Uncertainty propagation for deterministic models of biochemical networks using moment equations and the extended Kalman filter

Tamara Kurdyaeva and Andreas Milias-Argeitis

Molecular Systems Biology, Groningen Biomolecular Sciences & Biotechnology Institute, University of Groningen, Groningen, Netherlands

Differential equation models of biochemical networks are frequently associated with a large degree of uncertainty in parameters and/or initial conditions. However, estimating the impact of this uncertainty on model predictions via Monte Carlo simulation is computationally demanding. A more efficient approach could be to track a system of low-order statistical moments of the state. Unfortunately, when the underlying model is nonlinear, the system of moment equations is infinite-dimensional and cannot be solved without a moment closure approximation which may introduce bias in the moment dynamics. Here, we present a new method to study the time evolution of the desired moments for nonlinear systems with polynomial rate laws. Our approach is based on solving a system of low-order moment equations by substituting the higher-order moments with Monte Carlo-based estimates from a small number of simulations, and using an extended Kalman filter to counteract Monte Carlo noise. Our algorithm provides more accurate and robust results compared to traditional Monte Carlo and moment closure techniques, and we expect that it will be widely useful for the quantification of uncertainty in biochemical model predictions.

1. Introduction

Mechanistic mathematical models have become ubiquitous tools for analysing the dynamics of complex biochemical networks [1]. A critical challenge in the development and analysis of these models is that they often include several sources of uncertainty. For example, model parameters are often imprecisely known since they have to be inferred from noisy and incomplete experimental data [2]. On the other hand, the large heterogeneity observed in isogenic cell populations needs to be accounted for in models, typically with the introduction of uncertain initial conditions and/or parameters, or with the incorporation of stochastic effects in the model [3,4]. In such cases, it is often necessary to quantify the resulting uncertainty in model predictions, to analyse the robustness of the model, to understand how experimental outcomes are affected by population variability, or to design experiments that will help minimize the uncertainty in critical parts of a model [5].

Whenever the effects of stochasticity in the interactions of chemical species can be ignored (e.g. when they are present in sufficiently high copy numbers) and noise sources extrinsic to the modelled process vary slowly over the timescales of interest, it is possible to use a deterministic, ordinary differential equation (ODE) model to describe a biochemical network of interest. In this setting, uncertainty can be represented with a joint probability distribution over initial conditions and/or parameters [3,6,7]. This probability distribution can then be propagated through the system dynamics to map the initial uncertainty to future time points. The evolution of the probability distribution over the system states is governed by a linear first-order partial differential equation, also...
known as the Liouville equation [8,9]. Unfortunately, direct numerical solution of this equation is currently computationally feasible only for low-dimensional biochemical systems [10,11]. Instead of solving for the full probability distribution, one can work with a set of low-order statistical moments of the system states, since these quantities summarize key features of the full distribution, such as location, spread and skewness. Approaches based on surrogate modelling [12–15] have been proposed to approximate these moments, yet their implementation is relatively intricate, and they still require extensive computational resources. Moment estimation via Monte Carlo sampling is an alternative that is straightforward to implement [16], but it typically requires a large number of simulations to achieve sufficient accuracy. This can be an important limitation for the analysis of complex systems where simulation can be computationally expensive.

In place of approximating moments via surrogate models and simulation, one can also derive a set of ODEs describing the time evolution of these moments [8,17]. The impact of initial condition/parameter uncertainty can then be evaluated by solving these moment equations. However, when the underlying dynamical system is nonlinear, the derivation of moment equations results in an infinite-dimensional ODE system which cannot be solved, since moments up to any order \( n \) depend on moments of order greater than \( n \) [18]. To circumvent this well-known problem and arrive at a closed (finite-dimensional) system of equations, moment-closure techniques can be used [19]. These methods typically rely on the assumption that system variables approximately follow a known distribution (e.g. normal or lognormal), for which the relation between low- and high-order moments is known [20]. In practice, however, it is difficult to verify whether moment closure assumptions hold in a given case. The approximate moment estimates may diverge considerably from the true values when the moment closure assumptions are not valid, and in some cases, the closed system may even become unstable.

To circumvent these problems, we propose a new methodology for the propagation of initial condition and parameter uncertainty when the underlying biochemical network is described by polynomial rates. Our approach is based on tracking the time evolution of a set of low-order moments of the states by combining a small number of Monte Carlo simulations with the differential equations describing the low-order moment dynamics. The Monte Carlo simulations are used to estimate the high-order moments entering the equations of the tracked low-order moments. While this small number of simulations reduces the computational cost of our method, the high-order moment estimates introduce a large amount of uncertainty in the low-order moment system. We therefore employ an extended Kalman filter (EKF) [21] which is used to recursively estimate the posterior distribution of the low-order-moments of interest given (i) the system of low-order moment equations, (ii) the Monte Carlo estimates of the low- and high-order moments and (iii) the uncertainty associated with these Monte Carlo estimates. While the use of the EKF is based on our earlier work for a different class of models [22], the present study goes into much more depth into robust filter design and the comparison of the filter accuracy with ‘naive’ Monte Carlo (NMC) and moment closure. Through several tests with two dynamical systems of varying complexity, we demonstrate that, unlike moment closure, our moment estimation algorithm provides unbiased estimates of the tracked moments, and also outperforms NMC estimation in terms of accuracy.

Contrary to the laborious construction of surrogate models, moment equations can be automatically derived via symbolic calculation software, all necessary model simulations can be readily performed with available ODE integrators, and the Kalman filter calculations involve basic linear algebra operations. We therefore expect our method to be widely useful for the assessment of uncertainty in biochemical model predictions, as well as the construction of computationally efficient parameter estimation methods for nonlinear dynamical systems with initial condition and parameter uncertainty.

2. Results

In the subsections below, we will introduce the class of models considered in this work, the system of moment equations, and the main ingredients of our moment estimation algorithm: Monte Carlo simulation and EKF. Our presentation will be guided by the scheme shown on figure 1.

2.1. ODE model

We consider biochemical reaction networks that are described by a system of polynomial ordinary differential equations (ODEs), such as those arising from the application of mass-action kinetics [23]. The ODE formulation implies that the modelled network involves large numbers of reacting molecules to justify the use of continuous state space. We further assume that the system is well-mixed so that spatial effects can be ignored [23]. Such a model has the following form:

\[
\frac{dz}{dt} = \Psi(z, \theta), \quad z(0) = z_0,
\]

where \( z = (z_1, \ldots, z_s) \in \mathbb{R}^s \) denotes the state vector with initial condition \( z_0 = (z_1(0), \ldots, z_s(0)) \) and \( \Psi \) is a vector of polynomial functions of \( z \) and \( \theta \). The equations also include a vector of parameters \( \theta = (\theta_1, \ldots, \theta_p) \in \mathbb{R}^p \) which are assumed to remain constant with respect to time.

In such a model, uncertainty can be associated with initial conditions \( z_0 \), parameters \( \theta \) or both. When both initial conditions and parameters are uncertain, we can construct the augmented state vector \( x = (z_1, \ldots, z_s, \theta_1, \ldots, \theta_p) \in \mathbb{R}^d \) which contains all uncertain inputs. We can then rewrite the original system (2.1) in terms of the augmented state vector \( x \), where the parameters are assigned zero dynamics

\[
\frac{dx}{dt} = \begin{bmatrix} \frac{dz}{dt} \\ \frac{d\theta}{dt} \end{bmatrix} = \begin{bmatrix} \Psi(z, \theta) \\ 0 \end{bmatrix}, \quad x(0) = \begin{bmatrix} z_0 \\ \theta \end{bmatrix}.
\]

Using this state augmentation, any system with uncertain parameters can be cast into the form of a system with uncertain initial conditions. Therefore, we will henceforth only consider systems of the form

\[
\frac{dx}{dt} = \Phi(x), \quad x(0) = x_0.
\]

2.2. Derivation of the moment equations

We assume that the uncertainty around the initial conditions of (2.3) at time \( t = 0 \) can be modelled by a multivariate probability
distribution with density function \( f(0, x) = f_0(x) \). The form of this probability distribution can be based on theoretical knowledge or historical data available in the literature. With \( x_0 \) uncertain, future values \( x_t \) for \( t > 0 \) also become uncertain. The time evolution of the associated probability density function \( f(t, x_t) \) is governed by the Liouville equation [8]

\[
\frac{\partial f}{\partial t} + \sum_{i=1}^{d} \frac{\partial}{\partial x_i} (\Phi_i \cdot f) = 0, \tag{2.4}
\]

where \( \Phi \) and \( f \) are assumed to be continuously differentiable.

As an alternative to solving the Liouville equation, we will focus on estimating low-order statistical moments of the probability density function \( f(t, x_t) \), which can provide a good summary of the location and shape of \( f(t, x_t) \). Given any scalar differentiable function of the state, \( h(x) \), we can derive a differential equation that describes how the expectation of \( h(x) \) under \( f(t, x_t) \) evolves in time using the properties of the Liouville equation [8,18]. More specifically, if we denote by \( E_l[h(x)] \) the expectation of \( h(x) \) under the probability density \( f(t, x_t) \) (i.e. \( E_l[h(x)] = \int h(x)f(t, x_t)dx_t \)), it is known [8] that

\[
\frac{d}{dt} E_l[h(x)] = E_l \left[ \sum_{i=1}^{d} \frac{\partial}{\partial x_i} (\Phi_i \cdot h(x)) \right]. \tag{2.5}
\]

In the rest of the paper, we will consider functions \( h(x) \) which have one of the three basic forms:

1. \( h(x) = x_i \), for a positive integer \( i \), producing the \( i \)-th order moment of \( x_t \) which we denote by \( \mu_i \). For example, \( i = 1 \) gives the first-order moment (mean) of \( x_t \), \( \mu_1 \).
2. \( h(x) = x_i - \mu_i \), producing the centred \( i \)-th order moment of \( x_t \). For example, \( i = 2 \) gives the central second-order moment (variance) of \( x_t \). To distinguish central from uncentred moments, we will denote the central moments by \( \mu_i^c \).
3. \( h(x) = x_i x_j \), producing the cross moment \( \mu_{i,j}^{x_i x_j} \) of \( x_i \) and \( x_j \).

If we further assume that (2.3) has a polynomial right-hand side (that is, if every component of \( \Phi(x) \) is a polynomial function of \( x \)), we can use (2.5) to derive ODEs describing the time evolution of a set of desired moments. To do this, we need to observe that if \( \Phi(x) \) is a vector of polynomials, then \( \frac{d}{dt}(\mu_i(x)) \) will also be a polynomial of \( x \). This means that the right-hand side of (2.5) will contain moments of \( x \). An example derivation of moment equations for a simple polynomial system is provided in box 1.

**Box 1. Example derivation of moment equations.**

Let us consider the scalar system

\[
\frac{d}{dt} x = \theta_1 x^2 + \theta_2 x + \theta_3,
\]

and assume that initial condition \( x(0) \) follows a distribution \( f(0, x) = f_0(x) \), which implies that \( x(t) \) will follow a distribution \( f(t, x_t) \). Taking \( h(x) = x \), we can use (2.5) to write the differential equation for the evolution of the first moment of \( x \):

\[
\frac{d\mu_1}{dt} = \frac{d}{dt} E [x] = E \left[ \theta_1 x^2 + \theta_2 x + \theta_3 \right] = \theta_1 \mu_2 + \theta_2 \mu_1 + \theta_3.
\]

Likewise, taking \( h(x) = x^2 \) we can write the differential equation for the second moment of \( x \):

\[
\frac{d\mu_2}{dt} = \frac{d}{dt} E [x^2] = E \left[ 2\theta_1 x^2 + 2\theta_2 x + 2\theta_3 \right] = 2\theta_1 \mu_3 + 2\theta_2 \mu_2 + 2\theta_3 \mu_1.
\]

Finally, if we consider \( h(x) = (x - \mu_1)^2 \), we can obtain a differential equation for the variance of \( x \), \( \mu_2 \), by using (2.5) and the fact that \( \mu_2 = \mu_3 - (\mu_1)^2 \):

\[
\frac{d}{dt} \mu_2 = \frac{d}{dt} (\mu_2^2 - (\mu_1)^2) = 2\theta_1 \mu_3 - 2\theta_2 \mu_2 + 2\theta_3 \mu_1.
\]

2.3. The system of low-order moments

As the example of box 1 makes clear, whenever the ODEs of the underlying dynamical system (2.3) contain monomial terms of degree higher than 1, the differential equation for a moment of
order \( n \) will depend on moments of order higher than \( n \). Therefore, the system of moments up to order \( n \) for a general polynomial dynamical system cannot be closed; in other words, there can be no finite subset of moment equations that can exactly capture the evolution of moments up to order \( n \) [18].

More concretely, suppose that we would like to track the time evolution of a set of low-order uncentred moments up to a certain order \( n \), denoted by \( \mu^l \). Applying formula (2.5) to this vector of moments will lead to a set of differential equations that contain (uncentred) moments of order higher than \( n \). If we collect these moments in the vector \( \mu^H \), the system of low-order moment equations will take the form

\[
d\frac{d}{dt} \mu^l = F_{\mu}(\mu^l) + G_{\mu}(\mu^H), \tag{2.6}
\]

where \( F_{\mu}(\cdot) \) and \( G_{\mu}(\cdot) \) are linear functions of \( \mu^l \) and \( \mu^H \), respectively.

If \( \mu^l \) also contains some centred moments (e.g. variances), then, as the example in box 1 shows, the differential equations for \( \mu^l \) will take the form

\[
d\frac{d}{dt} \mu^l = F(\mu^l) + G(\mu^l, \mu^H), \tag{2.7}
\]

where \( F(\cdot) \) and \( G(\cdot) \) are nonlinear functions of \( \mu^l \) and \( \mu^H \), and \( \mu^H \) is still uncentred. Although (2.6) and (2.7) are equivalent formulations of the moment system, we chose to work with (2.7) since it provided more accurate moment estimates than (2.6) in our tests.

In general, it is impossible to solve (2.7) analytically or numerically without knowing \( \mu^H \) at every point in time. A possible solution is to express \( \mu^H \) as a function of \( \mu^l \), for example by making an assumption about the distribution \( f(t, \mu^l) \) or other properties of the system. Following such a moment closure approach, one can write and solve a finite system of moment equations. The moment closure problem for biochemical networks has been extensively studied when the underlying dynamical system is described by a Markov process [23], but the same approaches can be applied in the ODE case considered here. However, moment closure methods are notoriously tricky in practice, as it is difficult to estimate their performance for a given dynamical system [24].

The main point of our work lies in avoiding making any assumptions about the state distribution \( f(t, \mu^l) \). Instead, we propose to approximate the value of \( G(\mu^l, \mu^H) \) using a small set of simulated trajectories to estimate \( \mu^H \). These trajectories have initial conditions that are randomly drawn from \( f_{\mu}(\mu^l) \), and produce the moment estimates \( \hat{\mu}^H \). In this way, the original system of low-order moments (2.7) could be closed and solved numerically:

\[
d\frac{d}{dt} \mu^l = F(\mu^l) + G(\mu^l, \hat{\mu}^H). \tag{2.8}
\]

The Monte Carlo scheme for the generation of \( \hat{\mu}^H \) will be described next.

### 2.4. Monte Carlo simulations

In order to estimate the moments \( \hat{\mu}^H \), a Monte Carlo simulation scheme needs to be employed. In general, Monte Carlo estimates the moments of interest using a sample of independent random draws from the distribution of initial conditions \( f_0(x) \) of (2.3). An intuitive MC scheme (NMC) for performing this estimation is presented in figure 2a: using \( N \) random draws from \( f_0(x) \), we simulate (2.3) for each initial condition over the time period of interest \([0, T]\) and employ the whole ensemble of \( N \) simulation runs to estimate the desired moments, \( \hat{\mu}^H_{NMC}(t_j) \), over a desirable time grid \([0, t_1, t_2, \ldots , t_l = T]\). Using the same simulations, one can also obtain estimates of \( \hat{\mu}^H_{NMC}(t_j) \).

In contrast to the NMC scheme, our algorithm will make use of an alternative procedure to estimate the moments of interest, which we will call ‘Split Monte Carlo’ (Split MC, figure 2b). Instead of using the whole sample of \( N \) initial conditions to estimate the moments at all time points, in Split MC we generate \( J \) disjoint subsets of the sample of initial conditions, where \( J \) is the number of points in the simulation time grid. Each subset will therefore have size \( N/J \), and in this way the \( N \) simulations will be equally split among all points of the
time grid \( \{0, t_1, t_2, ..., t_J = T\} \). At every time step \( t_k, k = 1, ..., J \) the moments \( \bar{\mu}^i(t_k) \) and \( \bar{\mu}^{Hi}(t_k) \) are estimated using \( N/J \) simulations from the original MC sample.

A graphical comparison of the outputs of these two MC schemes is presented in figure 2c. As can be observed, NMC provides a smooth trajectory estimate for a moment of interest, but the errors with respect to the reference trajectory at different time points are very strongly correlated. This happens because the use of a small sample size in the practical implementation of this estimator introduces a finite-sample bias (i.e. a deviation from the true moment values) at \( t = 0 \); and this bias is propagated to later time points through the system dynamics. On the other hand, Split MC uses a different fraction of the initial sample to calculate the moment of interest at each time point, and therefore the deviations from the reference at different time points are independent from each other. For the same reason, however, the moment estimates \( \bar{\mu}^i \) and \( \bar{\mu}^{Hi} \) are considerably more noisy compared to the NMC estimates due to the smaller sample size used at each time point. Note that, although both MC schemes employ asymptotically unbiased moment estimators, they are affected by finite-sample bias in different ways, which results in different noise statistics over time. Furthermore, note that the computational cost of both MC schemes is the same if they both use \( N \) randomly drawn initial conditions. For reasons that will become clear below, we will use the Split MC estimates in our moment propagation algorithm, and benchmark the accuracy of our algorithm against NMC.

Independently of the MC scheme used, getting reliable estimates of the high-order moments \( \mu^i \) in (2.8) would require a large number of MC simulations. However, the computational cost of these simulations would outweigh the benefit of solving the moment ODEs in the first place. To preserve the computational advantage of using the moment ODEs, one should therefore use a relatively small number of MC simulations to estimate \( \mu^i \). In that case, however, the resulting noisy estimates \( \bar{\mu}^i \) would introduce large errors to the system of low-order moments (2.8).

To reduce the uncertainty produced by the small number of MC simulations in (2.8) and construct reliable estimates of \( \mu^i \), we made use of extended Kalman filtering [21], an approximate nonlinear filtering method that is described in the following.

### 2.5. Extended Kalman filter

The Kalman filter is a ubiquitous state estimation algorithm which recursively estimates the state of a dynamical system based on noisy observations of a (function of) the system state. The EKF is a version of the original algorithm that is used when the underlying dynamical system is nonlinear [21]. Prior to presenting the EKF formulation of the moment tracking problem, we define the state update and measurement equations that will be used in the EKF.

To write the state update equation, we should observe that the substitution of \( \mu^i \) by a Monte Carlo estimate \( \bar{\mu}^i \) in (2.8) introduces noise to the system of low-order moments. We can therefore assume that the low-order moments evolve according to system (2.7) corrupted by process noise \( w(t) \)

\[
\frac{d}{dt} \mu^i = F(\mu^i) + G(\mu^i, \mu^{Hi}) + w(t). \tag{2.9}
\]

In this system, the noise-corrupted term, \( \bar{\mu}^{Hi} \), has been substituted by the sum of a noiseless term, \( \mu^{Hi} \) and the process noise \( w(t) \). The latter models the variability in the Monte Carlo estimates of \( \mu^{Hi} \) and is described by a zero-mean white noise source with (incremental) covariance matrix \( Q(t) \).

To define the vector of observations, we assume that at each time point \( t_k, k = 1, ..., J \) the Monte Carlo estimates of the low-order moments, \( \bar{\mu}^i(t_k) \) (which we will henceforth denote by \( y(t_k) \) to conform to the EKF notation), are related to the true values \( \mu^i(t_k) \) via the equation

\[
y(t_k) = \mu^i(t_k) + v_k, \quad v_k \sim N(0, R_k), \tag{2.10}
\]

where \( v_k \) is zero-mean Gaussian white observation noise with covariance matrix \( R_k \). When a subset of the low-order moments \( \mu^i(t_k) \) is used in the filtering problem, we can write the output \( y(t_k) \) as

\[
y(t_k) = H \cdot \mu^i(t_k) + v_k, \tag{2.11}
\]

where the matrix \( H \) ‘selects’ the appropriate elements of \( \mu^i(t_k) \) and the dimension of \( v_k \) is adjusted accordingly. Finally, we assume that the two noise processes \( (w \text{ and } v) \) are independent.

The goal of the EKF is to recursively estimate \( \mu^i \) at time \( t_k \) based on the corresponding (posterior) estimate of \( \mu^i \) at \( t_{k-1} \) together with the observation vector \( y(t_k) \). In this way, we obtain an estimate \( \hat{\mu}^i(t_k) \), of the low-order moments at time \( t_k \), as well as the accuracy of the obtained estimates represented by the covariance matrix \( P(t_k) \). If we assume that \( M \) low-order moments \( \mu^i \) need to be tracked in time, our EKF scheme consists of the following steps:

1. **Initialization**

   a) Assign an initial probability density function \( f_0(x) \) to the state vector \( x \) in (2.3), with low-order moments \( \mu^i(0) \in \mathbb{R}^M \) which are known precisely, i.e. with zero uncertainty \( P(0) = 0_{M \times M} \).

   b) Simulate (2.3) over the time period of interest \( [0, T] \) for \( N \) initial conditions randomly drawn from \( f_0(x) \) using the Split MC scheme. Estimate \( \bar{\mu}^i \) and \( \bar{\mu}^{Hi} \) over a discrete-time grid \( \{0, t_1, t_2, ..., t_{J-1}, T\} \), using \( N/J \) simulations at each time point. The low-order moment estimates \( \{\bar{\mu}^i(t_k), k = 1, \ldots, J\} \) serve as the observations \( y(t_k) \) in (2.10). Note that the Split MC scheme ensures that \( v_k \) and \( v_{k+1} \) in (2.10) are independent for \( n \neq m \). The same independence property holds for the noise in the high-order moment estimates \( \bar{\mu}^{Hi}(t_k) \).

   c) Use the (full/non-split) MC sample to estimate the process and measurement noise covariance matrices \( Q \) and \( R \) over the time grid \( \{0, t_1, t_2, ..., t_{J-1}, T\} \). This time, all \( N \) simulations can be used at each time point. In this way, we obtain estimates of \( \bar{Q}(t_k) \) and \( \bar{R}(t_k) \) at each point of the time grid, denoted by \( \bar{Q}(t_k) \) and \( \bar{R}(t_k) \), respectively. The construction of these matrices makes use of the theory of unbiased estimators for population moments, and a detailed description can be found in the Methods section.

2. **EKF Prediction step**

   At the \( k \)-th time step, our knowledge of the low-order moments given the current and past measurements \( y(t_l), l \leq k \) is summarized by the posterior mean \( \mu^i_{k|k} \) and the posterior covariance matrix \( P_{k|k} \) (at \( t = 0, \mu^i_{0|0} := \mu^i(0) \) and \( P_{0|0} = 0_{M \times M} \)). Given the posterior information at \( t_k \), we then calculate an a priori estimate of the low-order moments and their associated uncertainty at \( t_{k+1} \) by integrating the
following equations [21] from $t_k$ up to $t_{k+1}$

\[
\begin{align*}
\frac{d\mu^L}{dt} &= F(\mu^L) + G(\mu^L, \tilde{\mu}^H(t_k)), \\
\frac{dP}{dt} &= A(t) \cdot P + P \cdot A'(t) + B(t) \cdot \tilde{Q}(t_k) \cdot B'(t),
\end{align*}
\]

with

\[
\mu^L(t_k) = \mu_{L|k}^0, P(t_k) = P_{k|k}^0.
\]

(2.12)

where $A(t) = \partial F/\partial \mu^L$ is evaluated at $\mu^L(t_k)$, $B(t) = \partial G/\partial \mu^H$ is evaluated at $(\mu^L(t_k), \tilde{\mu}^H(t_k))$, and the prime symbol denotes the matrix transpose. Note that the continuous-time equations are integrated by holding the estimates of the high-order moments, $\tilde{\mu}^H(t_k)$, and the process covariance matrix, $\tilde{Q}(t_k)$, constant over the interval $[t_k, t_{k+1}]$. Numerical solution of (2.12) up to $t_{k+1}$ provides the a priori estimates $\hat{\mu}_{L|k+1}^L(t_{k+1})$ and $P_{k+1|k}$. 

3. EKF update step

After observing $y(t_{k+1})$, the estimates of the low-order moments and their uncertainty are updated to obtain the posterior estimates $\hat{\mu}_{L|k+1}^L(t_{k+1})$ and $P_{k+1|k+1} = P(t_{k+1})$ according to the following rules [21]:

\[
\begin{align*}
K_{k+1} &= P_{k+1|k} A'(t) \left(HP_{k+1|k} A'(t) + \tilde{R}(t_{k+1})\right)^{-1} \\
\hat{\mu}_{L|k+1}^L &= \hat{\mu}_{L|k}^L + K_{k+1} \left(y(t_{k+1}) - H\hat{\mu}_{L|k}^L \right) \\
\text{and} \\
P_{k+1|k+1} &= (I - K_{k+1} H) P_{k+1|k},
\end{align*}
\]

(2.13)

where $K_k$ is the (time-varying) Kalman gain and $I$ denotes the $M \times M$ identity matrix. The prediction and update steps are iterated until the end of the simulation horizon at $t_f = T$. A summary of the EKF algorithm is presented in box 2.

**Box 2. EKF algorithm summary.**

**Input:** Split MC simulations of low- and high-order moments $\{(\mu^L(t_k), \tilde{\mu}^H(t_k)), k = 1, \ldots, J\}$ over time grid $[0, t_1, t_2, \ldots, t_J] = T$; low-order moments at $t = 0$: $\mu^L(0)$

**Initialization:** Set $\mu_{L|0}^L = \mu^L(0)$, $P_{0|0} = 0_{M \times M}$, $k = 0$

**Estimate covariance matrices**

\[\{\tilde{Q}(t_k), \tilde{R}(t_k), k = 1, \ldots, J\}\]

**While** $k < J$ do

Set $\hat{\mu}_{L|k}^L = \mu_{L|k}^L, P(t_k) = P_{k|k}^0$

**Prediction step:** solve

\[
\begin{align*}
\frac{d\mu^L}{dt} &= F(\mu^L) + G(\mu^L, \tilde{\mu}^H(t_k)), \\
\frac{dP}{dt} &= A(t) \cdot P + P \cdot A'(t) + B(t) \cdot \tilde{Q}(t_k) \cdot B'(t)
\end{align*}
\]

for $t \in [t_k, t_{k+1}]$

**Update step:**

\[
\begin{align*}
K_{k+1} &= P_{k+1|k} A'(t) \left(HP_{k+1|k} A'(t) + \tilde{R}(t_{k+1})\right)^{-1} \\
\hat{\mu}_{L|k+1}^L &= \hat{\mu}_{L|k}^L + K_{k+1} \left(y(t_{k+1}) - H\hat{\mu}_{L|k}^L \right) \\
\text{and} \\
P_{k+1|k+1} &= (I - K_{k+1} H) P_{k+1|k},
\end{align*}
\]

Set $k = k + 1$

**Output:** EKF posterior estimates of low-order moments and associated uncertainty: $\{(\mu_{L|k}^L, P_{k|k}), k = 1, \ldots, J\}$.
forms ($H_p$ and $R_p$), as well as the complexes $H_pR$ and $R_pH$ (section S1.1, electronic supplementary material). Since in the monofunctional design the histidine kinase $H$ does not modulate the dephosphorylation of $R_p$, the complex $R_pH$ is absent in this model, and dephosphorylation is carried out by an independent phosphatase (section S1.3, electronic supplementary material).

Experimental measurements have revealed large cell-to-cell variability in the concentrations of histidine kinase and response regulator in bacteria, as well as different degrees of basal activity in the unstimulated system. We therefore postulated that all initial conditions of the system prior to the onset of the stimulus are uncertain. To express this uncertainty, we assumed that all states at $t = 0$ follow independent lognormal distributions, whose parameters were defined based on literature evidence [26] (electronic supplementary material, sections S1.1 and S1.3). As we were interested in tracking the mean and variance of every state, the bifunctional TCS gave rise to 12 moment equations, while there were 10 equations for the monofunctional form. The moment equations of the bifunctional and monofunctional systems contained 16 and 10 high-order moment terms, respectively (electronic supplementary material, sections S1.2 and S1.4). To get reliable estimates of the actual trajectories of the mean and variance of the states in each system, we simulated the bi- and monofunctional TCS models with 400,000 Monte Carlo simulations. These reference moment trajectories acted as the ‘ground truth’ against which we compared the performance of the EKF.

We first evaluated the accuracy of our algorithm by varying the number input MC simulations. These simulations are used to generate the moment estimates needed by the EKF (box 2) via the Split MC scheme (§2.4). We investigated the sensitivity of the moment estimates with respect to changes in the input MC sample size by calculating the root mean squared error (RMSE) between the reference trajectories and the EKF-estimated values for the tracked low-order moments (Methods). Figure S1 of the electronic supplementary material summarizes the RMSE for the moment estimates of the bifunctional TCS model for different sizes of the input MC sample. The RMSE decreases rapidly as the input MC sample size increases, with 500 MC simulations providing small RMSE values for all tracked moments.

Since the small number of input MC simulations may render the EKF performance sensitive to the particular Monte Carlo sample, we next assessed the robustness of the mean and variance estimates in 50 independent runs of the EKF algorithm, each with a different input Monte Carlo sample of size 500. Besides the Split MC scheme described in §2.4, we also employed an additional resampling step (Methods) to boost the accuracy of the input moment estimates at each time point. Figures 4 and 5 display summaries of the EKF posterior trajectories for the mean and variance of TCS states. The EKF-based moment estimation algorithm captured the time evolution of the low-order TCS moments with high accuracy and minimal variability from run to run.

Several studies have shown that the concentration of $R_p$ in a bifunctional system is robust to variations in the concentration of the histidine kinase and the response regulator itself, when the response regulator is present in large excess compared to the kinase [27,28]. This property is not present in monofunctional systems [26]. To study the differences in the variance of $R_p$ between the bifunctional and monofunctional TCS designs, the dephosphorylation rate of the monofunctional system was adjusted to achieve the same mean $R_p$ levels in the two systems at steady-state, while keeping the initial distributions of all other states the same. As figure 5 shows the variance of $R_p$ in the bifunctional system is indeed smaller compared to the monofunctional scheme.

Next, we compared the accuracy of our algorithm with moment closure. For this comparison, we extended the original uncertainty propagation problem for the bifunctional TCS by additionally assuming that two parameters, the activation rate of the kinase ($k_i$) and the phosphorylation rate of the response regulator ($k_o$) are uncertain and follow independent lognormal distributions whose parameterization is defined in section S1.1 of electronic supplementary material. Figures S2 and S3 in electronic supplementary material show that the evolution of the means and variances of the TCS model obtained from our algorithm is considerably less biased than the estimates obtained from a multivariate lognormal moment closure scheme [20].

Finally, we turned to the use of linearization in the covariance propagation step of the EKF, which is a potential source of inaccuracy. To improve accuracy of covariance propagation, other Kalman filtering approaches have been proposed, such as the unscented Kalman filter (UKF) [29]. The UKF uses a deterministically selected set of ‘sigma points’ which are propagated in time using the original nonlinear dynamics instead of the linearization. However, as figures S4 and S5 of electronic supplementary material show, both the EKF and UKF displayed very similar performance, while the EKF produced more accurate moment estimates (especially, for the variances) during the fast initial phase of the response.

3.2. A MAPK cascade

Having established the functionality and robustness of our moment estimation algorithm with a small system, we evaluated its performance with a larger model for a mitogen-activated protein kinase (MAPK) cascade, described by 25 ODEs. The MAPK pathway is a ubiquitous signalling mechanism in eukaryotes, and consists of three protein kinases as shown on figure 6. The last kinase of this system is the MAPK (mitogen-activated protein (MAP) kinase). MAPK is activated upon phosphorylation by MEK (MAPKK, MAP kinase kinase), and MEK is in turn phosphorylated and activated by RAF (MAPKKK, MAP kinase kinase). This uppermost node receives several activating or inhibitory signals, both from upstream regulators and downstream nodes (e.g. MAPK) via feedback interactions. In the example considered here, active MAPK negatively regulates RAF by inducing an inhibitory phosphorylation, which results in the creation of a negative feedback loop [30].

To arrive at our final model equations, we extended the model presented in [31] (MODEL6615234250 in the BioModels database) by adding the negative interaction between doubly phosphorylated MAPK and RAF. The model simulates the rapid onset of an input signal to the uppermost node of the network (RAF) and its propagation through the signalling cascade. The full model equations are presented section S1.5 of the electronic supplementary material. We assumed that seven states with non-zero initial conditions (out of 25 states in total) are subject to uncertainty, which was expressed by independent lognormal distributions. Each lognormal distribution had a fixed mean equal to the nominal initial condition $IC$ provided with the model and a variance equal
To track the mean and variance of every MAPK state, a set of 50 moment equations was used (section S1.6, electronic supplementary material). This system of equations carried 84 additional high-order moments that had to be estimated via MC. To obtain reliable estimates of the desired low-order moments and form a ground truth for testing our algorithm, we performed 200 000 Monte Carlo simulations of the MAPK model with initial conditions randomly drawn from their corresponding distributions. We then compared the EKF performance in terms of accuracy and robustness to NMC [16].

To carry out this comparison, we let both algorithms use the exact same sets of input Monte Carlo samples. As a measure of relative performance, we calculated the RMSE between the ground truth moment trajectories and the estimates provided by the EKF or NMC (Methods). Our main focus was the time evolution of the mean and variance of all chemical species with non-zero abundance prior to signal onset, since these are the species that are mostly affected by initial condition uncertainty. To investigate the sensitivity of both methods to the input MC sample, we calculated the RMSE of the means and variances of all states with non-zero initial conditions for different input sample sizes, using 50 independent input MC runs for each sample size. The accuracy of the moment estimates is summarized in figures S6 and S7 in electronic supplementary material.

Overall, both the EKF-based method and NMC performed better with larger sample size in terms of accuracy. However, the RMSE values of the EKF were lower even for very small input MC datasets. Furthermore, the EKF estimates were overall less sensitive to the specific MC input sample compared to NMC, as can be seen by the tighter distribution of the RMSE values. Inspection of the plots of figures S6 and S7 shows that MC samples of size 500 offered a good compromise between computational costs and accuracy for the EKF-based method.

To compare how the EKF and NMC perform on a given input MC sample, we plotted their RMSE values obtained from 50 independent input MC runs with 500 simulations each. From figures 7 and 8, we can see that for all states with non-zero initial conditions the EKF achieved a smaller RMSE compared to NMC for the same input MC sample. Figure 9 illustrates in more detail the performance of the two-moment estimation algorithms with a MC input size of 500 simulations by showing the distribution of estimated moment trajectories versus the ground truth for three chemical species in the MAPK cascade. Consistent with the results

Figure 4. EKF-based trajectories of the mean and variance of the concentration of $H$, $H_p$, and $R$ for bifunctional (blue) and monofunctional (green) TCS. The colour bands display distribution percentiles of the EKF estimates over 50 independent runs where the percentiles range from 10th up to 90th with darker bands being close to the median and lighter bands close to the tails. Each run had an input Monte Carlo sample of size 500. The ground truth (reference) moment trajectories were obtained from 400 000 MC simulations.
of the strip plots (electronic supplementary material, figures S6 and S7), the distribution of the NMC moment estimates is typically wider than the EKF-based results, highlighting the fact that our EKF-based algorithm gives on average more consistent results compared to NMC.

The accuracy of the NMC can be increased by running a larger number of simulations, which will in turn increase the running time the method. On the other hand, the construction of the \( Q \) and \( R \) matrices for the EKF may also entail some computational cost, especially for large dynamical systems with many high-order moments. Therefore, depending on the size and complexity of the underlying dynamical system (which largely determines the run time of each model run), the efficiency of the ODE solver and the degree of parallelization that is possible, NMC may outperform the EKF approach in terms of running time for a given accuracy level. In such a case, the cost of our EKF-based algorithm can be reduced by decreasing the time needed to construct the noise covariances in at least two ways: (i) by estimating diagonal instead of full \( Q \) and \( R \) covariances from the MC input data, which greatly reduces the number of elements that need to be estimated at each time point, and (ii) by evaluating \( Q \) and \( R \) over a time grid that is much sparser than the EKF time grid. In the second case, matrix values at intermediate time points can be obtained using linear interpolation. Since the set of positive definite matrices is convex, the interpolated matrices will still be positive definite.
Figure S8 in electronic supplementary material shows that when diagonal covariance matrices were used in the EKF, the accuracy of the filter was somewhat reduced compared to an EKF that used full covariances. On the other hand, electronic supplementary material, figure S9 shows that interpolation of the covariance matrices on a sparse time grid that is a hundred times sparser than the original EKF time grid, produced moment estimates that were almost indistinguishable from the estimates based on the full EKF time grid.

Finally, we investigated whether we could improve upon the performance of the EKF by replacing it with the UKF. However, in this case, we encountered one of the key weaknesses of the UKF, related to the preservation of semi-positive definiteness of the posterior covariance matrix. Even when the polar decomposition was used to partially overcome this problem [32], the overall performance of the UKF was still quite poor.

4. Discussion

In this work, we addressed the problem of uncertainty propagation for ODE-based models of biochemical reaction networks with polynomial rates, such as systems with mass-action kinetics [23]. Using a set of low-order moments to summarize the state distribution of these systems, we introduced a new algorithm to trace the time evolution of these moments. Our approach is based on the formulation of a filtering problem to combine the information provided by a small number of Monte Carlo simulations and the low-order moment differential equations. In contrast to moment closure methods, our EKF-based algorithm does not rely on specific assumptions about the distribution of the states, and is thus more broadly applicable.

Contrary to [22], where EKF-based moment estimation for Markov chain models had been presented, the underlying dynamical system here is deterministic, which necessitated a
new Monte Carlo sampling scheme (Split MC) for the operation of the EKF. More crucially, however, the ad hoc simplifying assumptions of [22] about the form of the $Q$ and $R$ matrices and the relative strengths of process and measurement noise led invariably to poor results, with moment estimates becoming very noisy or even diverging to infinity. This fact suggests that EKF-based moment filtering for deterministic systems is sensitive to mis-specification of $Q$ and $R$.

To overcome this challenge and produce a robustly functional filter, we implemented unbiased moment estimators which directly estimate the (co)variances of the process and measurement noise vectors. Since we construct $Q$ and $R$ from the same set of Monte Carlo simulations that are used as input to the EKF, no additional Monte Carlo runs are needed to estimate the noise statistics.

Our analysis of the large MAPK system showed that our EKF-based algorithm outperforms the NMC approach for moment estimation by producing more accurate and robust estimates of the desired moments for the same number of Monte Carlo runs. It should be noted, however, that a comparison of the two methods in terms of computational cost at this point is more complicated (cf. implementation details in §5.1 of Methods). The simple computational tricks presented in the previous section can greatly speed up the EKF runs, should that be required in a particular case. Exploring additional options for further optimization of the EKF performance and accuracy is an interesting topic of future work.

We believe that the combination of Monte Carlo simulation with Kalman filtering to estimate low-order moments of uncertain ODE-based biochemical systems is a very promising approach to uncertainty propagation, which can be further developed in the future. Though we only focused on biochemical networks with polynomial rates in this work, the derivation of moment equations for systems with rational rates, which
5. Methods

5.1. Used software and implementation details
The derivation of moment equations for a given set of polynomial ODEs was completely automated using the Symbolic Toolbox of MATLAB. All Monte Carlo simulations of the models considered in the examples were performed using the IQM Tools v. 1.2.1 MATLAB toolbox [34,35] (IntiQuan GmbH, Basel), which makes use of the SUNDIALS CVODES solver [36]. All other necessary computations for the EKF (covariance matrix estimation, filtering steps) were performed in MATLAB R2018b.

The fact that the IQM Tools ODE solver is coded in C while our EKF algorithm was coded in MATLAB, does not make it at this point possible to directly compare the efficiencies of NMC and our EKF-based approach. This is due to the fact that a model run in C is around 50 times faster than a model run in MATLAB, while the EKF requires the estimation of $Q$ and $R$ matrices, which is also carried out in MATLAB. A complete implementation of the EKF pipeline in C should enable the comparison of its performance to C-based ODE integration.

5.2. Split Monte Carlo with resampling
As discussed in §2.4, the Split MC simulation scheme is used to ensure that $v(t_k)$ and $v(t_m)$ in (2.9) (as well as $v(t_k)$ and $v(t_m)$ in (2.10)) are independent whenever $k \neq m$. However, since at every time point $t_k$, $k = 1, \ldots, J$ only $N/J$ trajectories of the total MC sample of size $N$ are used to produce the estimates of $\tilde{\mu}^J(t_k)$ and $\tilde{\mu}^H(t_k)$ moments, this sampling scheme can produce very noisy moment estimates when $N/J$ becomes very small (e.g. smaller than 10).

To reduce the noise in the moment estimates, we introduced a resampling procedure whose idea is similar to bootstrapping. Specifically, at each time point $t_k$, we sample with replacement $k_N$, ($k_N < N$) trajectories from the original input MC sample of size $N$ to produce $\tilde{\mu}^J(t_k)$ and $\tilde{\mu}^H(t_k)$, $k_N$ can be larger than $N/J$, leading to smaller variability in the moment estimates. However,
contrary to the conventional bootstrap, $k_N$ should still be a small fraction of $N$ to ensure minimal dependence between moment estimates at different time points, and minimize the bias that can be introduced across time by the resampling process. This is necessary because our EKF implementation assumes that the noise vectors at different time points are fully independent and zero-mean.

5.3. Extended Kalman filter tuning

One challenging aspect of the EKF implementation is the choice of its design parameters, i.e. the covariance matrices for the process and observation noise components at each time step $t_k (Q(t_k)$ and $R(t_k)$, respectively), as well as the initial state estimate and its associated covariance. Proper choice of these parameters is crucial for both the stability and accuracy of the EKF [37].

In the EKF application considered here, the initialization of $\mu^0(t)$ and its associated covariance matrix $P(0)$ is straightforward, since the initial probability density function $f_0(x)$ of the system state is known. Therefore, the initial values of the low-order moments $\mu^0(t)$ are known precisely, which also implies that $P(0) = 0_{M \times M}$ (cf. with box 2 in the main text).

At each time point $t_k$, the covariance matrices $R(t_k)$ and $Q(t_k)$ represent the uncertainty associated with the Monte Carlo-based moment estimators $\mu^R(t_k)$ and $\mu^Q(t_k)$, respectively (cf. with §2.5). More concretely, the matrix $R(t_k)$ represents the covariance structure of the low-order moment estimates $\mu^R(t_k)$ comprising the output vector $y(t_k)$, while $Q(t_k)$ describes the covariance of the high-order moments $\mu^Q(t_k)$ entering the low-order moment equations.

Given that $\mu^R(t_k)$ and $\mu^Q(t_k)$ contain estimates of the true moments $\mu_R(t_k)$ and $\mu_Q(t_k)$, the covariance matrices $R(t_k)$ and $Q(t_k)$ will contain the covariances of these estimators. The precise form of these covariance expressions can be quite complex, especially when high-order (cross-)moments are considered. We therefore made use of the mathStatica tool [38] to derive them symbolically in Mathematica. The sample-based version of these covariance expressions were then calculated in two steps: first, the unbiased estimates of the necessary population moments were obtained using the full input Monte Carlo sample of size $N$. Then, the (co)variance estimates were calculated for the smaller Monte Carlo sample of size $k_N$ used at each time point $t_k$ in the context of Split MC and resampling schemes.

Below we illustrate the two-step construction of $R(t_k)$ and $Q(t_k)$, starting with the population moment estimators. All sample-based quantities are computed at a given time point $t_k$, dependence on $t_k$ has been removed to simplify the notation.

1. Unbiased estimator of the population product moment of the $p$-th power of $x_k$, the $q$-th power of $x_p$, the $r$-th power of $x_q$, and the $s$-th power of $x_r$:

$$\tilde{\mu}_{ijk} = \frac{1}{N} \sum_{k=1}^{N} x_k^p x_k^q x_k^r x_k^s .$$

Setting any of the exponents to zero in the above formula provides a product moment estimator for a subset of these four variables.

2. Unbiased estimator of the product of two population product moments: the product moment of the $p$-th power of $x_k$ and the $q$-th power of $x_q$, and the product moment of the $r$-th power of $x_r$ and the $s$-th power of $x_s$:

$$\tilde{\mu}_{ijkl} = \frac{1}{N(N-1)} \left( \sum_{k=1}^{N} x_k^p x_k^q - \sum_{k=1}^{N} x_k^r x_k^s \right) .$$

— Unbiased estimator of the product of three population moments: the product moment of $x_i$ and $x_j$, the first moment of $x_k$, and the first moment of $x_l$:

$$\tilde{\mu}_{ijkl} = \frac{1}{N(N-1)(N-2)} \left( \sum_{k=1}^{N} x_k x_k x_k x_k - \sum_{k=1}^{N} x_k x_k x_k x_k + 2 \sum_{k=1}^{N} x_k x_k x_k x_k \right) .$$

— Unbiased estimator of the product of four population moments: the first moment of $x_i$, the first moment of $x_j$, the first moment of $x_k$, and the first moment of $x_l$:

$$\tilde{\mu}_{ijkl} = \frac{1}{N(N-1)(N-2)(N-3)} \left( \sum_{k=1}^{N} x_k x_k x_k x_k - \sum_{k=1}^{N} x_k x_k x_k x_k + 2 \sum_{k=1}^{N} x_k x_k x_k x_k \right) .$$

— Unbiased estimator of the product of two sample means, $\tilde{\mu}_{ij}^{[1]}$ and $\tilde{\mu}_{ij}^{[2]}$:

$$\text{Cov} \left( \tilde{\mu}_{ij}^{[1]}, \tilde{\mu}_{ij}^{[2]} \right) = \frac{1}{k_N} (\tilde{\mu}_{ij}^{[1]} - \tilde{\mu}_{ij}^{[1]}) .$$

3. Unbiased estimator of the covariance of two sample variances, $\tilde{\sigma}_{ij}^{[1]}$ and $\tilde{\sigma}_{ij}^{[2]}$:

$$\text{Cov} \left( \tilde{\sigma}_{ij}^{[1]}, \tilde{\sigma}_{ij}^{[2]} \right) = \frac{1}{k_N} \left( \tilde{\sigma}_{ij}^{[1]} - \tilde{\sigma}_{ij}^{[1]} \right) .$$

4. Unbiased estimator of the covariance of two product moments, $\tilde{\mu}_{ij}^{[1]}$ and $\tilde{\mu}_{ij}^{[2]}$:

$$\text{Cov} \left( \tilde{\mu}_{ij}^{[1]}, \tilde{\mu}_{ij}^{[2]} \right) = \frac{1}{k_N} \left( \tilde{\mu}_{ij}^{[1]} - \tilde{\mu}_{ij}^{[1]} \right) .$$

The same estimators can be used to calculate some of the entries of $Q$. In that case, however, an additional estimator is required:

$$\tilde{\mu}_{ij}^{[1]} = \frac{1}{k_N} \left( \tilde{\mu}_{ij}^{[1]} - \tilde{\mu}_{ij}^{[1]} \right) .$$

The last formula can be extended in a straightforward way to the estimate covariance product moments, where each product moment involves more than two states.

It should be noted that our method treats uncertain parameters $\theta \in \mathbb{R}^p$ as additional uncertain states with zero dynamics (equation (2.2)). Furthermore, in the example presented in §3.1 the low-order moments of each parameter $\theta_i$ were introduced in the vector of observed moments $y$ (equation (2.10)).
estimators of $\mu_k$ and $\theta_k$, as well as covariances of estimators of state and parameter moments. The latter group of entries need to be estimated via Monte Carlo sampling (as described above), since the joint probability density function over the states and parameters, $f(t, z, \theta)$, is unknown for $t > 0$. The former group can be either estimated via Monte Carlo, or calculated exactly given the known (joint) parameter distribution. Note, however, that the second option may compromise EKF accuracy because of the disagreement between the theoretical parameter distribution and the empirical distribution defined via Monte Carlo. Therefore, the first, more stable option (Monte Carlo) is recommended.

Parameter uncertainty will also enter the low-order moment equations of the states $z \in \mathbb{R}^t$ through the expectations of products of the form $E_{\theta_k}[\theta_k z_k]$, where $i = 1, \ldots, p; j, k = 1, \ldots, s$. Since the joint distribution $f(t, z, \theta)$ is unknown for $t > 0$, these high-order moments will also be estimated via Monte Carlo simulation, using a sample from the known joint distribution of initial conditions and parameters. Matrix $Q$ contains (co)variances of the moment estimators that are used to estimate $\mu_k^{(1)}$, $\mu_k^{(1,1)}$, etc., where $E_{\theta_k}[\theta_k z_k^{(1)}] = E_{\theta_k}[\theta_k z_k]$. Overall, uncertain parameters are treated exactly like uncertain states in the low-order moment equations and in the construction of $Q$.

5.4. Extended Kalman filter settings

Table 1 summarizes the settings of the EKF algorithm which were used to produce the results for the TCS model (figures 4 and 5) and the MAPK model (figure 9):

| Model | Input MC | Resampling | Time step | Noise $\sigma$ | Observation |
|-------|----------|------------|-----------|---------------|-------------|
| TCS   | $N = 500$ | $u_0 = 100$ | 0.1       | -             | -           |
| MAPK  | $N = 500$ | $u_0 = 100$ | 0.5       | -             | -           |

5.5. Evaluation of EKF and NMC performance

To evaluate the moment estimation accuracy of the EKF-based and NMC methods, we used the root mean squared error (RMSE). This quantity was formed by the sum of the squared deviations between the estimated low-order moments $\hat{\mu}_k(t)$ and their reference (ground truth) values $\mu_k(t)$ at every time point $t_k$ of the simulation time grid as follows:

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{k=1}^{M} (\mu_k(t) - \hat{\mu}_k(t))^2}.$$ 

Data accessibility. This article has no additional data.

Authors’ contributions. T.K. and A.M.-A. designed the project, developed the method and wrote the paper. T.K. performed the computations.

Competing interests. We declare we have no competing interests.

Funding. This work was partly financed by the Netherlands Organization for Scientific Research (NWO) through a VIDI grant to A.M.-A. (project no. 016.Vidi.189.116).

References

1. Klipp E, Liebermeister W, Wierling C, Kowald A. 2016 Systems biology: a textbook. Weinheim, Germany: Wiley-VCH.
2. Vanlier J, Tiemann CA, Hilbers PAJ, van Riel NAW. 2013 Parameter uncertainty in biochemical models described by ordinary differential equations. Math. Biosci. 246, 305–314. (doi:10.1016/j.mbs.2013.03.006)
3. Llamos A, Gonzalez-Vargas AM, Versani C, Cinquemani E, Ferrari-Trecate G, Hersen P, Batt G. 2016 What population reveals about individual cell identity: single-cell parameter estimation of models of gene expression in yeast.PLoS Comput. Biol. 12, 1–18. (doi:10.1371/journal.pcbi.1004706)
4. Walther S. 2018 Estimation methods for heterogeneous cell population models in systems biology. J. R. Soc. Interface 15, 20180530. (doi:10.1098/rsif.2018.0530)
5. van Riel NAW. 2006 Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments. Brief. Bioinform. 7, 364–374. (doi:10.1093/bib/bbl040)
6. Spencer SL, Gaudet S, Albeck JG, Burke JM, Sorger PK. 2009 Non-genetic origins of cell-to-cell variability in TRAIL-induced apoptosis. Nature 459, 428–432. (doi:10.1038/nature08012)
7. Albeck JG, Burke JM, Aldridge BB, Zhang M, Lauffenburger DA, Sorger PK. 2008 Quantitative analysis of pathways controlling extrinsic apoptosis in single cells. Mol. Cell 30, 11–25. (doi:10.1016/j.molcel.2008.02.012)
8. Ehrendorfer M. 2006 In The Liouville equation and atmospheric predictability (eds T Palmer, R Hagedorn), pp. 59–98. Cambridge, UK: Cambridge University Press.
9. Weiße AT, Middleton RH, Huisingsa W. 2010 Quantifying uncertainty, variability and likelihood for ordinary differential equation models. BMC Syst. Biol. 4, 1–10. (doi:10.1186/1752-0509-4-1)
10. Weiße AT, Huisingsa W. 2011 Error-controlled global sensitivity analysis of ordinary differential equations. J. Comput. Phys. 230, 6824–6842. (doi:10.1016/j.jcp.2011.05.011)
11. Horenko I, Lorenz S, Schütte C, Huisingsa W. 2005 Adaptive approach for nonlinear sensitivity analysis of reaction kinetics. J. Comput. Chem. 26, 941–948. (doi:10.1002/jcc.20234)
12. Schillings C, Sunnåker M, Stelling J, Schwab C. 2015 Efficient characterization of parametric uncertainty of complex (bio)chemical networks.PLoS Comput. Biol. 11, 1–16. (doi:10.1371/journal.pcbi.1004457)
13. Gao Z, Hesthaven JS. 2011 Efficient solution of ordinary differential equations with high-dimensional parameterized uncertainty. Commun. Comput. Phys. 10, 253–278. (doi:10.4208/cicp.091010.s0910a)
14. Xu Y, Milli L, Sandu A, von Spakovsky MR, Zhao J. 2019 Propagating uncertainty in power system dynamic simulations using polynomial chaos. IEEE Trans. Power Syst. 34, 338–348. (doi:10.1109/TPWRS.2018.2865548)
15. Boas SE, Jimenez MIN, Merks RM, Blom JG. 2015 A global sensitivity analysis approach for morphogenesis models. BMC Syst. Biol. 9, 1–15. (doi:10.1186/s12918-014-0137-8)
16. Leith CE. 1974 Theoretical skill of Monte Carlo forecasts. Mon. Weather Rev. 102, 400–418. (doi:10.1175/1520-0493(1974)102<0409:TSOMCF>2.0.CO;2)
17. Bover DCC. 1978 Moment equation methods for stochastic kinetic models. J. Chem. Phys. 68, 4914–4930. (doi:10.1063/1.441147)
18. Epstein JS. 1969 Stochastic dynamic prediction. Tellus 21, 739–759. (doi:10.3402/tellusa.v21i6.10143)
19. Kuehn C. 2016 Moment closure—a brief review. In Control of self-organizing nonlinear systems, pp. 253–271. New York, NY: Springer.
20. Lakatos E, Ale A, Kirk PDP, Stumpf MPH. 2015 Multivariate moment closure techniques for stochastic kinetic models. J. Chem. Phys. 143, 094107. (doi:10.1063/1.4929337)
21. Simon D. 2006 Optimal state estimation: Kalman, H∞, and nonlinear approaches. Hoboken, NJ: John Wiley & Sons.
22. Russ J, Millius-Argetis A, Summers S, Lygeros J. 2011 Moment estimation for chemically reacting systems by extended Kalman filtering. J. Chem. Phys. 135, 165102. (doi:10.1063/1.3654133)
23. Munsky B, Hlavacek WS, Tsimring LS. 2018 Quantitative biology: theory, computational methods, and models. Cambridge, UK: MIT Press.
24. Schnoerr D, Sanguinetti G, Grima R. 2015 Comparison of different moment-closure approximations for stochastic chemical kinetics. J. Chem. Phys. 143, 185101. (doi:10.1063/1.4934990)
25. Batchelor E, Goulian M. 2003 Robustness and the cycle of phosphorylation and dephosphorylation in a two-component regulatory system. Proc. Natl Acad. Sci. USA 100, 691–696. (doi:10.1073/pnas.0234782100)
26. Igoshin OA, Alves R, Savageau MA. 2008 Hysteretic and graded responses in bacterial two-component signal transduction. Mol. Microbiol. 68, 1196–1215. (doi:10.1111/j.1365-2958.2008.06221.x)
27. Shinar G, Milo R, Martinez MR, Alon U. 2007 Input–output robustness in simple bacterial signaling systems. Proc. Natl Acad. Sci. USA 104, 19931–19935. (doi:10.1073/pnas.0706792104)
28. Maity AK, Bandyopadhyay A, Chaudhury P, Banik SK. 2014 Role of functionality in two-component signal transduction: a stochastic study. Phys. Rev. E 89, 032713. (doi:10.1103/PhysRevE.89.032713)
29. Julier SJ, Uhlmann JK. 1997 New extension of the Kalman filter to nonlinear systems. In Signal processing, sensor fusion, and target recognition VI, vol. 3068, pp. 182–193. International Society for Optics and Photonics.
30. Kholodenko BN, Hancock JF, Kolch W. 2010 Signalling ballet in space and time. Nat. Rev. Mol. Cell Biol. 11, 414–426. (doi:10.1038/nrm2901)
31. Levchenko A, Bruck J, Sternberg PW. 2000 Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. Proc. Natl Acad. Sci. USA 97, 5818–5823. (doi:10.1073/pnas.97.11.5818)
32. Higham NJ. 1988 Computing a nearest symmetric positive semidefinite matrix. Linear Algebra Appl. 103, 103–118. (doi:10.1016/0024-3795(88)90223-6)
33. Sauro HM. 2011 Enzyme kinetics for systems biology. Seattle, WA: Ambrosius Publishing.
34. IQM Tools. See http://www.intiquan.com/iqm-tools/.
35. Schmidt H, Jirstrand M. 2006 Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. Bioinformatics 22, 514–515. (doi:10.1093/bioinformatics/bti799)
36. CVODES: the sensitivity-enabled ODE solver in SUNDIALS.
37. Schneider R, Georgakis C. 2013 How to not make the extended Kalman filter fail. Ind. Eng. Chem. Res. 52, 3354–3362. (doi:10.1021/ie300415d)
38. Rose C, Smith MD. 2002 MathStatica: mathematical statistics with Mathematica. In Compstat, pp. 437–442. New York, NY: Springer.