A Class of Non-Gaussian State Space Models with Exact Likelihood Inference

BOYINA Leena   RAMESH Brian   TONIN Ireni

11 janvier 2026

Outline

The Autoregressive Gamma-Poisson Model (Overview)

A. The Observation (The Visible)

$$y_{it} \sim \mathcal{P}(h_t \cdot \tau_t \cdot e^{x'_{it}\beta})$$

- y_{it} : Event counts (defaults).
- τ_t : Exposure (e.g., active firms).
- $x'_{it}\beta$: Observable regressors.
- h_t : **Latent Process** (creates overdispersion & autocorrelation).

B. The State (The Invisible)

$$p(h_t|h_{t-1}) \sim \text{ARG } (\theta = (\phi, \nu, c))$$

- Discrete time equivalent of the **CIR** process (Finance).
- Ensures $h_t > 0$ (mandatory for Poisson intensity).

3. The Parameters $\theta = (\phi, \nu, c)$

- ϕ **(Persistence) :** Memory of the process ($0 < \phi < 1$).
  Interpretation : If $\phi \approx 1$, shocks (crises) persist ; if $\phi \approx 0$, risk is volatile/mean-reverting.
- ν **(Shape) :** Baseline level of average risk (h_t).
  Constraint : $\nu > 1$ (Feller Condition) to prevent intensity from sticking to zero.
- c **(Scale) :** Volatility of the risk (Variance of the variance).
  Interpretation : Controls how violently the market jumps between calm and crisis states.

The Auto-Regressive Gamma-Poisson Model (Cox Process)

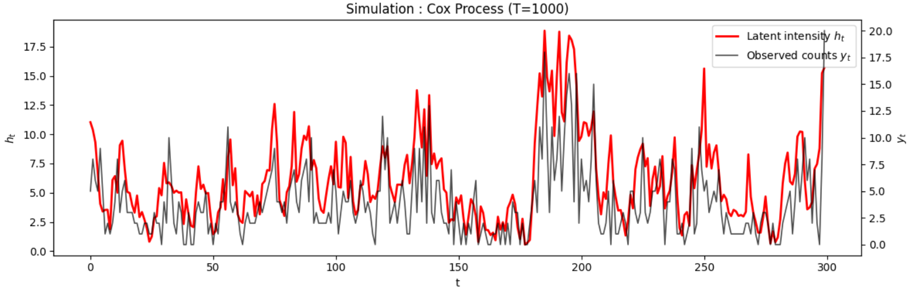


Figure – Cox Process (first 300 observations among 1000)

The Challenge : The "Intractable" Likelihood

The Goal : To estimate the parameters $\theta = (\nu, \phi, c)$, we must maximize the likelihood of the observed data $L(\theta) = p(y_{1:T}|\theta)$.

Why is this hard ?

- **Non-Gaussian :** We cannot use the Kalman Filter (which only works for Normal distributions).
- **High-Dimensional :** For $T = 1000$ observations, this is a 1000-dimensional integral.
- **No Analytical Solution :** Generally, this integral has no closed form.

The Research Question

How do we solve this ? We compare two competing philosophies :

1. **The Exact Trick (Creal, 2017) :** A specific mathematical derivation to solve the sum exactly.
2. **The Approximation (PMMH) :** A general-purpose simulation (Monte Carlo) to estimate the integral.

The Exact Approach (Creal, 2017)

The Key Insight : Transforming Continuous to Discrete.
Creal introduces an **auxiliary variable** z_t (Poisson) such that :

$$p(h_t|h_{t-1}) \longrightarrow p(h_t|z_t) \times p(z_t|h_{t-1})$$

- By conditioning on z_t, we can integrate out h_t analytically.
- The model transforms into a **Discrete Hidden Markov Model (HMM)**.
- The state space for z_t is infinite $(0, 1, \ldots, \infty)$, but we **truncate** it at a threshold Z (e.g., $Z = 200$).

Result : We obtain the **Exact Likelihood** (up to machine precision) using standard Forward-Backward recursions.

The Exact Filter Mechanism : Bridging Latent and Observed

To compute the likelihood exactly, we rely on the specific relationships between the variables h_t, z_t, and y_t.

A. The Latent Dynamics (From Continuous to Discrete)

The transition $h_{t-1} \to h_t$ is broken down into two steps using an intermediate integer z_t :

- **1. Inertia (Poisson) :** z_t captures the energy from the past.

$$z_t \mid h_{t-1} \sim \mathcal{P}\left(\frac{\phi h_{t-1}}{c}\right)$$

- **2. Diffusion (Gamma) :** h_t is regenerated around this inertia.

$$h_t \mid z_t \sim \Gamma(\nu + z_t, c)$$

B. The Conjugacy (Eliminating h_t)

The integral of the Poisson observation times the Gamma latent state yields a closed form :

$$p(y_t|z_t) = \int_0^\infty \underbrace{p(y_t|h_t)}_{\text{Poisson}(h_t)} \times \underbrace{p(h_t|z_t)}_{\text{Gamma}(h_t)} \; dh_t$$

Result : A **Negative Binomial** distribution. h_t disappears from the equation.

The Exact Filter : The Recursive Loop

The Likelihood L_t is the dot product of the **Observation** vector and the **Prediction** vector.

$$L_t = \sum_{k=0}^{Z} \underbrace{p(y_t \mid z_t = k)}_{\text{Observation}} \times \underbrace{p(z_t = k \mid y_{1:t-1})}_{\text{Prediction}}$$

1. The Observation Term

$$p(y_t \mid z_t)$$

The probability of the current data y_t given a specific latent level z_t.

2. The Prediction Term

$$p(z_t \mid y_{1:t-1})$$

Computed by propagating the **Previous Filter** (ω_{t-1}) through the transition.

Distribution : NegBin

$$y_t \mid z_t \sim \text{NB}(r_{obs}, p_{obs})$$

Parameters :

$$r_{obs} = \nu + z_t$$

$$p_{obs} = \frac{1}{1 + c\tau_t}$$

Transition : NegBin

$$z_t \mid z_{t-1} \sim \text{NB}(r_{trans}, p_{trans})$$

Parameters :

$$r_{trans} = \nu + z_{t-1} + \mathbf{y_{t-1}}$$

$$p_{trans} = \frac{1 + c\tau_{t-1}}{1 + c\tau_{t-1} + \phi}$$

The Recursive Link : The prediction at t uses the weights ω_{t-1} calculated at $t-1$:

$$p(z_t|y_{1:t-1}) = \sum_j p(z_t|z_{t-1} = j) \times \underbrace{\omega_{t-1}^{(j)}}$$

The Comparative Approach : Bootstrap Filter

When no exact trick exists, we use the **Particle Filter (SMC)**.

Algorithm Loop : Iterate for each observation y_t $(t = 1 \dots T)$:

1. **Propagation (Mutation) :** Draw N candidate particles from the transition model :
$$h_t^{(i)} \sim p(h_t | h_{t-1}^{(i)}) \quad \text{(Simulated via Poisson-Gamma)}$$
2. **Correction (Weighting) :** Assign weights based on the observation y_t :
$$w_t^{(i)} = p(y_t | h_t^{(i)}) \quad \text{(Poisson Density)}$$
3. **Likelihood Estimation :** The average weight gives the unbiased likelihood estimate :
$$\hat{L}_t = \frac{1}{N} \sum_{i=1}^{N} w_t^{(i)}$$
4. **Resampling :** Select particles proportional to $w_t^{(i)}$ to focus on probable regions.

Pros :

- Universal (works for any model).
- Easy to implement.

Cons :

- The estimated likelihood is **noisy**.
- Computational cost scales with N.

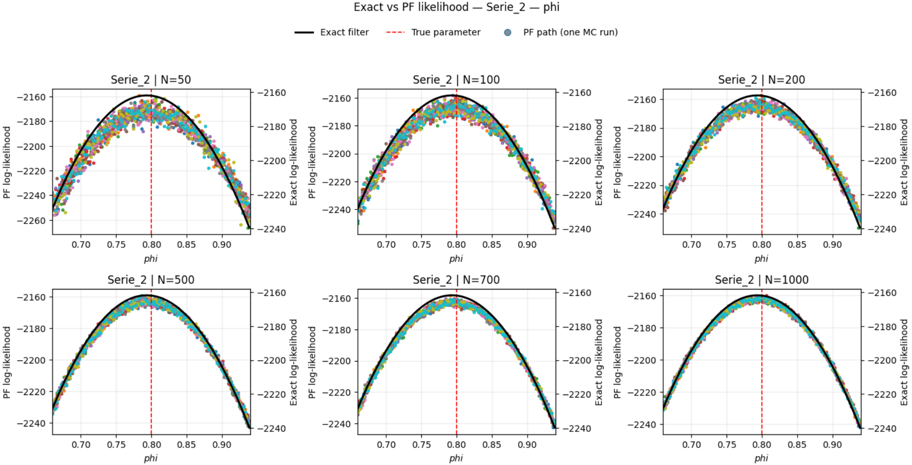


Figure – Contrast : Smooth Curve vs. Noisy Step Function

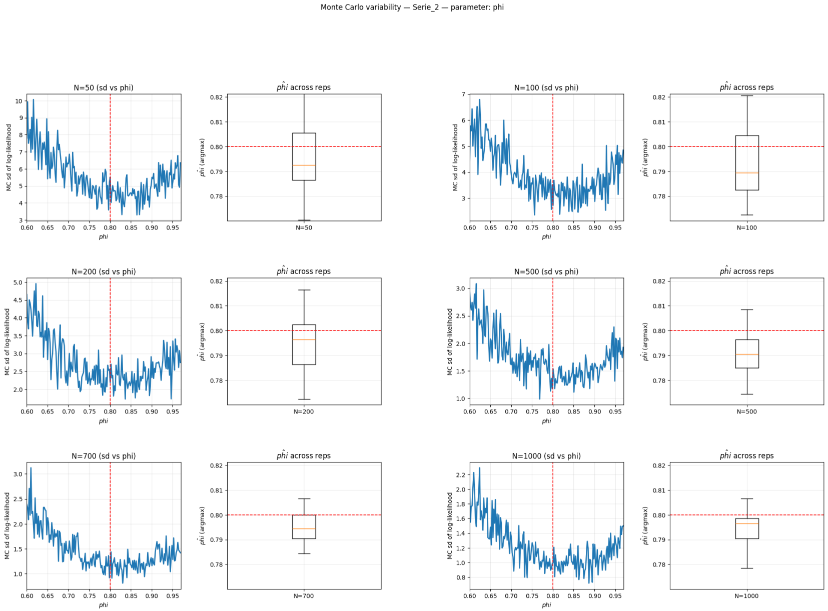


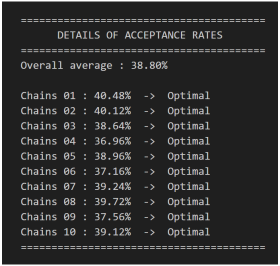
Figure – Variance of estimation of Bootstrap filter over $R = 30$ iterations.

The "Gold Standard" : RWMH on Exact Likelihood

We implemented a Random Walk Metropolis-Hastings (RWMH) algorithm targeting the exact posterior.

Implementation Results (20 Parallel Chains)

- **Acceptance Rate :** $\approx 40\%$ across all chains.
- This falls perfectly within the optimal theoretical range (20-50%), validating our tuning.



```
========================================
     DETAILS OF ACCEPTANCE RATES
========================================
Overall average : 38.80%

Chains 01 : 40.48%  ->  Optimal
Chains 02 : 40.12%  ->  Optimal
Chains 03 : 38.64%  ->  Optimal
Chains 04 : 36.96%  ->  Optimal
Chains 05 : 38.96%  ->  Optimal
Chains 06 : 37.16%  ->  Optimal
Chains 07 : 39.24%  ->  Optimal
Chains 08 : 39.72%  ->  Optimal
Chains 09 : 37.56%  ->  Optimal
Chains 10 : 39.12%  ->  Optimal
========================================
```

Figure – Acceptance rates of the mcmc chains computed

Analysis of Mixing :

- Even with the exact method, the Autocorrelation Function (ACF) decays slowly.
- **Interpretation :** This is intrinsic to the Cox model. The parameters (ϕ, ν, c) are structurally correlated, making exploration difficult.

The "Gold Standard" : RWMH on Exact Likelihood

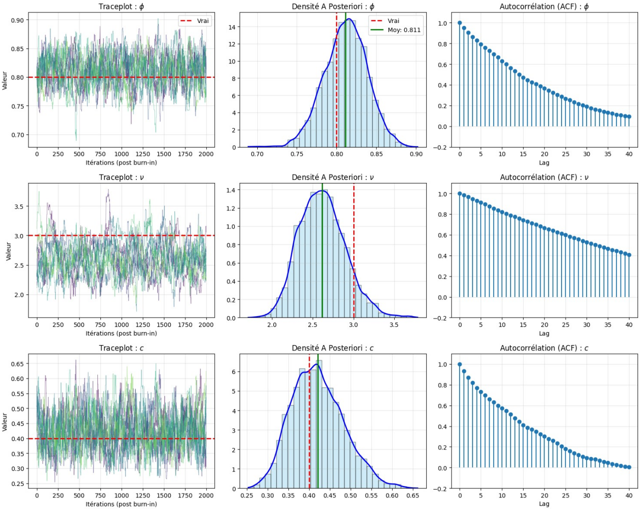


Figure – MCMC diagnostics showing good mixing (left), accurate parameter recovery (center), and decaying autocorrelation (right).

The PMMH Challenge : Variance vs. Mixing

Methodology : We implemented the **Particle Marginal Metropolis-Hastings (PMMH)** algorithm for the Creal et al. Cox model.

- The intractable likelihood $P(y|\theta)$ is estimated unbiasedly using a **Particle Filter** with N particles.
- **Trade-off :** The number of particles N controls the balance between computational speed and the variance of the likelihood estimator.

1. Impact of N on Mixing (Trace Plots & ACF) : Comparing Low N (200, red) vs. High N (1500, blue) :

- **Sticking Phenomenon :** With Low N, the variance of the likelihood estimate is high, leading to rare acceptances. The chain "sticks" (flat red lines).
- **Mixing :** High N reduces noise, allowing the chain to explore the posterior efficiently (fuzzy blue trace).
- **ACF :** The autocorrelation decays rapidly for High N, proving asymptotic independence of samples, whereas Low N remains highly correlated.

Numerical Stability & Robustness

2. Monte Carlo Variance (Stability Test) : Since PMMH is a stochastic algorithm, a single run is insufficient to validate convergence. We performed a stability test by running the algorithm **10 independent times**.

- **Observation :** The boxplots display the distribution of the estimated posterior means for each parameter (ϕ, ν, c, β).
- This step ensures "Numerical Common Sense" by quantifying the uncertainty inherent to the Monte Carlo method itself.

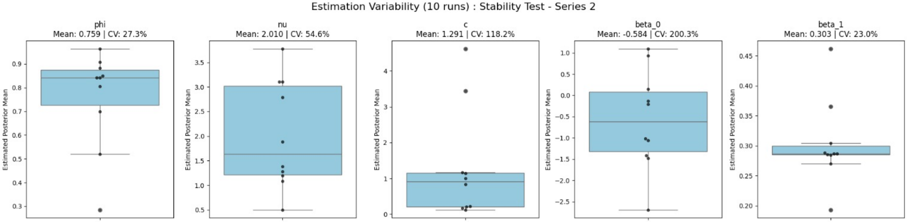


Figure – Robustness Check : Assessing Estimation Uncertainty

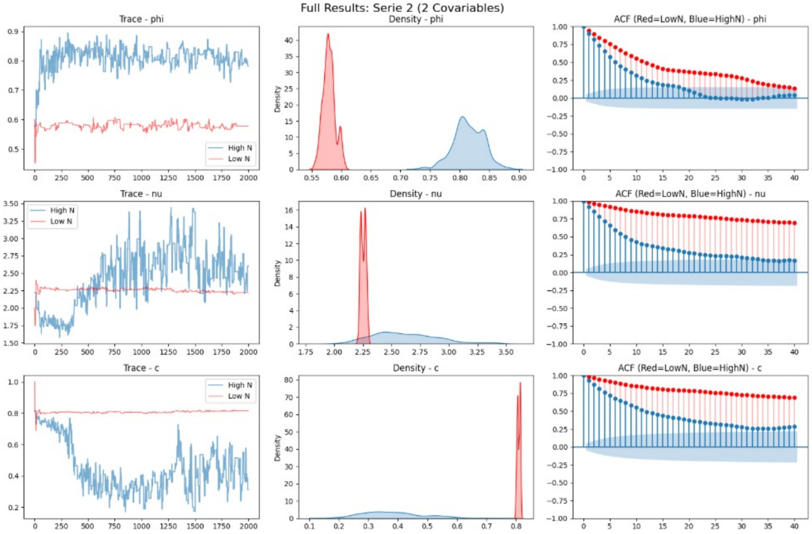


Figure – Impact of Particle Count (N) on PMMH Mixing Properties

Conclusion & Takeaways

- **Exact filtering** exploits model-specific conjugacy to compute the likelihood *exactly*, up to numerical precision.
- It provides a reliable **benchmark** for filtering distributions and likelihood-based inference.
- **Particle filters** are fully generic and widely applicable, but yield noisy likelihood estimates and require a large number of particles for accuracy.
- **MCMC on the exact likelihood** achieves superior mixing and parameter recovery compared to PMMH.
- **Key message :** when available, exact methods should be preferred for inference ; particle methods remain essential when no analytical structure can be exploited.

Thank you !