2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Dynamic Coarse Graining in Complex System Simulation

Yuzhen Xue, Pete J. Ludovice, Martha A. Grover

Abstract— **The high computational burden in complex system simulation, particularly for a polymer system, prohibits long term simulation that provides information to predict the system attributes. We propose a coarse graining simulation method, stemming from our earlier work on state reduction through a modified local feature analysis (LFA), so that, based on short term system dynamics, one can automatically identify a low number of "seeds" based on correlations in the dynamic motion of all states. The trajectories of the "seeds" are then extrapolated. A simple matrix transformation is proposed to calculate trajectories of the whole system from the extrapolated "seed" trajectories. As the recovered system dynamics are derived from the low dimensional seed trajectories, we call it coarse grained dynamics. Simulation is carried out to illustrate the application of the developed algorithm to the PVC polymer dynamics.**

I. Introduction

The high computational burden in complex system simulation prohibits long term simulation to predict the system attributes. In particular, for a polymer molecular dynamics (MD) simulation, composed of multiple polymer chains, which is defined by a large collection of discrete atoms, their interactions, and the resulting dynamic trajectories, the simulation time is typically on the order of nanoseconds, while the relaxation time of the polymer ranges from 1-1000 s [7]. The high computational burden in polymer simulation is due to simulating many atoms while resolving fast vibrational atomic time scales.

Intermediate order, which occurs in many polymers, e.g. polyvinyl chloride (PVC) [14] and poly(n-butyl methacrylate) as well as specialty polymers such as poly(trimethyl silyl propyne) [6], [22] and poly(norbornene) [1], [2], [9], [18], is a structural order that is in between the crystalline and amorphous states [13]. Polymers that show intermediate order have potential applications including membrane separation, photolithography and catalyst substrates. As the intermediate order is responsible for the useful properties of these polymers, in the polymer study we are interested in modeling the process for polymeric materials to evolve and reach intermediate levels of structural order. Intermediate order is in general detected from the wide angle x-ray diffraction (WAXD) curves. WAXD is an X-ray diffraction technique that is often used to determine the crystalline structure of polymers.

The initial object of our study is on PVC, where intermediate order produces a unique thermoreversible gelation property that makes PVC applicable to flexible applications. Some corporations are disrupting the order of highly crystalline polymers such as poly(paraphenylene) to create intermediate order because it results in a polymer with good mechanical properties that is easier to process than its crystalline counterpart.

As in general the computational capability does not allow us to simulate the process for a polymer to evolve enough to reach the desirable state, e.g. intermediate order, where we can analyze the system properties, we seek a model reduction approach to reduce the simulation burden.

Methods for coarse-graining of molecular simulations, by grouping nearby atoms, have been developed to speed up molecular dynamics simulations [4], [15], but due to the complexity of polymer tacticity, the size of the groups has been limited to one monomer. However, much greater speedup is still needed. An automated method for model reduction and system identification could provide a complementary approach for computational reduction, such as equation-free computing [11]. However, identifying an appropriate reduced-order state for equation-free computing remains a challenging problem [5], even for a simple fluid.

In our earlier work [19] we proposed a modified local feature analysis [16], [19], [21] algorithm and applied it in a polymer simulation. This algorithm automatically identifies "seed" atoms, based on correlations in the dynamic motion of all atoms. In this paper, we intend to extend our earlier work and propose a modeling method for molecular dynamics simulations of polymers, which combines features from equation-free computing with polymer coarse-graining. We call this method dynamic coarse graining. Although the algorithm to be proposed is applicable to any complex dynamic system, as our first concern is the polymer system, we will describe it in the polymer context.

II. Problem description

Consider a polymer system simulated with MD. For simplicity we will focus on the dynamics of the backbone carbons. Assume that the polymer system has N number of backbone carbons. After subtracting the overall translational and rotational motion, the dynamics of the polymer system is represented by its trajectory $\mathbf{x}(t) = \{x_1(t), x_2(t), ..., x_N(t)\}^T \in \mathbf{R}^{N\times 1}$, where $x_i(t)$ denotes the state value of the i^{th} backbone carbon at time t. Given that observations of s time steps are available, we have

$$\mathbf{x}(1:s) = \left\{ \begin{array}{c} x_1(1:s) \\ ... \\ x_N(1:s) \end{array} \right\} \in \mathbf{R}^{N\times s}$$

Moreover, as the motion of the polymer chains is three dimensional, we extend the motion of atoms at time i

School of Chemical & Biomolecular Engineering, Georgia Institure of Technology, Atlanta, GA 30332, yxue8@mail.gatech.edu, pete.ludovice@chbe.gatech.edu, martha.grover@chbe.gatech.edu

978-1-4577-0079-8/11/$26.00 ©2011 AACC

into three vectors, which in turn introduces the equivalent trajectory $\mathbf{x}(1:L) \in \mathbf{R}^{N\times L}$, where $L = 3s$. In general $L \gg N$. Then we construct the correlation matrix as

$$R_{i,j} = \left\langle \frac{x_i - \bar{x}_i}{std(x_i)}, \frac{x_j - \bar{x}_j}{std(x_j)} \right\rangle \triangleq \langle X_i, X_j \rangle$$

thus

$$R = X \times X^T$$

where $R \in \mathbf{R}^{N\times N}$, $X \in \mathbf{R}^{N\times L}$, $x_i, X_i \in \mathbf{R}^{1\times L}$, $\bar{x}_i$ denotes the mean value and $std(x_i)$ denotes the standard deviation of x_i. Note that in real applications, we can assume that R is full rank because of the noise contained in X due to the stochastic nature of MD.

We clarify that R is the matrix derived from a short term detailed simulation. The LFA based state reduction algorithm will be applied to R to pick out the "seeds", that is, the atoms with the most representative dynamics.

III. Dynamics Coarse Graining Based on LFA

A. Local Feature Analysis

The objective of LFA is to provide a topographic representation for all system states through a reduced basis set, i.e. local features. LFA stems from PCA (please see [19] or [16] for detailed description of PCA) and preserves all the information of PCA. However, unlike PCA which has a global basis set, i.e. basis that span over all the states, LFA provides a localized basis set. Moreover, as has been pointed out in [3] and [21], the LFA basis is more stable over the shift of sampling windows, e.g. windows of length L with different starting time.

Considering the correlation matrix $R \in \mathbf{R}^{N\times N}$ described in Sec. II, LFA kernel functions are defined as

$$K(l,k) = \sum_{i=1}^{r} \Psi_i(l) \frac{1}{\sqrt{\lambda_i}} \Psi_i(k)$$

where $k, l \in \{1, \cdots, N\}$, $K(l,k)$ is the (l,k) entry of matrix $K \in \mathbf{R}^{N\times N}$, λ_i and Ψ_i are the i^{th} eigenvalue and eigenvector of R respectively, and r is the number of dominant eigenvalues determined through PCA according to $\lambda_1 \geqslant \ldots \geqslant \lambda_r \gg \lambda_{r+1} \geqslant \ldots \geqslant \lambda_N$. Define the projection coefficients of any $\Phi \in \mathbf{R}^{N\times 1}$ onto the state variable x_l, $l = 1, \ldots, N$ as

$$o_l = \sum_{k=1}^{N} K(l,k)\Phi(k) \equiv \sum_{i=1}^{r} \frac{\Psi_i^T \Phi}{\sqrt{\lambda_i}} \Psi_i(l) \qquad (1)$$

where $o_l \in \mathbf{R}$ is the LFA output (feature). We can see that $K(l,k)$ is a topographic kernel as the output o_l from Φ only relies on the components of the eigenvectors corresponding to x_l. Letting $\mathbf{o} = [o_1, \cdots, o_N]^T$, we have

$$\mathbf{o} = K\Phi = \mathbf{\Psi}\Lambda^{-\frac{1}{2}}\mathbf{\Psi}^T\Phi \qquad (2)$$

where $\Lambda = diag(\lambda_1, \ldots, \lambda_r)$, and $\mathbf{\Psi} \in \mathbf{R}^{N\times r}$ is the matrix that contains the r dominant eigenvectors. Through LFA, any $\Phi \in \mathbf{R}^{N\times 1}$ can be approximated with minimum mean square error that is the same as in PCA, by

$$\Phi_{rec} = K^{(-1)}\mathbf{o} \qquad (3)$$

where $K^{(-1)}$ with entries $K^{(-1)}(l,k) = \sum_{i=1}^{r} \Psi_i(l)\sqrt{\lambda_i}\Psi_i(k)$, $k, l \in \{1, \cdots, N\}$, is the approximate inverse of K [16].

The reconstruction in (3) is with N local features, i.e. $\mathbf{o}$, such that the locality is achieved at the price that the number of features are extended to $N \gg r$, much larger than that in the PCA case. A sparsification step is thus critical in order to get rid of the redundant local features.

In our earlier work [19], a novel sparsification algorithm was derived. Rather than empirically searching based on multiple linear regression among the whole state set, see e.g. [12], [16], [20], [21], the newly developed algorithm in [19] involves a simple matrix calculation and offers clear theoretical foundation for sparsification.

B. LFA sparsification

In this section we briefly introduce the modified LFA sparsification algorithm proposed in [19]; interested readers may refer to [19] for more details.

Consider $X \in \mathbf{R}^{N\times L}$ in Section II with $rank(X) = N$, given any $\Phi \in \mathbf{R}^{N\times 1}$ there exists $P \in \mathbf{R}^{1\times L}$ such that $\Phi = XP^T$.

Defining the local feature matrix $\mathbf{O} \in \mathbf{R}^{N\times L}$ as

$$\mathbf{O} \triangleq KX = \mathbf{\Psi}\Lambda^{-\frac{1}{2}}\mathbf{\Psi}^T X \qquad (4)$$

it is easy to see that $rank(\mathbf{O}) = r$. From (2) and (4) we obtain

$$\mathbf{o} = \mathbf{O}P^T \qquad (5)$$

Defining

$$M \triangleq \mathbf{\Psi}^T\mathbf{O}$$

then M has the singular value decomposition

$$M = V\Sigma U^T \qquad (6)$$

where $U \in \mathbf{R}^{L\times r}$, $\Sigma \in \mathbf{R}^{r\times r}$, $V \in \mathbf{R}^{r\times r}$. It can easily be shown that there exists $\hat{\mathbf{O}} \in \mathbf{R}^{N\times r}$ such that

$$\mathbf{O} = \hat{\mathbf{O}}U^T \qquad (7)$$

The following proposition derived in [19] shows that the reconstruction (3) can be based on only r correctly chosen local features, i.e. $\mathbf{o}^r \in \mathbf{R}^{r\times 1}$.

Proposition 1: There exists a matrix $K_r^{(-1)} \in \mathbf{R}^{N\times r}$ and $\mathbf{o}^r \in \mathbf{R}^{r\times 1}$, the subvector of $\mathbf{o}$, such that

$$\Phi_{rec} = K^{(-1)}\mathbf{o} = K_r^{(-1)}\mathbf{o}^r$$

Defining the index set $\mathbf{S}$ that corresponds to the indices of entries in $\mathbf{o}$ that comprise $\mathbf{o}^r$, we have

$$\mathbf{S} \in \{\{k_1, \cdots, k_r\},\ rank(\hat{\mathbf{O}}_\mathbf{r}) = r\}$$

where $\hat{\mathbf{O}}_\mathbf{r} \in \mathbf{R}^{r\times r}$ is the submatrix of $\hat{\mathbf{O}}$ by including only the rows with indices $\{k_1, \cdots, k_r\}$. Given $\hat{\mathbf{O}}_\mathbf{r}$, the kernel function $K_r^{(-1)} = \mathbf{\Psi}\Lambda^{\frac{1}{2}}\mathbf{\Psi}_r\hat{\mathbf{O}}_\mathbf{r}^{-\mathbf{T}}\hat{\mathbf{O}}_\mathbf{r}^{-1}$, where $\mathbf{\Psi}_r \in \mathbf{R}^{r\times r}$ is the submatrix of $\mathbf{\Psi}$ by including the rows corresponding to index set $\mathbf{S}$.

Prop. 1 tells us that any Φ can be approximately reconstructed based on only r local features, $\mathbf{o}^r$, as long as the corresponding $\hat{\mathbf{O}}_{\mathbf{r}}$ matrix has $rank(\hat{\mathbf{O}}_{\mathbf{r}}) = r$. We clarify that the choice of index set $\mathbf{S}$ is not unique. The process of choosing the most representative r local features is called sparsification. There are practical considerations in favor of their judicious choice. In [16] it is suggested to choose indices set $\mathbf{S}$ such that $\mathbf{O}_{k_i}$, $\mathbf{O}_{k_j}$ are the least dependent for any $k_i \neq k_j$, k_i, $k_j \in \mathbf{S}$, where $\mathbf{O}_{k_i}$ is the ${k_i}^{\text{th}}$ row of $\mathbf{O}$. Such a set $\mathbf{S}$ is corresponding to the most representative local features.

Definition 1: The index of dependency between two vectors $\mathbf{O}_i$, $\mathbf{O}_j$ is defined as $\rho_{ij} = \frac{|\langle \mathbf{O}_i, \mathbf{O}_j \rangle|}{\|\mathbf{O}_i\| \|\mathbf{O}_j\|}$; a small ρ indicates that $\mathbf{O}_i$, $\mathbf{O}_j$ are highly independent.

C. Dynamic Coarse Graining Algorithm

By sparsification, we find index set $\mathbf{S} = \{k_1, \cdots, k_r\}$ through identifying the least dependent vectors $\mathbf{O}_{k_1}, \cdots, \mathbf{O}_{k_r}$, so that $\mathbf{o}^r = [o_{k_1}, \cdots, o_{k_r}]^T$ are the most representative local features, corresponding to "seed" states, $x_{k_1}, \cdots, x_{k_r}$, with kernel matrix $K_r^{(-1)} = \mathbf{\Psi} \Lambda^{\frac{1}{2}} \mathbf{\Psi}_r \hat{\mathbf{O}}_{\mathbf{r}}^{-\mathbf{T}} \hat{\mathbf{O}}_{\mathbf{r}}^{-1}$. As x_{k_l}, $l \in \{1, \cdots, r\}$, is the "seed" state and $x_{k_l}(1 : L)$, $l \in \{1, \cdots, r\}$ is the "seed" trajectory, from (4) we can view the counterpart $\mathbf{O}_{k_l}$, $l \in \{1, \cdots, r\}$, also as a time series of length L corresponding to the most representative local features, thus we call $\mathbf{O}_{k_l}$ the "seed" dynamics (trajectories).

Next we describe the dynamic coarse graining algorithm. The idea is to develop the relationship between trajectories of X_i, $i \in \{1, \cdots, N\}$ and "seed" trajectories $\mathbf{O}_{k_l}$, $l \in \{1, \cdots, r\}$ as well as to identify the time series model of "seed" trajectories so that, by extrapolating the seed trajectories, we can recover the dynamics of X_i, $i \in \{1, \cdots, N\}$ in the future time. As the recovery of the whole system dynamics is derived from the dynamics of r "seeds", we name our algorithm "dynamic coarse graining algorithm".

Define $\hat{X} \triangleq \mathbf{\Psi} \mathbf{\Psi}^T X$ as the filtered X, i.e. X without the non-principal components. We have the following claims.

Claim 1: The Euclidean norm between $\hat{X}$ and X is $\left\| \hat{X} - X \right\|_2 = \bigcirc(\sqrt{\lambda_{r+1}})$, where $\bigcirc(\sqrt{\lambda_{r+1}})$ means the order of $\sqrt{\lambda_{r+1}}$.

Proof: Proof is trivial using the definition of Euclidean norm of matrix. ■

From the above claim, we see that $\hat{X}$ can approximate X with error $\bigcirc(\sqrt{\lambda_{r+1}})$, which is small if λ_{r+1} is small.

Claim 2: $\mathbf{O}_{k_1}, \ldots, \mathbf{O}_{k_r}$ form the basis to reconstruct $\hat{X}$, and $\hat{X} = \mathbf{\Psi} \Lambda^{\frac{1}{2}} \mathbf{\Psi}^T C \begin{bmatrix} \mathbf{O}_{k_1} \\ \vdots \\ \mathbf{O}_{k_r} \end{bmatrix}$, where $C = \hat{\mathbf{O}} \hat{\mathbf{O}}_{\mathbf{r}}^{-1}$.

Proof: Proof is derived directly from the definition of $\mathbf{O}$, C and $\hat{X}$. ■

Now we see that the dynamics $\hat{X}_j$ of any state j can be reconstructed by the seed dynamics $\{\mathbf{O}_{k_1}, \cdots, \mathbf{O}_{k_r}\}$. That is, X can be reconstructed, with small 2-norm error $\sqrt{\lambda_{r+1}}$, through seed dynamics $\{\mathbf{O}_{k_1}, \cdots, \mathbf{O}_{k_r}\}$ during s time period.

Claim 3: Considering the system discussed above, choose $f_{l,p}$ so that

$$\mathbf{O}_{k_l}^{(p)}(t) = f_{l,p}(t), \; l \in \{1, \cdots, r\}, \, p \in \{x, y, z\}, 1 \leqslant t \leqslant s \tag{8}$$

where $\mathbf{O}_{k_l}^{(p)}(t)$ denotes to the entry in $\mathbf{O}_{k_l}$ that corresponds to the value in the p direction at time step t. (Recall the construction of X matrix in Sec. II, and note that vector $\mathbf{O}_{k_l}$ is composed of $\mathbf{O}_{k_l}^{(p)}(t)$, $1 \leqslant t \leqslant s$, $p \in \{x, y, z\}$.) Assuming that s is long enough so that (8) and the linear relationship in Claim 2 stay the same for $M \gg s$, then

$$\tilde{X}^{(p)}(M) = \mathbf{\Psi} \Lambda^{\frac{1}{2}} \mathbf{\Psi}^T \mathbf{C} \begin{bmatrix} f_{k_1,p}(M) \\ \vdots \\ f_{k_r,p}(M) \end{bmatrix} \tag{9}$$

where $\tilde{X}^{(p)}$ denotes the extrapolation of $\hat{X}^{(p)} \in \mathbf{R}^{N \times s}$, the submatrix of $\hat{X} \in \mathbf{R}^{N \times L}$ corresponding to the p direction.

Proof: Proof is obtained directly from Claim 2 and the assumption $\mathbf{O}_{k_l}^{(j)}(M) = f_{l,p}(M)$, $l \in \{1, \cdots, r\}$, $j \in \{x, y, z\}$. ■

This claim assumes that the complete system information can be obtained from the system dynamics over time period s and that the linear transformation in Claim 2 is enough to describe the relationship between $\hat{X}$ and $\mathbf{O}_{k_l}$, $l \in \{1, \cdots, r\}$. However, this is usually not true so we can only approximate $\tilde{X}^{(p)}$ based on $f_{l,p}$, $l \in \{1, \cdots, r\}$, $p \in \{x, y, z\}$. Through time series identification techniques, e.g. neural network, genetic algorithm, wavelet or simply linear fitting methods in the Matlab system identification toolbox, we can fit each time series $\mathbf{O}_{k_l}^{(p)}$ by $f_{l,p}$ so that

$$\mathbf{O}_{k_l}^{(p)}(1 : s) \approx f_{l,p}(1 : s), \; l \in \{1, \cdots, r\}, \, p \in \{x, y, z\}$$

Although the real system is not necessary linear, we can assume that, for relatively short time M, $s \ll M \ll \infty$, the linear relationship in Claim 2 can provide a good approximation. Thus we can approximate the dynamics of $\hat{X}$ at time M by (9).

Based on the discussion above, we propose the dynamic coarse graining algorithm:

Algorithm 1:

1) Record trajectory data for s time steps and construct R, $\mathbf{O}$ and $\hat{\mathbf{O}}$ matrices;
2) Find the most representative local feature indices $k_1, \cdots, k_r$ and calculate the $\mathbf{C}$ matrix;
3) Identify $f_{l,p}$ based on $\mathbf{O}_{k_l}^{(p)}(1 : s), l \in \{1, \cdots, r\}$;
4) Recover $\tilde{X}_j^{(p)}(t)$ in terms of $f_{l,p}(t)$, $l \in \{1, \cdots, r\}$ through (9), $s < t \leq M$, $j \in \{1, \cdots, N\}$

We remark here if the dynamics is a slow process, given limited time, we may not be able to collect enough data to model the whole system. Moreover, linear approximation to a nonlinear relationship will be inaccurate as time goes on. We call this the undersampling problem. Aware of this, we adopt a iterative scheme similar to that in [10], [17] to model the long term system dynamics.

Algorithm 2:

1) Run the detailed simulation for time duration s, adopt Algorithm 1 to choose seed dynamics $\mathbf{O}_{k_l}$ and extrapolate them to time step M, approximate $\tilde{X}(M)$ based on $\tilde{\mathbf{O}}_{k_l}(M)$.
2) Map $\tilde{X}(M)$ back to real trajectories $\mathbf{x}(\mathbf{M})$ by multiplying the standard deviation and adding the mean value; go to Step 1.

Note here the iterative algorithm allows the algorithm to represent system dynamics over different time duration s by selecting different seed dynamics and recovery formula (9).

IV. Simulation

A. Illustrative Example

We constructed a spring mass damper system as shown in Figure 1, where 10 masses were connected by 11 linear springs. In Figure 1, the initial position of masses were $x_1 = 0.05$, $x_{i+1} = x_i + 0.1$, $i \in \{1, \cdots, 10\}$. The spring equilibrium lengths were all at $E_i = 1/11$. Damping coefficients were all equal to 1. Spring constants were 950 for l_1 to l_3, 1000 for l_5 to l_7, 1050 for l_9 to l_{11} and 10^{-5} for l_4, l_8. A duration of 10s data was simulated and the position dynamics are shown in Figure 2. The first 1 s length data were used to calculate the correlation matrix R and the remaining 9 s simulation data were test data to compare with the recovered trajectories of all masses.

Through PCA we chose $r = 3$ in order to obtain small approximation error. According to the method discussed in Section III-B we picked three seed masses as shown in Figure 1. The corresponding seed dynamics $\mathbf{O}_{k_l}, l \in \{1, 2, 3\}$ were modelled by a modified version of the Matlab identification toolbox. A function set that contains sinusoid and exponential functions were used for modeling the seed dynamics $\mathbf{O}_{k_l}, l \in \{1, 2, 3\}$. Based on the identified functions, seed dynamics were extrapolated for an extra 9 s, then the recovered trajectories of $\hat{X}$ were calculated according to (9). The comparison between the modeled and real time series $\mathbf{O}_{k_l}$ over the first 1 s is shown in Figure 3. Figure 4 shows the comparison between the real and the recovered trajectories of $\hat{X}$, for Mass 2 and Mass 10 over the entire 10 s. The others were omitted due to space constraints, and they show similar results.

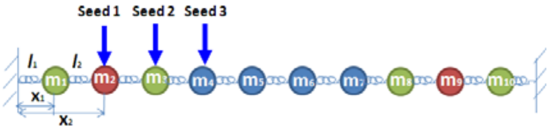

Figure 1. Mass spring damper system

As we can see from the construction of the system as well as in Figure 2, masses m_1 to m_3, m_8 to m_{10} and masses m_4 to m_7 have different dynamic behavior. So there should be at least one seed in each group. As we chose $r = 3$, dynamics of group m_1 to m_3, m_8 to m_{10} had been further decomposed into two different features that are represented by two different seeds, which matches what is shown in Figure 1. From Figure 3, we see that with the modified identification toolbox, the modelled seed dynamics fit the real seed dynamics very well. By comparing the real and recovered dynamics of $\hat{X}$ in Figure 4, we see that the proposed algorithm introduces very small estimation error even over the long time (9 s) extrapolation.

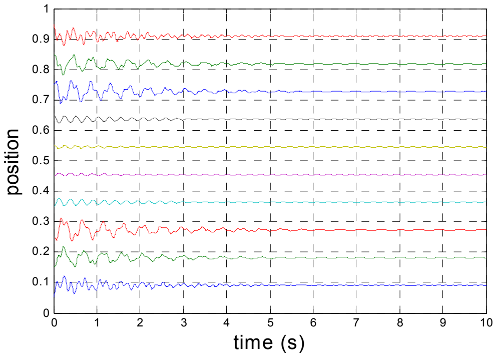

Figure 2. Position trajectories of masses.

Curves from low to high correspond to Masses 1 through 10.

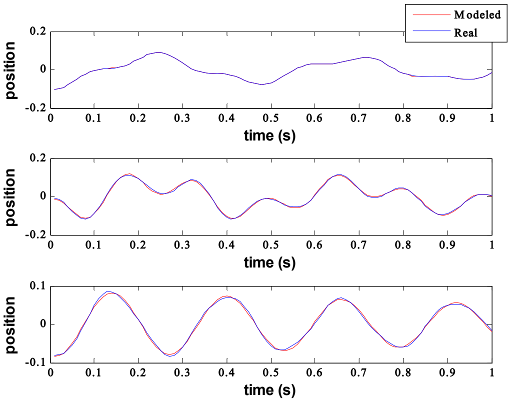

Figure 3. The modeled and true seed dynamics over 1 s.

In order, Seed 1 (Mass 2), Seed 2 (Mass 3), Seed 3 (Mass 4)

Positions are preprocessed, i.e. centered and normalized.

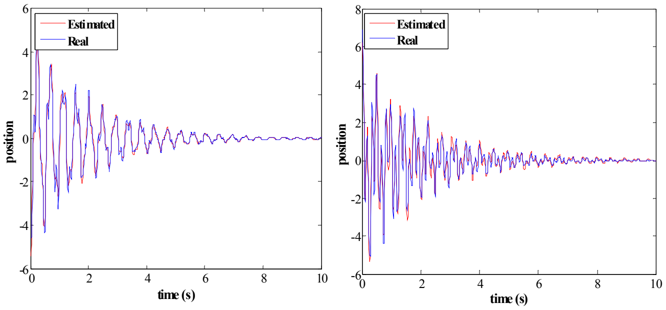

Figure 4. The real and estimated trajectories for

$\hat{X}_2$ (left) and $\hat{X}_2$ (right)

B. Polymer System Dynamics Simulation

We built the PVC polymer model and run the MD simulation in a simulation software MOE (Molecular Operating Environment).

Sixteen syndiotactic poly(vinyl chloride) polymer chains with 16 backbone carbons each were built in MOE within a periodic cell with size 20.48 × 20.96 × 20.32 $\mathring{A}$. Initial monomer conformations were constructed according to [8]. The potential energy of this system was then minimized using the steepest descent method, until the absolute value in the Eulidean norm of the total potential energy gradient was smaller than 0.05 kJ/mol. Then a 250 ps molecular dynamics simulation with a 2 fs simulation time step was performed at 500 K with the NPT ensemble. The pressure is set to be 2×10^6 Pa. We give in Figure 5 the conformations and WAXD curves at the initial time as well as at 250 ps for the real (fully simulated) system. We can see that at the beginning the polymer chains are very well aligned while when it reaches 250 ps the system conformation changed significantly and the chain arrangement is more disordered. Also the WAXD curve changed obviously from the initial time to 250 ps for the real system.

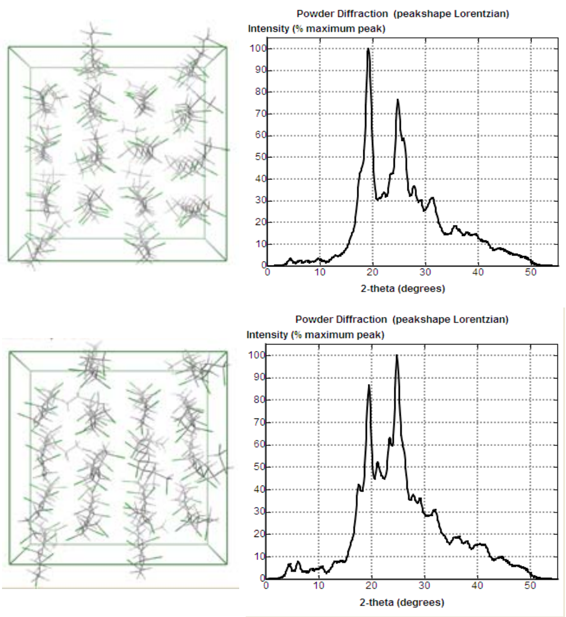

Figure 5. Full MD simulation for PVC

Top left: Conformation - 0 ps, Top right: WAXD - 0 ps

Lower left: Conformation - 250 ps, Lower right: WAXD - 250 ps

We studied different scenarios by varying the training/test time ratio. Note by training time we mean we ran the detailed MD simulation over this period while the test time period is when we extrapolated the system. Given similar estimation accuracy, the lower the training/test time ratio is, the more efficient of the proposed algorithm is.

In Scenario 1, we used the first 150 ps length data as training data and calculated correlation matrix R based on it; the remaining 100 ps simulation data were used for the test purpose. Through PCA we chose $r = 9$, and thus picked 9 seed atoms according to the sparsification approach. The corresponding seed dynamics $\mathbf{O}_{k_l}, l \in \{1, \cdots, 9\}$ were modeled by the modified Matlab identification toolbox with sinusoid and exponential function basis. Based on the identified functions, the seed dynamics were extrapolated for an extra 100 ps, then the recovered trajectories of $\tilde{X}$ were calculated according to (9). Taking the value of $\tilde{X}$ at 250 ps, and mapping it back to real trajectories $\mathbf{x}$ at 250 ps by multiplying the standard deviation as well as adding the mean value, we obtained the position of backbone carbons of the polymer system at 250 ps. Fixing the position of backbone carbon and minimizing the potential energy until the absolute value in the Eulidean norm of the total potential energy gradient was smaller than 0.01 kJ/mol, we obtained the recovered conformation for the polymer system at time 250 ps. With this conformation we calculated the WAXD curve of the recovered system at 250 ps. We repeated the above procedure so that training/test time is of 100 $ps/150$ ps in Scenario 2 or 55 $ps/195$ ps in Scenario 3. In both cases, we chose $r = 7$ manually as in PCA, by observing the gaps among eigenvalues of the R matrix.

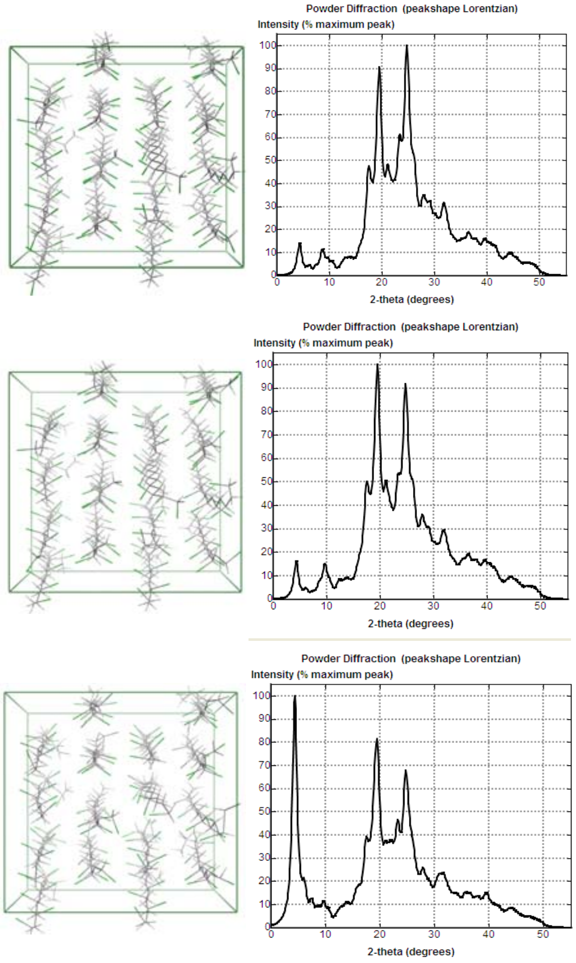

Figure 6. The extrapolated system at 250 ps

First row: Conformation and WAXD of Scn. 1 - 150/100 ps

Second row: Conformation and WAXD of Scn. 2 - 100/150 ps

Third row: Conformation and WAXD of Scn. 3 - 55/195 ps

We give in Figure 6 the extrapolated system conformation and WAXD curves at time 250 ps for the three scenarios. As we can see, in Scenario 1, the recovered system is almost equivalent to the real system at 250 ps, both in conformation and WAXD curves. The recovered system in Scenario 2 also provided a good approximation to the real system, as we can see from the comparison in conformation and WAXD, although the dominant peak has now changed. In Scenario 3, the recovered system provided a low quality approximation at 250 ps, as we can see clearly that the recovered system conformation is more ordered, which is obviously different from that of the real system. This might be due to that the system information contained in the dynamics over short training time is limited and not enough to represent the information over the long term, e.g. 195 ps here. The results in these three Scenarios indicate the sensitivity of the extrapolation accuracy to the training time. We will study this issue, as well as to choose the best time series identification basis functions/approaches, in our ongoing work.

It is also remarkable that only 9 "seed" atoms in Scenario 1 and 7 in Scenario 2 which represent 9 / 7 groups, are selected to reconstruct the whole system composed of 144 monomers. Compared to the method in [4], [15], which would introduce 144 groups, our simulation complexity is considerably reduced here.

Note in this example we did not include multiple iterations as in Algorithm 2. As long as the extrapolated system in one iteration can approximate the real system accurately, the iterations are used to repeat the process to retain the long time extrapolation accuracy.

V. Conclusions and Future Work

A coarse graining simulation method has been proposed so that, based on short term system dynamics, it can automatically identify a low number of "seeds" based on correlations in the dynamic motion of all states. The trajectories of "seeds" are then extrapolated. A simple matrix transformation has been proposed to calculate trajectories of the whole system from the extrapolated "seed" trajectories. Iteration of the proposed procedure, i.e. the procedure combining detailed simulation and coarse grained simulation, has been suggested to overcome the undersampling problem. The effectiveness of the proposed approach has been demonstrated from several simulations.

In our future work, we will characterize the local features (seed atoms) that comprise the intermediate order. We plan to study on more advanced time series identification approaches, to specify suitable time series identification basis functions, to develop a nonlinear relationship between the seed and the system dynamics, as well as to identify an optimal training/test time ratio so we can predict longer term dynamics from shorter training period accurately. The ultimate goal is to use the proposed algorithm to formulate mesoscale models that retain the dynamics and physics responsible for the intermediate order, and to use such models to explore the formation of intermediate order in new polymers.

Acknowledgment: Acknowledgment is made to the Donors of the American Chemical Society Petroleum Research Fund for support of this research. Moreover, the authors acknowledge Garrett Dowdy for his work on building the molecular dynamics simulation of PVC.

References

[1] S. Ahmed, S. Bidstrup, P. Kohl, and P. Ludovice, "Prediction of stereoregular poly(norbornene) structure using a long-range RIS model," *Makromol. Chem., Macromol. Symp.*, vol. 122, pp. 1–10, 1998.

[2] S. Ahmed, P. Kohl, and P. Ludovice, "Microstructure of 2,3-erythro di-isotactic polynorbornene from atomistic simulation," *J. Comp. and Theor. Polym. Sci.*, vol. 10, pp. 221–233, 2000.

[3] M. Balsera, W. Wriggers, Y. Oono, and K. Schulten, "Principal component analysis and long time protein dynamics," *Journal of Physical Chemistry*, vol. 100, pp. 2567–2572, 1996.

[4] R. Faller, "Automatic coarse graining of polymers," *Polymer*, vol. 45, pp. 3869–3876, 2004.

[5] Y. Frederix, G. Samaey, C. Vandekerckhove, T. Li, E. Nies, and D. Roose, "Lifting in equation-free methods for molecular dynamics simulations of dense fluids," *Discrete and Continuous Dynamical Systems, B*, vol. 11(4), pp. 855–874, 2009.

[6] J. R. Fried and D. K. Goyal, "Molecular simulation of gas transport in poly(1-(trimethylsilyl)-1-propyne)," *J. Polym. Sci., Polym. Phys.*, vol. 36, pp. 519–536, 1998.

[7] H. Fukunaga, J. Takimoto, and M. Doi, "A coarse-graining procedure for flexible polymer chains with bonded and nonbonded interactions," *Journal of Chemical Physics*, vol. 116, pp. 8183–8190, 2002.

[8] R. Hobson and A. Windle, "Crystallization and shape emulation in atactic poly(vinyl chloride) and polyacrylonitrile," *Polymer*, vol. 34(17), pp. 3582–3596, 1993.

[9] T. Hoskins, W. Chung, A. Agrawal, P. Ludovice, C. Henderson, L. Seger, L. Rhodes, and R. Shick, "Bis(trifluoromethyl)carbinol-substituted polynorbornenes: Dissolution behavior," *Macromolecules*, vol. 37(12), pp. 4512–4518, 2004.

[10] G. Hummer and I. G. Kevrekidis, "Coarse molecular dynamics of a peptide fragment: Free energy, kinetics, and long-time dynamics computations," *Journal of Chemical Physics*, vol. 118(23), pp. 10 762–10 733, 2003.

[11] I. G. Kevrekidis and G. Samaey, "Equation-free multiscale computation: algorithms and applications," *Annual Review of Physical Chemistry*, vol. 60, pp. 321–344, 2009.

[12] Y. Lee, K. Lee, and S. Pan, "Local and global feature extraction for face recognition," *Lecture Notes in Computer Science*, vol. 3546, pp. 219–228, 2005.

[13] P. Ludovice, S. Ahmed, J. V. Order, and J. Jenkins, "Simulation of intermediate order in polymer glasses," *Macromol. Symp.*, vol. 146, pp. 235–242, 1999.

[14] P. Ludovice and U. Suter, *In Computational Modeling of Polymers, Bicerano J., Ed.* New York: Marcel Dekker, 1992.

[15] G. Milano and F. Mullar-Plathe, "Mapping atomistic simulations to mesoscopic models: A systematic coarse-graining procedure for vinyl polymer chains," *Journal of Physical Chemistry, B*, vol. 109, pp. 18 609–18 619, 2005.

[16] P. S. Penev and J. J. Atick, "Local feature analysis: A general statistical theory for object representation," *Computation in Neural Systems*, vol. 7, pp. 477–500, 1996.

[17] R. Rico-Martínez, C. W. Gear, and I. Kevrekidis, "Coarse projective KMC integration: Forward/reverse initial and boundary value problems," *Journal of Computational Physics*, vol. 196(2), pp. 474–489, 2004.

[18] B. Wilks, W. Chung, P. Ludovice, M. Rezac, P. Meaking, and A. Hill, "Structural and free-volume analysis for alkyl-substituted palladium-catalyzed poly(norbornene): A combined experimental and Monte Carlo investigation," *Journal of Polymer Science Part B, Polymer Physics*, vol. 44, pp. 215–233, 2005.

[19] Y. Xue, P. J. Ludovice, and M. A. Grover, "Local feature analysis based clustering algorithm with application to polymer dynamics model reduction," *Accepted for the IEEE Conference on Decision and Control*, December, 2010.

[20] Z. Zhang and W. Wriggers, "Local feature analysis: A statistical theory for reproducible essential dynamics of large macromolecules," *Proteins: Structure, Function, and Bioinformatics*, vol. 64, pp. 391–403, 2006.

[21] ——, "Coarse-graining protein structures with local multivariate features from molecular dynamics," *Journal of Physical Chemistry, B*, vol. 112, pp. 14 026–14 035, 2008.

[22] T. Zheng and J. R. Fried, "Monte Carlo simulation of the sorption of pure and mixed alkanes in poly[1-(trimethylsilyl)-1-propyne]," *Separat. Sci. Tech.*, vol. 36, pp. 959–973, 2001.