I. INTRODUCTION

The description of a high-dimensional system by a model of reduced dimensionality is a well-established concept in classical and quantum mechanics.\textsuperscript{1,2} The treatment consists of two tasks: the decomposition of the total system in a relevant part (the “system”) and an irrelevant part (the “bath”) and the description of the dynamics of the system variables under the influence of the bath variables. Because the two tasks of the treatment are interdependent, the decomposition is a crucial step. For example, by restricting oneself to a too simple model of the relevant part, the reduced dynamics of this system can become quite complicated. This is because system and bath degrees of freedom are not clearly separated from each other and may thus be strongly coupled due to the absence of a clear time scale separation of system and bath variables.

In this work, we are concerned with the modeling of data from classical molecular dynamics (MD) simulations.\textsuperscript{3} Increasing computer power and improved computational methods now allow us to simulate systems up to \( \approx 10^8 \) atoms and to study biomolecular processes such as molecular recognition, folding, and aggregation up to a microsecond time scale. A popular strategy to analyze the resulting huge data sets is again to invoke a decomposition of the dynamics. This can be done, for example, by choosing some (in general, multidimensional) “reaction coordinate” \( x \) and considering the free energy landscape,\textsuperscript{4–8}

\[
\Delta G(x) \approx -k_B T \ln P(x),
\]

of the system, where \( P \) is the probability distribution of the MD trajectory along \( x \). Popular choices for the coordinate \( x \) include empirical “order parameters” such as the fraction of native contacts and the radius of gyration as well as linearly transformed coordinates obtained from a principal component analysis (PCA).\textsuperscript{9–12}

To account for the dynamics of the resulting reduced system (i.e., the second task), a generalized Langevin equation of the form

\[
x(t) = \int_0^t K(\tau)x(t-\tau)d\tau + F(t)
\]

can be derived by employing projection operator techniques.\textsuperscript{7} At the expense of a nonlocal bath memory kernel \( K(t) \), the generalized Langevin equation provides the exact time evolution of —an in principle arbitrary choice of— \( x(t) \). Recently, several attempts have been reported that approximately determine the memory kernel for biomolecular applications.\textsuperscript{13,14} Alternatively, one may try to choose the reaction coordinate \( x(t) \) such that it contains all slow large-amplitude motions of the molecule, while the bath coordinates only account for its high-frequency fluctuations. As a consequence of this time scale separation, it may be assumed that the memory kernel decays sufficiently fast such that the standard Langevin equation\textsuperscript{15}

\[
x(t) = h(x) + F(t)
\]

is recovered. Here the time evolution of \( x(t) \) is determined by
the deterministic field $h(x)$ and the stochastic driving $F(t)$. By including all important motion in the definition of $x(t)$, we have thus obtained a simpler form of the equation of motion at the expense of a higher dimensionality of the relevant system.

In a series of papers, we have recently studied the structure and conformational dynamics of short alanine peptides.\textsuperscript{16–22} Employing a dihedral angle-based PCA\textsuperscript{18,19} the free energy landscape was represented by the first few principal components $x_i(t)$ ($i = 1, \ldots, d$) of the MD trajectory. It has been shown\textsuperscript{18–21} that typically $d = 3–10$ principal components are required to incorporate all slow conformational motion in the reaction coordinate $x = (x_1, \ldots, x_d)^T$. As an illustrative example, Fig. 1 shows the time evolution of the first and tenth principal component, obtained from a 800 ns MD simulation of heptaalanine in explicit water.\textsuperscript{21} Requiring the fast large-amplitude motion of the reaction coordinates, it was found that in this case the reaction coordinate needs to include the first five principal components; hence $x_1(t)$ is a component of the reaction coordinate, while $x_{10}(t)$ represents a typical bath coordinate. Figure 1 clearly contrasts the slow large-amplitude motion of the reaction coordinate $x_1(t)$ with the high-frequency fluctuations of the bath coordinate $x_{10}(t)$. The former describes transitions between metastable conformational states of the peptide, while the latter accounts for fluctuations within such a metastable state. Thus, Fig. 1 nicely demonstrates the time and length scale separation between system and bath.

While the above discussed decomposition of the dynamics should facilitate a low-dimensional modeling of the system by Langevin equation (3), we are still left with the problem of determining its drift field $h(x)$ and stochastic driving $F(t)$. For one-dimensional problems, it is straightforward to directly calculate the global drift and diffusion functions from the corresponding Fokker–Planck equation.\textsuperscript{15,23} However, this strategy obviously becomes cumbersome in higher dimensional spaces since the estimation of the necessary probability densities becomes rather difficult. Following the approach of Ref. 24, the goal of this paper is to develop a practical algorithm to implement the multidimensional Langevin equation by employing standard methods from nonlinear time series analysis.\textsuperscript{15} To this end, we introduce a local estimation method to parametrize the $d$-dimensional vector fields $h(x)$ and $F(t)$ from a given MD trajectory.\textsuperscript{24,26} Adopting a 800 ns MD simulation of the folding of heptaalanine in explicit water, it is shown that a five-dimensional Langevin model represents a suitable description of the structure and conformational dynamics of the system. The virtues and limits of the approach are discussed in some detail.

\section*{II. THEORY AND METHODS}

\subsection*{A. Principle component analysis (PCA)}

PCA is a well-established method to reduce the dimensionality of a high-dimensional data set.\textsuperscript{27} Considering the dynamics of $M$ atoms, the basic idea is that the correlated internal motion is represented by the covariance matrix,

$$\sigma_{ij} = \langle (q_i - \langle q_i \rangle)(q_j - \langle q_j \rangle) \rangle,$$

where $q_1, \ldots, q_M$ are the mass-weighted Cartesian coordinates of the molecule and $\langle \cdots \rangle$ denotes the average over all sampled conformations.\textsuperscript{9–12} By diagonalizing the covariance matrix, we obtain $M$ eigenvectors $v^{(i)}$ and eigenvalues $\lambda_i$, which are rank ordered such that $\lambda_1$ represents the largest eigenvalue. The eigenvectors and eigenvalues of $\sigma$ yield the modes of collective motion and their amplitudes, respectively. The principal components

$$x_i = v^{(i)} \cdot q$$

of the data $q = (q_1, \ldots, q_M)^T$ can then be used, for example, to represent the free energy surface [Eq. (1)] of the system.

Studying the reversible folding and unfolding of pentalaanine in explicit water, Mu \textit{et al.}\textsuperscript{18} showed that a PCA using Cartesian coordinates did not yield the correct rugged free energy landscape because of an artifact of the mixing of internal and overall motion. As internal coordinates naturally provide a correct separation of internal and overall dynamics, they proposed a method, referred to as dPCA, which is based on the dihedral angles ($\phi_\alpha, \psi_\beta$) of the peptide backbone. To avoid problems arising from the circularity of these variables, a transformation from the space of dihedral angles $\{\varphi_n\}$ to a linear metric coordinate space (i.e., a vector space with the usual Euclidean distance) was built up by the transformation

$$q_{2n-1} = \cos \varphi_n,$$

$$q_{2n} = \sin \varphi_n,$$

where $n = 1, \ldots, K$ and $K$ is the total number of peptide backbone and side-chain dihedral angles used in the analysis. Recently, the theoretical foundations of the dPCA were established,\textsuperscript{19} the method was implemented in the GROMACS MD program,\textsuperscript{28} and various application have been presented.\textsuperscript{20,21,29–31}
B. Langevin algorithm

Starting point is the multidimensional Langevin equation,
\[ \mathbf{x}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{D}(\mathbf{x}(t)) \mathbf{e}(t), \]
(7)

where we decomposed the stochastic driving \( \mathbf{F}(t) \) of Eq. (3) into the diffusion operator \( \mathbf{D} \), which contains all spatial and temporal dependencies of the stochastic driving, and a Gaussian distributed space-independent white noise \( \mathbf{e} \). The noise has zero mean, \( \langle \mathbf{e}(t) \rangle = 0 \), and variance \( \sigma \), i.e.,
\[ \langle \mathbf{e}(t) \mathbf{e}(t') \rangle = \sigma^2 \delta(t - t') \delta_{ij}. \]
(8)

Working with simulated or measured data, we only have time discrete measurements \( \mathbf{x}_n = \mathbf{x}(t_0 + n \Delta t) \), where \( t_0 \) represents the starting time of the measurement and \( \Delta t \) denotes its time step, which we set \( \Delta t = 1 \) for simplicity. As a consequence, the time derivative in Eq. (7) is approximated by the difference quotient \( \Delta \mathbf{x}_n / \Delta t = \mathbf{x}_{n+1} - \mathbf{x}_n \), thus leading to
\[ \Delta \mathbf{x}_n = \mathbf{x}_{n+1} - \mathbf{x}_n = \mathbf{h}(\mathbf{x}_n) + \mathbf{D}(\mathbf{x}_n) \mathbf{e}_n. \]
(9)

As explained in Sec. I, the main task is to determine the vector fields \( \mathbf{h} \) and \( \mathbf{D} \) from the MD data. As the noise term \( \mathbf{e}_n \) is unknown, these quantities cannot be obtained directly (e.g., via a least-squares fit), but need to be calculated by exploiting the statistical properties of the noise. To this end, we define a local average \( \langle f(x) \rangle \) of the quantity \( f \) around a given point \( x \) by
\[ \langle f(x) \rangle = \frac{\sum f(x_i) \Theta(||x - x_i|| - \delta)}{\sum \Theta(||x - x_i|| - \delta)}, \]
where the sum is taken over all measured points. Due to the Heaviside step function \( \Theta \), the average is only performed in the neighborhood of size \( \delta \) around \( x \). Applied to Eq. (9), we obtain
\[ \langle \Delta \mathbf{x}_n \rangle = \langle \mathbf{h}(\mathbf{x}_n) \rangle + \langle \mathbf{D}(\mathbf{x}_n) \mathbf{e}_n \rangle. \]

Assuming that \( \mathbf{h} \) and \( \mathbf{D} \) are sufficiently smooth and that its neighborhood is sufficiently small, we can replace \( \langle \mathbf{h}(\mathbf{x}_n) \rangle \) by \( \mathbf{h}(\mathbf{x}_n) \) and \( \langle \mathbf{D}(\mathbf{x}_n) \mathbf{e}_n \rangle \) by \( \mathbf{D}(\mathbf{x}_n) \langle \mathbf{e}_n \rangle \). This yields
\[ \langle \Delta \mathbf{x}_n \rangle = \mathbf{h}(\mathbf{x}_n) + \mathbf{D}(\mathbf{x}_n) \langle \mathbf{e}_n \rangle, \]
(10)

which in the following will be employed to determine the Langevin fields \( \mathbf{h} \) and \( \mathbf{D} \).

Since the average over the noise is zero, \( \langle \epsilon(t) \rangle = 0 \), Eq. (10) directly yields the drift term
\[ \mathbf{h}(\mathbf{x}_n) = \langle \Delta \mathbf{x}_n \rangle. \]
(11)

To obtain an estimate of the diffusion operator \( \mathbf{D} \), we construct the covariance matrix
\[ \langle \Delta \mathbf{x}_n \Delta \mathbf{x}_n^\dagger \rangle = \langle \mathbf{h}(\mathbf{x}_n) + \mathbf{D}(\mathbf{x}_n) \mathbf{e}_n \rangle \langle \mathbf{h}(\mathbf{x}_n) + \mathbf{D}(\mathbf{x}_n) \mathbf{e}_n \rangle^\dagger \]
\[ = \mathbf{h}(\mathbf{x}_n) \mathbf{h}(\mathbf{x}_n)^\dagger + \langle \mathbf{D}(\mathbf{x}_n) \mathbf{e}_n \rangle \langle \mathbf{D}(\mathbf{x}_n) \mathbf{e}_n \rangle^\dagger \]
\[ + \mathbf{D}(\mathbf{x}_n) \langle \mathbf{e}_n \rangle \langle \mathbf{e}_n \rangle^\dagger \mathbf{D}(\mathbf{x}_n)^\dagger, \]
where we again used that \( \langle \epsilon(t) \epsilon(t') \rangle = 0 \) in the last line. Employing \( \langle \epsilon_i(t) \epsilon_i(t') \rangle = \sigma^2 \delta(t - t') \delta_{ij} \), we obtain
\[ \sigma^2 \mathbf{D}(\mathbf{x}_n) \mathbf{D}(\mathbf{x}_n)^\dagger = \langle \Delta \mathbf{x}_n \Delta \mathbf{x}_n^\dagger \rangle - \mathbf{h}(\mathbf{x}_n) \mathbf{h}(\mathbf{x}_n)^\dagger. \]
(12)

We note that the second term of this expression is absent in the corresponding result of Ref. 26, which assumed the limit \( \Delta t \to 0 \). As shown in Ref. 32, Eq. (12) can be regarded as the first-order correction to the result of Ref. 26. Finally, the diffusion operator \( \mathbf{D} \) is calculated from \( \mathbf{DD}^\dagger \) through a Cholesky decomposition. In summary, to estimate the fields \( \mathbf{h} \) and \( \mathbf{D} \) occurring in Langevin equation (9), we need to calculate the average and covariance matrix of position difference \( \Delta \mathbf{x}_n \).

Let us briefly comment on the assumptions that entered the estimation of the parameters. First, we assumed that we can replace \( \langle \mathbf{h}(\mathbf{x}_n) \rangle \) by \( \mathbf{h}(\mathbf{x}_n) \). This requires that the neighborhood over which the averaging is performed is sufficiently small. Second, we assumed that the statistical properties of the noise [Eq. (8)] hold for our local averaging procedure. This means that we have to average over sufficiently many points. As the latter condition is readily fulfilled when a sufficiently large neighborhood is chosen, the two requirements are contrary in practice. To warrant appropriate statistical averaging in a sufficiently small neighborhood, therefore a large data set is required. In the case of MD data, this is typically not a problem in the vicinity of a metastable state, since the trajectory stays there for considerable time. Transitions between metastable states, on the other hand, are typically fast and rare. Thus, these parts of the phase space are sampled poorly and the algorithm may run into problems there. The discussion shows that the chosen size of the neighborhood is a critical parameter in the practical implementation of the method. Moreover, it significantly determines the numerical effort of the algorithm, which for large \( N \) scales as \( N \log N \), \( N \) being the number of data points along the trajectory.

C. Langevin models

So far, the above described Langevin algorithm is quite general because we have not yet specified the physical meaning of vector \( \mathbf{x}_n = \mathbf{x}(t_n) \). In what follows, we discuss several useful choices for \( \mathbf{x}_n \), which together with Langevin equation (9) will serve as models to describe biomolecular dynamics.

As explained in Sec. I, a straightforward choice is to identify the coordinates \( x_i \) with the first \( d \) principal components \( y_i \) of a PCA obtained from a MD simulation,
\[ \mathbf{x}_n = (y_1(t_n), \ldots, y_d(t_n))^T. \]
(13)

As a consequence, Langevin equation (7), \( \dot{\mathbf{x}} = \mathbf{h} + \mathbf{D} \mathbf{e} \), and its discretized form Eq. (9) represent a first-order differential equation for coordinates \( y_i \). It can be regarded as a limiting case of the “standard” Langevin equation,
\[ \ddot{y}_i = v_i, \]
\[ m \ddot{v}_i = -\frac{\partial V}{\partial y_i} - \zeta_i \dot{v}_i + \mathbf{F}_i, \]
(14)

with masses \( m_i \), velocities \( v_i \), friction coefficients \( \zeta_i \), random forces \( \mathbf{F}_i \), and energy function \( V(y) \). As discussed by
Zwanzig,\textsuperscript{2} this equation reduces to Langevin equation (7) with Eq. (13), when we assume large friction and restrict ourselves to times much larger than $m_i/\zeta_i$. Equation (14) is appealing because its vector fields $\nabla V$, $\zeta$, and $F$ have a well-established physical interpretation. On the other hand, the velocity field $h$ and drift field $D$ of approximation (7) with Eq. (13) are much easier to estimate in practice.

To go beyond this approximation, we may identify the coordinates $x_i$ with phase-space variables $\{y_i, \dot{y}_i\}$ by setting

$$x_n = (y_1(t_n), \ldots, y_d(t_n), \dot{y}_1(t_n), \ldots, \dot{y}_d(t_n))^T.$$  \hspace{1cm} (15)

This way Langevin equation (7) is propagated in phase space rather than in coordinate space. It is important to note that Eq. (7) with Eq. (15) is more general than Eq. (14). Because fields $h(x_n)$ and $D(x_n)$ depend explicitly on phase-space variable $x_n$, this Langevin equation may also describe possible couplings of the velocities $\dot{y}_i$ to the bath. Note that velocity couplings do occur in standard constant-temperature MD simulations, where the velocities are rescaled by a thermostat.\textsuperscript{3} Only in the absence of these coupling, Eq. (7) with Eq. (15) reduces to Eq. (14).

In practice, the discretized form (9) of the Langevin equation is employed which approximates the time derivative in Eq. (7) by the difference quotient $\Delta x_n/\Delta t = x_{n+1} - x_n$. Rather than propagating the phase-space variables (15), we can therefore also use the vector

$$x_n = (y_1(t_n), \ldots, y_d(t_n), y_1(t_{n-1}), \ldots, y_d(t_{n-1}))^T.$$  \hspace{1cm} (16)

In fact, this definition is advantageous compared to Eq. (15). This is because the time-delayed variable $y_i(t_{n-1})$ contains relatively less noise than the difference $x_{n+1} - x_n$, where two noisy data are subtracted.

Equation (16) can be recognized as a special case of delay embedded variables, commonly applied in nonlinear time series analysis.\textsuperscript{25} Rather than using only variables at time step $t_n$, this means that also information from previous times $t_n, t_{n-1}, \ldots, t_{n-l}$ is employed. This leads to

$$x_n = (y_1(t_n), \ldots, y_d(t_n), y_1(t_{n-1}), \ldots, y_d(t_{n-1}), \ldots,$$

$$y_1(t_{n-l}), \ldots, y_d(t_{n-l}))^T,$$  \hspace{1cm} (17)

where the embedding dimension $l$ may be different for different variables $y_i$.\textsuperscript{35} Equation (17) represents the most general definition for the Langevin vector used in this work. In particular, it allows us to use a higher embedding dimension for the first principal components than for the remaining ones. This is often advantageous in practice because the first principal components contain the largest conformational fluctuations of the system and therefore—relatively speaking—less noise than the higher components.
III. A SIMPLE EXAMPLE

To illustrate the algorithm introduced above, we apply it to a well-established model problem, trialanine, whose conformation can be characterized by a single pair of $(\phi, \psi)$ backbone dihedral angles. Trialanine (Ala$_3$) in aqueous solution is a model peptide which has been the subject of numerous experimental$^{22,36-38}$ and computational$^{16,17,39}$ studies. To generate the angular distribution of $(\phi, \psi)$ of trialanine, we performed a 100 ns MD simulation at 300 K. We used the GROMACS program suite,$^{28}$ the GROMOS96 force field 43A1,$^{40}$ the simple point charge (SPC) water model,$^{41}$ and a particle-mesh Ewald$^{42}$ treatment of the electrostatics. Details of the simulation can be found in Ref. 17. The simulations show that mainly three conformational states are populated: the right-handed helix conformation $\alpha_R$ (15%), the extended conformation $\beta$ (39%), and the poly-L-proline II ($P_{II}$) helix-like conformation (42%). Although recent experimental data$^{22}$ indicate that the simulation overestimates the populations of $\alpha_R$ and $\beta$, we nevertheless adopt the MD data as a first simple yet nontrivial example.

Performing a dPCA on the central backbone dihedral angles of Ala$_3$, we obtain four principal components, the first two of which carry the main information on the conformational distribution.$^{19}$ Figure 2 shows the distribution $P(x_i)$ and the autocorrelation function $C_1(\tau) = \langle (x_1(\tau) - \langle x_1(0)\rangle) (x_1(0) - \langle x_1(\tau)\rangle) \rangle$ of the first two principal components $x_i$ ($i=1,2$). The distribution $P(x_i)$ exhibits two well-separated peaks corresponding to the conformational states $\alpha_R$ and $\beta/P_{II}$. The autocorrelation function $C_1(\tau)$ reveals that the time scale of transitions between these states is $\approx 200$ ps. The distribution $P(x_i)$ allows us to discriminate states $\beta$ and $P_{II}$. Transitions between these states occur on a time scale of $\approx 10$ ps.

We now wish to employ the 100 ns MD trajectory of $x_n = (x_1(t_n), x_2(t_n))^T$ as input data to determine the vector fields $h$ and $D$ of the corresponding two-dimensional Langevin equation of the system. To this end, we adopt the last point of the MD trajectory as first point $x_0 = x(t_0)$ of the Langevin trajectory. Then the next point, $x_1 = x(t_0 + \Delta t)$ is obtained from Langevin equation (9), where vector fields $h(x_0)$ and $D(x_0)$ are calculated via Eqs. (11) and (12) by including all points of the MD trajectory close to $x_0$ in the local average. We choose a time step of $\Delta t = 0.2$ ps and $k = 5$ nearest neighbors to define the size of this local average. Proceeding this way, we iterated another 100 ns into the future, in order to get as many points of the Langevin trajectory as we have from the MD trajectory. If the resulting Langevin simulation represents a true model of the original dynamics, the statistical and dynamical properties of MD and Langevin trajectory should be equivalent.

Figure 3 shows the outcome of such an iteration by comparing the time evolution of the first principal component $x_1$ obtained from the MD and the Langevin simulation, respectively. One clearly sees that the Langevin simulation nicely reproduces the main features of the time series, including the jumps between conformational states $\alpha_R$ and $\beta/P_{II}$ and the fluctuation within these states. Figure 2 shows that the Langevin simulation also reproduces almost quantitatively the distribution $P(x_i)$ and the autocorrelation function $C_1(\tau)$ of the first two principal components.

It is instructive to analyse the dynamical properties of the Langevin simulation in more detail. To this end, we employ a (rather coarse grained) clustering of the free energy landscape of Ala$_3$ by defining two conformational states, $\alpha_R$ and $\beta/P_{II}$. As mentioned above, these states are clearly separated by the first principal component $x_1$. Using this definition, we can evaluate the distribution of lifetimes of the states. As an example, Fig. 4(a) shows the lifetime distribu-
tion of the $\alpha_R$ state as obtained from the MD data. Due to the relatively infrequent transitions between $\alpha_R$ and $\beta/P_{II}$, the distribution is not well sampled by the 100 ns MD simulation. Performing a Langevin simulation of the same length, the resulting lifetime distribution looks quite similar within the given statistical accuracy.

As the propagation of a two-dimensional Langevin simulation is computationally much cheaper than the corresponding MD simulation including several thousands of atoms, it is an easy matter to increase the sampling of the lifetime distribution. By starting 10 000 trajectories in the $\alpha_R$ state, we readily obtain a lifetime distribution with an accuracy of $\approx 10^{-3}$. Using a logarithmic scale, Fig. 4(b) compares this distribution to a Poisson-type waiting time distribution,

$$P(\tau) \propto e^{-\tau/\tau_0},$$

where $\tau_0 = 320$ ps is the average lifetime as obtained from the MD trajectory. The form of the latter distribution assumes that the transitions between the two states can be described by a Markov process.\(^{15}\) The excellent agreement of the two lifetimes indicates that the Langevin simulation accurately accounts for the conformational transitions of the system.

Given that Langevin simulation reproduces the principal component distribution $P(x)$ [and thus also the free energy landscape $\Delta G(x) \propto \ln P(x)$], one may wonder whether this is a trivial consequence of the Langevin algorithm. After all, in the parametrization of the fields $h$ and $D$ the algorithm may exploit the complete MD trajectory [and thus also $\Delta G(x)$]. To answer this question, we recall that all information we use to calculate these fields at a given point $x_n$ is the $k$ previously obtained nearest neighbors of $x_n$. These $k$ points determine the future value of $x_n$ (i.e., $x_{n+1}$) through their average and covariance [see Eqs. (11) and (12)]. In other words, the important information given by these $k$ neighbors is not their local density but their local stability. That does not mean that the local density is completely unimportant. The higher the density, the closer the $k$ neighbors are located around $x_n$, thus resulting in a better realization of the assumptions used to derive Eq. (10). But as long as the local density is sufficiently high, its precise value does not affect the estimates of the Langevin fields. Hence, a correct construction of the free energy landscape by the Langevin algorithm is not a mere reproduction of the input data.

To demonstrate this property of the method, we perform a Langevin simulation that uses modified MD input data, in which large parts of the time when the system is in the $\beta/P_{II}$ state were removed. Comparing original and modified MD data, Fig. 5 shows that the modified distribution $P(x)$ equally populates the $\alpha_R$ and $\beta/P_{II}$ states, respectively. Also shown is the distribution of a Langevin trajectory which used the modified data as input. As anticipated in the discussion above, the Langevin distribution nicely reproduces the original MD distribution because the algorithm used the correct stability information. We note that this property of the Langevin algorithm is rather appealing because in practice the statistics of a MD simulation is often not converged.

Another nice feature of the algorithm is that it does not need a continuous MD trajectory as input data. In fact, from the description of the algorithm above, it is obvious that pairs of adjacent points (e.g., $x_n$ and $x_{n+1}$) suffice to parametrize the fields $h$ and $D$. To demonstrate this fact, we “scrambled” the $N$ data points of the MD trajectory into $N-1$ such pairs and repeated all above presented Langevin simulations using these scrambled data. As expected, the outcome of these simulations is identical to the Langevin simulations using the continuous MD trajectory. At first, this may seem surprising, since in the absence of time ordering of the data there is no direct information on the lifetime of conformational states. As explained above, though, only the stability information from the nearest neighbors of a given point $x_n$ is required to correctly determine the Langevin model. Again, this is a rather appealing property of the Langevin algorithm because it allows us to use data from enhanced-sampling Monte Carlo schemes such as replica exchange MD simulations.\(^{43}\)

### IV. APPLICATION TO HEPTAALANINE

In what follows, we apply the above described methodology to construct a Langevin equation that describes the folding of heptaaalanine (Ala$_7$). In a recent paper,\(^{21}\) we performed an 800 ns MD simulation of Ala$_7$ in aqueous solution at 300 K, using the GROMACS program suite,\(^{28}\) the GROMOS96 force field 43A1,\(^{40}\) the SPC water model\(^{41}\) and a particle-mesh Ewald\(^{42}\) treatment of the electrostatics (see Ref. 21 for details). Using geometric and kinetic clustering techniques, it was shown that a five-dimensional dPCA free energy surface is a suitable and accurate representation of the full-dimensional landscape. To demonstrate the time and length scale separation achieved by the PCA, Figs. 6 and 7 show the distribution $P(x)$ and the autocorrelation function $C_\tau(x)$ of the first six principal components $y_i$ ($i=1,\ldots,6$). The first five principal components exhibit multipeaked distributions and a nanosecond decay of the autocorrelation function. Beginning with the sixth component, the distributions become unimodal and the time scale is shorter than 100 ps.

As explained in Sec. I, we expect that an appropriate Langevin simulation of the conformational dynamics of Ala$_7$ should include sufficiently many (i.e., at least five) principal components, in order to warrant a time scale separation between system and bath degrees of freedom. To test this
Figure 6. (Color online) Distribution $P(x_i)$ of the first six principal components $x_i$ ($i = 1, \ldots, 6$) as obtained from a 800 ns MD trajectory of heptaalanine. Compared are original data from MD simulation (thick black line) and Langevin data using the smallest possible dimensionality (dashed blue line) and “full” (i.e., all five relevant components and their velocities) dimensionality (thin red line).

Figure 7. (Color online) Autocorrelation function $C_i(\tau)$ of the first six principal components $x_i$ ($i = 1, \ldots, 6$) as obtained from a 800 ns MD trajectory of heptaalanine. Compared are original data from MD simulation (thick black line) and Langevin data using the smallest possible (dashed blue line) and full (thin red line) dimensionality.

Assuming, we performed Langevin simulations using $d = 1, \ldots, 6$ principal components and a time step of $\Delta t = 1$ ps and $k = d + 1$ nearest neighbors. Figure 6 compares the resulting principal component distributions for various dimensionalities $d$ to the original MD data. Interestingly, we find that the lowest possible dimensionality [e.g., to reproduce $P(x_i)$ we need $d \geq 3$] is already sufficient to reproduce a distribution. That is, if we only wish to generate the correct statistical distribution of the data, we are free to use any dimensionality in the Langevin model. Figure 6 also displays results of a “full-dimensional” Langevin simulation that includes all five relevant principal components $y_i(t_n)$ as well as their time-delayed components $y_i(t_{n-1})$ to account for the velocities $y_i$ [see Eq. (16)]. Interestingly, we find that the increased dimensionality even leads to a somewhat inferior agreement with the MD reference data.

However, this finding does not extend to dynamical properties of the system. When we consider the autocorrelation functions $C_i(\tau)$, Fig. 7 shows that the full-dimensional Langevin simulation clearly does a better job to reproduce the time scales of the system than a calculation of lower dimensionality. Generally speaking, a too low dimensionality results in an overestimation of the decay of $C_i(\tau)$, since the nonlocal correlation of the dynamics is underestimated in the subspace, which results in a too short lifetime of the metastable states as we will see later on.

To discuss the conformation dynamics of Ala$_7$ in more detail, it is instructive to identify the metastable states of the system. As detailed in Ref. 21, we have performed $k$-means geometric clustering using various cluster sizes $K$. Using the resulting $K$ metastable states, we have calculated the corresponding transition matrix $T_{ij}(\tau)$, whose elements denote the probability of observing the system in state $j$ at time $t + \tau$ given that it is in state $i$ at time $t$. The eigenvalues of the transition matrix can be used to construct a kinetic clustering of the process. This is because eigenvalues close to unity, the so-called Perron eigenvalues, correspond to such metastable clusters, while small eigenvalues indicate the existence of kinetically unstable clusters. Systems showing hierarchical dynamics typically exhibit a clear gap between Perron and small eigenvalues.

Figure 8 shows the eigenvalues of the transition matrix constructed from $k$-means clustering of Ala$_7$ MD data using...
a cluster size $K=25$ and a lag time $\tau=1 \text{ ps}$. By construction, the largest eigenvalue is always 1, while all others lie in the interval $[0,1]$. From the MD data, we obtain 23 Perron eigenvalues, that is, for $K=23$ all clusters are metastable. Performing Langevin simulations with various numbers $d$ of included principal components, Fig. 8 reveals that it takes at least $d=4$ to qualitatively reproduce this behavior and $d=5$ to achieve quantitative agreement with the MD reference data. One- or two-dimensional Langevin models, on the other hand, largely fail to reproduce the correct metastabilities of the conformational states of Ala$_7$.

More detailed information on the conformational dynamics can again be gained by considering the lifetime distributions of specific conformational states of Ala$_7$. Choosing the most stable state 1 and the least stable state 23 as representative examples, Fig. 9 compares the lifetime distributions as obtained from MD and Langevin simulations, respectively. Again, we find that a one-dimensional Langevin model fails to account for the conformational dynamics of the system, as it considerably underestimates its lifetimes. The full-dimensional Langevin model, on the other hand, nicely reproduces the lifetime distributions of the MD reference data.

V. CONCLUSIONS

We have presented a systematic computational approach to describe the conformational dynamics of biomolecules in reduced dimensionality. The method is based on (i) the decomposition of a high-dimensional MD trajectory into a few system degrees of freedom and (many) bath degrees of freedom and (ii) a Langevin simulation of the resulting model. In the decomposition, the dimension of the system is chosen such that it contains all slow large-amplitude motions of the molecule, while the bath coordinates only account for its high-frequency fluctuations. Hence, a sufficiently large dimension of the model is essential to ensure a clear time scale separation of system and bath variables. As a consequence of a too small dimension, for example, it was found that Langevin simulations considerably underestimate the lifetimes of metastable states.

Employing methods from nonlinear time series analysis, we have presented a practical algorithm to perform the Langevin simulations. In particular, we have introduced a local estimation method to parametrize the Langevin vector fields describing drift and stochastic driving. Adopting a 800 ns MD simulation of the folding of heptaalanine in explicit water, it has been shown that a five-dimensional Langevin model correctly reproduces the statistical distribution and the autocorrelation function of all components of the reaction coordinate. Moreover, the lifetime distribution of various metastable conformational states was obtained correctly.

It is instructive to compare the Langevin approach with alternative methods to model biomolecular dynamics in reduced dimensionality. For example, if the process under consideration can be described by a Markov chain of metastable states, a suitable clustering combined with a simple master equation provides the complete information of the time evolution of the system (see Refs. 46–53 for recent applications of this approach to biomolecular processes). In biomolecular systems, however, the underlying assumption of a time scale separation between fast intrastate and slow interstate transitions may break down. This was also found to be the case for heptaalanine, where the minimal time scale to assure Markovian dynamics is already in the order of the conformational state’s lifetime. Working in continuous phase space, the Langevin approach does not require to define suitable metastable states.

Apart from a stochastic description, it may be also desirable to construct a deterministic model of biomolecular
free energy landscape from such enhanced-sampling techniques. This property of the Langevin algorithm allows us to use techniques from nonlinear dynamics theory to analyze the system. For example, a Lyapunov analysis of a deterministic model of peptide dynamics showed that the effective dimension of the system is rather small and may even decrease with chain length. However, the disadvantage of the deterministic modeling is that, starting with stochastically driven data, the noise reduction method needs to assume properties of the noise (or the bath), which are difficult to verify in practice. Depending on the system under consideration, it may therefore be not possible to construct an appropriate noise reduction at all.

The Langevin approach, on the other hand, directly accounts for the noise of the system. It is therefore more general, but it also requires more effort to determine the stochastic driving field of the Langevin equation. The algorithm does not require a continuous MD trajectory for the input data, but only needs pairs of adjacent trajectory points. This appealing property of the Langevin algorithm allows us to use data from enhanced-sampling Monte Carlo schemes such as replica exchange MD simulations. After constructing the free energy landscape from such enhanced-sampling schemes, Langevin simulations can, for example, be used to study the nonequilibrium structural evolution following a temperature jump of the system.

ACKNOWLEDGMENTS

We thank Alexandros Altis, Phuong H. Nguyen, and Moritz Otten for numerous inspiring and helpful discussions. This work has been supported by the Frankfurt Center for Scientific Computing, the Fonds der Chemischen Industrie, and the Deutsche Forschungsgemeinschaft.