International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-50, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

# Predictive Pontryagin Optimal Control of Nonlinear Fully Distributed DCS and CPS under Reliability and Uncertainty Constraints

**Ass.eng. Iliyan Vasilev, PhD[1], Eng. Nedko Perchemliev[2]**
[1]University of Chemical Technology and Metallurgy
[2]PhD Student, University of Chemical Technology and Metallurgy

**ABSTRACT:** The article provides an overview of the optimal control prediction framework for non-linear CPS and DCS in automation. The core issue is the fact that the present architecture of modern automation systems does not behave like a closed loop control system anymore. The design of these systems combines sensors, actuators, edge controllers, communication channels, digital twin, software-as-a-service, human involvement, and states of cyber-security. Thus, control design must not only address accuracy but should be a multicriteria design problem that accounts for reliability, uncertainty, probability mass, delay in communication, and energy costs. Four levels are considered in the proposed framework. The first level refers to forecasting based on hybrid ARIMA-ML model for a short horizon. The second level is concerned with estimating of risk states using Markov model/HMM. The contribution is a simulation-ready mathematical architecture in which each node solves a local Hamiltonian problem using predicted states, neighbour information and reliability constraints, while the global CPS behavior emerges through networked local decisions. The paper formulates the nonlinear dynamics, cost functional, Hamiltonian conditions, reliability constraints and evaluation protocol for smart factories, smart grids, intelligent buildings and smart campuses. The framework is positioned as a bridge between predictive maintenance and optimal distributed automation control.

**KEYWORDS:** pontryagin maximum principle, nonlinear optimal control, distributed control systems, cyber-physical systems, reliability, uncertainty, predictive maintenance.

## INTRODUCTION

Automation is changing from a hierarchical approach of supervisory control to networked and completely distributed controls. In conventional process control systems, coordination of field equipment and controllers takes place via supervisory hierarchy. In modern CPS or DCS systems, local devices have the ability to sense, compute, communicate and act. This architectural transition is visible in smart factories, power systems, intelligent buildings, transport infrastructures and smart campuses, where thousands of cyber-physical nodes exchange information and coordinate physical action. The CPS theory focuses on strong coupling of computations with physical processes, whereas distributed automation technologies and function block structures focus on partitioning the control tasks among several components (Lee, 2008; Rajkumar et al., 2010; Vyatkin, 2011). The problem here is that a distributed approach does not imply that an intelligent decision-making process occurs. Each of the local controllers will have a piece of information regarding the state of the whole system and will use delayed information from the neighbors to make decisions about controlling its own part of the process. If these local rules are not mathematically coordinated, they may reduce local error while increasing system-level risk. Therefore, the central problem of modern automation is not merely communication, but uncertainty-aware optimal coordination. This requires a theory in which local decisions are linked to global safety, reliability and performance.

Previous predictive and stochastic approaches in the authors articles provide important components of this problem. Hybrid ARIMA--ML forecasting can capture both linear and nonlinear structures in CPS time series and can improve predictive maintenance decisions (Zhang, 2003). Markov and hidden Markov models can represent discrete operational modes such as Normal, Warning, Overload, Fault and Recovery (Vasilev, 2026). Multilayer reliability modelling expands the reliability concept beyond hardware failure to include communication, software, cyber-security, data integrity and recovery (Rausand & Høyland, 2004). A stochastic distributed DCS/CPS framework can be usefull in explaination how local Markov transitions, Chapman–Kolmogorov composition, and Fokker–Planck uncertainty propagation interact within large automation networks. The missing layer is optimal control synthesis. Prediction and reliability assessment are valuable only if they are converted into control actions.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-50, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

This article proposes the use of Pontryagin's maximum principle as the optimization layer that transforms predicted states, reliability indices and uncertainty constraints into local optimal controls. The scope of the proposed framework covers networked CPS/DCS environments in which industrial assets and operational resources can be represented through RAMI 4.0 and IEC/EN 62264 information-integration models, while distributed sensor networks provide real-time data streams for monitoring, event identification, and reliability-aware predictive control (Ilieva & Metodiev, 2020; Vasilev & Metodiev, 2021). Pontryagin theory is suitable because it naturally handles nonlinear dynamics, constrained controls, terminal penalties and necessary optimality conditions expressed through a Hamiltonian system.

## THEORETICAL BACKGROUND

Optimization problem that maximizes some objective while minimizing the cost. The cost usually includes tracking error and control effort in automation and should contain additional terms in CPS/DCS: reliability degradation, confidence in data flow from cyber-secured sensors, communication latency and uncertainty. The classical design using feedback loops is crucial for stabilization analysis and implementation (Åström & Murray, 2008; Khalil, 2002). However, nonlinear distributed automation implies more general optimization since the process dynamics, the mode of operation and information properties evolve in time.

The maximum principle proposed by Pontryagin provides optimal criteria to such problems based on pairing up every state variable with its corresponding adjoint variable, using the Hamiltonian function which accounts for the immediate cost and the dynamics of the state variable. The maximum principle formulated by Pontryagin is more of a structure than an algorithm in that it enables formulation of any optimization problem in terms of coupled variables. This is the standard approach in deterministic and stochastic control (Bertsekas, 2017; Fleming & Rishel, 1975; Rawlings et al., 2017). The Hamiltonian should be augmented with future predictions for CPS control. Hybrid prediction will provide estimates of the expected load, residual errors, degradation rate and intensity of fault precursors. Prediction is crucial in that sense that ARIMA-like models account for linear dynamics, while machine-learning modules are responsible for nonlinear residual dependencies and abnormal operation regimes. Thus, prediction ceases to serve as monitoring, but becomes an argument of the optimization problem.

Finally, reliability should be treated as a control parameter. Contrary to common practice, reliability loss should be included directly in the cost. When the system loses reliability under some threshold, it should redistribute the load, reduce control effort, switch into safe operating mode, modify communication or order maintenance procedures. This perspective inherits the principles of reliability engineering in a modified way to be applicable to complex networked CPS prone to technical, informational and cyber-physical failures (Griffor et al., 2017).

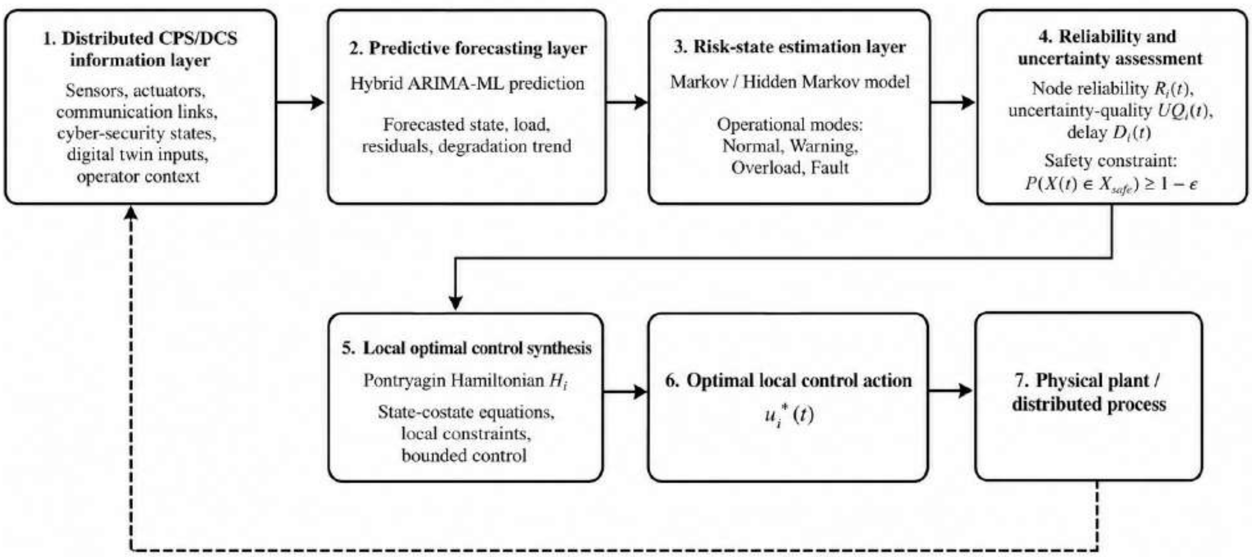


**Figure 1. Predictive Pontryagin optimal control architecture for fully distributed CPS/DCS. (author's source)**

**Motivation for Reliability-Aware Predictive Optimal Control in Distributed CPS/DCS**

CPS/DCS of the modern world work in an environment that substantially differs from those typical for conventional automation solutions. For instance, in classic architecture, it is assumed that there is full information about the overall state available to the controller, communication delay is zero, and uncertainty in the environment is considered a disturbance. Modern CPS/DCS use a fully distributed architecture where different nodes like sensors, actuators, edge controllers, software services, and cyber security are present, and each of them receives partial information about the global state. The discrepancy between assumptions made in classical optimal control theory and practical challenges of distributed automation leads to necessity to include multiple additional factors into consideration when designing distributed control systems. These factors include reliability degradation, uncertainty spreading, probability mass movement, delay, and quality of cyber-physical data, among others.

Predictive control becomes essential in this context. Forecasting future states, degradation trends and risk-state transitions allows each node to anticipate unsafe trajectories rather than react to them. Reliability indices and uncertainty measures must therefore become explicit arguments of the optimization problem. The motivation for the proposed framework is to unify prediction, reliability modelling and Pontryagin optimal control into a single structure that enables each node to compute locally optimal actions while contributing to globally safe and resilient behaviour.

***Architecture of Fully Distributed CPS/DCS***

A fully distributed CPS/DCS cannot be analyzed only using mathematical formulas as it behaves depending on the collaboration of multiple autonomous nodes. Every node is equipped with sensing, computing, communication and actuation abilities, which makes an architectural view obligatory for analyzing the interaction between local predictions, reliability, uncertainty, and optimal control within such a system.

Structurally, the network might be presented as a graph $G(V,E)$ where every node $i \in V$ holds a state $x_i(t)$,operational mode $z_i(t)$,a control policy $u_i(t)$, and an information quality indicator $q_i(t)$. Links $E$ represent communication channels, physical coupling or informational connection of any type. None of the nodes possess any global knowledge, and global behavior emerges due to local interactions.

The architecture is formed by three different analytical levels:

*(a) Local stochastic level (controlled Markov transitions).*

Every node undergoes the process of controlled Markov transition between various operational states, such as Normal, Warning, Overload, Fault, and Recovery. Transitions depend on measurement data, neighbor states, and information quality. Distributed control leads to changing transition probabilities in order to reduce unwanted transitions and increase recovery chances.

*(b) Temporal composite level (Chapman-Kolmogorov relation).*

All local transitions aggregate over time, as described by the Chapman-Kolmogorov equation. It models temporal properties of distributed control: numerous small, asynchronous decisions are combined into one global behavior after a period of time. Consistency of all local policies after such aggregation is ensured by this level.

*(c) Uncertainty propagation level (Fokker-Planck dynamics).*

Physical processes modeled by continuous random variables undergo stochastic differential equations, while their probability distributions change according to the Fokker-Planck equation. This level models disturbance correlation, uncertainty propagation and probability concentration due to feedback.

These three levels create a coherent architectural view, in which local interactions between predictions, reliabilities and optimizations result in global behavior of fully distributed CPS/DCS (**Figure 2**.).

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-50, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

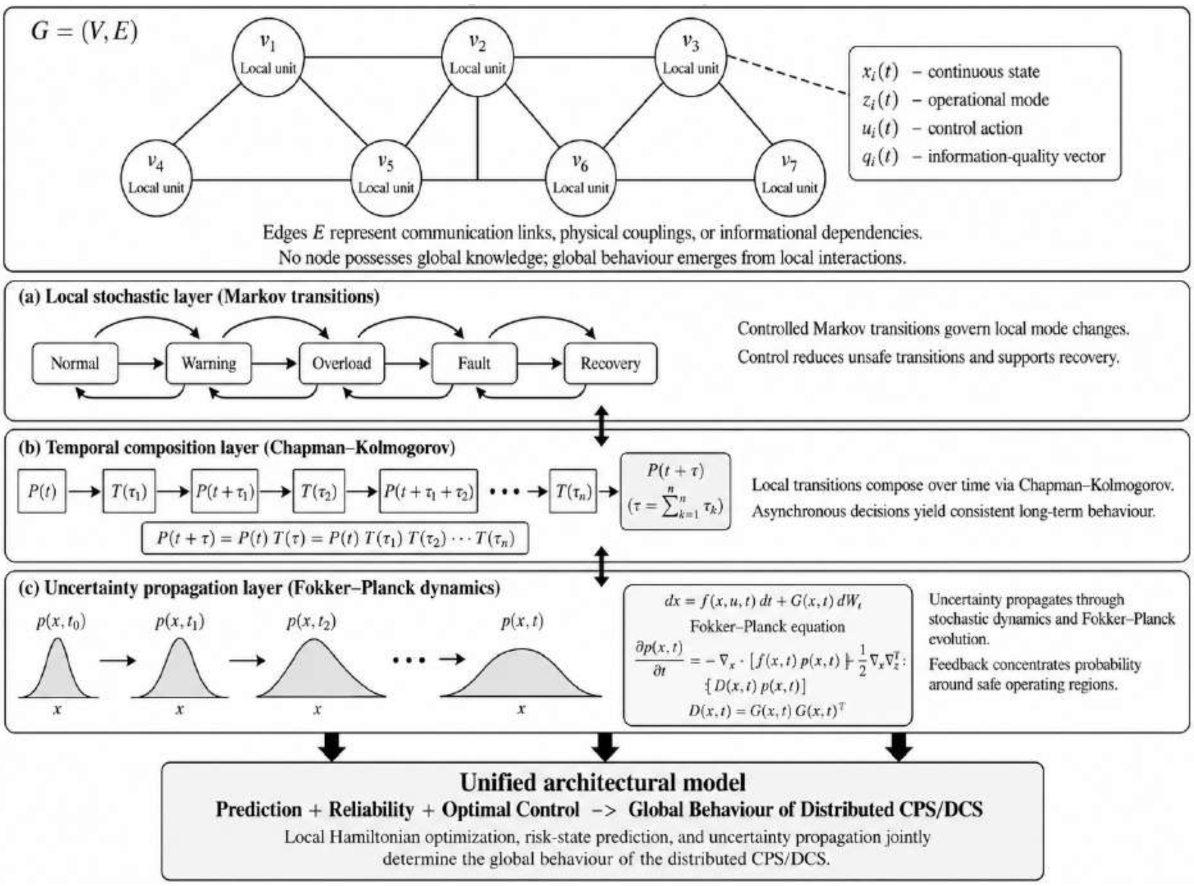


**Figure 2. Architectural representation of fully distributed CPS/DCS (author's source)**

***Proposed Nonlinear Distributed Model***

Let the automation system be represented by a graph G = (V, E), where V is the set of cyber-physical nodes and E denotes communication, informational or physical coupling. Each node i has a local continuous state $x_i(t)$, a local control $u_i(t)$, an estimated neighbourhood state x_Ni(t), an operational mode$z_i(t)$, and an information-quality vector $q_i(t)$. The vector $q_i(t)$ may include sensor confidence, packet-loss level, cyber-security trust, data freshness and actuator health. This is critical since the CPS nodes used in contemporary times take into account not only the data but also the reliability of data that forms the basis for these calculations (Lee & Seshia, 2017).

$$dx_i(t) = f_i(x_i(t), x_N i(t), u_i(t), q_i(t), t)dt + g_i(x_i(t), q_i(t), t)dW_i(t)$$

The first term expresses nonlinear drift caused by physical laws and control actions. The second term expresses diffusion, modelling disturbances, measurement noise, uncertain load and cyber-physical perturbations. At the discrete level, the local operational mode evolves through a controlled Markov transition:

$$P(z_i(t+1) = b | z_i(t) = a) = p_a b^{(i)}(u_i(t), x_N i(t), q_i(t)).$$

This equation has a direct automation interpretation: the local controller does not merely change the physical state; it changes the probability of transitions between operational modes. Good control should decrease the probability of unsafe transitions such as Warning -> Fault and increase the probability of recovery transitions such as Fault -> Recovery. The fully distributed control law is therefore written as:

$$u_i(t) = K_i(x_i(t), x_N i(t), z_i(t), q_i(t), r_i(t)), i = 1, \ldots, N.$$

The global action U(t) is the composition of all local controls. No single node needs the complete global state, but every node must solve a local problem that is consistent with network-level safety and reliability. This is the reason for selecting a local Hamiltonian formulation rather than a purely centralized optimization problem.

**Predictive Pontryagin Formulation**

This optimization problem tries to minimize the cost represented by an integral on the time interval [0, T]. This cost function includes tracking errors, control costs, loss due to unreliability, uncertainty, and delay. In the case of node i, the cost function becomes:

$$J_i = \int_0^T \left[\alpha_i e_i^2(t) + \beta_i u_i^2(t) + \gamma_i\big(1 - R_i(t)\big) + \delta_i UQ_i(t) + \eta_i D_i(t)\right] dt + \Phi_i\big(x_i(T)\big).$$

In this case, $e_i(t)$refers to tracking error or regulation error, $R_i(t)$ is the reliability index of the node, $UQ_i(t)$ refers to the uncertainty and quality cost, $D_i(t)$ is the communication delay or age cost, and $\Phi_i$ is the final penalty function. The factors $\alpha_i, \beta_i, \gamma_i, \delta_i$, and $\eta_i$represent the relative importance of the application's needs to the engineer designing the controller. While the smart grid may place a large weighting factor on safety of voltage and frequency, a smart factory might place more emphasis on throughput and reliability.

For each node, the Hamiltonian is defined as:

$$H_i = L_i(x_i, u_i, R_i, UQ_i, D_i, t) + \lambda_i^T f_i(x_i, x_N i, u_i, q_i, t).$$

The local necessary optimality conditions are:

$$\dot{x}_i^* = \frac{\partial H_i}{\partial \lambda_i}, \qquad \dot{\lambda}_i^* = -\frac{\partial H_i}{\partial x_i}, \qquad u_i^* = \arg\min_{u_i \in U_i} H_i$$

These conditions provide the bridge between prediction and actuation.

The forecast predicts the state and its uncertainty over the time horizon; the Markov model predicts the probability distribution of risk states; the reliability model predicts the progression of deterioration; and the Pontryagin approach predicts the control that minimizes the Hamiltonian function. In reality, the optimizer may take the form of numerical shooting, direct collocation, sequential quadratic programming, or an approximated local policy. The important thing is that control will be based on explicit criteria rather than heuristic response to alarm conditions.

***Reliability and Uncertainty Constraints***

Firstly, we should impose some reliability constraints. Local constraints imply that the value of the local reliability index at each node should stay above the certain threshold $R_{imin}$. Global constraints require that the probability mass of the state distribution stays within the safe region $X \quad X_{safe}$. We write these constraints as follows::

$$R_i(t) \geq R_{i,\min}, \qquad \mathbb{P}(X(t) \in X_{\text{safe}}) \geq 1 - \varepsilon, \qquad t \in [0, T]$$

In other words, the first constraint safeguards the critical local elements, while the second one helps to ensure the proper operation of the global automation network. In particular, the probability part is very important for distributed systems since, in such networks, the failure happens through the gradual change in the state distribution: from Normal to Warning to Overload and, finally, to the global fault. This concept is supported by stochastic modeling of CPS and the understanding of reliability as a multilayer, dynamic and actionable measure (Rausand & Høyland, 2004).

Essential conclusion should be taken into account of uncertainty because the CPS measurements and communications are noisy and imprecise. The uncertain state propagation in the continuous stochastic model is modeled by diffusion and the probability density function $p(X, t)$.Using the Fokker--Planck interpretation, it becomes possible for the designer to analyze how the feedback concentrates probability mass around the safe states and how the noise and delay disperse the state distribution towards the dangerous states (Fleming & Rishel, 1975; Øksendal, 2003).

**Table 1. Reliability-aware control objectives in automation CPS/DCS**

| Objective | Role | Meaning | Response |
|---|---|---|---|
| Tracking | $\alpha_i e_i^2(t)$ | Maintain reference values. | Adjust local actuation. |
| Energy | $\beta_i u_i^2(t)$ | Avoid aggressive control. | Smooth the control profile. |
| Reliability | $\gamma_i\big(1 - R_i(t)\big)$ | Penalize degradation. | Reconfigure or maintain. |

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-50, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

| | | | |
|---|---|---|---|
| Uncertainty | $\delta_i UQ_i(t)$ | Account for noise/data loss. | Use fused or conservative control. |
| Delay/data age | $\eta_i D_i(t)$ | Account for delayed data. | Event update or fallback. |

**Simulation-Ready Evaluation Design**

The proposed article can be validated through a MATLAB/Simulink or Python-based simulation of a distributed automation system. A suitable example is a smart factory cell composed of machines, conveyors, sensors, local controllers and edge gateways. Each node has continuous variables such as motor load, temperature, vibration or power consumption, and discrete modes such as Normal, Warning, Overload, Fault and Recovery. The communication graph defines which nodes exchange information. The simulation should include nominal operation, peak-load operation, early-stage degradation, communication delay and cyber-physical data-quality degradation.

This comparative analysis focuses on four different controllers, which include a classical PID controller, a linear-quadratic regulator, model predictive control, and a new approach of predictive Pontryagin control. PID is considered appropriate for a practical benchmark; LQR can be regarded as an instance of linear optimal feedback control; MPC can be viewed as a case of receding-horizon optimization under constraints; finally, our algorithm incorporates additional factors for reliability and uncertainty into the Hamiltonian. This is an apt comparison considering that dynamic programming, optimal control, and MPC are based on similar concepts but differ in approaches (Bertsekas, 2017; Bryson & Ho, 1975; Kirk, 2004).

The performance indicators should include integral absolute error, control energy, mean recovery time, percentage of time in safe modes, probability of transition to Fault, reliability index trajectory and computational time per node. An additional indicator is the Hamiltonian residual or optimality gap, which shows whether local numerical solutions remain consistent. The aim is not to claim universal superiority, but to demonstrate when the predictive Pontryagin formulation provides an advantage: high nonlinearity, partial information, degradation, uncertainty and safety-critical decision making..

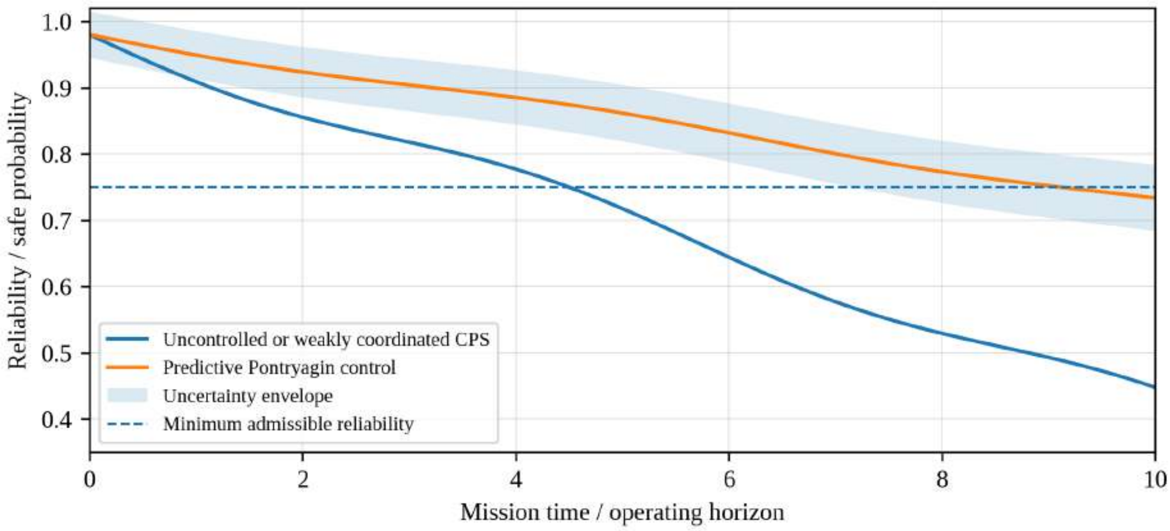


**Figure 2. Conceptual reliability effect of predictive Pontryagin control under uncertainty.**

**DISCUSSION**

One of the key theoretical contributions is the shift from prediction/reliability into control synthesis problem. In many cases in CPS, predictions are utilized to make decision about maintenance, while reliability is employed to measure performance once it has been operationalized. However, in the suggested paradigm both prediction and reliability will become components of Hamiltonian. This will make our control proactive because it will no longer wait for the failure to happen; it will instead punish trajectories leading to unreliable, uncertain or dangerous conditions. This approach would be reasonable to employ in automation problems where failure will follow warning closely. The second contribution is the local nature of the formulation. A fully distributed system cannot depend on a single optimization centre because that would reintroduce latency, communication burden and a single point of failure. With the use of the local Hamiltonian, it becomes possible for each node to determine its control action based on its own state, states of its neighbours and their reliability. In addition, the global performance is ensured by means of the coupling functions and the safety constraints that are used in common. This is in line with the concept of distributed automation (Vyatkin, 2011). The third contribution concerns uncertainty. In deterministic optimal control, a trajectory is assumed to be known once the initial condition and control are specified. In CPS/DCS, this assumption is too strong. Sensor noise, incomplete data, software delay, data integrity problems and cyber events generate multiple possible futures. The proposed framework handles this by combining predictive models, Markov states and stochastic diffusion. The role of Pontryagin control is then to select actions that guide the most probable futures toward safe and reliable regions rather than merely optimize one nominal trajectory. More specifically, the contribution overview of the **three strong parts**:

[1] ***Prediction and reliability analysis are embedded within control synthesis,*** not merely monitoring or maintenance. Prediction using ARIMA-ML, risk state prediction using Markov model/HMM and reliability index measures need to be incorporated directly into the Hamiltonian/cost function..

[2] ***The approach is distributed***, not centralized. Each CPS/DCS node uses local state, neighbour estimates, reliability, uncertainty, and mode probabilities to compute its own control action.

[3] ***The framework is positioned as a bridge between predictive maintenance and optimal distributed automation control***, especially for smart factories, smart grids, intelligent buildings, and smart campuses.

However, there are some limitations. The conditions posed by Pontryagin may be necessary, but they are not always sufficient to ensure that the optimization process is global. In addition, solving state–costate equations numerically may turn out to be quite challenging for large-scale networks. Furthermore, local optimization can lead to problems if there is a strong coupling, along with inadequate information transfer. Therefore, research needs to be carried out in the field of distributed computation, weight tuning, digital twins, and event-driven optimization.

**Practical Implementation in Automation Systems**

Implementation in a real automation environment should follow a layered design. The first layer collects sensor, actuator and communication data from PLCs, intelligent electronic devices, edge units and supervisory platforms. The second layer preprocesses the data and computes short-horizon forecasts, residuals and confidence intervals. Layer three deals with the realization of continuous-to-modes mappings and execution of the update of the local transition matrix. The next one computes reliability indices and quality parameters used for calculation of the Hamiltonian. Layer five solves the problem of local optimization and provides bounded control actions to actuators or sub-layers. Such a layered architecture allows easy integration of IEC 61499 compliant function blocks and/or digital-twin-based edge orchestrator services (Lee & Seshia, 2017; Vyatkin, 2011).

It should be noted that no solution of a two-point boundary value problem on each time step is provided in the implemented approach. Instead, each computational node utilizes its local finite-time horizon, exchanging compact data between them: probabilities of modes predictions, reliability indices, margins of uncertainties, as well as neighboring states. In case of systems featuring slow dynamics in such domains as energy management or building automation, the update of the Hamiltonian is done once in several minutes. However, for fast control loops, the role of Pontryagin layer will be that of the supervisor optimizing the Hamiltonian weights and constraints online, while actual control actions are computed in the PID and state feedback loops.

Safety certification and interpretability are also important. The proposed method is suitable for engineering use because the cost terms are explicit and each term has a clear physical meaning. If the controller increases energy consumption, the reason can be traced to reliability loss, uncertainty reduction or risk-state recovery. This is preferable to a purely black-box policy when automation engineers must explain why the system slowed a machine, isolated a communication segment or redistributed load. Thus, the framework may serve both as a control method and as an engineering governance tool.

**Future Research Directions**

The future directions could include finding numerical techniques to address the local Hamiltonian problem when the node dynamics are nonlinear, and the reliability index is updated dynamically. One of the potential techniques might involve using Pontryagin principles in conjunction with MPC where the former provides a structure for achieving optimization, and the latter provides the constraints to implement the algorithm efficiently through receding horizon approach. The other potential technique includes learning the local value function or costate variables using simulation data. A second research direction is stochastic validation. The proposed method should be tested not only on average performance but also on tail events: rare faults, communication bursts, cyber-physical data corruption and cascading failures. This is important because CPS reliability is often determined by low-probability/high-impact events. Monte Carlo testing, digital twins and hardware-in-the-loop simulation can be used to estimate whether the controller maintains P(X(t) in X_safe) above the required threshold under severe scenarios.

A third direction is standardization of reliability and uncertainty indicators. Different automation domains measure confidence, risk and degradation in different ways. A smart grid, a production line and an intelligent building have different physical variables, but they can share a common logic: normalize the indicators, transform them into reliability and uncertainty terms, embed them in the Hamiltonian, and evaluate whether local decisions improve global safety. Such a common logic would make predictive Pontryagin control transferable across automation applications.

## CONCLUSION

In this study, it is developed a predictive Pontryagin optimal control methodology for the class of nonlinear fully distributed CPS/DCS networks subject to reliability and uncertainty. This approach combines the use of prediction, probabilistic risk state model, reliability and Hamiltonian optimal control theory. The fundamental principle behind this approach is that any node in the distributed automation system should not only control its physical quantity but also influence transition of risk states, maintain reliability, and minimize uncertainty propagation. The result is a conceptually coherent and simulation-ready method for automation domains such as smart manufacturing, smart grids, intelligent buildings and smart urban infrastructures. Future research should focus on numerical implementation, real-time digital twins, stochastic validation and comparison with MPC and reinforcement-learning controllers.

## REFERENCES


1. Åström, K. J., & Murray, R. M. (2008). Feedback systems: An introduction for scientists and engineers. Princeton University Press.
2. Bertsekas, D. P. (2017). Dynamic programming and optimal control (4th ed.). Athena Scientific.
3. Bryson, A. E., Jr., & Ho, Y.-C. (1975). Applied optimal control: Optimization, estimation, and control. Hemisphere Publishing.
4. Fleming, W. H., & Rishel, R. W. (1975). Deterministic and stochastic optimal control. Springer.
5. Griffor, E. R., Greer, C., Wollman, D. A., & Burns, M. J. (2017). Framework for cyber-physical systems: Volume 1, overview (NIST Special Publication 1500-201). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.1500-201
6. Ilieva, N., & Metodiev, V. (2020). Computer sensor networks application for air quality events real time identification. *International Scientific Journal "Industry 4.0,"* **5**(4), 151–155.
7. Khalil, H. K. (2002). Nonlinear systems (3rd ed.). Prentice Hall.
8. Kirk, D. E. (2004). Optimal control theory: An introduction. Dover Publications.
9. Lee, E. A. (2008). Cyber physical systems: Design challenges. In Proceedings of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (pp. 363-369). IEEE. https://doi.org/10.1109/ISORC.2008.25
10. Lee, E. A., & Seshia, S. A. (2017). Introduction to embedded systems: A cyber-physical systems approach (2nd ed.). MIT Press.
11. Øksendal, B. (2003). Stochastic differential equations: An introduction with applications (6th ed.). Springer. https://doi.org/10.1007/978-3-642-14394-6
12. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., & Mishchenko, E. F. (1962). The mathematical theory of optimal processes. Interscience Publishers.
13. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber-physical systems: The next computing revolution. In Proceedings of the 47th Design Automation Conference (pp. 731-736). ACM. https://doi.org/10.1145/1837274.1837461

ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-50, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

14. Rausand, M., & Høyland, A. (2004). System reliability theory: Models, statistical methods, and applications (2nd ed.). Wiley.
15. Rawlings, J. B., Mayne, D. Q., & Diehl, M. M. (2017). Model predictive control: Theory, computation, and design (2nd ed.). Nob Hill Publishing.
16. Vasilev, P., & Metodiev, V. (2021). I4.0 component models based on IEC/EN 62264. *International Scientific Journal "Industry 4.0,"* **6**(5), 181–184.
17. Vyatkin, V. (2011). IEC 61499 as enabler of distributed and intelligent automation: State-of-the-art review. IEEE Transactions on Industrial Informatics, 7(4), 768-781. https://doi.org/10.1109/TII.2011.2166785
18. Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175. https://doi.org/10.1016/S0925-2312(01)00702-0