Inference of dynamic systems from noisy and sparse data via manifold-constrained Gaussian processes

Shihao Yang[a], Samuel W. K. Wong[b], and S. C. Kou[c,1]

[a]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332; [b]Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada; and [c]Department of Statistics, Harvard University, Cambridge, MA 02138

Edited by Wing Hung Wong, Stanford University, Stanford, CA, and approved March 5, 2021 (received for review September 30, 2020)

Parameter estimation for nonlinear dynamic system models, represented by ordinary differential equations (ODEs), using noisy and sparse data, is a vital task in many fields. We propose a fast and accurate method, manifold-constrained Gaussian process inference (MAGI), for this task. MAGI uses a Gaussian process model over time series data, explicitly conditioned on the manifold constraint that derivatives of the Gaussian process must satisfy the ODE system. By doing so, we completely bypass the need for numerical integration and achieve substantial savings in computational time. MAGI is also suitable for inference with unobserved system components, which often occur in real experiments. MAGI is distinct from existing approaches as we provide a principled statistical construction under a Bayesian framework, which incorporates the ODE system through the manifold constraint. We demonstrate the accuracy and speed of MAGI using realistic examples based on physical experiments.

parameter estimation | ordinary differential equations | posterior sampling | inverse problem

Dynamic systems, represented as a set of ordinary differential equations (ODEs), are commonly used to model behaviors in scientific domains, such as gene regulation (1), biological rhythms (2), spread of disease (3), ecology (4), etc. We focus on models specified by a set of ODEs

$$\dot{\boldsymbol{x}}(t) = \frac{d\boldsymbol{x}(t)}{dt} = \mathbf{f}(\boldsymbol{x}(t), \boldsymbol{\theta}, t), \quad t \in [0, T], \qquad [1]$$

where the vector $\boldsymbol{x}(t)$ contains the system outputs that evolve over time t and $\boldsymbol{\theta}$ is the vector of model parameters to be estimated from experimental/observational data. When $\mathbf{f}$ is nonlinear, solving $\boldsymbol{x}(t)$ given initial conditions $\boldsymbol{x}(0)$ and $\boldsymbol{\theta}$ generally requires a numerical integration method, such as Runge–Kutta.

Historically, ODEs have mainly been used for conceptual or theoretical understanding rather than data fitting as experimental data were limited. Advances in experimental and data collection techniques have increased the capacity to follow dynamic systems closer to real time. Such data will generally be recorded at discrete times and subject to measurement error. Thus, we assume that we observe $\boldsymbol{y}(\boldsymbol{\tau}) = \boldsymbol{x}(\boldsymbol{\tau}) + \boldsymbol{\epsilon}(\boldsymbol{\tau})$ at a set of observation time points $\boldsymbol{\tau}$ with error $\boldsymbol{\epsilon}$ governed by noise level σ. Our focus here is inference of $\boldsymbol{\theta}$ given $\boldsymbol{y}(\boldsymbol{\tau})$, with emphasis on nonlinear $\mathbf{f}$ where specialized methods that exploit a linear structure (e.g., refs. 5 and 6), are not generally applicable. We shall present a coherent, statistically principled framework for dynamic system inference with the help of Gaussian processes (GPs). The key to our method is to restrict the GPs on a manifold that satisfies the ODE system: Thus, we name our method MAGI (manifold-constrained Gaussian process inference). Placing a GP on $\boldsymbol{x}(t)$ facilitates inference of $\boldsymbol{\theta}$ without numerical integration, and our explicit manifold constraint is the key idea that addresses the conceptual incompatibility between the GP and the specification of the ODE model, as we shall discuss shortly when overviewing our method. We show that the resulting parameter inference is computationally efficient, statistically principled, and effective in a variety of practical scenarios. MAGI particularly works in the cases when some system component(s) is/are unobserved. To the best of our knowledge, none of the current available software packages that do not use numerical integration can analyze systems with unobserved component(s).

Overview of Our Method

Following the Bayesian paradigm, we view the D-dimensional system $\boldsymbol{x}(t)$ to be a realization of the stochastic process $\boldsymbol{X}(t) = (X_1(t), \ldots, X_D(t))$ and the model parameters $\boldsymbol{\theta}$ a realization of the random variable $\boldsymbol{\Theta}$. In Bayesian statistics, the basis of inference is the posterior distribution, obtained by combining the likelihood function with a chosen prior distribution on the unknown parameters and stochastic processes. Specifically, we impose a general prior distribution $\pi(\cdot)$ on $\boldsymbol{\theta}$ and independent GP prior distributions on each component $X_d(t)$ so that $X_d(t) \sim \mathcal{GP}(\mu_d, \mathcal{K}_d),\ t \in [0, T]$, where $\mathcal{K}_d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a positive definite covariance kernel for the GP and $\mu_d : \mathbb{R} \to \mathbb{R}$ is the mean function. Then, for any finite set of time points $\boldsymbol{\tau}_d$, $X_d(\boldsymbol{\tau}_d)$ has a multivariate Gaussian distribution with mean vector $\mu_d(\boldsymbol{\tau}_d)$ and covariance matrix $\mathcal{K}_d(\boldsymbol{\tau}_d, \boldsymbol{\tau}_d)$. Denote the observations by $\boldsymbol{y}(\boldsymbol{\tau}) = (\boldsymbol{y}_1(\boldsymbol{\tau}_1), \ldots, \boldsymbol{y}_D(\boldsymbol{\tau}_D))$, where $\boldsymbol{\tau} = (\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_D)$ is the collection of all observation time points, and each component X_d can have its own set of observation times $\boldsymbol{\tau}_d = (\tau_{d,1}, \ldots, \tau_{d,N_d})$. If the dth component is not observed, then $N_d = 0$, and $\boldsymbol{\tau}_d = \emptyset$. $N = N_1 + \cdots + N_D$ is the total number of observations. We note

Significance

Ordinary differential equations are a ubiquitous tool for modeling behaviors in science, such as gene regulation, biological rhythms, epidemics, and ecology. An important problem is to infer and characterize the uncertainty of parameters that govern equations. We present an accurate and fast inference method using manifold-constrained Gaussian processes, such that derivatives of the Gaussian process must satisfy the dynamics of the differential equations. Our method completely avoids the use of numerical integration and is thus fast to compute. Our construction is embedded in a principled statistical framework and is demonstrated to yield fast and reliable inference in a variety of practical problems. Our method works even when some system components are unobserved, which is a significant challenge for previous methods.

Author contributions: S.Y., S.W.K.W., and S.C.K. designed research; S.Y., S.W.K.W., and S.C.K. performed research; S.Y. and S.W.K.W. contributed new reagents/analytic tools; S.Y. and S.W.K.W. analyzed data; and S.Y., S.W.K.W., and S.C.K. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: kou@stat.harvard.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2020397118/-/DCSupplemental.

Published April 9, 2021.

that for the remainder of the paper, the notation t shall refer to time generically, while τ shall refer specifically to the observation time points.

As an illustrative example, consider the dynamic system in ref. 1 that governs the oscillation of Hes1 mRNA (messenger ribonucleic acid) (M) and Hes1 protein (P) levels in cultured cells, where it is postulated that an Hes1-interacting (H) factor contributes to a stable oscillation, a manifestation of biological rhythm (2). The ODEs of the three-component system $X = (P, M, H)$ are

$$\mathbf{f}(X, \boldsymbol{\theta}, t) = \begin{pmatrix} -aPH + bM - cP \\ -dM + \frac{e}{1+P^2} \\ -aPH + \frac{f}{1+P^2} - gH \end{pmatrix},$$

where $\boldsymbol{\theta} = (a, b, c, d, e, f, g)$ are the associated parameters. In Fig. 1, left-most panel, we show noise-contaminated data generated from the system, which closely mimics the experimental setup described in ref. 1: P and M are observed at 15-min intervals for 4 h, but H is never observed. In addition, P and M observations are asynchronous: Starting at time 0, every 15 min we observe P; starting at 7.5 min, every 15 min we observe M; P and M are never observed at the same time. It can be seen that the mRNA and protein levels exhibit the behavior of regulation via negative feedback.

The goal here is to infer the seven parameters of the system: a, b govern the rate of protein synthesis in the presence of the interacting factor; c, d, g are the rates of decomposition; and e, f are inhibition rates. The unobserved H component poses a challenge for most existing methods that do not use numerical integration but is capably handled by MAGI: The P and M panels of Fig. 1 show that our inferred trajectories provide good fits to the observed data, and the H panel shows that the dynamics of the entirely unobserved H component are largely recovered as well. We emphasize that these trajectories are inferred without any use of numerical solvers. We shall return to the Hes1 example in detail in *Results*.

Intuitively, the GP prior on $\boldsymbol{X}(t)$ facilitates computation as GP provides closed analytical forms for $\dot{\boldsymbol{X}}(t)$ and $\boldsymbol{X}(t)$, which could bypass the need for numerical integration. In particular, with a GP prior on $\boldsymbol{X}(t)$, the conditional distribution of $\dot{\boldsymbol{X}}(t)$ given $\boldsymbol{X}(t)$ is also a GP with its mean function and covariance kernel completely specified. This GP specification for the derivatives $\dot{\boldsymbol{x}}(t)$, however, is inherently incompatible with the ODE model because Eq. **1** also completely specifies $\dot{\boldsymbol{x}}(t)$ given $\boldsymbol{x}(t)$ (via the function $\mathbf{f}$). As a key contribution of our method, MAGI addresses this conceptual incompatibility by constraining the GP to satisfy the ODE model in Eq. **1**. To do so, we first define a random variable W quantifying the difference between stochastic process $\boldsymbol{X}(t)$ and the ODE structure with a given value of the parameter $\boldsymbol{\theta}$:

$$W = \sup_{t\in[0,T], d\in\{1,\dots,D\}} |\dot{X}_d(t) - \mathbf{f}(\boldsymbol{X}(t), \boldsymbol{\theta}, t)_d|. \qquad [2]$$

$W \equiv 0$ if and only if ODEs with parameter $\boldsymbol{\theta}$ are satisfied by $\boldsymbol{X}(t)$. Therefore, ideally the posterior distribution for $\boldsymbol{X}(t)$ and $\boldsymbol{\theta}$ given the observations $\boldsymbol{y}(\boldsymbol{\tau})$ and the ODE constraint, $W \equiv 0$, is (informally)

$$p_{\boldsymbol{\Theta}, \boldsymbol{X}(t)\,|\,W, \boldsymbol{Y}(\boldsymbol{\tau})}(\boldsymbol{\theta}, \boldsymbol{x}(t)\,|\,W = 0, \boldsymbol{Y}(\boldsymbol{\tau}) = \boldsymbol{y}(\boldsymbol{\tau})). \qquad [3]$$

While Eq. **3** is the ideal posterior, in reality W is not generally computable. In practice, we approximate W by finite discretization on the set $\boldsymbol{I} = (t_1, t_2, \dots, t_n)$ such that $\boldsymbol{\tau} \subset \boldsymbol{I} \subset [0, T]$ and similarly define $W_{\boldsymbol{I}}$ as

$$W_{\boldsymbol{I}} = \max_{t\in\boldsymbol{I}, d\in\{1,\dots,D\}} |\dot{X}_d(t) - \mathbf{f}(\boldsymbol{X}(t), \boldsymbol{\theta}, t)_d|. \qquad [4]$$

Note that $W_{\boldsymbol{I}}$ is the maximum of a finite set, and $W_{\boldsymbol{I}} \to W$ monotonically as $\boldsymbol{I}$ becomes dense in $[0, T]$. Therefore, the practically computable posterior distribution is

$$p_{\boldsymbol{\Theta}, \boldsymbol{X}(\boldsymbol{I})\,|\,W_{\boldsymbol{I}}, \boldsymbol{Y}(\boldsymbol{\tau})}(\boldsymbol{\theta}, \boldsymbol{x}(\boldsymbol{I})\,|\,W_{\boldsymbol{I}} = 0, \boldsymbol{Y}(\boldsymbol{\tau}) = \boldsymbol{y}(\boldsymbol{\tau})),$$

which is the joint conditional distribution of $\boldsymbol{\theta}$ and $\boldsymbol{X}(\boldsymbol{I})$ together. Thus, effectively, we simultaneously infer both the parameters and the unobserved trajectory $\boldsymbol{X}(\boldsymbol{I})$ from the noisy observations $\boldsymbol{y}(\boldsymbol{\tau})$.

Under Bayes' rule, we have

$$\begin{aligned} &p_{\boldsymbol{\Theta}, \boldsymbol{X}(\boldsymbol{I})\,|\,W_{\boldsymbol{I}}, \boldsymbol{Y}(\boldsymbol{\tau})}(\boldsymbol{\theta}, \boldsymbol{x}(\boldsymbol{I})\,|\,W_{\boldsymbol{I}} = 0, \boldsymbol{Y}(\boldsymbol{\tau}) = \boldsymbol{y}(\boldsymbol{\tau})) \\ &\propto P(\boldsymbol{\Theta} = \boldsymbol{\theta}, \boldsymbol{X}(\boldsymbol{I}) = \boldsymbol{x}(\boldsymbol{I}),\, W_{\boldsymbol{I}} = 0, \boldsymbol{Y}(\boldsymbol{\tau}) = \boldsymbol{y}(\boldsymbol{\tau})), \end{aligned}$$

where the right-hand side can be decomposed as

$$\begin{aligned} &P(\boldsymbol{\Theta} = \boldsymbol{\theta}, \boldsymbol{X}(\boldsymbol{I}) = \boldsymbol{x}(\boldsymbol{I}),\, W_{\boldsymbol{I}} = 0, \boldsymbol{Y}(\boldsymbol{\tau}) = \boldsymbol{y}(\boldsymbol{\tau})) \\ &= \pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \times \underbrace{P(\boldsymbol{X}(\boldsymbol{I}) = \boldsymbol{x}(\boldsymbol{I}) | \boldsymbol{\Theta} = \boldsymbol{\theta})}_{(1)} \\ &\quad \times \underbrace{P(\boldsymbol{Y}(\boldsymbol{\tau}) = \boldsymbol{y}(\boldsymbol{\tau}) | \boldsymbol{X}(\boldsymbol{I}) = \boldsymbol{x}(\boldsymbol{I}), \boldsymbol{\Theta} = \boldsymbol{\theta})}_{(2)} \\ &\quad \times \underbrace{P(\,W_{\boldsymbol{I}} = 0 | \boldsymbol{Y}(\boldsymbol{\tau}) = \boldsymbol{y}(\boldsymbol{\tau}), \boldsymbol{X}(\boldsymbol{I}) = \boldsymbol{x}(\boldsymbol{I}), \boldsymbol{\Theta} = \boldsymbol{\theta})}_{(3)}. \end{aligned}$$

The first term (1) can be simplified as $P(\boldsymbol{X}(\boldsymbol{I}) = \boldsymbol{x}(\boldsymbol{I}) | \boldsymbol{\Theta} = \boldsymbol{\theta}) = P(\boldsymbol{X}(\boldsymbol{I}) = \boldsymbol{x}(\boldsymbol{I}))$ due to the prior independence of $\boldsymbol{X}(\boldsymbol{I})$ and $\boldsymbol{\Theta}$;

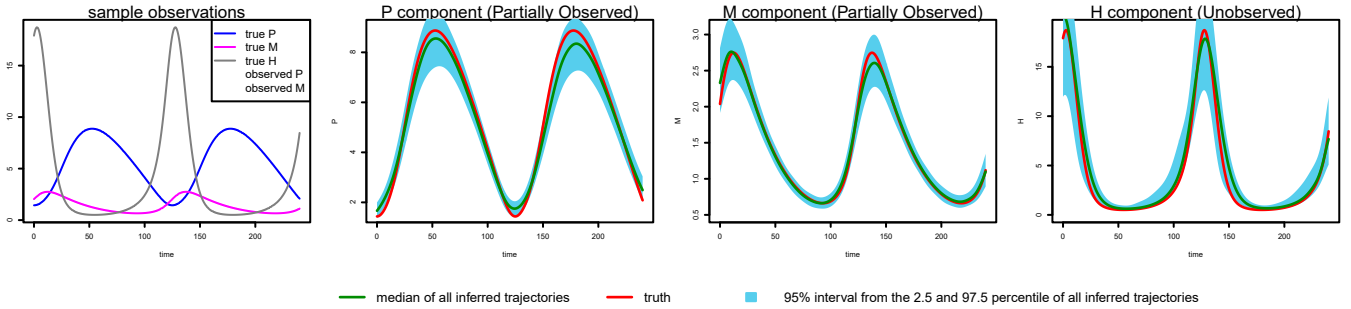


Fig. 1. Inference by MAGI for Hes1 partially observed asynchronous system on 2,000 simulated datasets. The red curve is the truth. MAGI recovers the system well, without the usage of any numerical solver: The green curve shows the median of the inferred trajectories among the 2,000 simulated datasets, and a 95% interval from the 2.5 and 97.5% of all inferred trajectories is shown via the blue area.

it corresponds to the GP prior on $\boldsymbol{X}$. The second term (2) corresponds to the noisy observations. The third term (3) can be simplified as

$$\begin{aligned}
&P(\boldsymbol{W_I}=0|\boldsymbol{Y}(\boldsymbol{\tau})=\boldsymbol{y}(\boldsymbol{\tau}),\boldsymbol{X}(\boldsymbol{I})=\boldsymbol{x}(\boldsymbol{I}),\boldsymbol{\Theta}=\boldsymbol{\theta})\\
&=P(\dot{\boldsymbol{X}}(\boldsymbol{I})-\mathbf{f}(\boldsymbol{x}(\boldsymbol{I}),\boldsymbol{\theta},t_{\boldsymbol{I}})=\mathbf{0}|\boldsymbol{Y}(\boldsymbol{\tau})=\boldsymbol{y}(\boldsymbol{\tau}),\boldsymbol{X}(\boldsymbol{I})=\boldsymbol{x}(\boldsymbol{I}),\boldsymbol{\Theta}=\boldsymbol{\theta})\\
&=P(\dot{\boldsymbol{X}}(\boldsymbol{I})-\mathbf{f}(\boldsymbol{x}(\boldsymbol{I}),\boldsymbol{\theta},t_{\boldsymbol{I}})=\mathbf{0}|\boldsymbol{X}(\boldsymbol{I})=\boldsymbol{x}(\boldsymbol{I}))\\
&=P(\dot{\boldsymbol{X}}(\boldsymbol{I})=\mathbf{f}(\boldsymbol{x}(\boldsymbol{I}),\boldsymbol{\theta},t_{\boldsymbol{I}})|\boldsymbol{X}(\boldsymbol{I})=\boldsymbol{x}(\boldsymbol{I})),
\end{aligned}$$

which is the conditional density of $\dot{\boldsymbol{X}}(\boldsymbol{I})$ given $\boldsymbol{X}(\boldsymbol{I})$ evaluated at $\mathbf{f}(\boldsymbol{x}(\boldsymbol{I}),\boldsymbol{\theta},t_{\boldsymbol{I}})$. All three terms are multivariate Gaussian: The third term is Gaussian because $\dot{\boldsymbol{X}}(\boldsymbol{I})$ given $\boldsymbol{X}(\boldsymbol{I})$ has a multivariate Gaussian distribution as long as the kernel $\mathcal{K}$ is twice differentiable.

Therefore, the practically computable posterior distribution simplifies to

$$\begin{aligned}
&p_{\boldsymbol{\Theta},\boldsymbol{X}(\boldsymbol{I})|\boldsymbol{W_I},\boldsymbol{Y}(\boldsymbol{\tau})}(\boldsymbol{\theta},\boldsymbol{x}(\boldsymbol{I})|\,\boldsymbol{W_I}=0,\boldsymbol{Y}(\boldsymbol{\tau})=\boldsymbol{y}(\boldsymbol{\tau})) \qquad \textbf{[5]}\\
&\propto\pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\Bigg[\right.\\
&+\underbrace{|\boldsymbol{I}|\log(2\pi)+\log|C_d|+\|x_d(\boldsymbol{I})-\mu_d(\boldsymbol{I})\|^2_{C_d^{-1}}}_{(1)}\\
&+\underbrace{|\boldsymbol{I}|\log(2\pi)+\log|K_d|+\left\|\mathbf{f}^{\boldsymbol{x},\boldsymbol{\theta}}_{d,\boldsymbol{I}}-\dot{\mu}_d(\boldsymbol{I})-m_d\{x_d(\boldsymbol{I})-\mu_d(\boldsymbol{I})\}\right\|^2_{K_d^{-1}}}_{(3)}\\
&\left.+\underbrace{N_d\log(2\pi\sigma_d^2)+\|x_d(\boldsymbol{\tau}_d)-y_d(\boldsymbol{\tau}_d)\|^2_{\sigma_d^{-2}}}_{(2)}\Bigg]\right\},
\end{aligned}$$

where $\|\boldsymbol{v}\|_A^2=\boldsymbol{v}^{\intercal}A\boldsymbol{v}$, $|\boldsymbol{I}|$ is the cardinality of $\boldsymbol{I}$, $\mathbf{f}^{\boldsymbol{x},\boldsymbol{\theta}}_{d,\boldsymbol{I}}$ is short for the dth component of $\mathbf{f}(\boldsymbol{x}(\boldsymbol{I}),\boldsymbol{\theta},t_{\boldsymbol{I}})$, and the multivariate Gaussian covariance matrix C_d and the matrix K_d can be derived as follows for each component d:

$$\begin{cases} C &= \mathcal{K}(\boldsymbol{I},\boldsymbol{I})\\ m &= {}'\mathcal{K}(\boldsymbol{I},\boldsymbol{I})\mathcal{K}(\boldsymbol{I},\boldsymbol{I})^{-1}\\ K &= \mathcal{K}''(\boldsymbol{I},\boldsymbol{I})-{}'\mathcal{K}(\boldsymbol{I},\boldsymbol{I})\mathcal{K}(\boldsymbol{I},\boldsymbol{I})^{-1}\mathcal{K}'(\boldsymbol{I},\boldsymbol{I}) \end{cases}, \qquad \textbf{[6]}$$

where ${}'\mathcal{K}=\frac{\partial}{\partial s}\mathcal{K}(s,t)$, $\mathcal{K}'=\frac{\partial}{\partial t}\mathcal{K}(s,t)$, and $\mathcal{K}''=\frac{\partial^2}{\partial s\partial t}\mathcal{K}(s,t)$.

In practice, we choose the Matern kernel $\mathcal{K}(s,t)=\phi_1\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{l}{\phi_2}\right)^{\nu}B_\nu\left(\sqrt{2\nu}\frac{l}{\phi_2}\right)$, where $l=|s-t|$, Γ is the Gamma function, B_ν is the modified Bessel function of the second kind, and the degree of freedom ν is set to be 2.01 to ensure that the kernel is twice differentiable. $\mathcal{K}$ has two hyperparameters ϕ_1 and ϕ_2. Their meaning and specification are discussed in *Materials and Methods*.

With the posterior distribution specified in Eq. **5**, we use Hamiltonian Monte Carlo (HMC) (7) to obtain samples of $\boldsymbol{X_I}$ and the parameters together. At the completion of HMC sampling, we take the posterior mean of $\boldsymbol{X_I}$ as the inferred trajectory and the posterior means of the sampled parameters as the parameter estimates. Throughout the MAGI computation, no numerical integration is ever needed.

Review of Related Work

The problem of dynamic system inference has been studied in the literature, which we now briefly review. We first note that a simple approach to constructing the "ideal" likelihood function is according to $p(\boldsymbol{y}(\boldsymbol{t})|\hat{\boldsymbol{x}}(\boldsymbol{t},\boldsymbol{\theta},\boldsymbol{x}(0)),\sigma)$, where $\hat{\boldsymbol{x}}(\boldsymbol{t},\boldsymbol{\theta},\boldsymbol{x}(0))$ is the numerical solution of the ODE obtained by numerical integration given $\boldsymbol{\theta}$ and the initial conditions. This approach suffers from a high computational burden: Numerical integration is required for every $\boldsymbol{\theta}$ sampled in an optimization or Markov chain Monte Carlo (MCMC) routine (8). Smoothing methods have been useful for eliminating the dependence on numerical ODE solutions, and an innovative penalized likelihood approach (9) uses a B-spline basis for constructing estimated functions to simultaneously satisfy the ODE system and fit the observed data. While in principle, the method in ref. 9 can handle an unobserved system component, substantive manual input is required as we show in *Results*, which contrasts with the ready-made solution that MAGI provides.

As an alternative to the penalized likelihood approach, GPs are a natural candidate for fulfilling the smoothing role in a Bayesian paradigm due to their flexibility and analytic tractability (10). The use of GPs to approximate the dynamic system and facilitate computation has been previously studied by a number of authors (8, 11–15). The basic idea is to specify a joint GP over $\boldsymbol{y},\boldsymbol{x},\dot{\boldsymbol{x}}$ with hyperparameters ϕ and then, provide a factorization of the joint density $p(\boldsymbol{y},\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{\theta},\phi,\sigma)$ that is suitable for inference. The main challenge is to find a coherent way to combine information from two distinct sources: the approximation to the system by the GP governed by hyperparameters ϕ and the actual dynamic system equations governed by parameters $\boldsymbol{\theta}$. In refs. 8 and 11, the factorization proposed is $p(\boldsymbol{y},\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{\theta},\phi,\sigma)=p(\boldsymbol{y}|\boldsymbol{x},\sigma)p(\dot{\boldsymbol{x}}|\boldsymbol{x},\boldsymbol{\theta},\phi)p(\boldsymbol{x}|\phi)p(\phi)p(\boldsymbol{\theta})$, where $p(\boldsymbol{y}|\boldsymbol{x},\sigma)$ comes from the observation model and $p(\boldsymbol{x}|\phi)$ comes from the GP prior as in our approach. However, there are significant conceptual difficulties in specifying $p(\dot{\boldsymbol{x}}|\boldsymbol{x},\boldsymbol{\theta},\phi)$: On one hand, the distribution of $\dot{\boldsymbol{x}}$ is completely determined by the GP given $\boldsymbol{x}$, while on the other hand, $\dot{\boldsymbol{x}}$ is completely specified by the ODE system $\dot{\boldsymbol{x}}=\mathbf{f}(\boldsymbol{x},\boldsymbol{\theta},t)$; these two are incompatible. Previous authors have attempted to circumvent this incompatibility of the GP and ODE system: Refs. 8 and 11 use a product of experts heuristic by letting $p(\dot{\boldsymbol{x}}|\boldsymbol{x},\boldsymbol{\theta},\phi)\propto p(\dot{\boldsymbol{x}}|\boldsymbol{x},\phi)p(\dot{\boldsymbol{x}}|\boldsymbol{x},\boldsymbol{\theta})$, where the two distributions in the product come from the GP and a noisy version of the ODE, respectively. In ref. 15, the authors arrive at the same posterior as refs. 8 and 11 by assuming an alternative graphical model that bypasses the product of experts heuristic; nonetheless, the method requires working with an artificial noisy version of the ODE. In ref. 12, the authors start with a different factorization: $p(\boldsymbol{y},\boldsymbol{x},\dot{\boldsymbol{x}},\boldsymbol{\theta},\phi,\sigma)=p(\boldsymbol{y}|\dot{\boldsymbol{x}},\phi,\sigma)p(\dot{\boldsymbol{x}}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\phi)p(\phi)p(\boldsymbol{\theta})$, where $p(\boldsymbol{y}|\dot{\boldsymbol{x}},\phi)$ and $p(\boldsymbol{x}|\phi)$ are given by the GP and $p(\dot{\boldsymbol{x}}|\boldsymbol{x},\boldsymbol{\theta})$ is a Dirac delta distribution given by the ODE. However, this factorization is incompatible with the observation model $p(\boldsymbol{y}|\boldsymbol{x},\sigma)$ as discussed in detail in ref. 16. There is other related work that uses GPs in an ad hoc partial fashion to aid inference. In ref. 13, GP regression is used to obtain the means of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ for embedding within an Approximate Bayesian Computation estimation procedure. In ref. 14, GP smoothing is used during an initial burn-in phase as a proxy for the likelihood, before switching to the ideal likelihood to obtain final MCMC samples. While empirical results from the aforementioned studies are promising, a principled statistical framework for inference that addresses the previously noted conceptual incompatibility between the GP and ODE specifications is lacking. Our work presents one such principled statistical framework through the explicit manifold constraint. MAGI is therefore distinct from recent GP-based approaches (11, 15) or any other Bayesian analogs of ref. 9.

In addition to the conceptual incompatibility, none of the existing methods that do not use numerical integration offer a practical solution for a system with unobserved component(s), which highlights another unique and important contribution of our approach.

Results

We apply MAGI to three systems. We begin with an illustration that demonstrates the effectiveness of MAGI in practical problems with unobserved system component(s). Then, we make comparisons with other current methods on two benchmark systems, which show that our proposed method provides more accurate inference while having much faster run time.

Illustration: Hes1 Model. The Hes1 model described in the Introduction demonstrates inference on a system with an unobserved component and asynchronous observation times. This section continues the inference of this model. Ref. 1 studied the theoretical oscillation behavior using parameter values $a = 0.022$, $b = 0.3$, $c = 0.031$, $d = 0.028$; $e = 0.5$, $f = 20$, $g = 0.3$, which leads to one oscillation cycle approximately every 2 h. Ref. 1 also set the initial condition at the lowest value of P when the system is in oscillation equilibrium (1): $P = 1.439$, $M = 2.037$, $H = 17.904$. The noise level in our simulation is derived from ref. 1 where the SE based on repeated measures is reported to be around 15% of the P (protein) level and M (mRNA) level, so we set the simulation noise to be multiplicative following a log-normal distribution with SD 0.15; throughout this example, we assume the noise level σ is known to be 0.15 from repeated measures reported in ref. 1. The H component is never observed. Owing to the multiplicative error on the strictly positive system, we apply our method to the log-transformed ODEs, so that the resulting error distributions are Gaussian. To the best of our knowledge, MAGI is the only one that provides a practical and complete solution for handling unobserved component cases like this example.

We generate 2,000 simulated datasets based on the above setup for the Hes1 system. The left-most panel in Fig. 1 shows one example dataset. For each dataset, we use MAGI to infer the trajectories and estimate the parameters. We use the posterior mean of $X_t = (P, M, H)_t$ as the inferred trajectories for the system components, which are generated by MAGI without using any numerical solver. Fig. 1 summarizes the inferred trajectories across the 2,000 simulated datasets, showing the median of the inferred trajectories of X_t together with the 95% interval of the inferred trajectories represented by the 2.5 and 97.5% percentiles. The posterior mean of $\boldsymbol{\theta} = (a, b, c, d, f, e, g)$ is our estimate of the parameters. Table 1 summarizes the parameter estimates across the 2,000 simulated datasets, by showing their means and SDs. Fig. 1 shows that MAGI recovers the system well, including the completely unobserved H component. Table 1 shows that MAGI also recovers the system parameters well, except for the parameters that only appear in the equation for the unobserved H component, which we will discuss shortly. Together, Fig. 1 and Table 1 demonstrate that MAGI can recover the entire system without any usage of a numerical solver, even in the presence of unobserved component(s).

Table 1. Parameter inference in the Hes1 partially observed asynchronous system based on 2,000 simulation datasets

θ	Truth	MAGI		Ref. 9	
		Estimate	RMSE	Estimate	RMSE
a	0.022	0.021 ± 0.003	**0.003**	0.027 ± 0.026	0.026
b	0.3	0.329 ± 0.051	**0.059**	0.302 ± 0.086	0.086
c	0.031	0.035 ± 0.006	**0.007**	0.031 ± 0.010	0.010
d	0.028	0.029 ± 0.002	**0.003**	0.028 ± 0.003	**0.003**
e	0.5	0.552 ± 0.074	0.090	0.498 ± 0.088	**0.088**
f	20	13.759 ± 3.026	**6.936**	604.9 ± 5084.8	5,117.0
g	0.3	0.141 ± 0.026	**0.162**	1.442 ± 9.452	9.519

Average parameter estimates based on MAGI and ref. 9 across the 2,000 simulated datasets are reported together with the SD. Parameter RMSEs are reported in the following column. Bold highlights the best method in terms of parameter RMSE for each parameter.

Metrics for assessing the quality of system recovery. To further assess the quality of the parameter estimates and the system recovery, we consider two metrics. First, as shown in Table 1, we examine the accuracy of the parameter estimates by directly calculating the root mean squared error (RMSE) of the parameter estimates to the true parameter value. We call this measure the parameter RMSE metric. Second, it is possible that a system might be insensitive to some of the parameters; in the extreme case, some parameters may not be fully identifiable given only the observed data and components. In these situations, it is possible that the system trajectories implied by quite distinct parameter values are similar to each other (or even close to the true trajectory). We thus consider an additional trajectory RMSE metric to account for possible parameter insensitivity and measure how well the system components are recovered given the parameter and initial condition estimates. The trajectory RMSE is obtained by treating the numerical ODE solution based on the true parameter value as the ground truth: First, the numerical solver is used to reconstruct the trajectory based on the estimates of the parameter and initial condition (from a given method); then, we calculate the RMSE of this reconstructed trajectory to the true trajectory at the observation time points. We emphasize that the trajectory RMSE metric is only for evaluation purpose to assess (and compare across methods) how well a method recovers the trajectories of the system components and that throughout MAGI, no numerical solver is ever needed.

We summarize the trajectory RMSEs of MAGI in Table 2 for the Hes1 system.

We compare MAGI with the benchmark provided by the B spline-based penalization approach of ref. 9. To the best of our knowledge, among all of the existing methods that do not use numerical integration, ref. 9 is the only one with a software package that can be manually adapted to handle an unobserved component. We note, however, that this package itself is not ready made for this problem: It requires substantial manual input as it does not have default or built-in setup of its hyperparameters for the unobserved component. None of the other benchmark methods, including refs. 11 and 15, provide software that is equipped to handle an unobserved component. Table 1 compares our estimates against those given by ref. 9 based on the parameter RMSE, which shows that the parameter RMSEs for MAGI are substantially smaller than ref. 9. Fig. 1 shows that the inferred trajectories from MAGI are very close to the truth. On the contrary, the method in ref. 9 is not able to recover the unobserved component H nor the associated parameters f and g; *SI Appendix*, Fig. S1 has the plots. Table 2 compares the trajectory RMSE of the two methods. It is seen that the trajectory RMSE of MAGI is substantially smaller than that of ref. 9. Further implementation details and comparison are provided in *SI Appendix*.

Finally, we note that MAGI recovers the unobserved component H almost as well as the observed components of P and M, as measured by the trajectory RMSEs. In comparison, for the result of ref. 9 in Table 2, the trajectory RMSE of the unobserved H component is orders of magnitude worse than those of P and M. The numerical results thus illustrate the effectiveness of MAGI in borrowing information from the observed components to infer the unobserved component, which is made possible by explicitly conditioning on the ODE structure. The self-regulating parameter g and inhibition rate parameter f for the unobserved component appear to have high inference variation across the simulated datasets despite the small trajectory RMSEs. This suggests that the system itself could be insensitive to f and g when the H component is unobserved.

Table 2. Trajectory RMSEs of the individual components in the Hes1 system, comparing the average trajectory RMSEs of MAGI and ref. 9 over the 2,000 simulated datasets

Method	*P*	*M*	*H*
MAGI	**0.97**	**0.21**	**2.57**
Ref. 9	1.30	0.40	59.47

The best trajectory RMSE for each system component is shown in bold.

Comparison with Previous Methods Based on GPs. To further assess MAGI, we compare with two methods: adaptive gradient matching (AGM) of ref. 11 and fast Gaussian process-based gradient matching (FGPGM) of ref. 15, representing the state of the art of inference methods based on GPs. For fair comparison, we use the same benchmark systems, scripts, and software provided by the authors for performance assessment and run the software using the settings recommended by the authors. The benchmark systems include the FitzHugh–Nagumo (FN) equations (17) and a protein transduction model (18).

FN model. The FN equations are a classic ion channel model that describes spike potentials. The system consists of $X=(V,R)$, where V is the variable defining the voltage of the neuron membrane potential and R is the recovery variable from neuron currents, satisfying the ODE

$$\mathbf{f}(X,\boldsymbol{\theta},t)=\begin{pmatrix} c(V-\frac{V^3}{3}+R) \\ -\frac{1}{c}(V-a+bR) \end{pmatrix},$$

where $\boldsymbol{\theta}=(a,b,c)$ are the associated parameters. As in refs. 11 and 15, the true parameters are set to $a=0.2$, $b=0.2$, $c=3$, and we generate the true trajectories for this model using a numerical solver with initial conditions $V=-1$, $R=1$.

To compare MAGI with FGPGM of ref. 15 and AGM of ref. 11, we simulated 100 datasets under the noise setting of $\sigma_V=\sigma_R=0.2$ with 41 observations. The noise level is chosen to be on similar magnitude with that of ref. 15, and the noise level is set to be the same across the two components as the implementation of ref. 11 can only handle equal-variance noise. The number of repetitions (i.e., 100) is set to be the same as ref. 15 due to the high computing time of these alternative methods.

The parameter estimation results from the three methods are summarized in Table 3, where MAGI has the lowest parameter RMSEs among the three. Fig. 2 shows the inferred trajectories obtained by our method: MAGI recovers the system well, and the 95% interval band is so narrow around the truth that we can only see the band clearly after magnification (as shown in Fig. 2, *Insets*). *SI Appendix* provides visual comparison of the inferred trajectories of different methods, where MAGI gives the most consistent results across the simulations. Furthermore, to assess how well the methods recover the system components, we calculated the trajectory RMSEs, and the results are summarized in Table 4, where MAGI significantly outperforms the others, reducing the trajectory RMSE over the best alternative method for 60% in V and 25% in R. We note that compared with the true parameter value, all three methods show some bias in the parameter estimates, which is partly due to the GP prior as discussed in ref. 15, and MAGI appears to have the smallest bias.

For computing cost, the average run time of MAGI for this system over the repetitions is 3 min, which is 145 times faster than FGPGM (15) and 90 times faster than AGM (11) on the same processor (we follow the authors' recommendation for running their methods) (*SI Appendix* has details).

Protein transduction model. This protein transduction example is based on systems biology where components S and S_d represent a signaling protein and its degraded form, respectively. In the biochemical reaction, S binds to protein R to form the complex S_R, which enables the activation of R into R_{pp}. $X=(S,S_d,R,S_R,R_{pp})$ satisfies the ODE

$$\mathbf{f}(X,\boldsymbol{\theta},t)=\begin{pmatrix} -k_1\cdot S-k_2\cdot S\cdot R+k_3\cdot S_R \\ k_1\cdot S \\ -k_2\cdot S\cdot R+k_3\cdot S_R+\frac{V\cdot R_{pp}}{K_m+R_{pp}} \\ k_2\cdot S\cdot R-k_3\cdot S_R-k_4\cdot S_R \\ k_4\cdot S_R-\frac{V\cdot R_{pp}}{K_m+R_{pp}} \end{pmatrix},$$

where $\boldsymbol{\theta}=(k_1,k_2,k_3,k_4,V,K_m)$ are the associated rate parameters.

We follow the same simulation setup as refs. 11 and 15 by taking $t=\{0,1,2,4,5,7,10,15,20,30,40,50,60,80,100\}$ as the observation times, $X(0)=(1,0,1,0,0)$ as the initial values, and $\boldsymbol{\theta}=(0.07,0.6,0.05,0.3,0.017,0.3)$ as the true parameter values. Two scenarios for additive observation noise are considered: $\sigma=0.001$ (low noise) and $\sigma=0.01$ (high noise). Note that the observation times are unequally spaced, with only a sparse number of observations from $t=20$ to $t=100$. Further, inference for this system has been noted to be challenging due to the nonidentifiability of the parameters, in particular K_m and V (15). Therefore, the parameter RMSE is not meaningful for this system, and we focus our comparison on the trajectory RMSE.

We compare MAGI with FGPGM of ref. 15 and AGM of ref. 11 on 100 simulated datasets for each noise setting (*SI Appendix* has method and implementation details). We plot the inferred trajectories of MAGI in the high-noise setting in Fig. 3, which closely recover the system. The 95% interval band from MAGI is quite narrow that for most of the inferred components, we need magnifications (as shown in Fig. 3, *Insets*) to clearly see the 95% band. We then calculated the trajectory RMSEs, and the results are summarized in Table 5 for each system component. In both noise settings, MAGI produces trajectory RMSEs that are uniformly smaller than both FGPGM (15) and AGM (11) for all system components. In the low-noise setting, the advantage of MAGI is especially apparent for components S, R, S_R, and R_{pp}, with trajectory RMSEs less than half of the closest comparison method. For the high-noise setting, MAGI reduces trajectory RMSE the most for S_d and R_{pp} (~50%). AGM (11) struggles with this example at both noise settings. To visually compare the trajectory RMSEs in Table 5, plots of the corresponding reconstructed trajectories by different methods at both noise settings are given in *SI Appendix*.

The run time of MAGI for this system averaged over the repetitions is 18 min, which is 12 times faster than FGPGM (15) and 18 times faster than AGM (11) on the same processor (we follow the authors' recommendation for running their methods) (*SI Appendix* has details).

Discussion

We have presented a methodology for the inference of dynamic systems, using manifold-constrained GPs. A key feature that

Table 3. Parameter inference in the FN model based on 100 simulated datasets

θ	MAGI		FGPGM (15)		AGM (11)	
	Estimate	RMSE	Estimate	RMSE	Estimate	RMSE
a	0.19 ± 0.02	**0.02**	0.22 ± 0.04	0.05	0.30 ± 0.03	0.10
b	0.35 ± 0.09	**0.17**	0.32 ± 0.13	0.18	0.36 ± 0.06	**0.17**
c	2.89 ± 0.06	**0.13**	2.85 ± 0.15	0.21	2.04 ± 0.14	0.97

For each method, average parameter estimates are reported together with SD; parameter RMSEs across simulations are also reported. Bold highlights the best method in terms of parameter RMSE for each parameter.

distinguishes our work from the previous approaches is that it provides a principled statistical framework, firmly grounded on the Bayesian paradigm. Our method also outperformed currently available GP-based approaches in the accuracy of inference on benchmark examples. Furthermore, the computation time for our method is much faster. Our method is robust and able to handle a variety of challenging systems, including unobserved components, asynchronous observations, and parameter nonidentifiability.

A robust software implementation is provided, with user interfaces available for R, MATLAB, and Python, as described in *SI Appendix*. The user may specify custom ODE systems in any of these languages for inference with our package by following the syntax in the examples that accompany this article. In practice, inference with MAGI using our software can be carried out with relatively few user interventions. The setting of hyperparameters and initial values is fully automatic, although may be overridden by the user.

The main setting that requires some tuning is the number of discretization points in $\boldsymbol{I}$. In our examples, this was determined by gradually increasing the denseness of the points with short sampler runs, until the results become indistinguishable. Note that further increasing the denseness of $\boldsymbol{I}$ has no ill effect, apart from increasing the computational time. To illustrate the effect of the denseness of $\boldsymbol{I}$ on MAGI inference results, an empirical study is included in *SI Appendix*, *Varying Number of Discretization*, where we examined the results of the FN model with the discretization set $\boldsymbol{I}$ taken to be 41, 81, 161, and 321 equally spaced points. The results confirm that our proposal of gradually increasing the denseness of $\boldsymbol{I}$ works well. The inference results improve as we increase $\boldsymbol{I}$ from 41 to 161 points, and at 161 points, the results are stabilized. If we further increase the discretization to 321 points, that doubles the compute time with only a slight gain in accuracy compared with 161 points in terms of trajectory RMSEs. This empirical study also indicates that as $W_{\boldsymbol{I}}$ becomes an increasingly dense approximation of W, an inference limit would be expected. A theoretical study is a natural future direction of investigation.

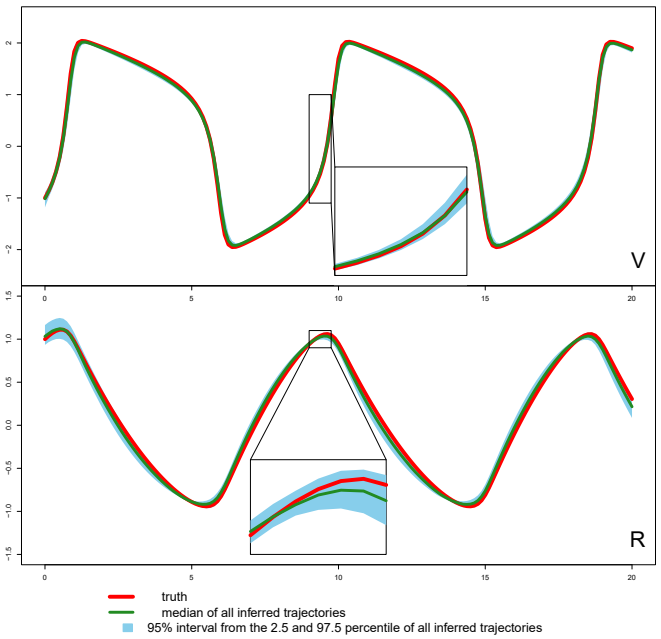


Fig. 2. Inferred trajectories by MAGI for each component of the FN system over 100 simulated datasets. The *Top* is the V component and the *Bottom* is the R component. The blue shaded area represents the 95% interval. *Insets* magnify the corresponding segments.

Table 4. Trajectory RMSEs of each component in the FN system, comparing the average trajectory RMSE of the three methods over 100 simulated datasets

Method	*V*	*R*
MAGI	**0.103**	**0.070**
FGPGM (15)	0.257	0.094
AGM (11)	1.177	0.662

The best trajectory RMSE for each system component is shown in bold. MAGI reduces the RMSE for 60% in component *V* and 25% in component *R* over the best alternative method.

We also investigated the stability of MAGI when the observation time points are farther apart. This empirical study, based on the FN model with 21 observations, is included in *SI Appendix*, *FN Model with Fewer Observations*. The inferred trajectories from the 21 observations are still close to the truth, while the interval bands become wider, which is expected as we have less information in this case. We also found that the denseness of the discretization needs to be increased (to 321 time points in this case) to compensate for the sparser 21 observations.*

An inherent feature of the GP approximation is the tendency to favor smoother curves. This limitation has been previously acknowledged (11, 15). As a consequence, two potential forms of bias can exist. First, estimates derived from the posterior distributions of the parameters may have some statistical bias. Second, the trajectories reconstructed by a numerical solver based on the estimated parameters may differ slightly from the inferred trajectories. MAGI, which is built on a GP framework, does not entirely eliminate these forms of bias. However, as seen in the benchmark systems, the magnitude of our bias in both respects is significantly smaller than the current state of the art in refs. 11 and 15.

We considered the inference of dynamic systems specified by ODEs in this article. Such deterministic ODE models are often adequate to describe dynamics at the aggregate or population level (19). However, when the goal is to describe the behavior of individuals [e.g., individual molecules (20, 21)], models such as stochastic differential equations (SDEs) and continuous-time Markov processes, which explicitly incorporate intrinsic (stochastic) noise, often become the model of choice. Extending our method to the inference of SDEs and continuous-time Markov models is a future direction we plan to investigate. Finally, recent developments in deep learning have shown connections between deep neural networks and GPs (22, 23). It could thus also be interesting to explore the application of neural networks to model the ODE system outputs $\boldsymbol{x}(t)$ in conjunction with GPs.

Materials and Methods

For notational simplicity, we drop the dimension index d in this section when the meaning is clear.

Algorithm Overview. We begin by summarizing the computational scheme of MAGI. Overall, we use HMC (7) to obtain samples of $\boldsymbol{X}_{\boldsymbol{I}}$ and the parameters from their joint posterior distribution. Details of the HMC sampling are included in *SI Appendix*, *Hamiltonian Monte Carlo*. At each iteration of HMC, $\boldsymbol{X}_{\boldsymbol{I}}$ and the parameters† are updated together with a joint gradient, with leapfrog step sizes automatically tuned during the burn-in period to achieve an acceptance rate between 60 and 90%. At the completion of

*This finding echoes the classical understanding that stiff systems require denser discretization (observations farther apart make the system appear relatively more stiff).

†The parameters here refer to θ and σ. If the noise level σ is known a priori, the parameters then refer to θ only.

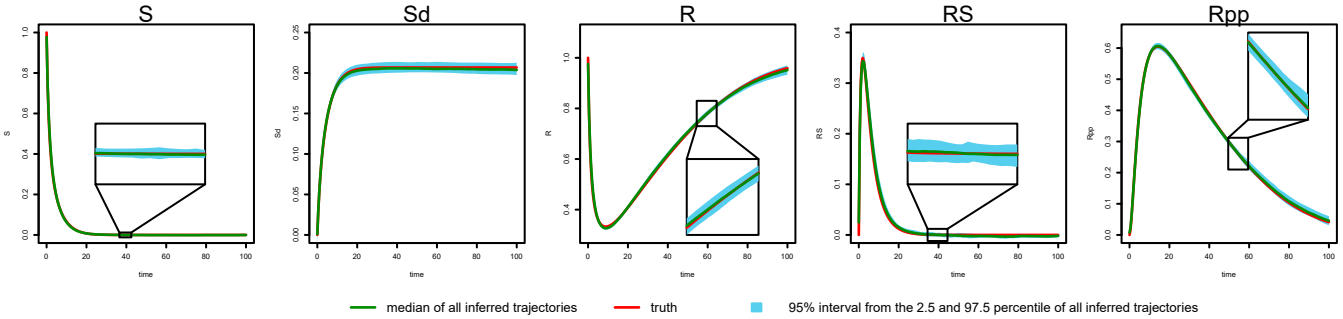


Fig. 3. Inferred trajectories by MAGI for each component of the protein transduction system in the high-noise setting. The red line is the truth, and the green line is the median inferred trajectory over 100 simulated datasets. The blue shaded area represents the 95% interval. *Insets* magnify the corresponding segment.

HMC sampling (and after discarding an appropriate burn-in period for convergence), we take the posterior means of $\boldsymbol{X_I}$ as the inferred trajectories and the posterior means of the sampled parameters as the parameter estimates. The techniques we use to temper the posterior and speed up the computations are discussed in *Prior Tempering* and *SI Appendix, Techniques for Computational Efficiency*.

Several steps are taken to initialize the HMC sampler. First, we apply a GP fitting procedure to obtain values of ϕ and σ for the observed components; the computed values of the hyperparameters ϕ are subsequently held fixed during the HMC sampling, while the computed value of σ is used as the starting value in the HMC sampler (if σ is known, the GP fitting procedure is used to obtain values of ϕ only). Second, starting values of $\boldsymbol{X_I}$ for the observed components are obtained by linearly interpolating between the observation time points. Third, starting values for the remaining quantities—$\boldsymbol{\theta}$ and ($\boldsymbol{X_I}$, ϕ) for any unobserved component(s)—are obtained by optimization of the posterior as described below.

Setting Hyperparameters ϕ for Observed Components. The GP prior $X_d(t) \sim \mathcal{GP}(\mu_d, \mathcal{K}_d)$, $t \in [0, T]$, is on each component $X_d(t)$ separately. The GP Matern kernel $\mathcal{K}(l) = \phi_1 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{l}{\phi_2}\right)^{\nu} B_\nu \left(\sqrt{2\nu}\frac{l}{\phi_2}\right)$ has two hyperparameters that are held fixed during sampling: ϕ_1 controls overall variance level of the GP, while ϕ_2 controls the bandwidth for how much neighboring points of the GP affect each other.

When the observation noise level σ is unknown, values of (ϕ_1, ϕ_2, σ) are obtained jointly by maximizing GP fitting without conditioning on any ODE information, namely

$$\begin{aligned}(\tilde{\phi}, \tilde{\sigma}) &= \arg\max_{\phi,\sigma} p(\phi, \sigma^2 | \boldsymbol{y}_{I_0}) \\ &= \arg\max_{\phi,\sigma} \pi_{\Phi_1}(\phi_1)\pi_{\Phi_2}(\phi_2)\pi_\sigma(\sigma^2) p(\boldsymbol{y}_{I_0} | \phi, \sigma^2), \end{aligned} \quad [7]$$

where $\boldsymbol{y}_{I_0} | \phi, \sigma \sim \mathcal{N}(0, \mathcal{K}_\phi + \sigma^2)$. The index set I_0 is the smallest evenly spaced set such that all observation time points in this component are in I_0 (i.e., $\boldsymbol{\tau} \subseteq I_0$). The priors $\pi_{\Phi_1}(\phi_1)$ and $\pi_\sigma(\sigma^2)$ for the variance parameter ϕ_1 and σ are set to be flat. The prior $\pi_{\Phi_2}(\phi_2)$ for the bandwidth parameter ϕ_2 is set to be a Gaussian distribution. 1) The mean μ_{Φ_2} is set to be half of the period corresponding to the frequency that is the weighted average of all of the frequencies in the Fourier transform of y on I_0 (the values of y on I_0 are linearly interpolated from the observations at $\boldsymbol{\tau}$), where the weight on a given frequency is the squared modulus of the Fourier transform with that frequency, and 2) the SD is set such that T is three SDs away from μ_{Φ_2}. This Gaussian prior on ϕ_2 serves to prevent it from being too extreme. In the subsequent sampling of $(\boldsymbol{\theta}, \boldsymbol{X_\tau}, \sigma^2)$, the hyperparameters ϕ are fixed at $\tilde{\phi}$, while $\tilde{\sigma}$ gives the starting value of σ in the HMC sampler.

If σ is known, then values of (ϕ_1, ϕ_2) are obtained by maximizing

$$\tilde{\phi} = \arg\max_{\phi} p(\phi | \boldsymbol{y}_{I_0}, \sigma^2) = \arg\max_{\phi} \pi_{\Phi_1}(\phi_1)\pi_{\Phi_2}(\phi_2) p(\boldsymbol{y}_{I_0} | \phi, \sigma^2) \quad [8]$$

and held fixed at $\tilde{\phi}$ in the subsequent HMC sampling of $(\boldsymbol{\theta}, \boldsymbol{X_\tau})$. The priors for ϕ_1 and ϕ_2 are the same as previously defined.

Initialization of $\boldsymbol{X_I}$ for the Observed Components. To provide starting values of $\boldsymbol{X_I}$ for the HMC sampler, we use the values of $\boldsymbol{Y_\tau}$ at the observation time points and linearly interpolate the remaining points in $\boldsymbol{I}$.

Initialization of the Parameter Vector θ When All System Components Are Observed. To provide starting values of $\boldsymbol{\theta}$ for the HMC sampler, we optimize the posterior Eq. **5** as a function of $\boldsymbol{\theta}$ alone, holding $\boldsymbol{X_I}$ and σ unchanged at their starting values, when there is no unobserved component(s). The optimized $\boldsymbol{\theta}$ is then used as the starting value for the HMC sampler in this case.

Settings in the Presence of Unobserved System Components: Setting ϕ, Initializing $\boldsymbol{X_I}$ for Unobserved Components, and Initializing θ. Separate treatment is needed for the setting of ϕ and initialization of $(\boldsymbol{\theta}, \boldsymbol{X_I})$ for the unobserved component(s). We use an optimization procedure that seeks to maximize the full posterior in Eq. **5** as a function of $\boldsymbol{\theta}$ together with ϕ and the whole curve of $\boldsymbol{X_I}$ for unobserved components while holding the σ, ϕ, and $\boldsymbol{X_I}$ for the observed components unchanged at their initial value discussed above. We thereby set ϕ for the unobserved component and the starting values of $\boldsymbol{\theta}$ and $\boldsymbol{X_I}$ for unobserved components at the optimized value. In the subsequent sampling, the hyperparameters are fixed at the optimized ϕ, while the HMC sampling starts at the $\boldsymbol{\theta}$ and the $\boldsymbol{X_I}$ obtained by this optimization.

Prior Tempering. After ϕ is set, we use a tempering scheme to control the influence of the GP prior relative to the likelihood during HMC sampling. Note that Eq. **5** can be written as

$$\begin{aligned} &p_{\Theta, X(I) | Y(\tau), W_I}(\boldsymbol{\theta}, \boldsymbol{x(I)} | \boldsymbol{y(\tau)}, W_I = 0) \\ &\propto p_{\Theta, X(I) | W_I}(\boldsymbol{\theta}, \boldsymbol{x(I)} | W_I = 0) p_{Y(\tau) | X(\tau)}(\boldsymbol{y(\tau)} | \boldsymbol{x(\tau)}). \end{aligned} \quad [9]$$

As the cardinality of $|\boldsymbol{I}|$ increases with more discretization points, the prior part $p_{\Theta, X(I) | W_I}(\boldsymbol{\theta}, \boldsymbol{x(I)} | W_I = 0)$ grows, while the likelihood part $p_{Y(\tau) | X(\tau)}(\boldsymbol{y(\tau)} | \boldsymbol{x(\tau)})$ stays unchanged. Thus, to balance the influence of the prior, we introduce a tempering hyperparameter β with the corresponding posterior

Table 5. Trajectory RMSEs of the individual components in the protein transduction system, by comparing the average RMSEs of the three methods over 100 simulated datasets

Method	S	S_d	R	S_R	R_{pp}
Low-noise case, $\sigma = 0.001$					
MAGI	**0.0020**	**0.0013**	**0.0040**	**0.0017**	**0.0036**
FGPGM (15)	0.0049	0.0016	0.0156	0.0036	0.0149
AGM (11)	0.0476	0.2881	0.3992	0.0826	0.2807
High-noise case, $\sigma = 0.01$					
MAGI	**0.0122**	**0.0043**	**0.0167**	**0.0135**	**0.0136**
FGPGM (15)	0.0128	0.0089	0.0210	0.0136	0.0309
AGM (11)	0.0671	0.3125	0.4138	0.0980	0.2973

The method achieving the best RMSE for each system component is shown in bold.

$$
\begin{aligned}
&p^{(\beta)}_{\Theta,\boldsymbol{X}_I|W_I,\boldsymbol{Y}_\tau}(\boldsymbol{\theta},\boldsymbol{x}_I|0,\boldsymbol{y}_\tau)\\
&\propto p_{\Theta,\boldsymbol{X}(I)|W_I}(\boldsymbol{\theta},\boldsymbol{x}(I)|W_I=0)^{1/\beta}\, p_{\boldsymbol{Y}(\tau)|\boldsymbol{X}(I)}(\boldsymbol{y}(\tau)|\boldsymbol{x}(I))\\
&\propto \pi_\Theta(\boldsymbol{\theta})\exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\left[N_d\log(2\pi\sigma_d^2)+\|x_d(\tau_d)-y_d(\tau_d)\|^2_{\sigma_d^{-2}}\right.\right.\\
&\qquad\left.\left.+\frac{1}{\beta}\left(\|x_d(I)-\mu_d(I)\|^2_{C_d^{-1}}+\left\|\mathbf{f}^{X,\theta}_{d,I}-\dot{\mu}_d(I)-m_d(x_d(I)-\mu_d(I))\right\|^2_{K_d^{-1}}\right)\right]\right\}. \qquad [10]
\end{aligned}
$$

A useful setting that we recommend is $\beta = D|I|/N$, where D is the number of system components, $|I|$ is the number of discretization time points, and $N=\sum_{d=1}^{D} N_d$ is the total number of observations. This setting aims to balance the likelihood contribution from the observations with the total number of discretization points.

Data Availability. All of the data used in the article are simulation data. The details, including the models to generate the simulation data, are described in *Results* and *SI Appendix*. Our software package, available at GitHub, https://github.com/wongswk/magi, also includes complete replication scripts for all of the data and examples.

ACKNOWLEDGMENTS. The research of S.W.K.W. is supported in part by Natural Sciences and Engineering Research Council of Canada Discovery Grant RGPIN-2019-04771. The research of S.C.K. is supported in part by NSF Grant DMS-1810914.

1. H. Hirata *et al.*, Oscillatory expression of the bHLH factor Hes1 regulated by a negative feedback loop. *Science* **298**, 840–843 (2002).
2. D. B. Forger, *Biological Clocks, Rhythms, and Oscillations: The Theory of Biological Timekeeping* (MIT Press, 2017).
3. H. Miao, C. Dykes, L. M. Demeter, H. Wu, Differential equation modeling of HIV viral fitness experiments: Model identification, model selection, and multimodel inference. *Biometrics* **65**, 292–300 (2009).
4. S. Busenberg, *Differential Equations and Applications in Ecology, Epidemics, and Population Problems* (Elsevier, 2012).
5. N. S. Gorbach, S. Bauer, J. M Buhmann. "Scalable variational inference for dynamical systems" in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds. (Curran Associates Inc., Red Hook, NY, 2017), pp. 4806–4815.
6. L. Wu, X. Qiu, Y.-X. Yuan, H. Wu, Parameter estimation and variable selection for big systems of linear ordinary differential equations: A matrix-based approach. *J. Am. Stat. Assoc.* **114**, 657–667 (2019).
7. R. M. Neal, "MCMC using Hamiltonian dynamics" in *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones, X. L. Meng, Eds. (Chapman & Hall/CRC Handbooks of Modern Statistical Methods, CRC Press, 2011), pp. 113–162.
8. B. Calderhead, M. Girolami, N. D. Lawrence, "Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes" in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, L. Bottou, Eds. (Curran Associates Inc., Red Hook, NY, 2008), pp 217–224.
9. J. O Ramsay, G. Hooker, D. Campbell, J. Cao, Parameter estimation for differential equations: A generalized smoothing approach. *J. Roy. Stat. Soc. B* **69**, 741–796 (2007).
10. P. Hennig, M. A. Osborne, M. Girolami, Probabilistic numerics and uncertainty in computations. *Proc. Math. Phys. Eng. Sci.* **471**, 20150142 (2015).
11. F. Dondelinger, D. Husmeier, S. Rogers, M. Filippone, "ODE parameter inference using adaptive gradient matching with Gaussian processes" in *International Conference on Artificial Intelligence and Statistics*, C. M. Carvalho, P. Ravikumar, Eds. (Machine Learning Research Press, Cambridge, MA, 2013), pp. 216–228.
12. D. Barber, Y. Wang, "Gaussian processes for Bayesian estimation in ordinary differential equations" in *International Conference on Machine Learning*, E. P. Xing, T. Jebara, Eds. (Machine Learning Research Press, Cambridge, MA, 2014), pp. 1485–1493.
13. S. Ghosh, S. Dasmahapatra, K. Maharatna, Fast approximate Bayesian computation for estimating parameters in differential equations. *Stat. Comput.* **27**, 19–38 (2017).
14. A. Lazarus, D. Husmeier, T. Papamarkou, "Multiphase MCMC sampling for parameter inference in nonlinear ordinary differential equations" in *International Conference on Artificial Intelligence and Statistics*, A. Storkey, F. Perez-Cruz, Eds. (Machine Learning Research Press, Cambridge, MA, 2018), pp. 1252–1260.
15. P. Wenk *et al.*, "Fast Gaussian process based gradient matching for parameter identification in systems of nonlinear ODEs" in *International Conference on Artificial Intelligence and Statistics*, K. Chaudhuri, M. Sugiyama, Eds. (Machine Learning Research Press, Cambridge, MA, 2019), pp. 1351–1360.
16. B. Macdonald, C. Higham, D. Husmeier, Controversy in mechanistic modelling with Gaussian processes. *Proc. Mach. Learn. Res.* **37**, 1539–1547 (2015).
17. R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J.* **1**, 445–466 (1961).
18. V. Vyshemirsky, M. A. Girolami, Bayesian ranking of biochemical system models. *Bioinformatics* **24**, 833–839 (2007).
19. T. G. Kurtz, The relationship between stochastic and deterministic models for chemical reactions. *J. Chem. Phys.* **57**, 2976–2978 (1972).
20. S. C. Kou, X. S. Xie, Generalized Langevin equation with fractional Gaussian noise: Subdiffusion within a single protein molecule. *Phys. Rev. Lett.* **93**, 180603 (2004).
21. S. C. Kou, B. J. Cherayil, W. Min, B. P. English, X. S. Xie, Single-molecule Michaelis-Menten equations. *J. Phys. Chem. B* **109**, 19068–19081 (2005).
22. A. Jacot, F. Gabriel, C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks" in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, S. Bengio *et al.*, Eds. (Curran Associates Inc., Red Hook, NY, 2018), pp. 8580–8589.
23. J. Lee *et al.*, "Deep neural networks as Gaussian processes" in *International Conference on Learning Representations*, I. Murray, M. Ranzato, O. Vinyals, Eds. (OpenReview.net, Amherst, MA, 2018), pp. 1–17.

PNAS
www.pnas.org

Supplementary Information for

Inference of dynamic systems from noisy and sparse data via manifold-constrained Gaussian processes

Shihao Yang, Samuel W. K. Wong and S. C. Kou

Corresponding author: S. C. Kou
E-mail: kou@stat.harvard.edu

This PDF file includes:

Supplementary text
Figs. S1 to S6
Tables S1 to S2

Supporting Information Text

This supporting information file presents techniques for efficient computation, a description of Hamiltonian Monte Carlo, further details and discussion for each of the dynamic system examples in the main manuscript, and additional empirical studies on varying the number of discretization points and reducing the number of observations.

Techniques for computational efficiency

After setting $\boldsymbol{\phi}$, the matrix inverses C_d^{-1}, K_d^{-1} can be pre-computed and held fixed in the sampling of $\boldsymbol{X}, \boldsymbol{\theta}, \sigma$ from the target posterior, Eq. (5) in the main text. Thus, the computation of Eq. (5) in the main text at sampled values of $(\boldsymbol{X}, \boldsymbol{\theta}, \sigma)$ only involves matrix multiplication, which has typical computation complexity of $O(n^2)$, where n is the matrix dimension (i.e., number of discretization points). Due to the short-term memory and local structure of Gaussian processes (GPs), the partial correlation of two distant points diminishes quickly to zero, resulting in the off-diagonal part of precision matrices C_d^{-1} and K_d^{-1} being close to zero. Similarly, m_d is the projection matrix of the Gaussian process to its derivative process, and since derivative is a local property, the effect from a far away point is small given one's neighboring points, resulting in the off-diagonal part of projection matrix m_d being close to zero as well. Therefore, an efficient band matrix approximation may be used on C_d^{-1}, K_d^{-1}, and m_d to reduce computation into $O(n)$, when calculating Eq. (5) in the main text at each sampled $(\boldsymbol{X}, \boldsymbol{\theta}, \sigma)$ with a fixed band size. In our experience, a band size of 20 to 40 is sufficient, and we recommend using an evenly spaced $\boldsymbol{I}$ for best results with the band matrix approximation and thus faster computation. In our implementation, a failure in the band approximation is automatically detected by checking for divergence in the quadratic form, and a warning is outputted to the user to increase the band size.

Hamiltonian Monte Carlo

Sampling procedure with HMC. We outline the HMC procedure for sampling from a target probability distribution. The interested reader may refer to Ref (7) for more thorough introduction to HMC.

First, suppose the target distribution has density $\pi_{\text{target}}(\boldsymbol{q}) = (1/Z)\exp(-U(\boldsymbol{q}))$, where Z is the normalizing constant, and $U(\boldsymbol{q})$ is the negation of the log target density. $U(\boldsymbol{q})$ has the physical interpretation of the "potential energy" at "position" $\boldsymbol{q}$. In MAGI, $\boldsymbol{q}$ is the collection of $\boldsymbol{X_I}$ and the parameters. When the noise level σ is known *a priori*, the parameters refer to $\boldsymbol{\theta}$ only; when σ is unknown, the parameters refer to $\boldsymbol{\theta}$ and σ. In MAGI the function $U(\cdot)$ is the negation of the log posterior density in Eq. (5) of the main text.

Second, momentum variables, $\boldsymbol{p}$, of the same dimension as $\boldsymbol{q}$, are introduced. A "kinetic energy" is defined to be $K(p) = \boldsymbol{p}^\intercal \boldsymbol{p}/2$.

Third, define the "Hamiltonian" to be $H(\boldsymbol{q}, \boldsymbol{p}) = U(\boldsymbol{q}) + K(\boldsymbol{p})$, and consider the joint density of $\boldsymbol{q}$ and $\boldsymbol{p}$, which is proportional to $\exp(-H(\boldsymbol{q}, \boldsymbol{p}))$. Under this construction, $\boldsymbol{q}$ and $\boldsymbol{p}$ are independent, where the marginal probability density of $\boldsymbol{q}$ is the target π_{target}, and the marginal probability density of $\boldsymbol{p}$ is Gaussian. We will then sample from this augmented distribution for $(\boldsymbol{q}, \boldsymbol{p})$.

We repeat the following three steps, that together compose one HMC iteration: (1) Sample $\boldsymbol{p}$ from the normal distribution $\mathcal{N}(0, I)$ since $K(\boldsymbol{p}) = \boldsymbol{p}^\intercal \boldsymbol{p}/2$ corresponds to a Gaussian kernel; (2) construct a proposal $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ for $(\boldsymbol{q}, \boldsymbol{p})$ by simulating the *Hamiltonian dynamics* using the leapfrog method (detailed in the next subsection), and (3) accept or reject $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ as the next state of $(\boldsymbol{q}, \boldsymbol{p})$ according to the usual Metropolis acceptance probability, $\min[1, \exp(-H(\boldsymbol{q}^*, \boldsymbol{p}^*) + H(\boldsymbol{q}, \boldsymbol{p}))]$.

After repeating the HMC iteration for the desired number of iterations, the sampled $\boldsymbol{q}$ are taken to be the samples from π_{target}. Recall $\boldsymbol{q}$ is the collection of $\boldsymbol{X_I}$ and the parameters in MAGI, so at the completion of HMC sampling, we have samples of $\boldsymbol{X_I}$ and the parameters. We finally take the posterior mean of $\boldsymbol{X_I}$ as the inferred trajectory, and the posterior means of the sampled parameters as the parameter estimates.

Leapfrog method for Hamiltonian dynamics. The generating of proposals in HMC is inspired by *Hamiltonian dynamics*. The leapfrog method is used to approximate the Hamiltonian dynamics.

One step of the leapfrog method with step size ϵ from an initial point $(\boldsymbol{q}_0, \boldsymbol{p}_0)$ consists of three parts. First, we make a half step for the momentum, $\tilde{\boldsymbol{p}} = \boldsymbol{p}_0 - (\epsilon/2)\nabla U(\boldsymbol{q})|_{\boldsymbol{q}=\boldsymbol{q}_0}$. Second, we make a full step for the position, $\boldsymbol{q}^* = \boldsymbol{q}_0 + \epsilon\tilde{\boldsymbol{p}}$. Third, we make a full step for the momentum using the gradient evaluated at the new position, $\boldsymbol{p}^* = \tilde{\boldsymbol{p}} - (\epsilon/2)\nabla U(\boldsymbol{q})|_{\boldsymbol{q}=\boldsymbol{q}^*}$.

The step size ϵ and the number of leapfrog steps can be tuned. In our MAGI implementation, we recommend fixing the number of leapfrog steps, and tuning the leapfrog step size automatically during the burn-in period to achieve an acceptance rate between 60% and 90%.

More details of the examples

Hes1 model. As stated in the main text, this system has three components, $X = (P, M, H)$, following the ODE

$$\mathbf{f}(X, \boldsymbol{\theta}, t) = \begin{pmatrix} -aPH + bM - cP \\ -dM + \frac{e}{1+P^2} \\ -aPH + \frac{f}{1+P^2} - gH \end{pmatrix}$$

where $\boldsymbol{\theta} = (a, b, c, d, e, f, g)$ are the associated parameters.

The true parameter values in the simulation are set as $a = 0.022$, $b = 0.3$, $c = 0.031$, $d = 0.028$; $e = 0.5$, $f = 20$, $g = 0.3$, which leads to one oscillation cycle approximately every 2 hours. The initial condition is set to be $P(0) = 1.438575$,

$M(0) = 2.037488$, $H(0) = 17.90385$. Recall that these settings, along with the simulated noise level, are derived from Ref (1), where the standard error based on repeated measures are reported to be around 15% of the P (protein) level and M (mRNA) level. Thus the simulation noise is set to be multiplicative following a log-normal distribution with standard deviation 0.15, since all components in the system are strictly positive. The number of observations is also set based on Ref (1), where P and M are observed at 15-minute intervals for 4 hours but the H component is entirely unobserved. In addition, the observations for P and M are asynchronous: starting at time 0, every 15 minutes we observe P; starting at the 7.5 minutes, every 15 minutes we observe M. Following our notation in the main text, $\boldsymbol{\tau}_1 = \{0, 15, 30, \ldots, 240\}$, $\boldsymbol{\tau}_2 = \{7.5, 22.5, 37.5, \ldots, 232.5\}$, and $\boldsymbol{\tau}_3 = \varnothing$. In total we have $N_1 = 17$ observations for P, $N_2 = 16$ observations for M, and $N_3 = 0$ observations for H; P and M are never observed at the same time. See Fig 1 (leftmost panel) of the main text for a visual illustration.

We provide additional details on how to set up MAGI, as applied to this system. Since the components are strictly positive, we first apply a log-transformation to the system so that the resulting noise is additive Gaussian. Define

$$\tilde{P} = \log P, \quad \tilde{M} = \log M, \quad \tilde{H} = \log H,$$

so that the transformed system is:

$$\frac{d\tilde{\boldsymbol{X}}(t)}{dt} = \begin{pmatrix} -a\exp(\tilde{H}) + b\exp(\tilde{M} - \tilde{P}) - c \\ -d + e\exp(-\tilde{M})(1 + \exp(2\tilde{P}))^{-1} \\ -a\exp(\tilde{P}) + f\exp(-\tilde{H})(1 + \exp(2\tilde{P}))^{-1} - g \end{pmatrix}.$$

We conduct all the inference on the log-transformed system, and transform back to the original scale only at the final step to obtain inferred trajectories on the original scale.

As described in "Setting hyper-parameters $\boldsymbol{\phi}$ for observed components" in the Materials and Methods, we consider the observed P component and the observed M component separately when setting their respective hyper-parameters $\boldsymbol{\phi}$. For P, since the observation time points are already equally spaced, we have $I_0 = \boldsymbol{\tau}_1 = \{0, 15, 30, \ldots, 240\}$; $\tilde{\boldsymbol{\phi}}$ is obtained by optimization of Eq (8) in the main text given $\boldsymbol{y}_{\boldsymbol{1},\boldsymbol{I_0}} = \boldsymbol{y}_{\boldsymbol{1},\boldsymbol{\tau_1}}$, and fixing the noise level σ at the true value of 0.15. For M, since the observation time points are also equally spaced, we have $I_0 = \boldsymbol{\tau}_2 = \{7.5, 22.5, 37.5, \ldots, 232.5\}$; $\tilde{\boldsymbol{\phi}}$ for M is obtained by optimization of Eq (8) in the main text, given $\boldsymbol{y}_{\boldsymbol{2},\boldsymbol{I_0}} = \boldsymbol{y}_{\boldsymbol{2},\boldsymbol{\tau_2}}$, and fixing the noise level σ at the true value of 0.15 as well.

Next, we consider the discretization set $\boldsymbol{I}$. In this example we use all observation time points as the discretization set, i.e., $I = \boldsymbol{\tau}_1 \cup \boldsymbol{\tau}_2 = \{0, 7.5, 15, 22.5, \ldots, 232.5, 240\}$. To initialize $\boldsymbol{X_I}$ for the observed component P and M, we follow the approach as described in Materials and Methods, using the values of $\boldsymbol{y_\tau}$ at the observation time points and linear interpolation for the remaining points in $\boldsymbol{I}$.

We set the hyper-parameter $\boldsymbol{\phi}$ and the initial values for the unobserved component H by maximizing the full likelihood function, Eq. (5) of the main text, as described in the *Materials and Methods* Section ("Settings in the presence of unobserved system components: setting $\boldsymbol{\phi}$, initializing $\boldsymbol{X_I}$ for unobserved components, and initializing $\boldsymbol{\theta}$").

To balance the contribution from the GP prior and that from the observed data, we use prior tempering (as described in the "Prior tempering" subsection of *Materials of Methods* of the main text). We set $\beta = D|I|/\sum_{d=1}^{D} N_d = 3$, since we have a total of 33 observations (17 observations for P, 16 observations for M, and 0 observations for H) and total of 33 discretization points (at times 0, 7.5, 15, ..., 240) for each of the 3 dimensions. Finally, priors for each parameter in $\boldsymbol{\theta}$ are set to be flat on the interval $(0, \infty)$.

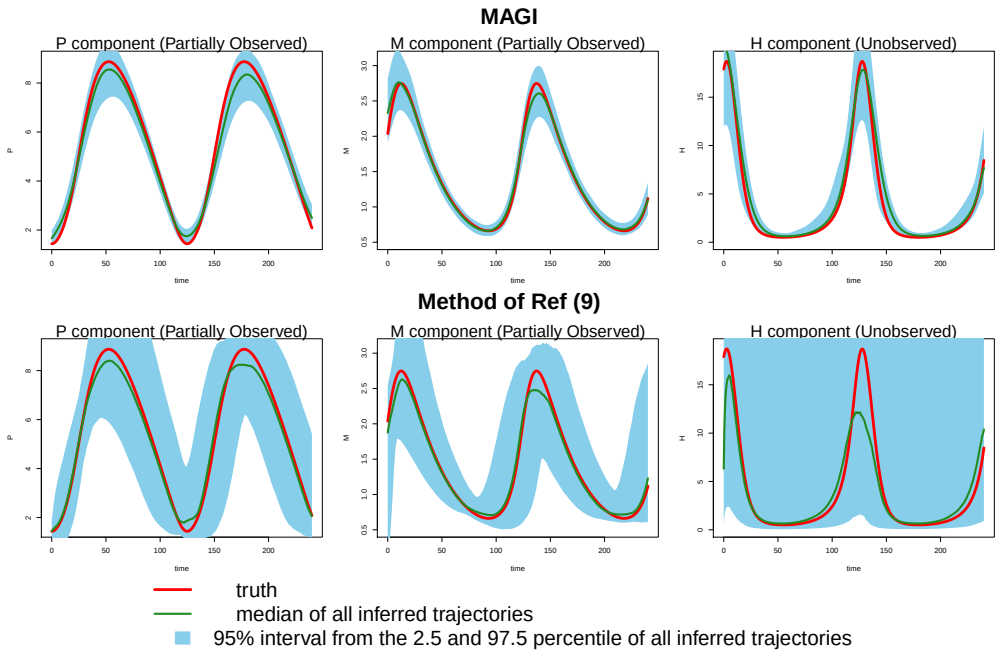
Having initialized the sampler for this system, we next provide details on HMC sampling to obtain our estimates of the trajectory and parameters. A total of 20000 HMC iterations were run, with the first 10000 discarded as burn-in. Each HMC iteration uses 500 leapfrog steps, where the leapfrog step size is drawn randomly from a uniform distribution on $[L, 2L]$ for each iteration. During the burn-in period, L is adaptively tuned: at each HMC iteration L is multiplied by 1.005 if the acceptance rate in the previous 100 iterations is above 90%, and L is multiplied by 0.995 if the acceptance rate in the previous 100 iterations is below 60%. To speed up computations, we use a band matrix approximation (see 'Techniques for computational efficiency' in this SI document) with band size 20. Using the draws from the 10000 HMC iterations after burn-in, the posterior mean of $X = (P, M, H)$ is our inferred trajectory for the system components at time points in $\boldsymbol{I}$, which are generated by MAGI without using any numerical solver; the posterior mean of $\boldsymbol{\theta} = (a, b, c, d, e, f, g)$ provides our parameter estimates.

We make comparisons with the B-spline-based penalization method of Ref (9), which provides the estimated parameters for a given dataset and ODE, but does not provide estimates for the system components (i.e., the trajectories) of the ODE. Thus, to infer the trajectories of system components implied by the method of Ref (9), we run the numerical solver for each parameter estimate (and initial values) produced by the method of Ref (9) to obtain the inferred trajectories for the system components. The method of Ref (9) also has hyper-parameters, in particular, the spline basis functions. The authors' R package `CollocInfer` does not provide the capability to fit spline basis functions if there are unobserved system components. Thus, to obtain results with unobserved components, we fit these spline basis functions using the true value of all system components at the observation time points in this study, which in fact gives the method of Ref (9) an additional advantage than in practice: in the analysis of real data, the true value of the system components is certainly unavailable. Specifically, we used the routines in the R package `CollocInfer` by Ref (9) twice: the first time, we supply the package with the fully-observed noiseless true values of all system components at the observation time points, and thus obtain the estimated B-spline basis functions as part of the package output; the second time, we supply the package with noisy data, together with the B-spline basis functions we obtained in the first run for the unobserved component, to get the final inference results. All other settings are kept at the default values in the package.

Even under this setting, the method of Ref (9) had difficulty recovering the system trajectories and parameters $\boldsymbol{\theta}$ (Figure S1, Table 1 of the main text). Figure S1 plots the inferred trajectories across the 2000 datasets, comparing the two methods side by side, where the method of Ref (9) is seen to have difficulty to recover the unobserved component H. Table 1 of the main text shows the parameter RMSE, where the method of Ref (9) has difficulty to recover the parameters f and g, which are associated with the unobserved component H. Even for the observed components P and M, the inferred trajectory of Ref (9) has much larger RMSE compared to MAGI (see Figure S1 and Table 2 of the main text).

Finally, we want to highlight that none of the other benchmark methods, for example, (11, 15), provides software that is equipped to handle an unobserved component.

Fig. S1. Inference for Hes1 partially observed asynchronized system on 2000 simulated datasets, comparing MAGI to the method of Ref (9). The green line is the median of the inferred trajectories across the 2000 simulated datasets. The blue shaded area represents the 95% interval represented by the 2.5 and 97.5 percentiles of the inferred trajectories. The upper panel is the result from MAGI, and the lower panel is result from the method of Ref (9).



FitzHugh-Nagumo (FN) Model. As stated in the main text, the FitzHugh-Nagumo (FN) model has two components, $X = (V, R)$, following the ODE

$$\mathbf{f}(X, \boldsymbol{\theta}, t) = \begin{pmatrix} c(V - \frac{V^3}{3} + R) \\ -\frac{1}{c}(V - a + bR) \end{pmatrix}$$

where $\boldsymbol{\theta} = (a, b, c)$ are the associated parameters.

Following the same simulation setup as Refs (11, 15), the initial conditions of the system are set at $X(0) = (V(0), R(0)) = (-1, 1)$, the true parameter values are set at $\boldsymbol{\theta} = (a, b, c) = (0.2, 0.2, 3)$, and the system is observed at the equally spaced time points from 0 to 20 with 0.5 interval, i.e, $\boldsymbol{\tau} = \{0, 0.5, 1, 1.5, \ldots, 20\}$. Simulated observations have Gaussian additive noise with $\sigma = 0.2$ on both components.

We provide additional details on how to set up MAGI, as applied to this system. As described in "Setting hyper-parameters $\boldsymbol{\phi}$ for observed components" in the Materials and Methods, the smallest index set that includes the observation time points is $\boldsymbol{I_0} = \boldsymbol{\tau} = \{0, 0.5, 1, 1.5, \ldots, 20\}$; then given $\boldsymbol{y_\tau}$, values of $(\tilde{\phi}, \tilde{\sigma})$ are obtained by optimizing Eq (7) in the main text. Next, we consider the discretization set $\boldsymbol{I}$. In this example we insert 3 additional equally spaced discretization time points between two adjacent observation time points, i.e., $\boldsymbol{I} = \{0, 0.125, 0.25 \ldots, 19.875, 20\}$, $|\boldsymbol{I}| = 161$ time points. As noted in the Discussion section of the main text, we successively increased the denseness of points in $\boldsymbol{I}$ and found that a further increase in the number of discretization points yielded only slightly better results as $\boldsymbol{I} = \{0, 0.125, 0.25 \ldots, 19.875, 20\}$. Next, to initialize $\boldsymbol{X_I}$ for the sampler, we follow the approach as described in Materials and Methods, using the values of $\boldsymbol{y_\tau}$ at the observation time points and linear interpolation for the remaining points in $\boldsymbol{I}$. Then, we obtain a starting value of $\boldsymbol{\theta}$ for the HMC sampler according to the "Initialization of the parameter vector $\boldsymbol{\theta}$ when all system components are observed" subsection in the main text. We apply tempering to the posterior distribution following our guideline in the "Prior tempering" subsection in the main text, where $\beta = D|\boldsymbol{I}|/\sum_{d=1}^{D} N_d = (161 \times 2)/(41 \times 2)$. Finally, the prior distributions for each parameter in $\boldsymbol{\theta}$ are set to be flat on $(0, \infty)$.

Having initialized the sampler for this system, we run HMC sampling to obtain our estimates of the trajectory and parameters. A total of 20000 HMC iterations were run, with the first 10000 discarded as burn-in. Each HMC iteration uses 100 leapfrog

steps, where the leapfrog step size is drawn randomly from a uniform distribution on $[L, 2L]$ for each iteration. During the burn-in period, L is adaptively tuned: at each HMC iteration L is multiplied by 1.005 if the acceptance rate in the previous 100 iterations is above 90%, and L is multiplied by 0.995 if the acceptance rate in the previous 100 iterations is below 60%. To speed up computations, we use a band matrix approximation (see 'Techniques for computational efficiency' in this SI document) with band size 20. Using the draws from the 10000 HMC iterations after burn-in, the posterior mean of $X = (V, R)$ is our inferred trajectory for the system components at time points in $\boldsymbol{I}$, which are generated by MAGI without using any numerical solver; the posterior mean of $\boldsymbol{\theta} = (a, b, c)$ provides our parameter estimates.

For the two benchmark methods, we strictly follow the authors' recommendation. Specifically, for FGPGM of Ref (15), we run their provided software with all settings as recommended by the authors: the standard deviation parameter γ there for handling potential mismatch between GP derivatives and the system is set to 3×10^{-4}, a Matern52 kernel is used, and 300000 MCMC iterations are run. We treat the first half of the iterations as burn-in, and use the posterior mean as the estimate of the parameters and initial conditions. For AGM of Ref (11), the observation noise level is assumed to be known and fixed at their true values (as this method cannot handle unknown noise level), and 300000 MCMC iterations are run. We treat the first half of the iterations as burn-in, and use the posterior mean of the sampled values of the parameters and initial conditions as their respective estimates.

As described in "Metrics for assessing the quality of system recovery" in the main text, the *parameter RMSE* is the root mean squared error (RMSE) of the parameter estimates to the true parameter value. To visualize the parameter estimates of different methods, we plot the histogram of estimated parameters for each of the methods in Figure S2. The red line indicates the true value of each parameter (a, b, c), and the histograms show the distributions of the corresponding parameter estimates over the 100 simulated datasets. For MAGI (upper panel), the red lines lie close to the histogram values for each parameter, indicating that statistical bias is small; the spreads of the histogram values illustrate the variances of the estimates. For FGPGM (15) (middle panel), the red lines lie close to the histogram values for each parameter, indicating that statistical bias is small; the spreads of the histogram values are visibly wider compared to the upper panel, showing larger variances of the estimates. For AGM (11) (lower panel), the relatively narrow spreads of the histogram values indicate that the variances of the parameter estimates are small; however, for parameters a and c the histogram values are much further from the true values, indicating a larger statistical bias than the other two methods.

As described in "Metrics for assessing the quality of system recovery" in the main text, the *trajectory RMSE* is computed for each method based on its estimate of the parameters and initial conditions. Recall that the trajectory RMSE treats the numerical ODE solution based on the true parameter values as the ground truth, and is obtained as follows: first, the numerical solver is used to reconstruct the trajectory based on the estimates of the parameter and initial condition from a given method; then, the RMSE of this reconstructed trajectory to the true trajectory at the observation time points is calculated. To visualize the trajectory RMSEs shown in Table 4 of the main text for each method, Figure S3 plots the true trajectory (red lines) and the 95% interval of the reconstructed trajectories (gray bands) over the 100 simulated datasets for MAGI, FGPGM of Ref (15), and AGM of Ref (11). For MAGI (upper panel), the gray bands closely follow the true trajectories for both components, showing that the statistical bias of the reconstructed trajectories is small; the bands are also quite narrow, showing that the variance in the reconstructed trajectories is low. For FGPGM (15) (middle panel), the gray bands largely follow the true trajectories for both components, showing that the statistical bias of the reconstructed trajectories is small; however, the bands are visibly wider compared to the upper panel for both components, indicating larger variances in the reconstructed trajectories. For AGM (11) (lower panel), the gray bands do not capture the true trajectory for either component, which indicates there is clear statistical bias in the reconstructed trajectories, and the bands are also much wider than the other two methods indicating a higher variance; this is probably due to the underlying statistical bias in the parameter estimates as seen in the lower panel of Figure S2.

Fig. S2. Histograms of the estimated $\boldsymbol{\theta}$ of the FN system over 100 simulated datasets. Three methods are compared. Upper panel: MAGI. Middle panel: FGPGM of Ref (15). Lower panel: AGM of Ref (11). The red line is the true parameter value.

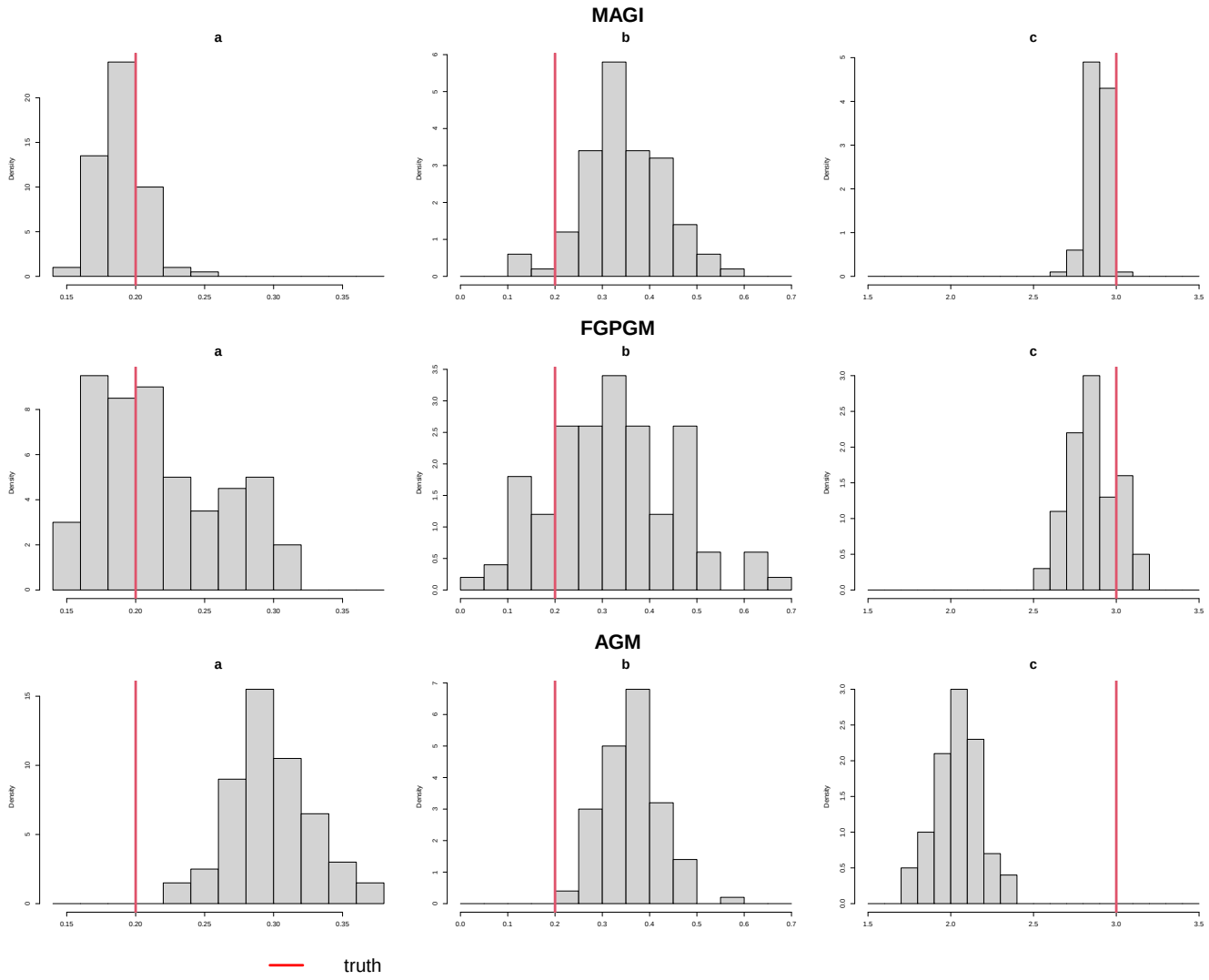
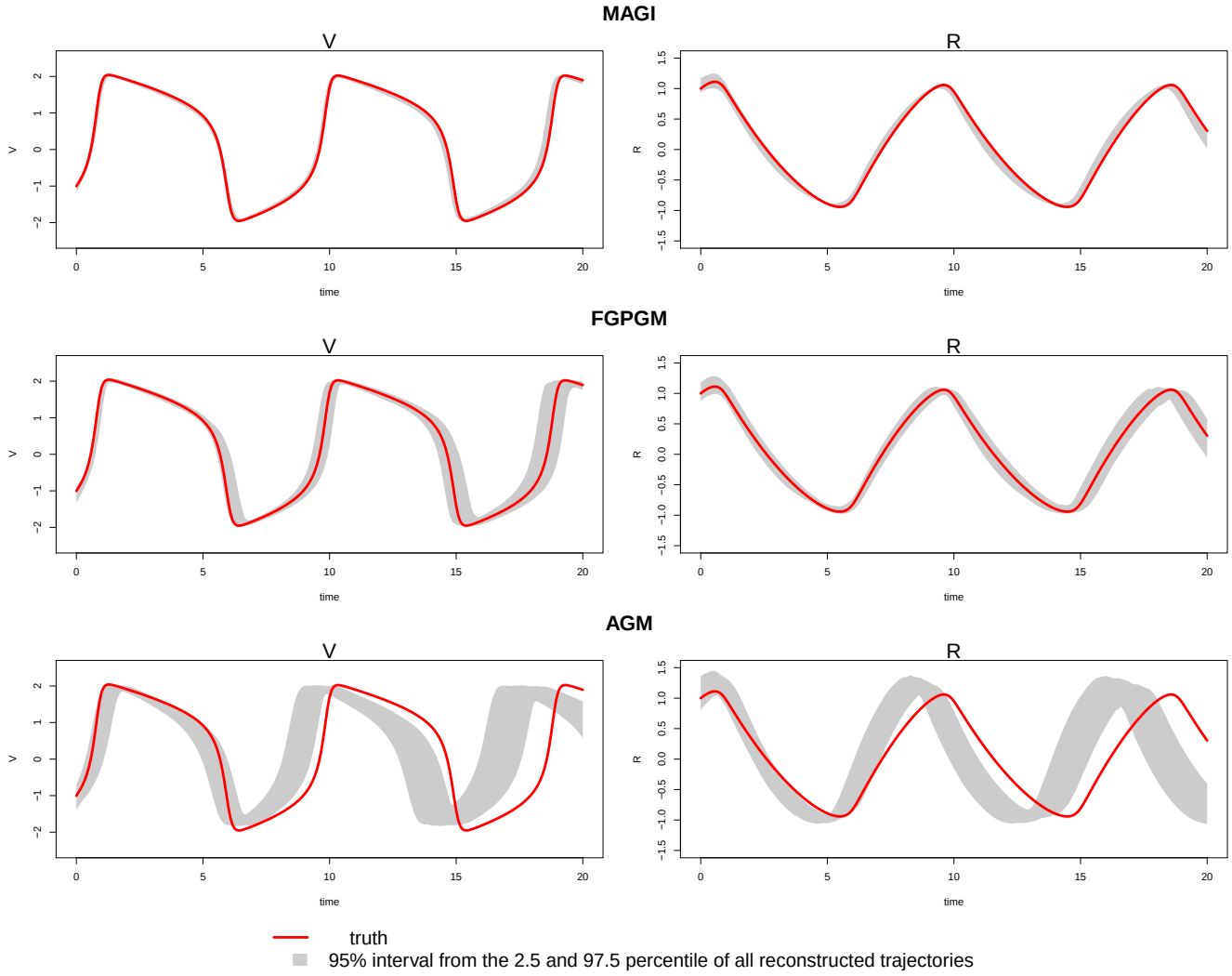


Fig. S3. Reconstructed trajectories by the numerical solver for each component of the FN system from three methods. Upper panel: MAGI. Middle panel: FGPGM of Ref (15). Lower panel: AGM of Ref (11). The red line is the true trajectory. The grey area is a 95% interval represented by the 2.5 and 97.5 percentiles.



Protein transduction model. As stated in the main text, the protein transduction model has five components, $X = (S, S_d, R, S_R, R_{pp})$, following the ODE

$$\mathbf{f}(X, \boldsymbol{\theta}, t) = \begin{pmatrix} -k_1 \cdot S - k_2 \cdot S \cdot R + k_3 \cdot S_R \\ k_1 \cdot S \\ -k_2 \cdot S \cdot R + k_3 \cdot S_R + \frac{V \cdot R_{pp}}{K_m + R_{pp}} \\ k_2 \cdot S \cdot R - k_3 \cdot S_R - k_4 \cdot S_R \\ k_4 \cdot S_R - \frac{V \cdot R_{pp}}{K_m + R_{pp}} \end{pmatrix},$$

where $\boldsymbol{\theta} = (k_1, k_2, k_3, k_4, V, K_m)$ are the associated rate parameters.

Following the same simulation setup as in (11, 15), the initial conditions of the system are $X(0) = (1, 0, 1, 0, 0)$, the true parameter values are $\boldsymbol{\theta} = (0.07, 0.6, 0.05, 0.3, 0.017, 0.3)$, and the system is observed at the time points

$$t = \{0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100\}.$$

In the low noise scenario, simulated observations have Gaussian additive noise with $\sigma = 0.001$, while in the high noise scenario $\sigma = 0.01$. As noted in the main text, inference for this system is challenging due to the non-identifiability of the parameters, so the comparison of different method focuses on the trajectory recovery rather than the parameter RMSE.

We provide additional details on how to set up MAGI, as applied to this system. Recall that the observation times are unequally spaced. Thus, as described in "Setting hyper-parameters $\boldsymbol{\phi}$ for observed components" in the Materials and Methods, we take $\boldsymbol{I_0} = \{0, 1, 2, \ldots, 99, 100\}$, which is the smallest index set with equally spaced time points that includes the observation times, and use linear interpolation between the observations $\boldsymbol{y_\tau}$ to obtain $\boldsymbol{y_{I_0}}$; given $\boldsymbol{y_{I_0}}$, values of $(\tilde{\boldsymbol{\phi}}, \tilde{\sigma})$ are obtained by optimization. Next, we consider the discretization set $\boldsymbol{I}$. In this example we insert 1 additional equally spaced discretization time point between two adjacent time points in $\boldsymbol{I_0}$, i.e., $\boldsymbol{I} = \{0, 0.5, 1 \ldots, 99.5, 100\}$, $|\boldsymbol{I}| = 201$ time points. As noted in the Discussion, we successively increased the denseness of points in $\boldsymbol{I}$ and found that a further increase in the number of discretization points did not continue to offer improved results compared to this setting of $\boldsymbol{I}$. Next, to initialize $\boldsymbol{X_I}$ for the sampler, we follow the approach as described in Materials and Methods, using the values of $\boldsymbol{y_\tau}$ at the observation time points and linear interpolation for the remaining points in $\boldsymbol{I}$. Then, we obtain a starting value of $\boldsymbol{\theta}$ for the HMC sampler according to "Initialization of the parameter vector $\boldsymbol{\theta}$ when all system components are observed". We apply tempering to the posterior following our guideline in "Prior tempering", so that $\beta = D|\boldsymbol{I}|/\sum_{d=1}^{D} N_d = (201 \times 5)/(15 \times 5)$. Finally, priors for each parameter in $\boldsymbol{\theta}$ are set to be uniform on the interval $[0, 4]$ as in Ref (15).

Having initialized the sampler for this system, we run HMC sampling to obtain samples of the trajectory and parameters. A total of 20000 HMC iterations were run, with the first 10000 discarded as burn-in. Each HMC iteration uses 100 leapfrog steps, where the leapfrog step size is drawn randomly from a uniform distribution on $[L, 2L]$ for each iteration. During the burn-in period, L is adaptively tuned: at each HMC iteration L is multiplied by 1.005 if the acceptance rate in the previous 100 iterations is above 90%, and L is multiplied by 0.995 if the acceptance rate in the previous 100 iterations is below 60%. To speed up computations, we use a band matrix approximation (see 'Techniques for computational efficiency' in this SI document) with band size 40. Using the draws from the 10000 HMC iterations after burn-in, the posterior mean of $X = (S, S_d, R, S_R, R_{pp})$ is our inferred trajectory for the system components, which are generated by MAGI without using any numerical solver; the posterior mean of $\boldsymbol{\theta} = (k_1, k_2, k_3, k_4, V, K_m)$ provides our parameter estimates.

We compare MAGI with FGPGM of Ref (15) and AGM of Ref (11) on 100 simulated datasets for each of the two noise settings. All methods use the same priors for $\boldsymbol{\theta}$, namely uniform on $[0, 4]$ as used previously in Ref (15). We strictly follow the authors' recommendation for running their methods. Specifically, for FGPGM of Ref (15), we run their provided software with all settings as recommended by the authors: the standard deviation parameter γ there for handling potential mismatch between GP derivatives and the system is set to 10^{-4}, a sigmoid kernel is used, and 300000 MCMC iterations are run. We treat the first half of the iterations as burn-in, and use the posterior mean as the estimate of the parameters and initial conditions. For AGM of Ref (11), the observation noise level is assumed to be known and fixed at their true values (as this method cannot handle unknown noise level), and 300000 MCMC iterations are run. We treat the first half of the iterations as burn-in, and use the posterior mean as the estimate of the parameters and initial conditions.

As described in "Metrics for assessing the quality of system recovery" in the main text, the *trajectory RMSE* is computed for each method based on its estimate of the parameters and initial conditions. Recall that the trajectory RMSE treats the numerical ODE solution based on the true parameter values as the ground truth, and is obtained as follows: first, the numerical solver is used to reconstruct the trajectory based on the estimates of the parameter and initial condition from a given method; then, the RMSE of this reconstructed trajectory to the true trajectory at the observation time points is calculated. To visualize the trajectory RMSEs shown in Table 4 of the main text for each method, Figures S4 and S5 (for the low and high noise cases, respectively) plot the true trajectory (red lines) and the 95% interval of the reconstructed trajectories (gray bands) over the 100 simulated datasets for MAGI, FGPGM of Ref (15), and AGM of Ref (11).

In the low noise case (Figure S4), the gray bands for MAGI (top panel) closely follow the true trajectories for all five system components, showing that the statistical bias of the reconstructed trajectories is small overall. The interval bands are also very narrow, indicating that the variance in the reconstructed trajectories is low. For FGPGM (15) (middle panel), the gray bands largely follow the true trajectories for most of the system components, indicating that the statistical bias of the reconstructed trajectories is small for most of the time range; however, there is clearly visible bias for the second half of the time period ($t = 50$ to $t = 100$) for R and R_{pp}. The interval bands are also narrow, indicating that the variance in the

reconstructed trajectories is low. For AGM (11) (lower panel), the gray bands are unable to capture the true trajectories, indicating there is significant statistical bias in the reconstructed trajectories. The wide interval bands indicate a high variance in the reconstructed trajectories as well; note that the 97.5 percentile of AGM also exceeds the visible upper limit of the plots for S_d and R.

Inference is more challenging in the high noise case (Figure S5). For MAGI (upper panel), the gray bands still closely follow the true trajectories for all five system components, showing that the statistical bias of the reconstructed trajectories is small overall, with some slight bias for R_{pp}. The interval bands are wider than the corresponding low noise case but still relatively narrow for all the components, indicating that the variance in the reconstructed trajectories is low. For FGPGM (15) (middle panel), the gray bands largely follow the true trajectories for all the system components, showing that the statistical bias of the reconstructed trajectories is small overall. The interval bands are, however, significantly wider than the upper panel; the variance in the reconstructed trajectories of FGPGM is thus much increased compared to that of MAGI. For AGM (11) (lower panel), the gray bands are again unable to capture the true trajectories, which indicates there is significant statistical bias in the reconstructed trajectories. The wide interval bands indicate a high variance in the reconstructed trajectories; note that the 97.5 percentile of AGM also exceeds the visible upper limit of the plots for S_d and R.

Fig. S4. Reconstructed trajectories by the numerical solver for each component of the protein transduction system from three methods, in the low noise case. Upper panel: MAGI. Middle panel: FGPGM of Ref (15). Lower panel: AGM of Ref (11). The red line is the true trajectory. The grey area is the 95% interval represented by the 2.5 and 97.5 percentiles.

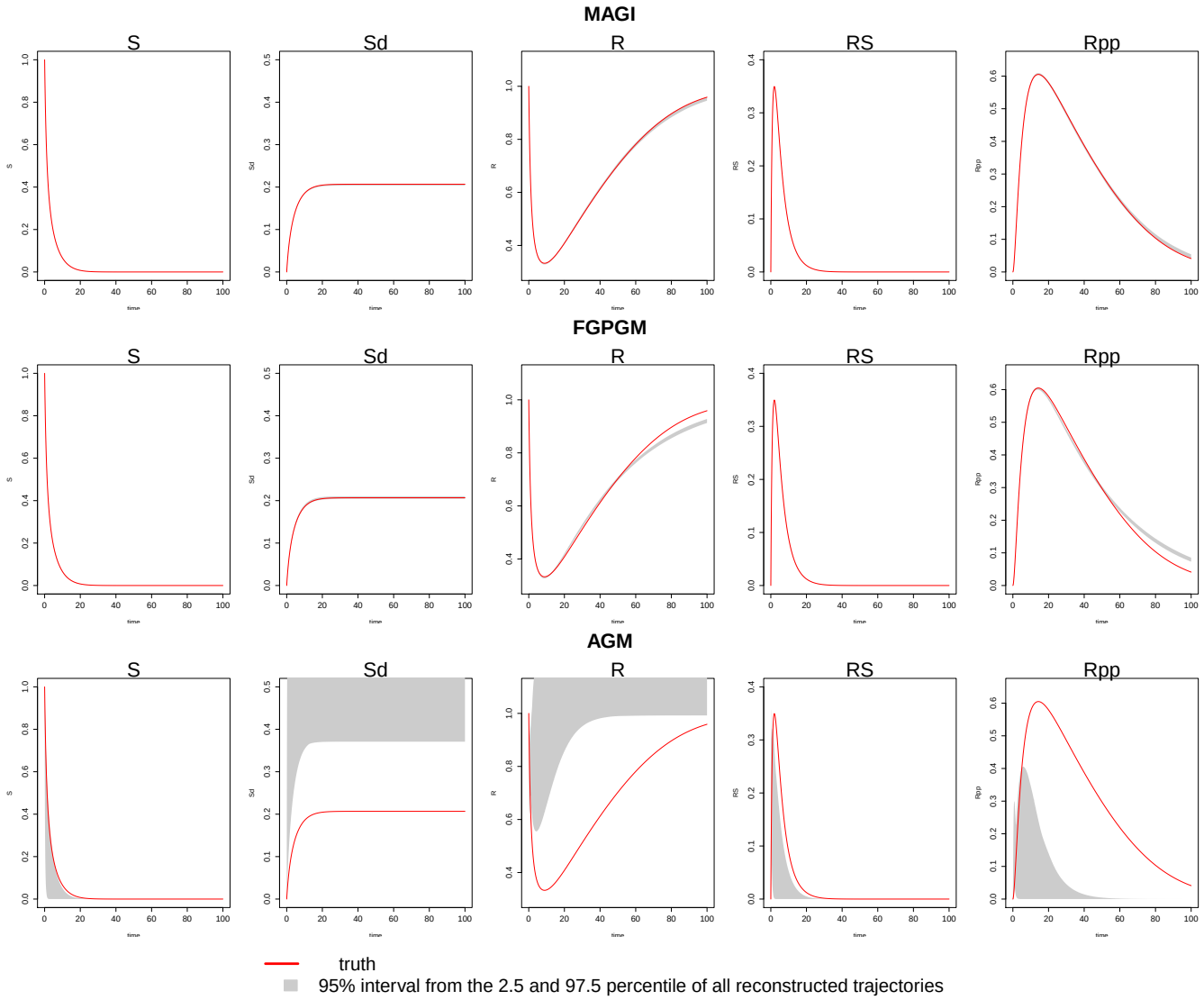
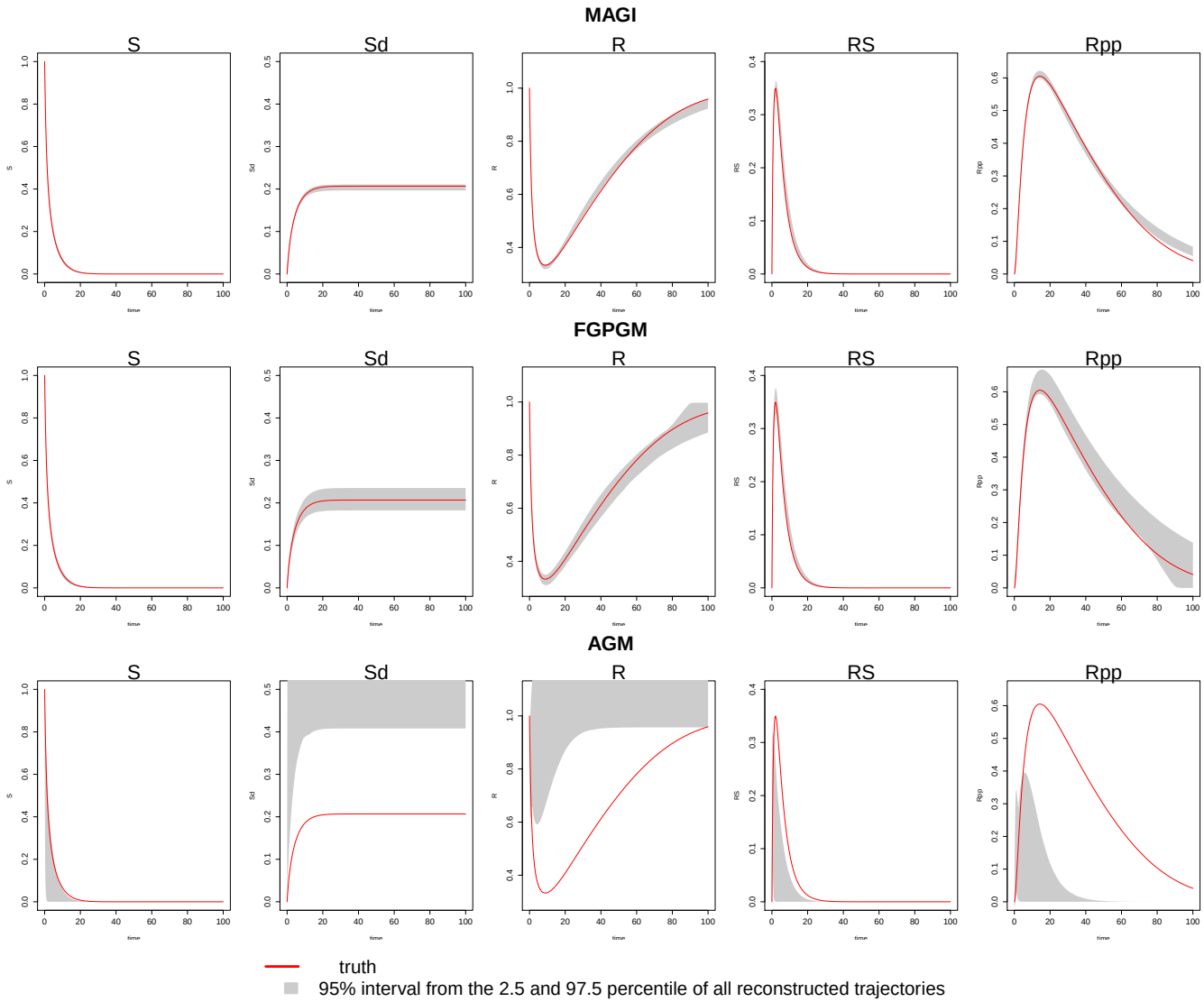


Fig. S5. Reconstructed trajectories by the numerical solver for each component of the protein transduction system from three methods, in the high noise case. Upper panel: MAGI. Middle panel: FGPGM of Ref (15). Lower panel: AGM of Ref (11). The red line is the true trajectory. The grey area is the 95% interval represented by the 2.5 and 97.5 percentiles.



Varying number of discretization

In this section we empirically study the effect of replacing W by $W_{\boldsymbol{I}}$. Specifically, we examine the results from varying the number of discretization points in $\boldsymbol{I}$ in the context of the FN model example.

As discussed in the main text, the number of discretization points in $\boldsymbol{I}$ is the main setting that requires some tuning. In our examples, this was determined by gradually increasing the denseness of the points with short sampler runs, until the results become stabilized. Note that further increasing the denseness of $\boldsymbol{I}$ has no ill effect, apart from increasing the computational time.

Here we illustrate the effect of $\boldsymbol{I}$ by varying the number of discretization points, using the same dataset of the FN system presented in the main text. The result is summarized in Table S1. The results in the main text Tables 3 and 4 are based on 161 discretization points. As can be seen, the inference results improve as we increase $\boldsymbol{I}$ from 41 to 161 points, and at 161 points the results are stabilized. If we further increase the discretization to 321 points, that doubles the compute time with only a slight gain in accuracy compared to 161 points in terms of the RMSEs.

Table S1. Results of FN model inference based on the same 100 simulated datasets as in the main text, with varying number of discretization points (41, 81, 161, 321) equally spaced in time. The results presented in the main text use 161 discretization points.

number of	parameter a		parameter b		parameter c		trajectory RMSE		run time
discretizations	Estimate	RMSE	Estimate	RMSE	Estimate	RMSE	V	R	(minutes)
41	0.20 ± 0.03	0.026	0.24 ± 0.08	0.091	2.83 ± 0.12	0.211	0.358	0.146	0.84
81	0.19 ± 0.02	0.020	0.34 ± 0.09	0.165	2.82 ± 0.07	0.199	0.270	0.142	1.67
161	0.19 ± 0.02	0.020	0.35 ± 0.09	0.172	2.89 ± 0.06	0.128	0.103	0.070	3.13
321	0.19 ± 0.02	0.020	0.33 ± 0.09	0.162	2.92 ± 0.06	0.097	0.072	0.051	5.94

FN model with fewer observations

In this section we study the FN system with 21 observations, which is fewer than the 41 observations presented in the main text. This investigation aims to answer two questions: (1) how does MAGI perform when the number of observations is more sparse, and (2) how does MAGI perform if the observation time points are spaced farther apart?

Following the same setup as the FN system in the main text, we now consider the scenario where 21 observations are made at equally spaced time points from 0 to 20, i.e, $\boldsymbol{\tau} = \{0, 1, \ldots, 20\}$. When applying MAGI, the discretization set I was determined by successively increasing its denseness (with short sampler runs), until the results become stabilized. The numerical results show that in this scenario with sparser observations that are also farther apart, a higher number of discretization points is needed for the results to be stabilized. Specifically for this example with 21 observations, 321 points in the discretization set $\boldsymbol{I}$, i.e., $\boldsymbol{I} = \{0, 0.0625, 0.125, \ldots, 20\}$ are needed to obtain stable inference results. The required increase in discretization seen here echos the classical understanding that stiff systems require denser discretization (observations farther apart make the system appear relatively more stiff).

The inference results are presented in Table S2. The trajectory RMSE is 0.128 for V component and 0.107 for R component, which is roughly $\sqrt{2}$ times the trajectory RMSE for that of 41 observations as reported in the main text. The $\sqrt{2}$ factor is expected, as we halved the number of observations. Further visualization in Figure S6 suggests that the inferred trajectory is quite close to the true system, while the interval bands become wider, which is expected as we have less information in this case.

Table S2. Results of FN model inference based on 100 simulated datasets, each with 21 observations. Average parameter estimates are reported together with standard deviations; parameter RMSEs across simulations are also reported; trajectory RMSEs for the two components are reported as well. The true parameters are set to $a = 0.2, b = 0.2, c = 3$, as in the main text.

number of	number of	parameter a		parameter b		parameter c		trajectory RMSE		run time
observations	discretizations	Estimate	RMSE	Estimate	RMSE	Estimate	RMSE	V	R	(minutes)
21	321	0.19 ± 0.03	0.029	0.44 ± 0.15	0.280	2.79 ± 0.16	0.261	0.128	0.107	5.81

Software implementation

User interfaces for MAGI are available for R, MATLAB, and Python at the Github repository https://github.com/wongswk/magi. Detailed instructions are provided therein for using our package with custom ODE systems specified in any of these languages. Detailed instructions are also provided for replicating all of our results and figures provided in the paper.

Fig. S6. Inferred trajectories by MAGI for each component of the FN system over 100 simulated datasets, each with 21 observations. The red line is the truth, and the green line is the median inferred trajectory over 100 simulated datasets. The blue shaded area represents the 95% interval. The black dots indicate the observations across 100 simulated datasets.