Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria

doi: 10.3850/ESREL2026061419_esrel26-p26410-cd

Towards automated Markov modeling and simulation of black-box systems

Adarsh Ratan Thakur

Safety and Resilience of Technical Systems, Fraunhofer Institute for High-speed Dynamics EMI, Germany.
E-mail: adarsh.ratan.thakur@emi.fraunhofer.de

Ivo Häring, Mirjam Fehling-Kaschek

Fraunhofer Institute for High-speed Dynamics EMI, Germany.
E-mail: ivo.haering@emi.fraunhofer.de , mirjam.fehling-kaschek@emi.fraunhofer.de,

Ever-increasing operational data from interconnected smart technical systems create opportunities for data-driven modelling. Yet inferring reliable system state dynamics from system level data alone, without knowledge of internal structure, functional design, or subsystem models, remains a fundamental challenge in black-box modelling. While the existing literature addresses specific aspects of this challenge, a unified methodology integrating these interdependent tasks into a closed-loop, criterion-driven workflow with automated convergence is absent. To address this gap, the present paper presents a seven-step workflow that systematically integrates scattered approaches into a closed-loop, iteratively converging process. The workflow covers: First, define system state, subsystem state and transition model-space based on system design assumptions. Second, generation of system state time histories. Third, disturbance and measurement modeling are incorporated to represent noise, partial observability, and environmental effects explicitly. Fourth, model structure and parameter learning include Bayesian inference via Markov chain Monte Carlo (MCMC) simulation. Fifth, hidden state estimation and forecasting. Sixth, evaluation and uncertainty are assessed to know the quality of model. Seventh, criterion-driven deployment monitoring with automated iteration until convergence. Each workflow step is supported by reviewed literature, clarifying methods, dependencies between steps and research gaps. Comparison with five existing approaches confirms that no prior work unifies structure selection, parameter learning, convergence criteria, and feedback routing for black-box reliability systems. Key research gaps are identified in automated state space construction. The method provides structured guidance for practitioners modeling partially observable systems, emphasizing transparency, reproducibility, and systematic iteration for black-box system under non-stationary conditions.

Keywords: Black-box state modelling, Transition parametrization, MCMC simulation, Automated convergence, System health monitoring, Uncertainty quantification.

1. Introduction

Modern technical systems are increasingly characterized by high complexity, interconnectivity, and continuous operation under changing conditions. In safety-critical applications such as energy infrastructure, industrial systems, and cyber-physical networks, system-level data is measured continuously through embedded sensors (Ross, 2019; Trivedi, 2001). However, detailed knowledge about internal structures and transition mechanisms often remains unavailable, requiring analysis under black-box conditions using only aggregated observations. The inference of states and dynamics from black-box data is fundamentally ill-posed: structurally different models can explain the same observations equally well (Zheng et al., 2022). Despite this, Markov and hidden Markov models (HMMs) remain widely adopted because they provide interpretable stochastic frameworks for system states, transitions, and uncertainties (Ross, 2019; Rabiner, 1989; Trivedi, 2001). However, existing work treats isolated tasks state-space definition, parameter estimation, state inference, uncertainty quantification independently without documenting their interdependencies (Gamiz et al., 2023; Zheng et al., 2022). Choices during state-space construction constrain parameter identifiability, observation models determine distinguishable states, and uncertainty depends on all prior assumptions. Without a unified framework, practitioners risk inconsistent assumptions and unreliable models.

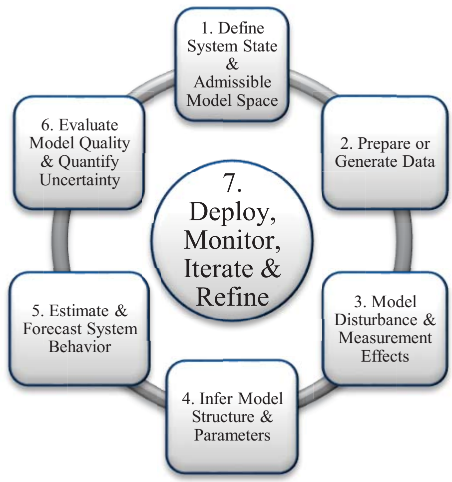
This paper addresses this gap by presenting a seven-step methodological framework that systematically connects the interdependent tasks of black-box Markov modeling into a closed-loop, automatically iterating process. The term *automated* in this work refers specifically to the criterion-driven iterative convergence of the complete workflow: Step 7 evaluates the current model against formally defined quality metrics and convergence criteria, identifies the step where assumptions or model components fail via diagnostic routing rules, and triggers targeted re-execution of earlier steps without requiring manual intervention at each iteration cycle. The process repeats until convergence to an optimal model within the predefined model space is achieved. This automation distinguishes the proposed workflow from existing approaches that treat each modeling step as an isolated, manually supervised task.

The contributions are: (1) a closed-loop workflow unifying structure selection, parameter learning, convergence criteria, and feedback routing for black-box reliability systems; (2) formal cross-step identifiability constraints linking steps theoretically; (3) MCMC-based Bayesian inference as alternative to EM within the workflow; (4) explicit method-selection criteria at each step. Table 4 confirms that no prior approach integrates all these capabilities.

2. Overview of the proposed seven-step workflow Methodology

The proposed seven-step workflow provides a systematic framework for Markov-based modelling of black-box systems. The workflow addresses the gap identified in Section 1 by explicitly connecting interdependent modelling tasks and embedding them within an automated convergence loop: state-space definition constrains parameter identifiability (Steps 1–4), observation models determine distinguishable states (Steps 3–5), uncertainty propagates from all prior steps (Steps 1–6), and the automated iteration mechanism (Step 7) closes the loop by evaluating model quality and triggering targeted re-execution when convergence criteria are not met.

Figure1. Overview of the proposed seven-step methodology



The framework makes modelling assumptions explicit at each step, documenting what is assumed, why, and how assumptions propagate. Steps 1–3 establish foundations: state-spaces and transitions (Step 1), representative data (Step 2), observation processes (Step 3). Steps 4–6 comprise inference: parameter learning (Step 4), state estimation (Step 5), uncertainty quantification (Step 6). Step 7 implements automated deployment with criterion-driven iteration. Table 1 illustrates the workflow. Table 1 summarizes each step.

Table 1 Overview of the proposed seven-step system state modelling workflow

Step	Description	Primary purpose
1	System state and model-space definition	Define admissible state representations, transitions, and modeling assumptions.
2	Data preparation and data generation	Prepare experimental or synthetic data suitable for model inference and validation.
3	Disturbance and measurement modeling	Explicitly represent noise, partial observability, and environmental effects.
4	Learning of model structure and parameters	Infer transition models and parameters within the defined model space.
5	State estimation and forecasting	Compute most probable hidden states and future system behavior.
6	Evaluation and uncertainty quantification	Assess model quality, confidence and predictive reliability.
7	Deployment, monitoring, and iterative refinement	Maintain model validity under non-stationary system conditions.

3. Seven-step methodology for black-box system state modeling

3.1. System state and model space definition

In black-box system modeling, learning from data is meaningful only if admissible system states and transitions are defined in advance. Without such constraints, multiple incompatible models can explain the same data (ill-posed inference). Step 1 defines the system state representation and admissible transition structure before data-driven modeling. Practically, this involves modeling the system as a multi-state Markov process built from subsystem health states using product state spaces with exponential or Weibull transitions. State aggregation or structural simplification limits state-space growth (Dayar & Orhan, 2016; Zheng et al., 2022; Ross, 2019; Trivedi, 2001). The number of hidden states should be initialized based on domain knowledge (typically 2–5 for degradation processes) and subsequently refined using information criteria within the automated iteration loop. The choice between discrete-time (DTMC) and continuous-time (CTMC) representations follows from the data sampling regime: fixed-interval sampled data supports DTMC formulations, while asynchronous event-time data requires CTMC with generator matrices. If observed sojourn times deviate significantly from exponential distributions (testable via Kolmogorov-Smirnov tests), semi-Markov extensions should be considered. The identifiability constraint $K \leq |Y|$ (Allman et al., 2009) must be verified at this stage. **Aim:** Define a constrained and interpretable model space providing a reliable foundation for subsequent modeling.

3.2. Data preparation and data generation

This step ensures data compatibility with assumed Markov models. Step 2 prepares real-world or generates synthetic data consistent with modeling assumptions. For synthetic data generation, two approaches are available: direct simulation samples an initial state from the initial distribution and propagates through the transition matrix at each time step, generating observations from the emission model at each state. This is computationally straightforward and sufficient for simple single-component systems. For complex multi-subsystem models where the product state space is large, MCMC methods using Gibbs sampling provide an alternative by resampling each hidden state from its conditional distribution given its two temporal neighbors, producing representative trajectories that respect the full joint distribution of the model (Andrieu et al., 2003). The generated synthetic data with known ground-truth hidden state sequences enables controlled validation of the learning algorithms in Steps 4–6: the estimated parameters and decoded states can be compared against the true generating model. For operational data, preprocessing must address missing values (interpolation or modified forward algorithms that skip missing observations), irregular sampling intervals (already handled by the CTMC framework via time-dependent transition matrices), and resolution limits while preserving the temporal dependence structure essential for Markov modeling. Feature extraction should produce time series consistent with assumed emission distributions. Time-to-event distributions may be constructed using Weibull models for failure and degradation behavior (Zhang, 2017; NIST/SEMATECH, 2023). **Aim:** Obtain data representations consistent with model assumptions and suitable for reliable inference and validation.

3.3. Disturbance and measurement modeling

This step distinguishes underlying system dynamics from the measurement process. In black-box settings, system states are not directly observable, and measurements are affected by noise, disturbances, operating conditions, and sensor limitations. If not modeled, parameter estimation and state inference become biased and overly optimistic. Hidden Markov models separate state evolution (governed by Markov dynamics) from observations (generated through probabilistic emission models). Time-varying or covariate-dependent HMMs capture external factors' influence on degradation behavior (Wang et al., 2025). Accurate observation process modeling is essential for reliable state estimation and decision support (Gamiz et al., 2023). The choice of HMM variant depends on system characteristics: standard HMM for autonomous degradation without external covariates; Input-Output HMM (IOHMM) when external inputs or covariates affect transitions; Factorial HMM for multiple concurrent independent degradation mechanisms; Hierarchical HMM for systems with nested operating modes. The identifiability constraint that emission distributions must be linearly independent across states (Gassiat & Rousseau, 2016) must be verified: two degradation states producing statistically indistinguishable observations cannot be separated regardless of estimation method. **Aim:** Define observation and disturbance models correctly relating system states to measured data under partial observability.

3.4 Learning of model structure & parameters

This step infers system structure, transition models, and parameter within the predefined model space, as Markov-based state estimation and forecasting depend entirely on learned model quality. Two principal estimation paradigms are available. Maximum likelihood via Expectation-Maximization: The Baum-Welch algorithm (a special case of EM for HMMs) iteratively estimates transition and emission parameters by alternating between computing expected state occupancies (E-step) and maximizing the expected complete-data log-likelihood (M-step) (Rabiner, 1989). Statistical reliability and computational performance depend on data length, state-space size, and optimization strategy (Yang et al., 2017; Perdry et al., 2024). EM is computationally efficient and scalable but provides only point estimates without uncertainty quantification and is susceptible to local optima.

Bayesian inference via Markov chain Monte Carlo (MCMC): When full uncertainty quantification is required or data is limited, Bayesian inference via MCMC provides posterior distributions over all model parameters (Andrieu et al., 2003). For finite-state HMMs typical in reliability (3–5 degradation states), the standard approach is Gibbs sampling with Forward-Filtering Backward-Sampling (FFBS) (Scott, 2002). Dirichlet priors $Dir(\alpha_1, \ldots, \alpha_K)$ are placed on each row of the transition matrix, exploiting conjugacy: if row i has prior $Dir(\alpha_1, \ldots, \alpha_K)$ and n_{ij} transitions from state i to j are observed in the imputed hidden state sequence, the posterior is $Dir(\alpha_1 + n_{i1}, \ldots, \alpha_K + n_{iK})$. The Gibbs sampler alternates between: (i) sampling hidden state sequences from their full conditional posterior using FFBS, given current parameter values; and (ii) updating transition and emission parameters from their conjugate posteriors given the imputed hidden states. Multiple parallel chains are run, convergence is assessed via the Gelman-Rubin $\hat{R}$ statistic ($\hat{R} < 1.05$), and posterior summaries (meaning 95% credible intervals) are computed from combined post-burn-in samples. For rare failure states that are poorly sampled under standard MCMC, targeted stochastic gradient approaches can over-sample observations from rare regimes (Ou et al., 2024). For complex state-space models where FFBS is intractable, Particle MCMC methods provide a general alternative requiring only that the model be simulable (Andrieu et al., 2010). **Identifiability constraint (Steps 1,3 → Step 4).** Parameter identifiability requires jointly that the transition matrix has full rank and that emission distributions are linearly independent. An HMM with K states and κ emission categories have $K(K + \kappa - 2)$ free parameters (Cole, 2019). If Step 4 detects signs of non-identifiability degenerate transition rows, near-identical emissions the automated routing reduces K in Step 1. **Method-selection guidance.** EM/Baum-Welch when data is abundant ($T >> K^2$) and computational speed is prioritized; Bayesian MCMC when data is limited relative to model complexity, informative priorities are available, or full posterior uncertainty quantification is required. A practical hybrid strategy uses EM for initial screening of candidate model structures and MCMC for final parameter estimation and uncertainty quantification of the selected model. **Aim:** To learn data-consistent model structures and parameters within the predefined Markov model space, with appropriate uncertainty characterization.

3.5. State estimation and forecasting

This step infers the most probable observable and hidden system states and their future evolution. Once a Markov model has been learned, raw observations alone do not provide direct access to system conditions. Filtering and smoothing algorithms estimate current and past system states from observation sequences, while dynamic programming approaches such as Viterbi decoding reconstruct the most likely state trajectories and support short- to medium-term forecasting (Viterbi, 1967; Jurafsky & Martin, 2023). These methods assume fixed transition and observation models and depend directly on the correctness of earlier steps. **Method-selection guidance.** For discrete-state HMMs: the Forward-Backward algorithm provides marginal posterior state probabilities at each time step (required for EM parameter learning), while the Viterbi algorithm provides the single most likely state sequence (required for diagnosis). For continuous state-space models: Kalman filter for linear-Gaussian systems; particle filters for non-linear or non-Gaussian cases. When Bayesian parameter estimation (Step 4) is used, state estimation should integrate over parameter uncertainty by averaging state probabilities across posterior samples. **Aim:** To estimate observable and hidden system state trajectories and forecast future system behavior consistent with the learned model and available observations.

3.6. Evaluation and uncertainty quantification

This step evaluates the reliability of the learned model and its state estimates. Since safety-related decisions cannot rely on point estimates alone, uncertainty must be explicitly quantified. Fisher Information or Hessian-based confidence intervals provide frequentist uncertainty bounds for transition and emission parameters (Ly et al., 2017). Bayesian approaches provide posterior distributions for parameters and predictions, enabling direct uncertainty propagation from model structure to forecasts (Murphy, 2012; Gelman et al., 2013). When MCMC-based inference is employed in Step 4, posterior predictive distributions over any derived reliability quantity system reliability, mean time to failure, steady-state availability are obtained by computing that quantity for each posterior sample. Credible intervals on transition probabilities and probability statements about safety thresholds flow directly from the MCMC output. **Model comparison:** AIC for short time series, BIC for large samples, DIC or WAIC for Bayesian models computed directly from MCMC output. Cross-validation for time-series HMMs require sequence-level blocking to preserve temporal dependence. **Aim:** To quantify confidence in the learned model and its prediction for uncertainty.

3.7 Automated deployment, monitoring, and iterative refinement

This step constitutes the automation mechanism that distinguishes this framework from manual, step-by-step modeling approaches. It ensures that the learned model remains valid when system behavior or operating conditions change and implements the criterion-driven iteration loop. **Process-level objective.** The automated workflow optimizes $J(m, \theta)$ over candidate model structures $m \in M$ and parameters θ: (a) negative held-out log-likelihood; (b) $BIC(m) = -2\ log\ L + |\theta|\ log\ T$, penalizing complexity; or (c) Bayesian model evidence via bridge sampling. **Convergence criteria.** Iteration terminates when: (i) relative improvement $|J_k - J_{k-1}|/|J_{k-1}| < \varepsilon$ (e.g., $\varepsilon = 10^{-3}$); (ii) all parameters satisfy convergence diagnostics (Gelman-Rubin for MCMC); (iii) no concept drift detected over N consecutive monitoring windows; or (iv) maximum iterations reached. **Diagnostic routing rules.** When convergence fails, Step 7 diagnoses the source: (a) emission residuals show systematic patterns (multimodality, non-random structure) → re-execute Step 3 with revised emissions; (b) degenerate state occupancy (near-zero posterior probability) indicates overspecification with $K(K + \kappa - 2)$ redundant parameters (Cole, 2019) → re-execute Step 1 with reduced K; (c) likelihood plateaus but residuals retain temporal structure → re-execute Step 1 with increased K; (d) parameters show high posterior variance despite adequate data → Simplify model structure (Step 1) or acquire additional data (Step 2); (e) concept drift detected via KL-divergence or Page-Hinkley test (Gama et al., 2014) → update emissions (Step 3) or state definitions (Step 1) through adaptive learning (Zhang et al., 2016). This criterion-driven loop eliminates manual trial-and-error, providing a reproducible, auditable path from black-box data to a deployed Markov model. **Aim:** Maintain model validity and drive the workflow toward convergence through automated, criterion-based iteration.

4. Method-Selection Guidance, Algorithm, and Comparison

Table 2 formalizes the automated convergence loop as Algorithm 1. Table 3 provides method-selection decision criteria at each workflow step. Table 4 compares the proposed workflow against five existing approaches, confirming that no prior work unifies structure selection, parameter learning, formal convergence criteria, and feedback routing for black-box reliability systems.

Table 2 Algorithm 1: Automated Markov Modelling

Input: Observations Y, model space $M = \{K_{min}, \ldots, K_{max}\}$, tolerance ε
Output: Optimal model m^*, parameters θ^*, state estimates
1: for K = K_{min} to K_{max} do
2: Define state space S_K and constraints (Step 1)
3: Prepare data for K-state model (Step 2)
4: Specify emission model B_K (Step 3)
5: Verify: $K \leq \|Y\|$ and emissions linearly independent
6: if identifiability violated then skip K, continue
7: Estimate θ_K via EM; compute $J(K) = BIC(K)$ (Step 4)
8: Diagnose residuals and state occupancies (Step 6)
9: If degenerate states then flag K as over specified
10: end for
11: Select $K^* = argmin\, J(K)$ among valid candidates
12: Refine $\theta_{\{K^*\}}$via MCMC Gibbs-FFBS (Step 4)
13: Compute state estimates and posteriors (Steps 5-6)
14: Deploy; monitor for drift (Step 7)
15: If drift detected then diagnose → route to step 1
16: Re-execute from identified step with updated M
17: return m^*, θ^*, state estimates, credible intervals

Table 3 Method selection decision matrix

Step	Key decision	Decision criterion	Recommended method
1	State count K	Domain knowledge + IC	Init K=2–5; refine via AIC (T<500) or BIC (T>500)
1	DTMC vs. CTMC	Sampling regime	Fixed-interval → DTMC; event-time → CTMC
3	HMM variant	System structure	Covariates → IOHMM; nested modes → HHMM; independent → Factorial
4	EM vs. MCMC	Data volume + UQ	T>>K² → EM; limited data or UQ → MCMC; hybrid for screening
5	Filtering method	State space type	Discrete → Fwd - Bwd/Viterbi; linear → Kalman; nonlinear → particle
6	Model comparison	Estimation approach	Frequentist: AIC/BIC; Bayesian: DIC/WAIC
7	Convergence metric	Application	Held-out likelihood, BIC, or Bayesian model evidence

Table 4 Comparison of proposed workflow

Approach	Structure selection	Parameter learning	Convergence criteria	Feedback routing	Reliability focus
HDP-HMM (Teh et al., 2006)	Yes	Yes (Gibbs)	No	No	No
Guo & Yang (2008)	Yes	No (assumed)	No	No	Yes (white-box)
Gamiz et al. (2023)	No (pre-set)	Yes (BW)	No	No	Yes
Auto-HMM	Yes (IC sweep)	Yes (EM)	No	No (linear)	No
Tappler et al. (2019)	Yes (iterative)	Yes	Yes (stat.)	Yes	No (verification)
Proposed workflow	Yes (criterion)	Yes (EM+ MCMC)	Yes (formal)	Yes (diagnostic)	Yes

5. Research gap and prioritization

While the framework unifies methods with formal convergence and identifiability constraints, challenges remain. (i) *Automated state-space construction:* nonparametric Bayesian approaches (HDP-HMM) or reversible-jump MCMC could determine K and parameters simultaneously. (ii) *Formal convergence guarantees:* criteria are empirically motivated; establishing theoretical guarantees remains open. (iii) *Routing rule formalization:* linking diagnostics to provably optimal re-execution targets would strengthen automation. (iv) *Scalability:* product state-spaces face exponential growth for many subsystems. (v) *Empirical validation* across diverse domains with public datasets is needed. These gaps represent future research opportunities; the current work provides structured methodological foundation.

6. Conclusion

This paper presented a seven-step workflow for automated Markov modelling of black-box systems. The central contribution is the closed-loop automation: Step 7 evaluates model quality against formal convergence criteria, diagnoses inadequacies via routing rules grounded in HMM identifiability theory (Allman et al., 2009; Gassiat & Rousseau, 2016; Cole, 2019), and triggers targeted re-execution without manual intervention. The paper further contributes a formalized convergence algorithm (Table 2), a method-selection decision criteria (Table 3) and explicit novelty positioning against existing approaches (Table 4). Future work focuses on software implementation, nonparametric Bayesian methods for state-space construction, empirical validation across domains, and formal convergence analysis.

Acknowledgement

This work was carried out within the framework of the EU-funded FOURIER doctoral programme and has received funding from the European Union's Marie Skłodowska-Curie Actions under Grant Agreement No. 101169429. The authors acknowledge the use of automated literature search and screening tools to support the identification and organization of relevant publication.

References

Allman, E. S., Matias, C., & Rhodes, J. A. (2009). Identifiability of parameters in latent structure models with many observed variables. Annals of Statistics, 37(6A), 3099–3132.

Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. Machine Learning, 50, 5–43.

Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society: Series B, 72(3), 269–342.

Cole, D. J. (2019). Parameter redundancy and identifiability in hidden Markov models. METRON, 77, 105–118.

Dayar, T., & Orhan, M. C. (2016). Cartesian product partitioning of multi-dimensional reachable state spaces. Probability in the Engineering and Informational Sciences, 30(4), 413–430.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44, 1–37.

Gamiz, M. L., Limnios, N., & Segovia-García, M. (2023). Hidden Markov models in reliability and maintenance: A review. European Journal of Operational Research, 304(3), 818–836.

Gassiat, E., & Rousseau, J. (2016). Nonparametric finite translation mixtures with dependent regime. Bernoulli, 22(1), 193–212.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian Data Analysis (3rd ed.). CRC Press.

Guo, H., & Yang, X. (2008). Automatic creation of Markov models for reliability assessment of safety instrumented systems. Reliability Engineering & System Safety, 93(6), 829–837.

Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed.). Stanford University.

Ly, A., Marsman, M., Verhagen, J., Grasman, R. P. P. P., & Wagenmakers, E.-J. (2017). A tutorial on Fisher information. Journal of Mathematical Psychology, 80, 40–55.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

NIST/SEMATECH. (2023). e-Handbook of Statistical Methods. National Institute of Standards and Technology.

Ou, Z., Young, A., Sen, S., & Dunson, D. (2024). Targeted stochastic gradient MCMC for HMMs with rare latent states. Bayesian Analysis, advance online.

Perdry, H., Génin, M., & Lajarge, C. (2024). Comparing hidden Markov models: A comparative study of HMM parameter estimation methods. Computational Statistics & Data Analysis, 189, 107843.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257–286.

Ross, S. M. (2019). Introduction to Probability Models (12th ed.). Elsevier.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: Recursive computing in the 21st century. Journal of the American Statistical Association, 97(457), 337–351.

Tappler, M., Aichernig, B. K., Bacci, G., Eichlseder, M., & Larsen, K. G. (2019). Probabilistic black-box reachability checking. Formal Methods in System Design, 53, 191–227.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. Journal of the American Statistical Association, 101(476), 1566–1581.

Trivedi, K. S. (2001). Probability and Statistics with Reliability, Queuing, and Computer Science Applications (2nd ed.). Wiley.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13(2), 260–269.

Wang, L., Yang, M., & Andrews, J. (2025). Degradation hidden Markov models with time-varying parameters for partially observed systems. IEEE Transactions on Reliability, 74, 1–14.

Yang, F., Balakrishnan, S., & Wainwright, M. J. (2017). Statistical and computational guarantees for the Baum-Welch algorithm. Journal of Machine Learning Research, 18, 1–53.

Zhang, Z. (2017). Parametric regression model for survival data: Weibull regression. Annals of Translational Medicine, 5, 484.

Zhang, Z., Li, X., & Jia, X. (2016). Adaptive hidden Markov models for bearing fault diagnosis under time-varying conditions. Mechanical Systems and Signal Processing, 80, 219–233.

Zheng, C., Zeng, Z., & Zio, E. (2022). Analysis of simplification in Markov state-based models for reliability assessment. Reliability Engineering & System Safety, 222, 108441.