International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

A Stochastic Framework for Fully Distributed Control Systems and CPS: From Local State Transitions to Global Uncertainty Propagation

Ass. Eng. Iliyan Vasilev, Phd
University of Chemical Technology and Metallurgy, Sofia

ABSTRACT: The transition from hierarchical automation toward fully distributed Distributed Control Systems (DCS) and Cyber-Physical Systems (CPS) creates a new class of engineering problems in which local intelligence, networked coordination and physical dynamics must operate under uncertainty. In these systems, control is no longer concentrated in a single supervisory unit. Instead, sensors, controllers, actuators, edge devices and cyber agents cooperate through local decisions and partial information. This article develops a stochastic framework for examining fully distributed DCS/CPS by linking three levels of analysis: how local states shift through Markov transitions, how short-term decisions accumulate over time through the Chapman–Kolmogorov relation, and how uncertainty spreads in continuous processes through the Fokker–Planck equation. All these aspects indicate that a distributed system is something much more than a mere configuration of communication; it is an adaptive, stochastic controller organism, where its global functioning arises from many small local decisions that modify probabilities and paths. The design should then consider how these local decisions affect one another over time, rather than simply interconnecting devices. The proposed framework is then analyzed from the perspectives of stability, resilience, communication delay, cyber-security, scalability, energy efficiency and digital-twin-based prediction. The result is a theoretical foundation suitable for smart factories, smart grids, intelligent buildings, autonomous transport systems and future smart urban infrastructures. The added interpretative value of the framework is that each equation is treated not only as a formal mathematical relation, but also as a design logic. Markov probabilities are interpreted as local operational tendencies, Chapman-Kolmogorov composition as the logic of accumulated distributed decisions, and Fokker-Planck dynamics as the evolution of confidence, risk and uncertainty in the whole cyber-physical network.

KEYWORDS: cyber-physical systems, distributed control systems, Markov transitions, Chapman–Kolmogorov equation, Fokker–Planck equation, stochastic control.

INTRODUCTION

Distributed Control Systems and Cyber-Physical Systems are moving toward a new architectural stage. Earlier industrial control systems were usually designed around hierarchical layers: field devices, programmable logic controllers, supervisory control, manufacturing execution systems and enterprise planning layers. This architecture remains useful, but it becomes insufficient when the physical infrastructure is large, mobile, heterogeneous and continuously reconfigurable. Smart factories, smart grids, intelligent transport systems and smart campuses contain thousands or millions of interacting devices. Their behavior depends on local sensing, communication, computation, actuation and feedback. In such environments, centralized control alone creates bottlenecks, single points of failure and unacceptable latency. The emerging alternative is a fully distributed or totally distributed control architecture. In this architecture, every relevant node can observe part of the environment, exchange information with neighboring nodes, estimate its own state, participate in decision-making and influence the physical process through local control actions. The global behavior is not imposed by a single controller. It emerges from the composition of many local decisions. This shift creates new possibilities for scalability, resilience, flexibility and autonomy, but it also creates new analytical difficulties.

A useful way to strengthen the theoretical foundation of fully distributed CPS/DCS is to position them within the broader evolution of cyber-physical systems research. Cyber-Physical Systems are typically defined as systems combining computational, communications, and physical processes with feedback from cyber to physical and vice versa, resulting in new constraints on timing, reliability, security, and controller design (Lee, 2008; Rajkumar et al., 2010). However, in industrial settings based on networked automation, intelligence is not confined anymore to centralized control but rather distributed over controllers, intelligent devices, edge computing nodes, and autonomous agents (Ge et al., 2017; Mangharam & Pajic, 2013). These considerations are also confirmed by the CPS framework literature, which highlights timing, trustworthiness, uncertainty of information, resilience, and inter-

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

operability among different CPS domains as key issues for CPS design (Griffor et al., 2017; Wollman et al., 2017). From the point of view of automation, it is also supported by the standards of IEC 61499 modeling approach and function block architecture, which facilitates the distribution of control functions on several hardware/software entities (Monroy Cruz et al., 2023; Tkáčik et al., 2024; Vyatkin, 2011).

Formal Stability and Resilience Conditions in Fully Distributed CPS/DCS

Fully distributed CPS/DCS works in uncertain conditions both on the levels of sensing, communication, computation, and actuation. For that reason, any qualitative terms like "stability of behavior" or "robustness of operation" become insufficient. Any formal approach requires the definition of stability and resilience under uncertainty propagation and accumulation of the local decision-making. Three important aspects are necessary in this situation: stochastic stability of operational modes, mean-square stability of continuous dynamics, and probabilistic resilience of the distributed structure.

(a) Stochastic stability of operational modes.

Here Pstands for a matrix defining transition between operational modes (Normal, Warning, Overload, Fault, and Recovery). System is called stochastically stable when distribution $\pi_t$converges to stationary π^*:

$$\lim_{t\to\infty} \pi_t = \pi^*$$

with major probability being assigned to acceptable modes in the limit distribution. Thus, stochastic stability represents the limit properties of distributed architecture and demonstrates the effect of local control, communication delays or maintenance procedures upon its operation mode.

(b) Mean-square stability of continuous dynamics.

For continuous dynamical variables described by a stochastic differential equation

$$dx(t) = \mu(x,t), dt + \sigma(x,t), dW(t)$$

mean-square stability means that there exist positive constant $C$and λsuch that

$$\mathbb{E}[|x(t)|^2] \le Ce^{-\lambda t}|x(0)|^2.$$

In a distributed environment, these conditions must be satisfied not separately for each node but for the overall network dynamics due to uncertainty propagation through communication or physical processes.

(c) Probabilistic resilience of the distributed architecture.

Resilience means the ability of the system to return back to its normal operation mode (i.e. safe region X_{safe}) after the disturbance with guaranteed speed

$$P(X_t \in X_{\text{safe}}) \to 1 \quad \text{as } t \to t_r.$$

The above formulation reflects the practical requirements to the system and means that even with malfunctioning nodes or sensing errors the system recovers successfully.

(d) Control-induced shaping of transition probabilities.

Unlike global control strategies, distributed control affects stability via changing transition probabilities locally. Let $\Delta p_{ab}^{(i)}$represent the difference between probabilities of transitions from state $a$to state $b$under the effect of the control strategy ($p_{ab}^{\text{controlled}}$) and without it ($p_{ab}^{\text{uncontrolled}}$). Then resilient controller must satisfy the following inequality:

- $\Delta p_{ab}^{(i)} < 0$ for all transitions considered as unsafe (e.g., Warning → Fault),
- $\Delta p_{ab}^{(i)} > 0$ for recovery transitions (e. g., Fault → Recovery).

The main difficulty with a fully distributed CPS/DCS is that it cannot be treated simply as a collection of devices linked together. In practice, it behaves as a stochastic dynamical system. Its state shifts over time, individual components may fail and later recover, communication can slow down or be disrupted, and physical measurements are always affected by noise. Because of this, traditional deterministic block diagrams do not capture the real behaviour of such systems. A useful framework has to show how local states evolve, how short actions accumulate into long-term behaviour, and how uncertainty moves through the network. The article approaches this problem by bringing together three mathematical ideas. Markov transitions describe how a component is likely to move from one operational state to another. The Chapman–Kolmogorov relation shows how these short-term transitions combine across time. The Fokker–Planck equation captures how uncertainty spreads in continuous variables. Taken together, these three elements form a coherent way to describe and analyse fully distributed DCS/CPS.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

The central idea is that mathematical interpretation must connect three questions that engineers face in practice. ***First, what is the local tendency of a component under its current conditions? Second, what happens when many such local tendencies are composed over time? Third, how much uncertainty remains around the predicted system trajectory?*** A fully distributed system is trustworthy only when these three questions are answered consistently.

This view is consistent with contemporary CPS frameworks that treat CPS as engineered interacting networks of physical and computational components and emphasize cross-cutting aspects such as timing, trustworthiness, security, resilience, data and functional uncertainty. In this article, these aspects are reorganized into a stochastic control perspective in which distributed intelligence is measured by the ability of the system to maintain useful global behavior under local uncertainty.

From Distributed Control to Fully Distributed Cyber-Physical Control

In a traditional DCS, the distribution of tasks usually means that input/output handling and control loops are spread across several controllers, but the overall structure still follows a hierarchy. A fully distributed CPS/DCS changes this picture. It removes the need for a single point where all decisions converge. Instead, each node takes on a degree of autonomy, acting as a small decision-making unit. These nodes coordinate through direct communication with their neighbors, through consensus routines, event-triggered updates, distributed optimization, and increasingly through local edge-level intelligence.

Let the system be represented as a graph:

$$G = (V, E), V = 1,2,\dots,N$$

where V is the set of cyber-physical nodes and E is the set of communication, informational or physical coupling links. Each node i has a local state $x_i(t)$, a local control action $u_i(t)$, and a set of neighbors N_i. The global system state is therefore:

$$X(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T.$$

A centralized controller would generate a global action from the entire system state, u(t) = K(X(t)). In a fully distributed system, the control law is local:

$$u_i(t) = K_i\left(x_i(t), x_{Ni}(t), m_i(t), r_i(t)\right)$$

where $x_{Ni}(t)$ denotes the states or estimates received from neighboring nodes, $m_i(t)$ denotes messages and $r_i(t)$ denotes the local objective or reference signal. The global control vector is the composition of all local controls:

$$U(t) = [u_1(t), u_2(t), \dots, u_N(t)]^T.$$

The central research challenge is to determine when these local rules produce stable, resilient and efficient global behavior. This challenge cannot be solved by topology alone. It requires a mathematical theory of stochastic state evolution, because each local decision is made under uncertainty and each local state may influence other states through delayed, noisy or incomplete information (Table 1.).

Table 1. Analytical layers of the proposed framework

Layer	Mathematical object	Main question	Engineering meaning
Local state evolution	$p_{ij} = P(X_{t+1} = j \mid X_t = i)$	What is the next state of a component?	Transitions among normal, warning, overload, fault and recovery modes.
Temporal composition	$P_{ij}(t+s) = \Sigma_k P_{ik}(t)P_{kj}(s)$	How do short decisions combine over time?	Long-term behavior generated by sequences of local actions.
Uncertainty propagation	$\partial p/\partial t = -\partial(\mu p)/\partial x + \frac{1}{2}\partial^2(\sigma^2 p)/\partial x^2$	How does uncertainty evolve?	Noise, delay and disturbance form a probability field.

Markov Transitions as Local Stochastic Causality

The first mathematical level is the Markov transition. In a discrete-state system, the state at time t is denoted by X_t. If the set of possible states is S, a transition from state i to state j is described by:

$$p_{ij} = P(X_{t+1} = j | X_t = i)$$

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

In engineering terms, this is not merely a probability. It is a compact representation of local cyber-physical causality. For example, if i represents normal operation and j represents overload, then p_{ij} is the probability that the system will move from normal operation to overload during the next time step. This probability may depend on physical load, machine wear, environmental disturbance, local controller behavior, communication latency and data quality.

For a single node i, the local transition matrix can be written as:

$$P^{(i)} = [p^{(i)}{}_a b], p^{(i)}{}_a b = P(x_i(t+1) = b | x_i(t) = a).$$

In a fully distributed CPS/DCS, however, a node rarely evolves independently. The transition probability of node i is influenced by neighboring states and by messages arriving through the communication network. Therefore, the local transition probability becomes conditional on local and neighboring information:

$$p^{(i)}{}_a b(t) = P\left(x_i(t+1) = b | x_i(t) = a, x_j(t), j \in N_i, m_i(t)\right).$$

This equation is important because it shows that distributed control transforms the Markov matrix from a fixed object into an adaptive object. The probabilities are not static; they are shaped by load balancing, local control policies, cyber-security mechanisms, maintenance strategies and reconfiguration after faults.

If we assume a smart-factory machine that can operate in four modes: OK, Warning, Overload and Fault. A local Markov model can be used to estimate how likely the machine is to move from one mode to another. If one of the neighboring machines breaks down, then the additional burden is likely to cause a change in probability, with events like warning-overload or overload-fault being more likely to occur. However, with a decentralized controller, the task may be moved around, or the speed may be lowered. In practical terms the controller is changing the transition probabilities: it lowers the chances of unsafe jumps and raises the odds of recovery. Viewed this way, the Markov model becomes a practical link between hands-on engineering judgment and formal control design—letting local actions be evaluated by how they reshape the system's probabilistic behaviour rather than by instantaneous measurements alone. It connects operational health, fault probability, resilience and control action in one mathematical object. In fully distributed systems, this local object is replicated across many nodes, and the global behavior emerges from their interaction.

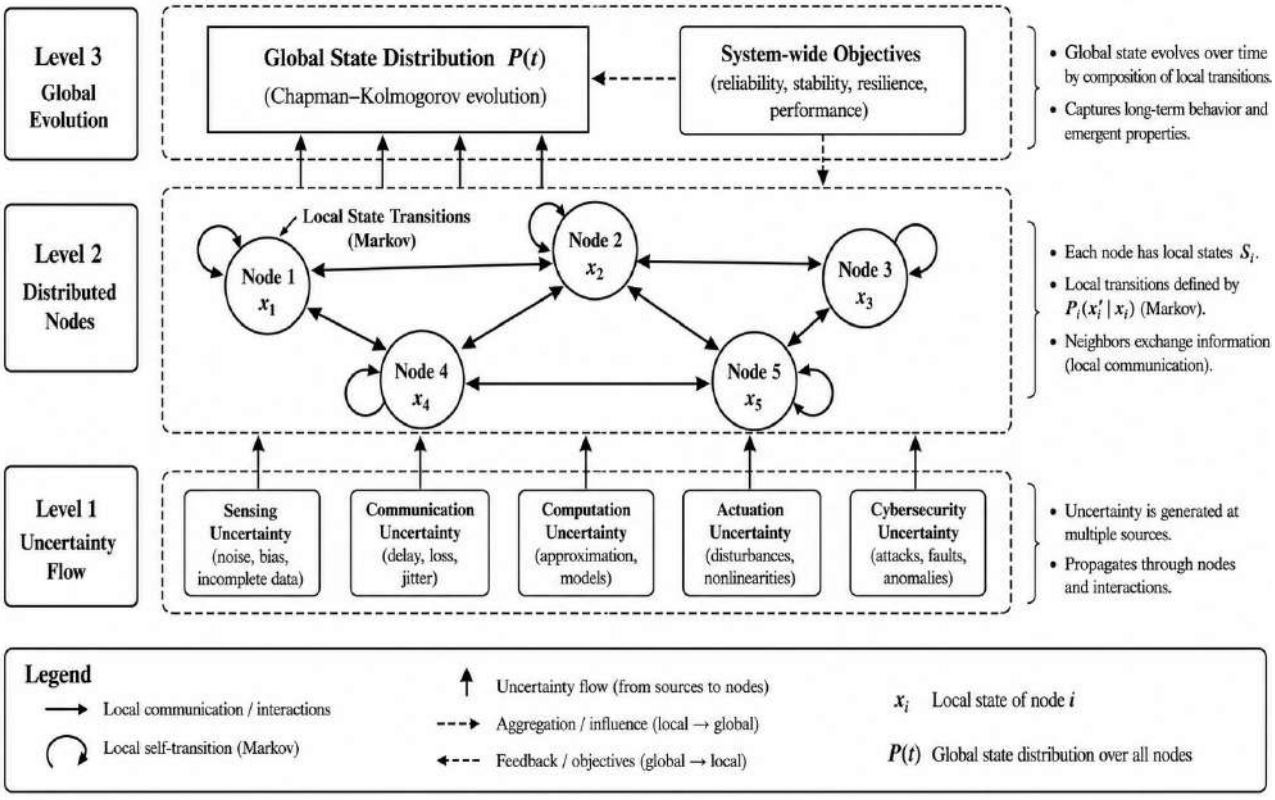

Figure 1. Integrated architectural model of continuous CPS reliability governance.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

Interpretative logic of local state transitions

The key value of a Markov transition is that it turns the informal language engineers use "the machine is getting unstable," "it's drifting toward overload," "it's close to failing" into something that can be measured. The transition matrix captures these tendencies as probabilities between states, giving a compact picture of how the system typically reacts to extra load, noise, delayed information or imperfect local control. **A second, equally important reading is the control reading**: the same probabilities describe what the controller is trying to achieve. In other words, transition entries are not only passive descriptors of system behaviour but also levers for intervention—good control reduces the probability of unsafe jumps (e.g., Warning→Overload) and increases the probability of desirable moves (e.g., Fault→Recovery). This perspective makes the matrix a concise tool for assessing performance, since minor variations in the matrix correspond to the effects of local policies, communication, and maintenance procedures, thereby allowing engineers to evaluate different controllers based on how they reduce the risk in the system, as opposed to just looking at instantaneous measures. They do not only describe what the system does on its own; they also show what the controller is trying to influence. A distributed controller is not simply keeping the machine in its current state—it is actively reshaping the transition structure. Good control lowers the likelihood of unsafe moves and raises the chances of recovery:

$$\Delta p^{(i)}{}_a b = p^{(i)}{}_a b, controlled - p^{(i)}{}_a b, uncontrolled.$$

For undesirable transitions, such as Normal to Fault or Warning to Fault, a successful controller should make $\Delta p^{(i)}$_ab negative. For desirable transitions, such as Fault to Recovery or Warning to Normal, it should make $\Delta p^{(i)}$_ab positive. This gives a logical performance criterion for distributed control without requiring every node to know the complete global state.

A third way to read the transition structure concerns the role of each individual node. In a fully distributed architecture, no component has a full picture of the entire system, and none is expected to. Each node can influence only a small part of the overall dynamics, and its responsibility is to adjust the probabilities it can directly affect. When many nodes do this in parallel, the global behaviour that emerges is a result of countless small, local corrections. This perspective links stochastic modelling with engineering practice: every controller becomes a modest probability-shaping element within the larger CPS/DCS.

Evolution of the Global State Distribution

While a Markov transition captures how a single component moves from one state to another in one step, a distributed CPS is often better understood by looking at how operational states are distributed across the entire network. Let π_t denote a row vector that represents the probability distribution of a component, a subsystem, or the full architecture over M possible operational modes:

$$\pi_t = [\pi_t(1), \pi_t(2), \dots, \pi_t(M)].$$

If *P*is the transition matrix, the distribution evolves according to

$$\pi_{t+1} = \pi_t P.$$

This relation describes the **system-level operating regime** rather than the condition of any single device. Instead of asking whether one machine is overloaded or unstable, the engineer can evaluate what fraction of the entire CPS is likely to operate in Normal, Warning, Overload, or Fault modes at the next time step. This perspective is essential in environments such as smart factories, microgrids, intelligent buildings, and distributed transport systems, where global performance emerges from the interaction of many autonomous nodes.

A long-term distribution, when it exists, satisfies

$$\pi^* = \pi^* P.$$

The stationary distribution provides a **statistical picture of the system's long-run operating tendencies**. In a resilient distributed architecture, the stationary probability of acceptable modes should dominate the probability of critical or unstable modes. Any change in control policies, maintenance strategies, communication delays, or network topology alters the transition matrix and therefore shifts π^*. For this reason, π^*can be interpreted as a **macroscopic indicator of how well the distributed control logic performs over extended horizons**.

Interpretation of System-Level Operating Regimes

Consider this vector as the overall health indicator of the system. An unusual sensing device or one faulty machine may be just a minor anomaly, but the transition of probability mass from *Normal* to *Warning* on π_t is the sign of trouble for the entire system. If mass piles up in *Overload* or *Fault*, the network is no longer experiencing isolated glitches — interactions and constraints at the system level are beginning to dominate. This view matters because distributed systems rarely fail all at once. Degradation usually

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

shows up first as a slow redistribution of modes long before any dramatic cascade. This gives us the significance of $\pi_t a$ as an important early indicator: through the analysis of the way that it moves, it is possible for a controller to recognize a coming system vulnerability while everything seems to be alright in most of the nodes. In other words, by monitoring π_t, we are able to take preventive actions.

Chapman–Kolmogorov Composition as the Time Logic of Distributed Control

The second layer of the framework is the Chapman–Kolmogorov equation:

$$P_{ij}(t+s) = \Sigma_k P_{ik}(t) P_{kj}(s).$$

The Chapman–Kolmogorov relation states that the probability of moving from state i to state j over a longer interval $t+s$can be obtained by summing over all possible intermediate states k. In a distributed CPS/DCS, this is more than a formal identity. It captures the idea that long-term system behaviour is built from many short, local transitions.

A completely distributed system will not generally have to issue one universal command to the whole system. Rather, it will develop through a series of little changes, such as when a sensor detects a discrepancy, a controller tweaks an actuator, a neighbor adjusts its own calculations, another controller transfers some load, and finally the physical system reacts. The result that the system reaches at last is the total effect of these little actions. The Chapman-Kolmogorov equation captures this idea, in illustrating how many local decisions made with incomplete information lead to the overall evolution of the CPS.

If transition matrices vary over time, the global transition from time 0 to time T is:

$$P(0,T) = P(0)P(1)\dots P(T-1).$$

The state distribution then evolves as:

$$\pi_T = \pi_0 P(0)P(1)\dots P(T-1).$$

This formulation becomes particularly important in adaptive and event-triggered control, where the transition probabilities themselves shift whenever the system experiences a significant event—a sudden load increase, a detected cyber intrusion, a sensor malfunction, a new production command, or a change in the communication graph. The Chapman–Kolmogorov relation ensures that these updates remain time-consistent: whatever model is used for a long horizon must align with the behaviour implied by its shorter segments. From a practical standpoint, the equation helps determine whether local control actions still make sense once they accumulate over time. A distributed system may look stable when examined one step at a time yet become unstable when many such steps are composed. The opposite can also occur short-term deviations may be harmless if the long-term transition structure pulls probability mass back toward safe operating regions. For this reason, Chapman–Kolmogorov analysis is central to evaluating reliability, resilience, and the predictive accuracy of digital-twin models.

The proposed stochastic interpretation is also supported by previous work on stochastic hybrid and probability-density-based control. Distributed CPS can be viewed as hybrid systems because they combine continuous physical variables, discrete operational modes, network events, and control decisions that occur asynchronously across different nodes (Bujorianu, 2018). Markov transitions can thus prove useful for modeling discrete switches between modes, whereas the method of manipulating the probability density function gives us a more general way of controlling the distribution of the system states than controlling just a single state trajectory (Ren et al., 2019). It is in this vein that the Fokker–Planck equation proves helpful not only as a model of diffusion but also as an analytical instrument for monitoring the effects of uncertainty, disturbances, and noise on the evolution of probability density in the CPS (Annunziato & Borzì, 2018). All this helps support our thesis that in completely distributed CPS/DCS, a good performance of local controllers should include their effectiveness in controlling transition probabilities and probability mass.

Interpretation of temporal coherence

The Chapman-Kolmogorov equation provides a simple and pragmatic way to test temporal consistency. after being performed by numerous nodes over many iterations each activity that is rational on an individual level may result in the system moving to an undesired state. Take, for instance, the shedding of load from any node at increased levels of risk as seen from the perspective of the particular node alone; while the initial decision seems reasonable, its consequence results in an overload elsewhere, perhaps resulting in oscillation or the emergence of another critical point. The problem with the Chapman-Kolmogorov equation can therefore be identified easily enough – ***does the sensible local rule work when one looks at its total effect?*** Engineering-wise, state k is relevant as well. It is not only the variable that acts as the summation index, but it could be a point for action. If the system tends

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

to move from Normal to Fault through Warning, then Warning is the state where control must act. A totally distributed control architecture should identify such intermediate states and increase the probability of returning from them to safe operation.
This suggests that there is a design rule to follow, which states that distributed control must be measured not just on a one-step basis but also based on:

$$E[Risk(X_{t+s})|X_t] < E[Risk_0(X_{t+s})|X_t], s > 1$$

$Risk_0$ refers to the risk associated with the uncontrolled or poorly controlled process. This equation represents the necessity of having reduced future accumulative risk rather than just improving the immediate situation.

Fokker–Planck Dynamics as the Geometry of Uncertainty

The third layer concerns continuous stochastic dynamics. Many CPS variables are continuous: voltage, frequency, temperature, pressure, position, velocity, vibration, energy consumption and flow. In such cases, a stochastic differential equation can represent the local process:

$$dx(t) = \mu(x,t)dt + \sigma(x,t)dW(t),$$

where μ(x,t) is the drift term, σ(x,t) is the diffusion coefficient and W(t) is a Wiener process. The drift describes the directed tendency of the system, usually generated by physical laws and control actions. The diffusion describes noise, disturbance and uncertainty. The probability density p(x,t) of the state evolves according to the Fokker–Planck equation:

$$\partial p(x,t)/\partial t = -\partial[\mu(x,t)p(x,t)]/\partial x + ½\partial^2[\sigma^2(x,t)p(x,t)]/\partial x^2.$$

This equation changes the interpretation of the system. Instead of representing the CPS as a single trajectory, it represents the CPS as a moving probability cloud. The cloud may become narrower if feedback reduces uncertainty, or wider if noise, attacks, delay or poorly coordinated control increase uncertainty. The Fokker–Planck equation therefore describes not only the state of the system, but the state of knowledge about the system.
In distributed CPS/DCS, the multidimensional version is more appropriate. For the global state X(t), let:

$$dX(t) = F(X,t)dt + G(X,t)dW(t).$$

The associated probability density $p(X,t)$evolves as:

$$\partial p(X,t)/\partial t = -\nabla \cdot [F(X,t)p(X,t)] + ½\nabla\nabla : [D(X,t)p(X,t)], D = GG^{\mathrm{T}}.$$

F(X,t)stands for the distributed drift resulting from local controllers and the actual interaction dynamics among agents. On the other hand, the diffusion matrix D(X,t)describes how uncertainty is organized within the network. If the disturbance influences the network separately, then D is nearly a diagonal matrix. When disturbances are correlated—whether through shared communication links, mechanical coupling, or electrical interconnections—off-diagonal terms appear. In this way, the diffusion matrix becomes a compact mathematical picture of how uncertainty is linked and transmitted across the network. The Fokker–Planck equation is particularly valuable for digital twins. A deterministic digital twin predicts one possible future. A stochastic digital twin predicts a distribution of possible futures. This allows the system to estimate the probability of overload, the probability of fault propagation, the expected time to recovery and the confidence level of a control decision.

Interpretation of drift, diffusion and safe probability mass

The Fokker-Planck equation can be interpreted as a competition between concentration and dispersion. The drift term expresses the system tendency created by control, physical laws and coupling. It moves the probability density toward desired regions of the state space. The diffusion term expresses the loss of precision caused by noise, uncertainty and disturbances. It spreads the density and increases the probability that the system may enter unsafe regions.
For engineering design, the goal is not to remove uncertainty completely, because this is impossible in real CPS/DCS environments. The goal is to keep sufficient probability mass inside the safe operational region X_{safe}. This can be expressed as:

$$P_{safe}(t) = \int_{X_safe} p(X,t)dX.$$

A good distributed (or a robust) CPS must ensure that $Psafe(t)$remains above the threshold despite any noise, delay, or failure in some local nodes. In other words, the network should still be "mostly working" despite any problems from its components. This view accepts that single sensors or machines will sometimes misbehave, but the whole system must keep most of its probability mass inside safe limits. Also, uncertainty isn't just random fuzz: **the diffusion matrix Drecords how disturbances travel between parts of the system**—off-diagonal entries show when two machines tend to vary together (for example, two machines on the same conveyor or two microgrids linked by a weak line). The practical goal is simple: design local controllers and the communication

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

between them so the system stays safe even when sensors wobble, messages lag, or parts fail. Focus on keeping the **probability of safe operation high**, and pay special attention to how disturbances travel—identify the components that spread risk and protect or decouple them so local problems don't become systemwide failures.

Fokker–Planck analysis therefore helps reveal not only the magnitude of uncertainty but also its pathways—how it spreads, where it concentrates, and which parts of the network are most tightly coupled.

An Integrated Model for Fully Distributed DCS/CPS

The three mathematical layers can be combined into one hybrid stochastic model. At the node level, a cyber-physical component may be described by:

$$dx_i(t) = f_i\big(x_i(t), u_i(t), x_{Ni}(t)\big)dt + g_i\big(x_i(t)\big)dW_i(t).$$

The distributed control action is:

$$u_i(t) = K_i\big(x_i(t), \hat{x}_{Ni}(t), r_i(t), q_i(t)\big),$$

where $\hat{x}_{Ni}(t)$ denotes estimated neighbor states and $q_i(t)$ represents quality indicators such as communication confidence, sensor reliability or cyber-security status. This extension is important because modern CPS nodes should not react only to measured values; they should also react to the credibility of the information they receive.

At the discrete operational level, each node has a mode variable $z_i(t)$ belonging to a finite set such as Normal, Warning, Overload, Fault and Recovery. The mode evolves according to a controlled Markov transition:

$$P(z_i(t+1) = b | z_i(t) = a) = p^{(i)}{}_a b\big(u_i(t), x_{Ni}(t), q_i(t)\big).$$

Thus, continuous dynamics and discrete operational modes coexist. The continuous model describes physical evolution, while the Markov model describes operational health. The Fokker–Planck equation predicts the evolution of uncertainty in continuous states, while Chapman–Kolmogorov composition predicts long-term mode probabilities. This hybrid formulation is suitable for cyber-physical production lines, smart microgrids, autonomous traffic networks and distributed energy systems.

Interpretative synthesis of the three layers

The integrated model as a complete decision cycle. Aspects of an integrated model as a whole cycle of decision-making. While the continuous state is a representation of the physical state trajectory, the Markov mode captures its classification by operations. The distributed control uses the information provided by the former and probabilistic transitions between latter. As such, there is a loop of influences among all elements: control impacts the continuous state, the change in continuous state impacts mode transitions, and modified mode transitions impact future local decision-making. Therefore, one must note the importance of designing controllers for impacting not only the continuous state but also possible probabilistic behavior. Over time, lots of local decisions occur, while Chapman-Kolmogorov is responsible for aggregation of these decisions, whereas Fokker-Plank is responsible for propagation of uncertainty associated with the entire process of operations and physical processes. This demonstrates the strong importance of maintaining a close relationship between the logical action and physical process since delay in message influences the decision on controlling, changed command alters the physical process like temperature, vibration, or flow, and change in state influences mode transitions.

[1] A clear trend is that "distributed" increasingly means distributed intelligence rather than just distributed hardware. Where older DCS designs spread controllers across a plant, modern CPS push decision-making to the edge: embedded AI, local digital twins and smarter controllers let nodes estimate, decide and adapt without waiting for a central supervisor. That reduces latency and improves resilience, but it also raises a new requirement—mathematical guarantees that local autonomy will not, by itself, create global instability.

[2] The second trend is the movement toward platform-independent distributed automation models. Standards and design approaches associated with function blocks and distributed application mapping, such as IEC 61499-based modeling, support the decomposition of control logic across multiple hardware components. This trend strengthens the need for formal models that can describe how many small local controllers form one coherent cyber-physical system.

[3] The third trend is uncertainty-aware control. Future CPS will not only compute the most likely state; they will compute distributions, confidence intervals and risk levels. This trend is visible in probability-density-function control, stochastic model predictive control and digital twins that estimate possible futures rather than a single trajectory. The Fokker–Planck equation is central to this trend because it directly models the evolution of probability density.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

[4] The fourth trend is security-control integration. Cyber-security is no longer a separate layer added after control design. In fully distributed CPS, attacks alter measurements, delays, actuator commands and transition probabilities. A false-data injection attack, for example, can transform the measured state from $x_i(t)$ to $x_i(t)+a_i(t)$, causing the local controller to compute a wrong action. A denial-of-service attack may remove messages from the neighbor set and effectively change the communication graph. Therefore, security must be modeled as part of the system dynamics.

[5] The fifth trend is resilience-by-design. Fully distributed systems are expected to continue operating after partial failures. This does not mean that every component remains functional; it means that the system redistributes functions, isolates faults and recovers acceptable performance. Stochastic transition models are useful here because they allow the engineer to distinguish between failure probability, recovery probability and cascading risk.

[6] The sixth trend is sustainability-aware distributed control. Smart factories and smart grids must optimize not only productivity and stability, but also energy consumption, peak demand, thermal load and carbon footprint. A distributed control law can include energy terms in its local objective, but the global effect depends on the interaction of all local decisions. This makes the stochastic composition of local actions an important tool for sustainable CPS design.

Think of the integrated model as a closed decision loop. The continuous state records the physical evolution. The Markov mode groups that evolution into operational categories. Distributed controllers act on both the trajectory and the switching probabilities. Delays and uncertainty change control actions, which in turn change physics and mode-switching likelihoods. And the transition matrix along with πt become succinct and action-oriented representations of where risks are being concentrated. The Chapman-Kolmogorov theorem emphasizes the fact that repeated isolated decisions made over time lead to global implications, and it is when all those individually rational decisions are harmful as a whole. This means that control systems need to account for both uncertainties and delays (controllers must be uncertainty- and delay-aware), communication systems must have a richer semantic meaning in terms of confidence and potential risks in the future, and designing should focus on modifying probabilities locally.

Multi-Aspect Analysis of Fully Distributed Systems

A fully distributed DCS/CPS should be analyzed across several dimensions. The NIST CPS framework emphasizes cross-cutting aspects such as function, timing, data, trustworthiness and uncertainty. In the present article, these concerns are translated into stochastic control indicators. The purpose is not to replace system engineering analysis, but to provide a mathematical layer that supports it (Table 2.).

Table 2. Multi-aspect analytical indicators for fully distributed DCS/CPS

Aspect	Possible mathematical indicator	Interpretation
Stability	$E[V(X_{t+1})-V(X_t)\|X_t] \le -\alpha V(X_t)$	Expected decrease of a Lyapunov function under local distributed actions.
Resilience	$R(t)=1-N\_f(t)/N$	Fraction of the network that remains non-failed or operationally acceptable.
Communication delay	$u_i(t)=K_i(x_i(t),m_{ji}(t-\tau_{ji}))$	Local actions depend on delayed neighbor information.
Cyber-security	$\tilde{x}_i(t)=x_i(t)+a_i(t)$	Attack signals distort measurements and modify control decisions.
Uncertainty	$\partial p/\partial t = -\nabla\cdot(Fp)+½\nabla\nabla:(Dp)$	The probability density spreads, contracts or shifts under stochastic dynamics.
Scalability	Complexity per node $\approx O(\|N_i\|)$	Local computation should depend mainly on neighbor degree, not global system size.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

Stability definition for a decentralized CPS in terms of stochastic stability requires checking whether local rules can drive the network to a desired operating state. Let us denote by V(X) a non-negative scalar which represents the system's energy, error or risk. One useful condition is that the expected value of V decreases within one time step:

$$\mathbb{E}[V(X_{t+1}) - V(X_t) \mid X_t] \leq -\alpha V(X_t), \alpha > 0.$$

In continuous time the stochastic generator gives the instantaneous expected change of V:

$$\mathcal{L}V(X) = \nabla V(X)^T F(X) + \frac{1}{2}\mathrm{Tr}\ ,(G(X)^T \nabla^2 V(X) G(X)).$$

If $\mathcal{L}V(X) \leq -cV(X)$ for some $c > 0$ outside a neighbourhood of the equilibrium, then the distributed system is stable in a probabilistic sense: despite noise, delays and local autonomy, the global tendency is to reduce deviation or risk over time.

Resilience addresses what happens after a disturbance. Let $N_f(t)$ denote the number of failed nodes among N total nodes and define a simple resilience index

$$R(t) = 1 - \frac{N_f(t)}{N}.$$

An increase in $\mathbb{E}[R(t)]$ following a disruption indicates recovery. From a Markov perspective, resilience requires nonzero and sufficiently large transition probabilities from Fault to Recovery and from Recovery to Normal, together with a coupling topology that prevents neighbouring failures from triggering uncontrolled cascades. Thus, resilience is a property of both the transition structure and the network interconnections.

Communication delays alter the information available to local controllers and therefore change effective decision rules. If a message from node j to node i is delayed by τ_{ji}, a typical control law can be written as

$$u_i(t) = K_i(x_i(t), m_{ji}(t - \tau_{ji}), j \in \mathcal{N}_i).$$

Small delays allow near-real-time coordination, but larger delays force controllers to act on stale information and raise the chance of unsafe transitions. We can make latency an explicit part of safety analysis by using a local risk function that converts delay and information quality into higher transition intensities. In other words, when messages arrive late or with low confidence, the model should reflect an increased probability of moving toward Warning or Fault states. It follows from the above discussion that the following three practical recommendations may be suggested for design: choose control algorithms and communication protocols such that a Lyapunov-type reduction is obtained with respect to the given stochastic process despite the presence of noise, delays, and partial observability, thus giving an actual stability condition. Organize transition probabilities and network interaction such that recovery actions will be guaranteed to succeed with high probability and without any possibility of catastrophic cascade failures. Ensure that the controllers are aware of the influence of the delay on their performance and utilize the information about delay properly when constructing control algorithms and transition probabilities. It should be noted now that the concepts of stability, resilience, and delay are interconnected in one language: the language of expected Lyapunov reduction, resilience measures, and delayed transition probabilities.

$$Risk_i(t) = \alpha\tau_i(t) + \beta{\sigma_i}^2(t) + \gamma L_i(t) + \eta A_i(t),$$

where τ_i is delay, σ_i^2 is local noise intensity, L_i is load and A_i is an attack or anomaly indicator. When $Risk_i(t)$ exceeds a threshold, the node should enter a protective mode, reduce load, increase diagnostic sampling or request neighbor support.

The logical relationship between these aspects is recursive. The instability of the system decreases its capability to recover since the system with increased deviations is more incapable of making efficient corrective changes; communication time delays complicate distinguishing real information from any replay or manipulation attempts; and excessive energy saving could lead to the exhaustion of resources and decreased safety margins. Hence, the study of various factors cannot be viewed as an independent process. On the contrary, the entire process should be approached from the perspective of coupling, where improving the efficiency of one factor can pose a risk for another. The implication is that all the mentioned factors should be addressed simultaneously because when making decisions about autonomy and coordination, one has to bear in mind the impact of one factor on another. Thus, one may approach all these factors as constrains of a single decision-making process. This makes fully distributed DCS/CPS a multi-objective stochastic control problem rather than a conventional automation problem.

Smart Factory Example

Consider a smart factory with N machines. Each machine has four operational modes: OK, Warning, Overload and Fault. Let the distribution of machine states at time t are:

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

$$\pi_t = [\pi_O K, \pi_W arning, \pi_O verload, \pi_F ault].$$

A possible transition matrix without advanced distributed control is:

P = [[0.90, 0.07, 0.02, 0.01],
[0.30, 0.50, 0.15, 0.05],
[0.10, 0.30, 0.40, 0.20],
[0.15, 0.25, 0.10, 0.50]].

If the current distribution is π_0 = [0.80, 0.10, 0.07, 0.03], the next distribution is $\pi_1 = \pi_0 P$. This gives an expected picture of factory health after one time step. If distributed control is introduced, local controllers can reduce unsafe transitions by load sharing, predictive maintenance or local speed reduction. Mathematically, this means that p_OK,Overload, p_Warning,Fault and p_Overload,Fault should decrease, while p_Warning,OK and p_Fault,Recovery should increase.

The same example can be extended to continuous variables. Let $x_i(t)$ be the vibration amplitude, temperature or load of machine i. A stochastic model is:

$$dx_i(t) = f_i\big(x_i(t), u_i(t), x_{Ni}(t)\big)dt + g_i\big(x_i(t)\big)dW_i(t).$$

If the diffusion term increases because of sensor noise or mechanical degradation, the probability density of $x_i(t)$ widens. The digital twin can then estimate whether the probability of crossing a critical threshold is increasing. A maintenance action is not triggered only when the fault occurs, but when the probability mass near the fault region becomes too large.

In essence, the Markov and the Fokker-Planck models complement each other in such a way that Markov transitions represent the states in which the model operates and their likelihoods, while Fokker-Planck represents the process of evolution leading to boundary conditions set out by the transitions. Many transient Markovian steps combined together by means of Chapman-Kolmogorov give us a picture of how local uncertainties on individual machines are accumulated and affect the long-term factory reliability. What does it mean? Increasing probability of a transition to the state marked as Warning means loss of resilience rather than imminent failure, and if at the same time Fokker–Planck density gets broader, the uncertainty increases, resulting in higher probability of a transition beyond a particular boundary. The correct reaction would be precautionary – reduce load, rebalance tasks, check sensors or increase monitoring rate before transitioning to the Fault state. To implement a distributed intelligent factory, we need to learn how to manage probability distributions – keep local density of states narrow within the safe interval without overloading neighbouring controllers.

Smart Grid Interpretation

In the smart grid the same principles appear in microgrids, distributed energy resources, batteries, electric vehicles and demand-response devices. The cyber layer—communication, computation and control—actively shapes physical flows in generation, transmission and distribution, and physical disturbances feed back into cyber decisions. This two-way coupling means that cyber actions change power flows and voltages, which in turn alter mode probabilities and trigger further control actions. Understanding and managing the grid therefore requires models that link discrete operational modes, continuous uncertainty, and the temporal composition of many local decisions.

A node may represent a microgrid, inverter, storage unit or controllable load. Its state may include voltage, frequency, state of charge, local demand and communication status. A local controller attempts to maintain constraints such as:

$$V_m in \le V_i(t) \le V_m ax, f_m in \le f_i(t) \le f_m ax.$$

At the same time, the node may participate in a distributed optimization problem:

$$min\Sigma_i[c_i u_i^2(t) + d_i(x_i(t) - x_i *)^2 + e_i E_i(t)],$$

subject to physical dynamics, communication constraints and safety bounds. Stochastic analysis is necessary because renewable generation, demand and communication availability are uncertain. A Markov model can describe discrete states such as normal, congested, islanded and blackout-risk. A Fokker–Planck model can describe the probability density of frequency deviation or voltage deviation. A digital twin can use both to predict whether the grid is moving toward a safe operating region or toward cascading instability. Control distributed within the smart grid needs to take into account two opposing requirements: independence from one side, synchronization from another. A microgrid is capable of controlling its operations related to battery and inverter operations independently. These operations affect the boundaries between adjacent networks. Discrete changes in modes – grid-connected, isolated, congested, faulty modes are accounted for by the Markov layer. Continuous changes in voltage and frequency, on the other hand, are modeled by the Fokker–Planck layer. A probabilistic approach to defining system stability emerges from this.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

Security and Trustworthiness in a Fully Distributed Architecture

Security is a structural concern in fully distributed CPS/DCS because attacks can alter the same variables used for control. A false-data injection attack can be represented as:

$\tilde{x}_i(t)=x_i(t)+a_i(t)$,

where $a_i(t)$ is an attack signal. The local controller computes $u_i(t)=K_i(\tilde{x}_i(t))$ instead of $K_i(x_i(t))$. Even if the controller is mathematically stable under correct data, the attacked system may become unstable because the perceived state differs from the true state.

According to Markov theory, a security threat represents a perturbation in the transition matrix,

$$P \mapsto \tilde{P}, \tilde{p}_{\text{Normal,Fault}} > p_{\text{Normal,Fault}},$$

so that the probability of reaching a faulty regime is higher. In contrast, considering the Fokker-Planck picture, a perturbation due to security threats may affect the continuous dynamical model: it can affect the drift coefficient, increase the effect of the diffusion coefficient, or generate multimo dality in the distribution of states. Denial of service can prevent messages from being delivered and therefore affect the structure of the graph G; similarly, replay attacks force the local controllers to take into account only an old value of the estimated state. In light of these considerations, security control cannot limit itself to just monitoring (it should be framed as the task of ***preserving safe probability distributions over time***, i.e., maintaining the joint mode probabilities and state densities within acceptable bounds).

A residual-based local detection signal can be written as:

$r_i(t)=y_i(t)-\hat{y}_i(t)$.

If $|r_i(t)|$ exceeds a threshold δ_i, the node may reduce trust in the received data, switch to a local safe mode or request verification from neighbors. In a fully distributed architecture, trust becomes dynamic and local. The system does not simply trust or distrust a component; it continuously estimates the reliability of information streams and adjusts control accordingly.

Trust should be treated as a graded input, not a binary gate. A message can be weighted by its confidence, its consistency with a node's local prediction, and its agreement with neighbouring observations. Extend the control law from plain state feedback to ***trust-weighted feedback*** by multiplying incoming measurements or commands with a trust weight that reflects these factors. The weights are updated online by employing basic consistency checks or probability weighting such that unreliable inputs have minimal impact on the outcome. This helps maintain decentralisation and at the same time enables the network to devalue faulty or outdated information. Thus, a smoother process control emerges whereby local estimates and remote information inputs are weighted based on their level of reliability:

$$u_i(t) = K_i(x_i(t), w_{ji}(t)m_{ji}(t), j \in N_i), 0 \leq w_{ji}(t) \leq 1.$$

The weight $w_{ji}(t)$ expresses the current trust assigned by node i to information from node j. This makes cyber-security part of the mathematical control structure. A suspected message receives lower influence, which reduces the probability that corrupted data will shift the system toward unsafe states.

Digital Twins as Probabilistic Observers

Both the Markov and the Fokker-Planck descriptions have their part in the analysis of factory reliability. Markov jumps split the process into distinct modes and give the probabilities of transitioning from one to another. Fokker-Planck equations model the smooth uncertainty leading the factory towards such transition points. Combining the small steps of transitions using Chapman-Kolmogorov helps effectively estimate how fluctuations on the machine level accumulate to long-term process evolution. Interpretation fives us crucial answers: increased probability for a Warning event in the Markov description is the sign of decreased stability, but not of imminent failure. On the other hand, when combined with an increased uncertainty shown through Fokker-Planck, the risk of surpassing critical points grows. In such a case, the appropriate response should be preventative: lighten the load, balance the workload, calibrate sensors, or increase monitoring frequency to avoid a Fault event. The approach to managing a factory should be probabilistic, which means keeping control over the distributions: individual controllers must ensure that their local state density does not exceed safe intervals and work together to prevent overloading their neighbours.

The same principles apply in the smart grid. Microgrids, distributed energy resources, batteries, electric vehicles and demand-response devices all embody fully distributed CPS control. The cyber layer—communication, computation and control—actively shapes physical flows in generation, transmission and distribution, and physical disturbances feed back into cyber decisions. This two-way coupling means cyber actions change power flows and voltages, which in turn alter mode probabilities and trigger

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

further control actions. Managing the grid therefore requires models that link discrete operational modes, continuous uncertainty, and the temporal composition of many local decisions.

From a logical standpoint, distributed control in the grid must balance local autonomy with network synchrony. A microgrid can optimise battery and inverter behaviour locally, but those actions change the boundary conditions seen by neighbouring units. While the Markov layer models the discrete mode switching events (connected, isolated, congested, and faulted modes), the Fokker-Planck layer models uncertainty with regard to the continuous variables (voltage, frequency). This forms the basis for a probabilistic notion of grid stability, which states that controls act on the probability of the occurrence of an excursion or mode switch instead of simply imposing strict limits. In terms of Markov processes, the attack is modeled by perturbation of the transition matrix, thereby increasing the probability of falling into the Fault mode. In terms of Fokker-Planck dynamics, an attack could change the drift term, amplify the diffusion term, or introduce multimodal uncertainty in the state probability distribution.

A digital twin for a fully distributed CPS/DCS should act as a probabilistic observer, not merely a geometric or deterministic replica. Let $X(t)$be the true physical state and $\hat{X}(t)$the twin's estimate; the estimation error $e(t) = X(t) - \hat{X}(t)$has a probability density $p(e,t)$whose evolution can be written in Fokker–Planck form. This lets the twin answer probabilistic questions—what is the chance a machine will overload in the next hour, that a microgrid will violate voltage limits, or that sensor data have been compromised. The twin thus bridges local stochastic control and strategic decision-making. The digital twin itself will be becoming increasingly more distributed over time. Each node should maintain its own digital twin and broadcast the probabilities and the degree of certainty within the estimates made to neighboring nodes. The overall digital twin becomes an outcome of connecting all the above models. In this case, the idea of Chapman-Kolmogorov and Fokker-Planck consistency can serve as guiding rules to connect the uncertain components of the twins. Thus, a distributed digital twin will be serving as an epistemic component, which represents certain beliefs and uncertainties concerning the status of the system in question.

This architecture is especially relevant for large-scale infrastructure. Cities, factories and energy networks cannot always stream all raw data to a central model in real time. It is often more efficient for local twins to transmit compressed probabilistic knowledge—predicted fault probabilities, confidence intervals, safe margins—rather than raw measurements. Designing digital twins in this way ties directly into Markov and Fokker–Planck modeling and makes probabilistic, distributed decision-making practical at scale.

DISCUSSION

The proposed framework argues that fully distributed DCS/CPS must be treated as stochastic, networked and adaptive systems. At the local level, Markov transitions provide the language for operational change; Chapman–Kolmogorov composition supplies the temporal logic by which many local decisions aggregate into long-term behavior; and the Fokker–Planck equation supplies the continuous language of uncertainty, representing the system not as a single point but as a probability distribution over states.

This viewpoint prevents a frequent pitfall. Distributed control is not simply a communication issue – the real problem lies in statistical coordination. Many localized decisions, each made using partial information and noise, have to be combined into one stable stochastic process. This involves ***probabilistic guarantees***, ***stochastic stability proof, Lyapunov conditions, and Chapman-Kolmogorov consistency*** to allow the designer to foresee the effect of combining the localized rules over time. This needs prediction-based digital twins that incorporate uncertainty propagation using Fokker-Planck type models where certainty and distributions act as control parameters. Decentralization increases scalability and reduces latency, but introduces risks of coordination if localized rules do not lead to decreased uncertainty and prevention of cascading transitions. Trustworthiness is then a quantitative concept: a decentralized CPS can be trusted only if mode probabilities and state distributions are kept within proven bounds in the presence of delays, noise, and adversarial disruptions.

Similarly, a probabilistic approach is applicable to digital twins, smart grids, and security issues related to CPS. Indeed, the use of digital twins in CPS is important due to their role in coupling physical assets with virtual models, real-time data, simulations, and predictive reasoning. Therefore, this technology is particularly useful in smart manufacturing and Industry 4.0 settings (Tao et al., 2018; Tao et al., 2019). Within the framework of cyber-physical power systems, the intricate relationship among the communication channel, the control procedure, and physics makes it necessary to incorporate randomness in the model in the light of distributed energy sources and smart-grid testbeds, along with other considerations (Abdelmalak et al., 2022; Cintuglu et al., 2017). In terms of security issues, it may be possible to integrate the stochastic nature through false data injections, denial-of-service attacks, replay

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

attacks, and using compromised data which affects the state actually measured by the controller and hence affects transition probability and stability (Alguliyev et al., 2018; Pasqualetti et al., 2013; Xing & Shen, 2024; Yu et al., 2023).

Putting the framework into practice requires data and continual model updating. Transition probabilities must be estimated from historical operation, simulation, expert judgment or online learning. Drift and diffusion terms should be identified from physical models and sensor streams. Cyber-security parameters must be revised as threats evolve. Thus the model is not a fixed formula but a dynamic modeling environment that evolves with the system it represents. The main theoretical contribution is interpretative integration. Markov theory, Chapman–Kolmogorov composition and Fokker–Planck dynamics are often treated in isolation, but in fully distributed CPS/DCS they describe different facets of the same phenomenon: local components change modes, those changes compose over time, and uncertainty evolves around the resulting physical trajectory. Treating these three levels together yields a more logical and realistic representation of distributed cyber-physical intelligence. The framework also implies a certain sequence for system design. In the first place, modes need to be defined, and the probabilities of transitions between them estimated. Next, it needs to be discovered how the local controllers affect such probabilities. Then, there is a need to study temporal transitions composition. Finally, it needs to be considered how to deal with continuous uncertainties and the probability mass. The above sequence helps turn abstract stochastic formulas into a systematic engineering approach for fully distributed architectures.

CONCLUSION

Fully distributed architectures of DCS and CPS are undoubtedly one of the most important directions in contemporary automation and smart infrastructures development. Such a design promises scalability, reliability, flexibility and self-sufficiency, but also requires a new level of mathematical insight into the properties of complex emerging systems. Stochastic modelling using Markovian transitions, Chapman-Kolmogorov compositions, and Fokker-Planck dynamics provides the basis for such a conceptual understanding of fully distributed CPS/DCS. The critical interpretive finding is that a fully distributed CPS/DCS should be modelled as a probabilistic architecture. Local controls influence transition probabilities; communication influences transition composition; and estimation and feedback influence the probability density function. All this working together results in stability and reliability. On the contrary, mere distributed systems do not guarantee distributed intelligence. Viewed together with graph models, distributed control laws and digital twins, the Markov, Chapman–Kolmogorov and Fokker–Planck layers provide a strong basis for analysing stability, resilience, communication delay, cyber-security, scalability and sustainability. The central conclusion is that the future of fully distributed DCS/CPS lies in uncertainty-aware distributed intelligence: systems that are probabilistic, adaptive and networked, and whose quality is measured by their ability to guide probability mass toward safe, efficient and resilient operating regions despite noise, faults, delays and attacks.

REFERENCES

1. Abdelmalak, M., Venkataramanan, V., & Macwan, R. (2022). A survey of cyber-physical power system modeling methods for future energy systems. *IEEE Access, 10*, 99875–99896. doi:10.1109/ACCESS.2022.3206830
2. Alguliyev, R., Imamverdiyev, Y., & Sukhostat, L. (2018). Cyber-physical systems and their security issues. *Computers in Industry, 100*, 212–223. doi:10.1016/j.compind.2018.04.017
3. Annunziato, M., & Borzì, A. (2018). A Fokker–Planck control framework for stochastic systems. *EMS Surveys in Mathematical Sciences, 5*(1/2), 65–98. doi:10.4171/EMSS/27
4. Bujorianu, M. L. (2018). Towards a modelling language for distributed control of cyber-physical systems. *IFAC-PapersOnLine, 51*(23), 432–437. doi:10.1016/j.ifacol.2018.12.074
5. Cintuglu, M. H., Mohammed, O. A., Akkaya, K., & Uluagac, A. S. (2017). A survey on smart grid cyber-physical system testbeds. *IEEE Communications Surveys & Tutorials, 19*(1), 446–464. doi:10.1109/COMST.2016.2627399
6. Ge, X., Yang, F., & Han, Q.-L. (2017). Distributed networked control systems: A brief overview. *Information Sciences, 380*, 117–131. doi:10.1016/j.ins.2015.07.047
7. Griffor, E. R., Greer, C., Wollman, D. A., & Burns, M. J. (2017). *Framework for cyber-physical systems: Volume 1, Overview* (NIST SP 1500-201). National Institute of Standards and Technology. doi:10.6028/NIST.SP.1500-201
8. Lee, E. A. (2008). Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)* (pp. 363–369). IEEE. doi:10.1109/ISORC.2008.25

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 09 Issue 06 June 2026
DOI: 10.47191/ijcsrr/V9-i6-01, Impact Factor: 8.048
IJCSRR @ 2026
www.ijcsrr.org

9. Mangharam, R., & Pajic, M. (2013). Distributed control for cyber-physical systems. *Journal of the Indian Institute of Science, 93*(3), 353–387. No DOI found.
10. Monroy Cruz, E., García Carrillo, L. R., & Cruz Salazar, L. A. (2023). Structuring cyber-physical systems for distributed control with IEC 61499 standard. *IEEE Latin America Transactions, 21*(2), 251–259. doi:10.1109/TLA.2023.10015217
11. NIST Cyber-Physical Systems Public Working Group. CPS Framework Release 1.0 and Timing Framework for Cyber-Physical Systems. National Institute of Standards and Technology.
12. NIST Cyber-Physical Systems Public Working Group. Framework for Cyber-Physical Systems: Volume 1, Overview. NIST Special Publication 1500-201, 2017. DOI: 10.6028/NIST.SP.1500-201.
13. Pasqualetti, F., Dörfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control, 58*(11), 2715–2729. doi:10.1109/TAC.2013.2266831
14. 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. doi:10.1145/1837274.1837461
15. Ren, M., Zhang, Q., & Zhang, J. (2019). An introductory survey of probability density function control. *Systems Science & Control Engineering, 7*(1), 158–170. doi:10.1080/21642583.2019.1588804
16. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., & Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology, 94*, 3563–3576. doi:10.1007/s00170-017-0233-1
17. Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: Correlation and comparison. *Engineering, 5*(4), 653–661. doi:10.1016/j.eng.2019.01.014
18. Tkáčik, M., Jadlovský, J., Jadlovská, S., Jadlovská, A., & Tkáčik, T. (2024). Modeling and analysis of distributed control systems: Proposal of a methodology. *Processes, 12*(1), 5. doi:10.3390/pr12010005
19. 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. doi:10.1109/TII.2011.2166785
20. Wollman, D. A., Weiss, M. A., Li-Baboud, Y., Griffor, E. R., & Burns, M. J. (2017). *Framework for cyber-physical systems: Volume 3, Timing Annex* (NIST SP 1500-203). National Institute of Standards and Technology. doi:10.6028/NIST.SP.1500-203
21. Xing, W., & Shen, J. (2024). Security control of cyber–physical systems under cyber attacks: A survey. *Sensors, 24*(12), 3815. doi:10.3390/s24123815
22. Yu, Z., Gao, H., Cong, X., Wu, N., & Song, H. H. (2023). A survey on cyber–physical systems security. *IEEE Internet of Things Journal, 10*(24), 21670–21686. doi:10.1109/JIOT.2023.3289625

Cite this Article: Vasilev, I. (2026). A Stochastic Framework for Fully Distributed Control Systems and CPS: From Local State Transitions to Global Uncertainty Propagation. International Journal of Current Science Research and Review, 9(6), pp. 2943-2957. DOI: https://doi.org/10.47191/ijcsrr/V9-i6-01