Estimating Hidden Dynamics in Stochastic Reaction Networks from Partial Observations

Mingkai Yu (Melchizedek Education and Research Institute, MEERI)
joint work with Muruhan Rathinam (UMBC)

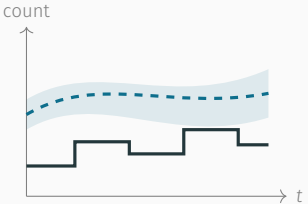
ICM@ICM 2026 Satellite Conference
Morgan State University
June 2, 2026

Problem overview

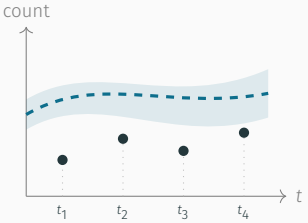
Model: A continuous-time Markov chain $Z(t)$ on $\mathbb{Z}_+^n$. It jumps as reactions fire.

Data: We watch some coordinates exactly, either continuously, or at snapshots.

Goal: Combine model and data to recover the distribution of what we can't see.



Continuous-time: the whole path of Y is seen.



Discrete-time: Y is seen only at $t_1, \ldots, t_k$.

Solid = observed Y | Dashed = hidden process | Band = our uncertainty

Stochastic reaction network: a Continuous-time Markov Chain (CTMC)

Stochastic reaction network: n molecular **species** and m **reaction channels**. The copy-number vector $Z(t) \in \mathbb{Z}_+^n$ is a **continuous-time Markov chain**, characterized by

- **Stoichiometric vectors** $\nu_j \in \mathbb{Z}^n$: reaction j sends $Z(t) = Z(t-) + \nu_j$;
- **Propensities** $a_j(z)$: $P\{Z(t+h) = z + \nu_j \mid Z(t) = z\} = a_j(z)\,h + o(h)$.

Example: gene expression (S = mRNA, A = protein), $Z(t) = (\#A(t), \#S(t))$.

$$S \xrightarrow{c_1} S + A,$$
$$\varnothing \xrightarrow{c_2} S,$$
$$S \xrightarrow{c_3} \varnothing.$$

$$\nu = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix},$$

$$a_1(z) = c_1 z_2, \quad a_2(z) = c_2, \quad a_3(z) = c_3 z_2.$$

From the master equation to stochastic simulation

The probability mass function $p(t, z) = P\{Z(t) = z\}$ evolves by the **Kolmogorov's forward equations** also known as the **chemical master equations** (CME):

$$p'(t, z) = \sum_{j=1}^{m} a_j(z - \nu_j)\, p(t, z - \nu_j) - \sum_{j=1}^{m} a_j(z)\, p(t, z). \tag{1}$$

The state space is large or infinite, so the CME is intractable.

We instead draw sample paths with the **Stochastic Simulation Algorithm** [Gillespie, 1977]. From the initial state z_0, repeat until the terminal time T:

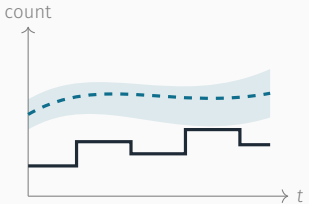
- **When?** Wait an exponentially distributed holding time with total rate $\lambda_0(z) = \sum_{j=1}^{m} a_j(z)$.
- **Which?** Fire reaction j with probability $a_j(z)/\lambda_0(z)$, then update $z \leftarrow z + \nu_j$.

Problem Statement

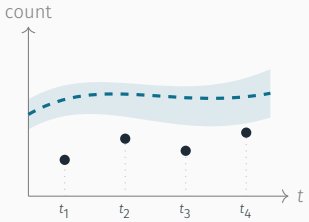
Exact (noiseless) observation of the last n_2 species.
$Z(t) = (X(t), Y(t))$: $X(t) \in \mathbb{Z}_+^{n_1}$ **hidden**, $Y(t) \in \mathbb{Z}_+^{n_2}$ **observed**.

Scenario 1 (continuous). $\pi(t, x) = P\{X(t) = x \mid Y(s) = y(s),\ 0 \le s \le t\}$.

Scenario 2 (snapshots). $\pi(t, z) = P\{Z(t) = z \mid Y(t_i) = y_i\}, \quad t \in [0, T]$.



hidden $X(t)$ | observed $Y(t)$



hidden $Z(t)$ | snapshots $Y(t_i)$

Continuous observations: filtering equations to Monte Carlo

The conditional distribution $\pi(t, x)$ satisfies the **filtering equations**: a system of nonlinear ODEs with jumps.

Two obstacles:

- Infeasible to solve directly when the state space is large or infinite.
- Nonlinearity $\Rightarrow \pi$ is *not* the law of a Markov process.

Fix: unnormalized conditional distribution $\rho(t, x)$:

- ρ satisfies a linear evolution and recovers π: $\pi(t, x) = \rho(t, x) \big/ \sum_{\tilde{x}} \rho(t, \tilde{x})$.
- ρ is the mean of a Markov process (V, w) we can *simulate* an unbiased weighted Monte Carlo (particle) filter.

Filtering equations: Fristedt, Jain & Krylov (2007). Finite-state CTMC: Confortola & Fuhrman (2013). Unnormalized form & reaction-network Monte Carlo simulation: **Rathinam & Yu (2021)**.

Particle filter: simulating $(V(t), w(t))$

V is a proxy for the hidden X and w a weight. Estimate ρ with N_s i.i.d. particles $\hat{\rho}(t,x) = \frac{1}{N_s}\sum_{i=1}^{N_s} 1_{\{x\}}(V^{(i)}(t))\, w^{(i)}(t)$.

Between observed jumps $(t_k \le t < t_{k+1})$:

- Evolve V base on the network with all observable reactions removed.
- Decay the weight: $w(t) = w(t_k)\exp\Big\{-\int_{t_k}^{t} a^{\mathcal{O}}(V(s), y(t_k))\, ds\Big\}$.

At an observed jump t_k**:** pick $j \in \mathcal{O}_k$ uniformly $(1/|\mathcal{O}_k|)$ and set

$$V(t_k) = V(t_k-) + \nu'_j, \qquad w(t_k) = w(t_k-)\, a_j(V(t_k-),\, y(t_{k-1})),$$

where $\mathcal{O}_k = \{\, j \in \mathcal{O} \mid \nu''_j = y(t_k) - y(t_k-) \,\}$.

Result [Rathinam & Yu, 2021]

$\mathbb{E}[\,1_{\{x\}}(V(t))\, w(t)\,] = \rho(t,x)$ for all $t \ge 0$, x, where ρ solves the unnormalized filtering equations.

Numerical example: estimating hidden states

Example 1: Gene Expression (S = mRNA and A = protein)

$$S \xrightarrow{c_1} S + A, \varnothing \xrightarrow{c_2} S, S \xrightarrow{c_3} \varnothing,$$

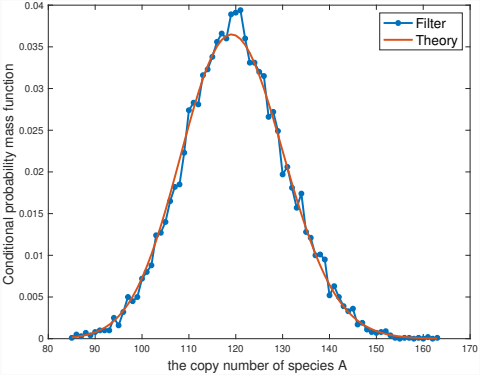


Figure 1: Example 1 (Species S is observed in continuous-time, while the species A is unobserved). Thus, $Z(t) = (X(t), Y(t)) = (\#A(t), \#S(t))$. Estimated conditional probability distribution of $A(T)$ compared with the exact theoretical function at time $T = 20$. $z_0 = (0, 5)$, parameter vector $c = (1, 5, 1)$, filter sample size $N_s = 10,000$. Theoretical distribution is computed precisely by $\pi(t, x) = \frac{\lambda^x e^{-\lambda}}{x!}$, where $\lambda = \int_0^t c_1 y(t) dt$.

Numerical example: estimating hidden states

Example 2: Genetic Toggle Switch

$$\varnothing \xrightarrow{a_1} S_1,\ S_1 \xrightarrow{a_2} \varnothing,\ \varnothing \xrightarrow{a_3} S_2,\ S_2 \xrightarrow{a_4} \varnothing.$$

with $a_1(z) = \frac{\alpha_1}{1+z_2^\beta}$, $a_2(z) = z_1$, $a_3(z) = \frac{\alpha_2}{1+z_1^\gamma}$, $a_4(z) = z_2$.

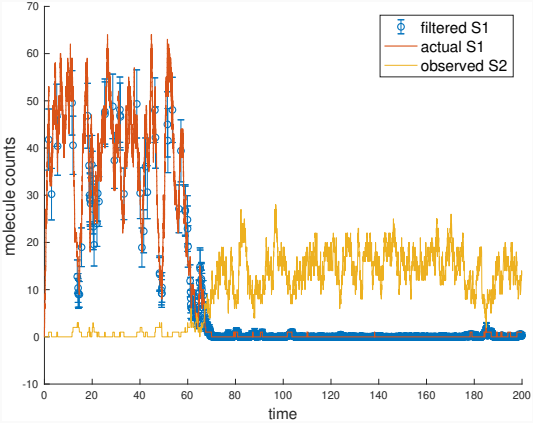


Figure 2: Example 2 (Species S_2 is observed in continuous-time, while the species S_1 is unobserved). Estimation of copy number of S_1 at time t based on one observed trajectory of S_2 over $[0, t]$. Estimates are shown at jump times of the observed species S_2 along with their 68% confidence intervals. Filter sample size $N_s = 10,000$. $Z(0) = (0, 0)$, $\alpha_1 = 50$, $\alpha_2 = 16$, $\beta = 2.5$, $\gamma = 1$.

Numerical example: Bayesian parameter estimation

Propensity functions $a_j = a_j(x, y, c)$ could contain unknown constants.

- Start with a prior distribution $\bar{\mu}$ on the parameter space
- Treat c as a constant-valued stochastic process $C(t)$
- Apply our algorithm to Markov process $Z(t) = (X(t), C(t), Y(t))$

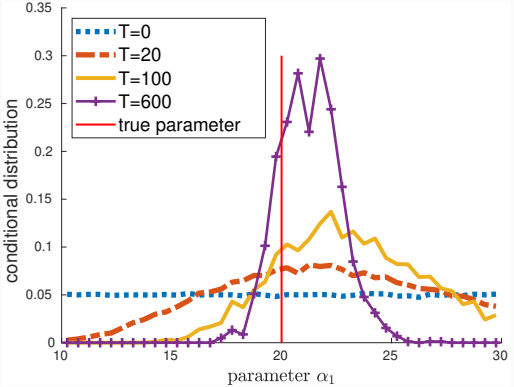


Figure 3: Example 2. The posterior conditional probability density function of parameters α_1 and in the genetic toggle example. The nominal parameter values were $\alpha_1 = 20$, $\alpha_2 = 9$, $\beta = 2.5$, $\gamma = 1$.

Discrete in time observations: Why the obvious filter collapses

Snapshot setting: initial law μ on $Z(0)$, one terminal observation $Y(T) = y$.
Estimate $\pi(t, z) = P\{Z(t) = z \mid Y(T) = y\}$, $z \in \mathbb{Z}_+^n$, $t \in [0, T]$.

Lab process: $Z = (X, Y)$, . Observation $Y(T) = y$ is maded based on lab process.

Method/Filter process: $\tilde{Z} = (U, V)$, Monte Carlo simulations.

Naive accept/reject. Simulate N_s free paths to T; keep only those with $V(T) = y$:

$$\widehat{\pi}(t, z) = \frac{\sum_i \mathbf{1}_{\{z\}}\big(\tilde{Z}^{(i)}(t)\big)\, W^{(i)}}{\sum_i W^{(i)}}, \quad W^{(i)} = \mathbf{1}_{\{y\}}(V^{(i)}(T)).$$

The chance of landing on a given snapshot is tiny, so almost every weight is 0.

Conditional Propensity

Simulate under a measure $\mathbb{P}_y$ that makes $\{Y(T) = y\}$ likely, and correct with a Radon–Nikodym (Girsanov) weight $L^{(i)}$.

$$\hat{\pi}(t, z) = \frac{\sum_{i=1}^{N_s} 1_{\{z\}}(\tilde{Z}^{(i)}(t))\, L^{(i)}\, W^{(i)}}{\sum_{i=1}^{N_s} L^{(i)}\, W^{(i)}}.$$

The "ideal" tilt is the conditional propensity

$$\lambda_j(t, z \,|y) = \lim_{h \to 0} \frac{P\{Z(t+h) = z + \nu_j \,|\, Z(t) = z, Y(T) = y\}}{h} = a_j(z) \frac{P\{Y(T) = y \,|\, Z(t) = z + \nu_j\}}{P\{Y(T) = y \,|\, Z(t) = z\}}.$$

It is general intractable. Golightly–Sherlock (2019) seeks to sample directly from the conditional distribution via the use of approximated conditional propensity functions.

Our approach: Targeting algorithm (overview)

- **Lab process:** $Z = (X, Y)$, $Z(t) = Z(0) + \nu R(t)$, where R is reaction counts.
- **Method/Filter process:** $\tilde{Z} = (U, V)$, $\tilde{Z}(t) = \tilde{Z}(0) + \nu I(t)$, where I is reaction counts.
- **Goal:** Construct a proxy process $\tilde{Z} = (U, V)$ such that $V(T) = y$, always reaching the target state.
- **Idea 1:** The partial final state constraint $V(T) = y$ can be translated into **reaction counts** constraints for some of the reactions.
- **Idea 2:** If the value of an inhomogenous Poisson process at time $T > 0$ is known, we can simulate exact conditional realizations of its path on $[0, T]$.

Targeting Algorithm: Reaction Counts Constraints

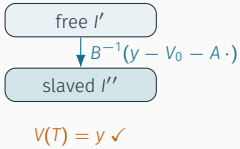
Reaction counts $I(t)$ drive the filter process: $\widetilde{Z}(t) = \widetilde{Z}(0) + \nu\, I(t)$. On the observed block,

$$V(T) = V_0 + \nu''\, I(T), \qquad \nu'' = [A \mid B], \quad B \in \mathbb{Z}^{n_2 \times n_2} \text{ invertible.}$$

Split $I = (I', I'')$ into free (I') and slaved (I'') counts. Conditioning on $V(T) = y$ pins the slaved counts:

$$\boxed{I''(T) = B^{-1}\big(y - V_0 - A\, I'(T)\big)}.$$

So the observation becomes a *linear constraint on reaction counts.* Choose the free counts freely; the slaved counts are determined — every particle reaches the target y by construction.



Targeting Algorithm: Inhomogeneous-Poisson Proposal

Under the simulation measure $\mathbb{P}_0$, proposed reaction counts I are an m-variate inhomogeneous Poisson process with intensity $\lambda(t, z_0, y)$.

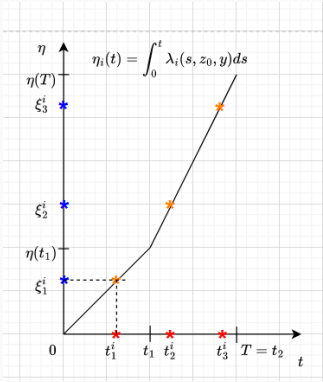
- **Predict.** Draw $\widetilde{Z}_0$ and the free counts $K' = I'(T)$.
- Compute $K'' = B^{-1}(y - V_0 - A\,K')$. Accept if $K'' \in \mathbb{Z}_+^{m_2}$, else redraw.
- **Correct.** Weight by the Poisson p.m.f. of the slaved counts,

$$W = \rho(K''), \qquad \rho \sim \text{Poisson}\Big(\int_0^T \lambda''(t, z_0, y)\, dt \Big).$$

Targeting Algorithm: Interpolation of inhomogeneous Poisson processes

- Having obtained a value for $\tilde{Z}_0$ and $I(T)$ along with a weight $W > 0$, obtain $I(t)$ for $0 \leq t \leq T$ by **"interpolation"**.
- $I(t)$ on $[0, T]$ that has same intensity as $R(t)$ and satisfies $I(T) = K$.

- Let $\eta_i(t) = \int_0^t \lambda_i(s, z_0, y)ds$.
- Generate ξ_l^i for $i = 1, 2, \ldots, m$ and $l = 1, 2, \ldots, K_i$ be an i.i.d. collection of random variables uniformly distributed on $[0, \eta_i(T)]$ and independent of all else. Let $t_l^i = \eta_i^{-1}(\xi_l^i)$. Set $I_i(t) = \sum_{l=1}^{K_i} 1_{[t_l^i, \infty)}(t)$.



Compute the Girsanov change of measure process $\tilde{L}(t)$:

$$\tilde{L}_j(t) = \left(\prod_{i \geq 1} \frac{a_j(\tilde{Z}(\tilde{T}_i^j-))}{\lambda_j(T_i^j-, \tilde{Z}_0, y)} 1_{\{\tilde{T}_i^j \leq t\}} \right) \times \exp\left(\int_0^t (\lambda_j(s, \tilde{Z}_0, y) - a_j(\tilde{Z}(s))\, ds \right), \quad \widetilde{L} = \prod_{j=1}^m \widetilde{L}_j,$$

where $\tilde{T}_i^j$ for $i = 1, 2, \ldots$ are the ith jump time of I_j.

Main Result : Rathinam and Yu

$$P\{Z(t) = z \,|\, O\} = \frac{\mathbb{E}_0[1_{\{z\}}(\tilde{Z}(t))\, \tilde{L}(T)\, W | \, O]}{\mathbb{E}_0[\tilde{L}(T)\, W | \, O]}$$

where $O = \{Y(T) = y\}$ is the observed event.

Numerical example: A pure death process

- $S \xrightarrow{c} \varnothing$. Propensity $a(x) = cx$. One species which is observed. One reaction channel which is not free. $R = R''$ and $R''(T) = y_0 - y$.
- Exact formula is known for $\lambda_j(t, z \,|y)$ and depends on time t.
- Approximate conditional propensity. Keep constant propensity between jump events.
- Total Variation Error (TVE) = $\mathbb{E}\left[\sum_z |\hat{\pi}(t,z) - \pi(t,z)| \,\middle|\, Y(T) = y\right]$.
- Effective Sample Fraction $= \frac{N_e}{N_s} = \frac{1}{N_s}\frac{\left(\sum_{i=1}^{N_s} W^i\right)^2}{\sum_{i=1}^{N_s}(W^i)^2}$.

Method	Observation	Total Variation Error (95% interval)	Effective Sample Fraction (ESF)	Runtime (seconds
Naive	$x_T = 368$	1.1353, [1.1077, 1.1630]	0.027	2.3
Targeting ($\overline{\lambda}_1$)	(common observation)	0.2037, [0.1995, 0.2080]	0.919	2.6
Targeting ($\overline{\lambda}_2$)		0.2072, [0.2029, 0.2116]	0.900	2.7
CP (approx)		0.3485, [0.3414, 0.3556]	0.322	2.4
CP (exact)		0.1957, [0.1920, 0.1994]	1.000	2.4
Naive	$x_T = 404$	1.9036, [1.8934, 1.9139]	0.002	2.4
Targeting ($\overline{\lambda}_1$)	(rare observation)	0.1979, [0.1942, 0.2016]	0.923	2.6
Targeting ($\overline{\lambda}_2$)		0.2297, [0.2248, 0.2347]	0.654	2.4
CP (approx)		0.3351, [0.3275, 0.3427]	0.324	2.1
CP (exact)		0.1909, [0.1872, 0.1947]	1.000	2.3

Table 1: Pure death process. We took $x_0 = 1000$, $c = 2$, $T = 0.5$, $t = 0.2$, $\Delta t = 0.02$. We used $N_r = 100$ trials, while each trial has filter sample size $N_s = 1000$. We note that in our experiment for the rare observation case, the naive method has only 80% of the trials provided meaningful results, and the rest of 20% runs of the naive filter failed when none of the sample landed at the target state.

Summary & open problems

Contributions

- First Monte Carlo particle filters in the reaction network setting for exact partial observation of a CTMC — continuous time (2021) and snapshots (2024).
- Targeting: free/slaved splitting + inhomogeneous-Poisson proposals + exact bridge $\Rightarrow$ less weight degeneracy.
- Same machinery gives Bayesian parameter posteriors.

Open directions

- Efficient scaling to many hidden coordinates.
- Parameter identifiability analysis.
- Colloborative study in systems biology and epidemiology.

Thank you!

Questions and comments welcome.

References

- M. Rathinam, M. Yu. *State and parameter estimation from exact partial state observation in stochastic reaction networks.* J. Chem. Phys. **154**, 034103 (2021).
- M. Rathinam, M. Yu, *Stochastic filtering of reaction networks partially observed in time snapshots.* J. Comput. Phys. **515**, 113265 (2024).
- D. T. Gillespie. *Exact stochastic simulation of coupled chemical reactions.* J. Phys. Chem. **81**, 2340 (1977).
- F. Confortola, M. Fuhrman. *Filtering of continuous-time Markov chains with noise-free observation and applications.* Stochastics **85**, 216 (2013).
- A. Golightly, C. Sherlock. *Efficient sampling of conditioned Markov jump processes.* Stat. Comput. **29**, 1149 (2019).
- D'Ambrosio, Fang, Gupta, Kumar, Khammash. *Filtered finite state projection.* SIAM J. Appl. Dyn. Syst. (2024).
- Fang, Gupta, Khammash. *Stochastic filtering for multiscale stochastic reaction networks.* J. Comput. Phys. **467**, 111441 (2022).

Back up: SEIR model (epidemic spread)

- Reaction network for the spread of an infectious disease:

$$S + I \xrightarrow{\beta} E + I, \qquad E \xrightarrow{\kappa} I, \qquad I \xrightarrow{\gamma} R.$$

- State $Z(t) = (\#S, \#E, \#R, \#I)$ for *susceptible* / *exposed* / *infected* / *recovered*. Only the infected population I is observed exactly; S, E are hidden.
- Propensities (mass action), with $N = z_1 + z_2 + z_3 + z_4$ the conserved total population:

$$a_1(z) = \frac{\beta z_1 z_4}{N}, \qquad a_2(z) = \kappa z_2, \qquad a_3(z) = \gamma z_4.$$

- **State estimation:** recover hidden S, E, R from the observed I-trajectory
- **Parameter estimation:** Bayesian inference of κ (reciprocal of the incubation period) under a uniform prior.

Back up: continuous-time filtering equations

$\pi(t,x)$ satisfies a nonlinear system of ODEs with jumps:

- Let $t_k, k = 1, 2, \ldots$ be the successive jump times of $y(t)$.
- On interval $[t_k, t_{k+1})$

$$\begin{aligned}\pi'(t,x) = &\sum_{j\in\mathcal{U}} \pi(t, x-\nu_j')\, a_j(x-\nu_j', y(t_k)) - \sum_{j\in\mathcal{U}} \pi(t,x)\, a_j(x, y(t_k)) \\ &- \pi(t,x)\left(a^{\mathcal{O}}(x, y(t_k)) - \sum_{\tilde{x}} a^{\mathcal{O}}(\tilde{x}, y(t_k))\pi(t,\tilde{x})\right) \quad \forall x \in \mathbb{Z}_+^{n_1}.\end{aligned} \tag{2}$$

- At jump times t_k

$$\pi(t_k, x) = \frac{\sum_{l\in\mathcal{O}_k} a_l(x-\nu_l', y(t_{k-1}))\, \pi(t_k-, x-\nu_l')}{\sum_{\tilde{x}} \sum_{l\in\mathcal{O}_k} a_l(\tilde{x}, y(t_{k-1}))\, \pi(t_k-, \tilde{x})} \quad \forall x \in \mathbb{Z}_+^{n_1}, \tag{3}$$

where $\mathcal{O}_k = \{j \in \mathcal{O} \mid \nu_j'' = y(t_k) - y(t_k-)\}$.

Issues:

- Hard to compute when state space of X is large or infinite.
- Nonlinear evolution does not correspond to a Markov process.

Back up: unnormalized conditional distribution $\rho(t, x)$

Define $\rho(t, x)$ which satisfies linear ODEs with jumps:

- On interval $[t_k, t_{k+1})$,

$$\rho'(t, x) = \sum_{j \in \mathcal{U}} \rho(t, x - \nu_j')\, a_j(x - \nu_j', y(t_k)) - \sum_{j \in \mathcal{U}} \rho(t, x)\, a_j(x, y(t_k)) \\ - \rho(t, x)\, a^{\mathcal{O}}(x, y(t_k)) \quad \forall x \in \mathbb{Z}_+^{n_1}. \tag{4}$$

- At jump times t_k,

$$\rho(t_k, x) = \frac{1}{|\mathcal{O}_k|} \sum_{j \in \mathcal{O}_k} a_j(x - \nu_j', y(t_{k-1}))\, \rho(t_k-, x - \nu_j'), \quad x \in \mathbb{Z}_+^{n_1}, \tag{5}$$

  where $\mathcal{O}_k = \{j \in \mathcal{O} \mid \nu_j'' = y(t_k) - y(t_k-)\}$.

Advantages of $\rho(t, x)$

- $\pi(t, x) = \rho(t, x) / \sum_{\tilde{x}} \rho(t, \tilde{x})$.
  $\rho(t, x)$ is called **unnormalized conditional distribution**.
- We can relate $\rho(t, x)$ to a Markov process.