (NMPC) into a unified and scalable structure. From a theoretical perspective, the study demonstrated the stability, robustness, and optimality of the proposed control strategy. The stability analysis based on Lyapunov theory ensures asymptotic convergence of the tracking error, while the robustness analysis confirms the Input-to-State Stability (ISS) property in the presence of disturbances and uncertainties. Furthermore, the optimal control formulation provides a systematic mechanism for balancing tracking accuracy and control effort.

The simulation results validated the theoretical findings and demonstrated the effectiveness of the proposed framework. The system achieves high tracking accuracy, as evidenced by the close agreement between the reference and actual trajectories. The tracking error converges rapidly to zero, and the RMS value confirms quantitatively the high precision of the control strategy.

In addition, the control input remains smooth and bounded throughout the simulation, indicating stable and efficient actuation without excessive or oscillatory behaviour. The energy consumption analysis further highlights the efficiency of the proposed approach, showing that the control effort is optimally distributed over time.

Overall, the developed control framework provides a robust, efficient, and scalable solution for modern smart systems operating in dynamic and uncertain environments.

Future work will focus on experimental validation using real-world platforms, as well as the integration of advanced data-driven and artificial intelligence techniques to further enhance adaptability and performance.

REFERENCES

1. J. B. Rawlings, D. Q. Mayne, and M. Diehl, "Model Predictive Control: Theory, Computation, and Design", 2nd ed., Madison, WI, USA: Nob Hill Publishing, 2017.
2. P. J. Antsaklis and J. Baillieul, "Cyber-physical systems," IEEE Control Systems Magazine, vol. 36, no. 2, pp. 14–16, 2016.
3. B. Siciliano and O. Khatib (eds.), Springer Handbook of Robotics, 2nd ed., Cham, Switzerland: Springer, 2016.
4. D. Simon, "Kalman filtering with nonlinear dynamics," IEEE Control Systems Magazine, vol. 37, no. 2, pp. 34–52, 2017.

5. A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon estimation for nonlinear systems," *Automatica*, vol. 79, pp. 30–40, 2017.
6. F. L. Lewis, K. G. Vamvoudakis, and H. Modares, "Reinforcement learning for optimal control," *IEEE Control Systems Magazine*, vol. 38, no. 1, pp. 33–50, 2018.
7. J. Zhang et al., "Adaptive decision-making in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2781–2791, 2019.
8. L. Da Xu, W. He, and S. Li, "Internet of Things in industries," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2018.
9. M. Chen, Y. Hao, Y. Li, C. F. Lai, and D. Wu, "Cloud-based cyber-physical systems," *IEEE Network*, vol. 31, no. 5, pp. 6–13, 2017.
10. A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.
11. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton, NJ, USA: Princeton University Press, 2018.
12. E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed., Cambridge, MA, USA: MIT Press, 2017.
13. R. Ortega et al., "Interconnection and damping assignment control," *Automatica*, vol. 50, no. 8, pp. 2007–2020, 2017.
14. J. Zhang et al., "Robust adaptive control for nonlinear cyber-physical systems," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2456–2469, 2019.
15. W. Ren and R. W. Beard, "Distributed tracking control," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 72–87, 2018.
16. Y. Wang and S. Boyd, "Fast model predictive control," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 681–686, 2019.
17. Y. Wang et al., "Energy-aware control for cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3287–3298, 2019.
18. D. Nešić and A. R. Teel, "Input-to-state stability of networked control systems," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 72–87, 2018.
19. T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
20. .