MODEL REDUCTION FOR A POWER GRID MODEL

JING LI AND PANOS STINIS∗
Pacific Northwest National Laboratory
Richland, WA 99354, USA

(Communicated by Kevin Lin)

ABSTRACT. We examine the complexity of constructing reduced order models for subsets of the variables needed to represent the state of the power grid. In particular, we apply model reduction techniques to the DeMarco-Zheng power grid model. We show that due to the oscillating nature of the solutions and the absence of timescale separation between resolved and unresolved variables, the construction of accurate reduced models becomes highly non-trivial because one has to account for long memory effects. In addition, we show that a reduced model that includes even a short memory is drastically better than a memoryless model.

1. Introduction. The importance of stability of the power grid cannot be overstated (see, e.g., [6, 15, 19, 16, 12, 20] for a collection of recent articles on various aspects of power grid stability). The study of stability of the power grid usually involves large-scale computations. With this in mind, in the current work we examine the feasibility of accurate reduced order models for systems of equations describing power grid dynamics. While model reduction approaches for power grid dynamics have had a long history (see, e.g., [5, 14, 13] and references therein), they have mostly been based on assumptions of time-scale separation, space-scale separation, approximate linearization, etc. In the current work, we do not make such assumptions and we examine the problem of model reduction for power grid models in its more general form. In particular, we focus on the DeMarco-Zheng power grid model [22]. The DeMarco-Zheng model, when augmented by an appropriate line failure mechanism, can be used to study cascade failures. Here, we examine the DeMarco-Zheng model without the line failure mechanism and investigate how to construct reduced order models for subsets of the state variables.

The absence of clear timescale separation between the state variables renders the construction and subsequent simulation of a reduced model to be non-trivial. In particular, there are long memory effects [7, 4]. This means that constructing an accurate reduced model for a subset of the variables, called resolved variables,
requires accounting for the history of those variables. In fact, we show that if we want the reduced model to retain its accuracy, the length of the history that we need to account for turns out to be equal to the interval of time for which we wish to evolve the reduced model. This is one of the most challenging cases of model reduction because we have to find a way to accurately represent such long memory effects.

We employ the Mori-Zwanzig (MZ) formalism (see, e.g., [2]) which in principle allows the construction of reduced order models with any memory length. This facilitates the investigation of the effect of truncating the memory length as well as the cost needed to obtain an accurate representation of the memory. Our results for the DeMarco-Zheng model are informative, as we now explain.

First, as mentioned above, accurate prediction by a reduced model requires the length of the memory to be equal to the time interval we wish to obtain a prediction for. Second, even if we retain a long memory, the simulation of the integro-differential equations of the reduced model (the integral part is due to the memory) requires an adequately small timestep to remain stable. Third, if we truncate the memory length, the reduced model quickly loses accuracy for time intervals longer than the truncated memory length, but it does not lose its stability provided we use a small enough timestep. Fourth, even the inclusion of a short memory is better than a memoryless reduced model. Finally, at least for the initial conditions examined here, we observe that while the DeMarco-Zheng model is nonlinear, the memory is accurately represented using a linear function of the resolved variables (we offer a short qualitative analysis for this behavior). This is interesting because it can lead to simplification of the reduced model, which is not initially obvious.

Motivated by the encouraging results we obtained for the DeMarco-Zheng model and to put our results in context, we used a simple example of a single particle coupled linearly to a heat bath of linear oscillators put forth by Zwanzig (see, e.g., Section 1.6 in [23]), where the exact reduced model can be constructed analytically. For this simple example, we found the same qualitative features in the behavior of the reduced model as in that for the considerably more complicated DeMarco-Zheng model. In this sense, the results we obtained for the reduced order model of the DeMarco-Zheng model are optimal even though we had to perform certain approximations to facilitate the computations.

The paper is organized as follows. In Section 2, we present the MZ formalism and explain our choice of a projection operator. In Section 3, we present the n-bus DeMarco-Zheng model as well as the 3-bus version that we used for our numerical experiments. Section 4 contains the construction of the reduced order model, results of its simulation, a short qualitative analysis for its behavior, and numerical results for the exact reduced model for the simple example of a single particle coupled linearly to a heat bath of linear oscillators. Finally, Section 5 contains a short discussion of the results and suggestions for future work.

2. The Mori-Zwanzig formalism. We begin in Section 2.1 with a brief presentation of the Mori-Zwanzig (MZ) formalism for constructing reduced models of systems of differential equations (see [2, 3, 4] for more details). The MZ formalism belongs in the class of projection methods. Therefore, the choice of projection operator can significantly affect the final form of the reduced models. Section 2.2 contains a discussion of various projection operators used in the literature as well as the choice we made in the current work.
2.1. The Mori-Zwanzig equation. Suppose we are given the system
\[ \frac{du(t)}{dt} = R(t, u(t)), \] (1)
where \( u = \{u_k\}, \) \( k \in H \cup G \) with initial condition \( u(0) = u_0. \) The unknown variables (modes) are divided into two groups: one group is indexed in \( H \) and the other indexed in \( G. \) Our goal is to construct a reduced model for the modes in the set \( H. \) The system of ordinary differential equations we are given can be transformed into a system of linear partial differential equations
\[ \frac{\partial \varphi_k}{\partial t} = L \varphi_k, \quad \varphi_k(u_0, 0) = u_{0k}, \quad k \in H \cup G \] (2)
where \( L = \sum_{k \in H \cup G} R_k(u_0) \frac{\partial}{\partial u_0} \) \( k. \) The solution of (2) is given by \( u_k(u_0, t) = \varphi_k(u_0, t). \)
Using semigroup notation, we can rewrite (2) as
\[ \frac{\partial}{\partial t} e^{tL} u_{0k} = L e^{tL} u_{0k} \]
where we have used Dyson’s formula
\[ e^{tL} = e^{tQL} + \int_0^t e^{(t-s)L} P Le^{sQL} Q L u_{0k} ds, \quad k \in H, \] (3)
The solution of (2) can be rewritten as
\[ \frac{\partial}{\partial t} e^{tL} u_{0k} = e^{tQL} \int_0^t e^{(t-s)L} P Le^{sQL} Q L u_{0k} ds. \] (4)
Equation (3) is the Mori-Zwanzig identity. Note that this relation is exact and is an alternative way of writing the original ODE. It is the starting point of our approximations. Of course, we have one such equation for each of the resolved variables \( u_k, k \in H. \) The first term in (3) is usually called Markovian because it depends only on the values of the variables at the current instant. The second is called “noise,” and the third is called “memory” (see [4] for a discussion of the significance of each term).
If we write
\[ e^{tQL} Q L u_{0k} = w_k, \]
\( w_k(u_0, t) \) satisfies the equation
\[ \begin{align*}
\frac{\partial}{\partial t} w_k(u_0, t) &= Q L w_k(u_0, t) \\
w_k(u_0, 0) &= Q L u_{0k} = R_k(u_0) - (PR_k)(\hat{u}_0).
\end{align*} \] (5)
If we apply the projection operator \( P \) on (5) we find
\[ P \frac{\partial}{\partial t} w_k(u_0, t) = P Q L w_k(u_0, t) = 0, \]
because \( PQ = 0. \) Also, for the initial condition,
\[ P w_k(u_0, 0) = P Q L u_{0k} = 0 \]
by the same argument. Thus, the solution of (5) is at all times orthogonal to the range of \( P. \) We call (5) the orthogonal dynamics equation (see more details in [2]).
Because the solutions of the orthogonal dynamics equation remain orthogonal to the range of $P$, we can project the Mori-Zwanzig equation (3) and find

$$\frac{\partial}{\partial t} P e^{tL} u_0 = P e^{tL} P L u_0 + P \int_0^t e^{(t-s)L} P L e^{sQ} L u_0 ds. \quad (6)$$

We will not present here more details about how to start from Eq. (6) and construct reduced models of different orders for a general system of ODEs. Such constructions have been documented thoroughly elsewhere (see, e.g., [3]). However, we will provide such details for the specific example of the power grid equations in Section 4.2.

2.2. The choice of the projection operator $P$. Before we proceed, we would like to comment on the choices of the projection operator $P$ that have appeared in the literature as well as the choice that we have opted for in the current work. As we have seen in Eq. (3), what the MZ formalism offers is a way to decompose the RHS of the equations for the resolved variables. This decomposition involves three terms. The choice of the projection operator affects how the information content of the RHS of the equation is distributed among these three terms. The popular choices for the projection operator (see, e.g., [3]) include the conditional expectation and the finite-rank projection with respect to an invariant measure for the system. The projection operator based on the conditional expectation also comes with the added property of being optimal in an $L^2$ sense. One uses these projection operators when the objective is to produce trajectories for the resolved variables starting from initial conditions sampled from the invariant measure or when one wants to study the relaxation to the invariant measure starting from a non-typical initial condition.

The projection operator $P$ we have chosen is defined as

$$(P f)(\hat{u}_0) = P(f(u_0)) = P(f(\hat{u}_0, \tilde{u}_0)) = f(\hat{u}_0, \tilde{u}_0).$$

What this definition means is that our projection operator, when applied to a function of the initial conditions, assigns the value $\tilde{u}_0$ to the unresolved variables, where $\tilde{u}_0$ is a chosen vector of values for the unresolved variables. One way to think about this that makes contact with other choices for the projection operator that have appeared in the literature, is that we have used a projection operator which is defined with respect to a measure on the initial conditions that has a delta measure for the unresolved variables centered at $\tilde{u}_0$. Thus, it does not allow fluctuations in the initial condition of the unresolved variables. This particular choice of projection operator with $\tilde{u}_0 = 0$ has been used successfully by the current authors previously to tackle a variety of problems from detection and tracking of singularities to locating bifurcations and uncertainty quantification (see, e.g., [8, 10, 11, 17, 18]).

Our choice of operator comes with a few distinctive features. First, unlike the more popular choices of the projection operator presented above, our chosen operator commutes with nonlinear functions. This property turns out to be convenient for the analytical calculations of some, but not all, of the expressions that appear in the MZ formalism (we will return to this point in Section 4.2.3). Second, for the case $\tilde{u}_0 = 0$, our choice of projection operator is the Galerkin projection that sets the unresolved variables to zero for all times. However, note that our reduced models incorporate the memory terms too, which are there to account for the interaction of the resolved and unresolved variables. Third, because our choice of projection operator does not allow fluctuations in the initial conditions of the unresolved variables, when we apply the projection operator to the MZ equation to cancel the
noise term, we get an equation that is valid \textit{pathwise}. In other words, the projected, noiseless MZ equation (6) is valid for the prediction of each trajectory that starts with whatever initial conditions we have chosen for the resolved variables and \( \tilde{u}^0 \) for the unresolved variables.

We will return to the discussion of the choice of the projection operator at the end of Section 4.1.

3. The DeMarco-Zheng model for the power grid.

3.1. The \( n \)-bus system. The state of the \( n \)-bus DeMarco-Zheng model for the power grid can be described in terms of a vector \( u = (\omega_g, \delta_g, \delta_l, V^l) \) where \( \omega_g \) contains the angular velocities of the generators, \( \delta_g, \delta_l \) contain the bus voltage phase angles with respect to an arbitrary synchronous reference frame of the generators and loads, respectively, and \( V^l \) contains the voltage magnitudes of the loads [22].

We define the energy function \( \Phi \) by

\[
\Phi(u) = \frac{1}{2} \omega_g^T M_g \omega_g + (P_g + P^a_g) \cdot \delta_g + (P_l + P^a_l) \cdot \delta_l + (Q_g + Q^a_l) \cdot \ln V^l
\]

where \( \ln V^l \) is the \( \ln V^l \). Also, \( P_g, P_l \) is the active power injected into the generator and load buses, respectively, \( Q^a_l \) is the reactive power injected into the load buses, and \( P^a_g, P^a_l, Q^a_l \) is the power absorbed by the generator and load buses, respectively.

The gradient of \( \Phi \) is given by

\[
\nabla_u \Phi(u) = \begin{bmatrix}
M_g \omega_g \\
P_g + P^a_g \\
P_l + P^a_l \\
V^l \cdot (Q_g + Q^a_l)
\end{bmatrix}
\]

where \( (V^l)^{-1} = V^l \). We also define the full-rank, negative semi-definite matrix \( A \) given by

\[
A = \begin{bmatrix}
-M_g^{-1} D_g M_g^{-1} & -M_g^{-1} & 0 & 0 \\
-M_g^{-1} & 0 & 0 & 0 \\
0 & 0 & -D_l^{-1} & 0 \\
0 & 0 & 0 & -1/\epsilon * I
\end{bmatrix}
\]

where \( M_g, D_g, D_l, \epsilon \) are system parameters and \( I \) is the identity matrix. With the definitions (7)-(9), the dynamic equations for the DeMarco-Zheng model are given by

\[
\frac{du}{dt} = A \nabla_u \Phi(u).
\]

Given the negative semi-definiteness of the matrix \( A \), we see that

\[
\frac{\partial \Phi}{\partial t} = \nabla_u \Phi(u) \cdot \frac{du}{dt} = \nabla_u \Phi^T(u) A \nabla_u \Phi(u) \leq 0.
\]

We note that the DeMarco-Zheng model was developed to study the problem of cascading failure, and its energy function contains an additional term to account for the possible line failure between nodes [22]. In the current work, we are not interested in constructing reduced models for a system where line failures may occur, so we omit this extra term from the energy function. Reduced models for systems with possible line failures will be described in a future publication.
3.2. The 3-bus system. In this section, we focus on the $n=3$ case, i.e., a 3-bus system (see [22] and Fig. 1). In particular, Bus 1 is the reference generator. The voltage magnitude $V_1$ and the voltage phase angle $\delta_1$ are fixed, while the injected powers $P_1$ and $Q_1$ vary according to the system. Bus 2 is a regular generator. The voltage magnitude $V_2$ and the active power injected $P_2$ are fixed, while the reactive power $Q_2$ and voltage phase angle $\delta_2$ vary. Bus 3 is a load bus. The injected powers $P_3$ and $Q_3$ are fixed while the voltage magnitude $V_3$ and voltage phase angle $\delta_3$ vary. With this arrangement, there are five variables in the system: $\omega_1$, $\omega_2$, $\alpha_2 = \delta_2 - \delta_1$, $\alpha_3 = \delta_3 - \delta_1$, $V_3$. Thus, $u = (\omega_1, \omega_2, \alpha_2, \alpha_3, V_3)^T$.

The energy function $\Phi_{3bus}(u)$ is

$$\Phi_{3bus}(u) = \frac{1}{2} M_1 \omega_1^2 + \frac{1}{2} M_2 \omega_2^2 + \frac{1}{2} (b_1 + b_2) V_1^2 + \frac{1}{2} (b_1 + b_3) V_2^2 + \frac{1}{2} (b_2 + b_3) V_3^2 - b_1 V_1 V_2 \cos(\alpha_2) - b_2 V_1 V_3 \cos(\alpha_3) - b_3 V_2 V_3 \cos(\alpha_3 - \alpha_2) + P_2 \alpha_2 + P_3 \alpha_3 + Q_3 \ln(V_3)$$

(12)

The gradient $\nabla_u \Phi_{3bus}(u) = [\frac{\partial \Phi_{3bus}}{\partial \omega_1}, \frac{\partial \Phi_{3bus}}{\partial \omega_2}, \frac{\partial \Phi_{3bus}}{\partial \alpha_2}, \frac{\partial \Phi_{3bus}}{\partial \alpha_3}, \frac{\partial \Phi_{3bus}}{\partial V_3}]^T$ is given by

$$\frac{\partial \Phi_{3bus}}{\partial \omega_1} = M_1 \omega_1,$$

(13)

$$\frac{\partial \Phi_{3bus}}{\partial \omega_2} = M_2 \omega_2,$$

(14)

$$\frac{\partial \Phi_{3bus}}{\partial \alpha_2} = b_1 V_1 V_2 \sin(\alpha_2) + b_3 V_2 V_3 \sin(\alpha_2 - \alpha_3) + P_2,$$

(15)

$$\frac{\partial \Phi_{3bus}}{\partial \alpha_3} = b_2 V_1 V_3 \sin(\alpha_3) + b_3 V_2 V_3 \sin(\alpha_3 - \alpha_2) + P_3,$$

(16)

$$\frac{\partial \Phi_{3bus}}{\partial V_3} = (b_2 + b_3) V_3 - b_2 V_1 \cos(\alpha_3) - b_3 V_2 \cos(\alpha_3 - \alpha_2) + \frac{Q_3}{V_3}.$$  

(17)
The matrix $A$ is given by

$$
A = \begin{bmatrix}
-M_1^{-1}D_1M_1^{-1} & 0 & M_1^{-1} & 0 \\
0 & -M_2^{-1}D_2M_2^{-1} & -M_2^{-1} & 0 \\
-M_1^{-1} & M_2^{-1} & 0 & 0 \\
0 & 0 & 0 & -D_3^{-1} \\
0 & 0 & 0 & -\epsilon^{-1}
\end{bmatrix}.
$$

The values of the parameters $M_1, D_1, M_2, D_2, D_3,$ and $\epsilon$ used for the numerical experiments are provided in Table 1.

With the definitions (12)-(18), the dynamics for the 3-bus system are given by

$$
\frac{du}{dt} = A\nabla_u \Phi_{3\text{bus}}(u).
$$

The power flow equations are given by

$$
P_i = \sum_j B_{i,j}V_iV_j \sin(\delta_i - \delta_j) \text{ for all buses } (i = 1, 2, 3)
$$

$$
Q_i = -\sum_j B_{i,j}V_iV_j \cos(\delta_i - \delta_j) \text{ for loading buses } (i = 3)
$$

where $B_{i,j}$ are parameters. Note that with appropriately chosen $B_{i,j}$, the fixed-point equations $\nabla_u \Phi_{3\text{bus}}(u) = 0$ coincide with the power flow equations.

4. **Numerical results.** We want to examine the behavior of MZ reduced order models for the case of the DeMarco-Zheng model of the power grid. We focus on the case of a 3-bus system. In Section 4.1, we present results for the full-order 3-bus system, which helps us show the absence of time scale separation between the different variables of the system. In Section 4.2, we present numerical results for the MZ reduced order model of the 3-bus system when we resolve only three of the five variables of the system.

4.1. **Full-order system.** The 3-bus system was simulated with the forward Euler scheme with initial conditions given by

$$
\omega_1(0) = 0, \quad \omega_2(0) = 0, \quad \alpha_2(0) = -0.16, \quad \alpha_3(0) = -0.3 \quad V_3(0) = 0.8.
$$

The values for the parameters of the 3-bus system are listed in Table 1. In this study, we choose the variables associated with the generators as the resolved variables, and the variables associated with the loads as the unresolved variables. As a result, the unresolved variables are $\alpha_3$ and $V_3$.

Figures 2 and 3 show the evolution of quantities associated with the first and second generators (resolved) and the load (unresolved), respectively. The purpose of showing these results is to establish that there is absence of timescale separation in the evolution of the resolved and unresolved variables. This is very important for the construction of the reduced model because it leads to long memory effects. In
particular, by examining Fig. 2(c) and Fig. 3(a) for the evolution of the resolved variable $\alpha_2$ and the unresolved variable $\alpha_3$, respectively, we see that they oscillate on very similar timescales. As a result, we expect that the memory term will be oscillating for a long time instead of decaying quickly as in problems with timescale separation. In Section 4.2, we will establish (1) that the memory is indeed long and (2) that naive treatment of such long memory effects leads to rapid loss of accuracy of the reduced model.

![Figure 2](image)

**Figure 2.** Full system - Evolution of resolved variables. a) $\omega_1$ for generator 1, b) $\omega_2$ for generator 2 and c) $\alpha_2$ for generator 2.

Now that we have seen the temporal behavior of the variables of the 3-bus system and before we proceed with the construction of the reduced model in the next section, we would like to make a couple comments about our choice of resolved variables and of the measures used to define the projection operator $P$ (recall the discussion in Section 2.2).

First, the choice of the resolved and unresolved variables is critical in model reduction. For large power grid models, the number of generator buses is much smaller than the number of load buses. So, one would be primarily interested in resolving the generator buses while treating most (if not all) of the load buses as unresolved. Also, for reasons of predicting possible overloading of the generator buses, one would like to have reduced order models for the evolution of the generator buses. That is the motivation behind our choice here.
We note that one may possibly find collective resolved variables that are complex functions of the generator and load bus variables and for which the resulting reduced model may be simpler. However, these collective variables may not have a clear interpretation. In contrast, the resolved (and unresolved) variables we have chosen have a clear interpretation.

Second, let us examine the effect of choosing a non-invariant measure to define the projection operator \( P \). As we can see from Eq. (11), the rate of change of the energy function of the DeMarco-Zheng model is non-positive. In fact, the DeMarco-Zheng model can be used to study the relaxation to the solution of the power flow equations, which are the fixed point equations of the RHS of the model (as is mentioned at the end of Section 3.2). This fixed point can be used to define a trivial invariant measure because the solution of the model from this fixed point does not change. However, we do not use this measure to define the projection. Even if we did, the reduced order model would still have a long memory. This is because we use as the initial condition a perturbation of the fixed point used to define the invariant measure. Thus, the reduced model would still have to account for the relaxation to the fixed point of the resolved and unresolved variables, among which there is no timescale separation.

4.2. Reduced system with three resolved variables.

4.2.1. The MZ formalism. To derive the MZ reduced model for the 3-bus system, we first rewrite it in the notation of the MZ formalism given in Section 2. As stated in Section 4.1, we choose the variables associated with the generators as the resolved variables, and those associated with the loads as the unresolved variables. As a result, the unresolved variables are \( \alpha_3 \) and \( V_3 \). As explained in Section 2.2, for any function \( f(\omega_{10}, \omega_{20}, \alpha_{30}, V_{30}) \) of the initial conditions of all the variables, we define the projection operator \( P \) as \( P f(\omega_{10}, \omega_{20}, \alpha_{30}, V_{30}) = f(\omega_{10}, \omega_{20}, \alpha_0, V_0) \). This means that we assign to the unresolved variables \( \alpha_3 \) and \( V_3 \) the values \( \alpha_0 \) and \( V_0 \). There is no variance allowed for the values of the unresolved variables. We note that in our numerical examples, we also tested the case of allowing the unresolved variables to have a small variance around \( \alpha_0 \) and \( V_0 \) but found no significant change in the performance of the reduced MZ models. We
decided to use the projection operator, which does not allow any variance because it facilitates the derivation of expressions needed for the MZ reduced models.

Even if we change the initial values for the resolved variables, the projection operator will assign to the unresolved variables the same values $\alpha_0^3$ and $V_0^3$. This is a Galerkin-type projection, but instead of assigning to the unresolved variables the value 0, we assign $\alpha_0^3$ and $V_0^3$. However, note the important difference that in a Galerkin model, the unresolved variables would be assigned the same value for all times. As explained in Section 2.2, we allow the unresolved variables to evolve, and their interaction with the resolved variables is accounted for through the memory term.

Let

\[
R_1 = a_{1,1}M_1\omega_1 + a_{1,3}[b_1V_1V_2\sin(\alpha_2) + b_3V_2V_3\sin(\alpha_2 - \alpha_3) + P_2]
+ a_{1,4}[b_2V_1V_3\sin(\alpha_3) + b_3V_2V_3\sin(\alpha_3 - \alpha_2) + P_3],
\]

\[
R_2 = a_{2,1}M_2\omega_2 + a_{2,3}[b_1V_1V_2\sin(\alpha_2) + b_3V_2V_3\sin(\alpha_2 - \alpha_3) + P_2],
\]

\[
R_3 = a_{3,1}M_3\omega_1 + a_{3,2}M_2\omega_2,
\]

\[
R_4 = a_{4,1}M_1\omega_1 + a_{4,4}[b_2V_1V_3\sin(\alpha_3) + b_3V_2V_3\sin(\alpha_3 - \alpha_2) + P_3],
\]

\[
R_5 = a_{5,5}[b_2 + b_3]V_3 - b_2V_1\cos(\alpha_3) - b_3V_2\cos(\alpha_3 - \alpha_2) + Q_3^3,
\]

where $a_{i,j}$ is the $(i,j)$-th element of the matrix $A$ (see Eq. (18)). Using (23)-(27), the system of equations (19) for the 3-bus system can be rewritten as

\[
\frac{du}{dt} = R(u).
\]  

4.2.2. The Markovian term. The system (28) is the starting point for the MZ reduced models. Recall that in the MZ formalism, all expressions are computed at $t = 0$, i.e., using the initial conditions (see Eq. (3)). The Markovian terms for the MZ reduced model equations for the three resolved variables $\hat{u} = (\omega_1, \omega_2, \alpha_2)^T$ are given by

\[
PL\omega_{10} = PR_1(u_0) = R_1(\hat{u}_0) = a_{1,1}M_1\omega_{10} + a_{1,3}[b_1V_1V_2\sin(\alpha_{20})
+ b_3V_2V_3\sin(\alpha_{20} - \alpha_3^0) + P_2]
+ a_{1,4}[b_2V_1V_3\sin(\alpha_3^0) + b_3V_2V_3\sin(\alpha_3^0 - \alpha_{20}) + P_3],
\]

\[
PL\omega_{20} = PR_2(u_0) = R_2(\hat{u}_0) = a_{2,1}M_2\omega_{20} + a_{2,3}[b_1V_1V_2\sin(\alpha_{20})
+ b_3V_2V_3\sin(\alpha_{20} - \alpha_3^0) + P_2],
\]

\[
PL\omega_{30} = PR_3(u_0) = R_3(\hat{u}_0) = a_{3,1}M_1\omega_{10} + a_{3,2}M_2\omega_{20} + a_{3,3}M_3\omega_{30} + a_{3,4}M_3\omega_{33},
\]

4.2.3. The memory term. We continue with the presentation of some details for the expressions in the memory term. We define the inner product

\[
(f, g) = \int fg\,\rho.
\]

where $\rho$ is the joint probability measure with respect to the initial conditions for all the variables in the full system. This can be any measure we choose—for the 3-bus system, we have chosen it to be a Gaussian for the following specific purpose. The projection operator $P$ we used sets the initial conditions for the unresolved variables equal to some pre-chosen values. In order to compute the memory kernel, we utilize a finite-rank projection operator $\hat{P}$ that approximates $P$. Because the original operator
For a function \( \varphi_j(u_0, t) \) of the initial conditions and time, the finite-rank projection reads (see, e.g., [3])

\[
(\mathbb{P} \varphi_j)(\hat{u}_0, t) = \sum_{\nu \in I} (\varphi_j(u_0, t), h^\nu(\hat{u}_0)) h^\nu(\hat{u}_0),
\]

where \( h^\nu(\hat{u}_0) \) are tensor product Hermite polynomials up to some order \( p \), \( \nu \) is the multi-index \( \nu = (\nu_1, \nu_2, \nu_3) \) with \( |\nu| = \sum_{i=1}^3 \nu_i \), and \( I \) is the index set up to order \( p \), i.e., \( I = \{ \mu | |\mu| \leq p \} \). For the 3-bus system, the highest-order \( p \) that we consider for the basis functions is 5 for a total of \( \frac{(3+5)!}{3!5!} = 56 \) basis functions.

For each \( j \leq 3 \), the component \( F_j(u_0, t) \) denotes the solution of the orthogonal dynamics

\[
\frac{\partial}{\partial t} F_j(u_0, t) = QL F_j(u_0, t) = LF_j(u_0, t) - PL F_j(u_0, t),
\]

\[ F_j(u_0, 0) = QL u_{0j} = R_j(u_0) - PL u_{0j}. \]  

Eq. (34) is equivalent to the Dyson formula [2]:

\[
F_j(u_0, t) = e^{tL} F_j(u_0, 0) - \int_0^t e^{(t-s)L} PL F_j(u_0, s) ds.
\]

Eq. (35) is a Volterra integral equation for \( F_j(u_0, t) \). To proceed, we replace the projection operator \( \mathbb{P} \) with the finite-rank projection operator \( \mathbb{P} \) and find

\[
K_j(\hat{u}_0, s) = PL F_j(u_0, s) \approx \mathbb{P} LF_j(u_0, s) = \sum_{\nu \in I} b^\nu_j(s) h^\nu(\hat{u}_0),
\]

where

\[
b^\nu_j(s) = (LF_j(u_0, s), h^\nu(\hat{u}_0)).
\]

Consequently,

\[
e^{(t-s)L} \mathbb{P} LF_j(u_0, s) = \sum_{\nu \in I} b^\nu_j(s) h^\nu(e^{(t-s)L} \hat{u}_0).
\]

We substitute \( e^{(t-s)L} \mathbb{P} LF_j(u_0, s) \) for \( e^{(t-s)L} PL F_j(u_0, s) \) in Eq. (35), multiply both sides by \( L \), and take the inner product with \( h^\mu(\hat{u}_0) \); the result is (dropping the approximation sign)

\[
(LF_j(u_0, t), h^\mu(\hat{u}_0)) = (Le^{tL} F_j(u_0, 0), h^\mu(\hat{u}_0)) - \int_0^t \sum_{\nu \in I} b^\nu_j(s)(Le^{(t-s)L} h^\nu(\hat{u}_0), h^\mu(\hat{u}_0)) ds.
\]

Eq. (38) is a Volterra integral equation for the function \( b^\nu_j(t) \), which can be rewritten as follows:

\[
b^\nu_j(t) = f^\nu_j(t) - \int_0^t \sum_{\nu \in I} b^\nu_j(s) g^{\nu\mu}(t - s) ds,
\]
where
\[ f^\mu_j(t) = (L e^{tL} f_j(u_0,0), h^\mu(\hat{u}_0)), \quad g^{\nu\mu}(t) = (L e^{tL} h^\nu(\hat{u}_0), h^\mu(\hat{u}_0)). \]

The functions \( f^\mu_j(t) \) and \( g^{\nu\mu}(t) \) can be found by averaging over a collection of experiments or simulations, with initial conditions drawn from the initial distribution. In this example, we use a sparse grid quadrature rule for the multi-dimensional integrals [21]. For the reduced model, we used a three-dimensional sparse grid with a total of 681 points.

Note that we need to clarify how the expressions \( L e^{tL} f_j(u_0,0) \) and \( L e^{tL} h^\nu(\hat{u}_0) \) which appear in the expressions for \( f^\mu_j(t) \) and \( g^{\nu\mu}(t) \) can be estimated. Both of these expressions involve an application of the operator \( L \) which differentiates w.r.t. to the initial conditions. To proceed with the differentiation, we use the formula (see [2])
\[
LG(u(u_0,t)) = \sum_{r=0}^M R_r(u_0) \frac{\partial}{\partial u_0} G(u(u_0,t)) = \sum_{r=0}^M R_r(u(u_0,t)) (\frac{\partial G}{\partial u_0})(u(u_0,t)),
\]
which holds for any function \( G(u(u_0,t)) \) of the solution at time \( t \) and where \( u(u_0,t) = e^{tL} u_0 \). For example, for \( L e^{tL} f_j(u_0,0) \) we find
\[
L e^{tL} f_j(u_0,0) = \sum_{r=0}^M R_r(u(u_0,t)) (\frac{\partial F}{\partial u_0})(u(u_0,t)).
\]

Finally, we perform one more projection to eliminate the noise term (see Section 2). There are two ways to compute the effect of the second projection on the memory term, in particular how to compute
\[
\int_0^t P e^{(t-s)L} K_j(\hat{u}_0,s) ds.
\]
We present both ways and discuss them comparatively. The first way is to use the property of the projection operator \( P \) so that it commutes with a nonlinear function. Thus, from the expression \( K_j(\hat{u}_0,s) = \sum_{\nu \in I} b^\nu_j(s) h^\nu(\hat{u}_0) \) (see Eq. (36)), we get
\[
\int_0^t P e^{(t-s)L} K_j(\hat{u}_0,s) ds = \sum_{\nu \in I} \int_0^t b^\nu_j(s) h^\nu(P e^{(t-s)L} \hat{u}_0) ds. \tag{41}
\]

The expression for the memory term in Eq. (41) is in the form of a convolution sum. Thus, to evaluate it for any time \( t \), we must keep the history of values \( h^\nu(P e^{(t-s)L} \hat{u}_0) \) from time 0 to time \( t \), which becomes increasingly expensive as time progresses. On the other hand, if the memory extends only to \( t_{memory} \) units of time in the past, then one needs to keep only the recent \( t_{memory} \) units of time of the history of \( h^\nu(P e^{(t-s)L} \hat{u}_0) \). The expression for the memory becomes
\[
\int_0^t P e^{(t-s)L} K_j(\hat{u}_0,s) ds = \sum_{\nu \in I} \int_0^{t_{memory}} b^\nu_j(s) h^\nu(P e^{(t-s)L} \hat{u}_0) ds. \tag{42}
\]

The second way to compute the effect of the second projection on the memory term is to use the approximation of the projection operator \( P \) by the operator \( \hat{P} \), and the
memory term becomes
\[
\int_0^t P e^{(t-s)L} K_j(\hat{u}_0, s) ds = \int_0^t \sum_{\nu\mu\in I} b_j(\nu\mu)(s) \gamma^{\nu\mu}(t-s) h^\nu(\hat{u}_0) ds,
\]
(43)
where
\[
\gamma^{\nu\mu}(t) = (e^{tL} h^\nu(\hat{u}_0), h^\mu(\hat{u}_0)).
\]
(44)
We can rewrite the memory term as
\[
\int_0^t P e^{(t-s)L} K(\hat{u}_0, s) ds = \int_0^t B(s) \Gamma(t-s) h(\hat{u}_0) ds,
\]
(45)
where for each instant \(s\), \(K(\hat{u}_0, s)\) is the vector with elements \(K_j(\hat{u}_0, s), j = 1, 2, 3\), \(B(s)\) is a rank-4 tensor with elements \(b_j(\nu)(s) = (LF_j(u_0, s), h^\nu(\hat{u}_0))\), and \(\Gamma(t-s)\) is a rank-6 tensor with elements \(\gamma^{\nu\mu}(t-s) = (e^{(t-s)L} h^\nu(\hat{u}_0), h^\mu(\hat{u}_0))\). Also, \(h(\hat{u}_0)\) is a rank-3 tensor with elements \(h^\nu(\hat{u}_0)\). The advantage of this representation is that we can rewrite it as
\[
\int_0^t P e^{(t-s)L} K(\hat{u}_0, s) ds = \left( \int_0^t B(s) \Gamma(t-s) ds \right) h(\hat{u}_0).
\]
(46)
The expression in parentheses in Eq. (46) can be computed offline and stored as a function of time \(t\). Then, it can be used to compute the memory term for different initial conditions extremely efficiently. The drawback of this approach is that unlike the first way, even if the memory is finite with length \(t_{\text{memory}}\), to compute \(\int_0^{t_{\text{memory}}} B(s) \Gamma(t-s) ds\) for \(t > t_{\text{memory}}\), we need to continue calculating \(\Gamma(t)\). For, say, \(t = t_{\text{memory}} + \epsilon\) (for \(\epsilon > 0\)) and \(s = 0\), the function \(\Gamma(t-s)\) in the integrand will need to be evaluated for an argument that is equal to \(t-s = t_{\text{memory}} + \epsilon > t_{\text{memory}}\). Thus, one needs to compute \(\Gamma(t)\) for \(t > t_{\text{memory}}\). This calculation again involves the evolution of the full system for multiple initial conditions, which is very expensive. For our example of the 3-bus system that involves an extremely long memory, we chose the second way because it makes the simulation of the reduced order model very efficient.

With this choice for the memory term, we obtain for the 3-bus system the following MZ reduced model,
\[
\frac{d}{dt}\hat{u}(t) = \hat{R}(\hat{u}(t)) + \int_0^t B(s) \Gamma(t-s) h(\hat{u}_0) ds, \quad \hat{u}(0) = \hat{u}_0.
\]
(47)
The vector \(\hat{R}(\hat{u}(t))\) contains the Markovian terms (29)-(31), and \(\hat{u}_0\) is the initial condition of the resolved variables.

Figure 4 shows the evolution of the memory integral for \(\omega_1\) and \(\omega_2\). The memory integral keeps oscillating for a long time, which makes the construction of accurate reduced models more difficult.

4.2.4. Numerical schemes for the reduced model. After setting up the reduced model (47), we need to solve it to obtain the evolution of the resolved variables. The system (47) contains integro-differential equations. Thus, we need to also decide how to evaluate the memory term, which is given by a (convolution) integral. We tested two different numerical implementations for the evaluation of the memory term.

The first scheme employs a forward Euler scheme both for the Markovian and the memory terms. Also, we approximate the memory integral with the trapezoidal
rule. In particular, we have

\[
\hat{u}_j(t_{n+1}) = \hat{u}_j(t_n) + \Delta t R_j(\hat{u}(t_n)) + \frac{\Delta t^2}{2} \sum_{i=0}^{n-1} \left( \sum_{\nu\in I} b^\nu_j(t_i) \gamma^{\nu\mu}(t_n - t_i) h^\mu(\hat{u}_0) + \sum_{\nu\in I} b^\nu_j(t_{i+1}) \gamma^{\nu\mu}(t_n - t_{i+1}) h^\mu(\hat{u}_0) \right)
\]

(48)

where \( j = 1, 2, 3 \) for the three resolved variables \( \omega_1, \omega_2, \alpha_2 \) and \( t_n = (n-1)\Delta t \). The numerical results we present used \( \Delta t = 5 \times 10^{-5} \) and \( \Delta t = 10^{-4} \).

The second scheme employs a forward Euler scheme for the Markovian term and an implicit Euler scheme for the memory term. Again, we approximate the memory integral with the trapezoidal rule. In particular, we have

\[
\hat{u}_j(t_{n+1}) = \hat{u}_j(t_n) + \Delta t R_j(\hat{u}(t_n)) + \frac{\Delta t^2}{2} \sum_{i=0}^{n} \left( \sum_{\nu\in I} b^\nu_j(t_i) \gamma^{\nu\mu}(t_n - t_i) h^\mu(\hat{u}_0) + \sum_{\nu\in I} b^\nu_j(t_{i+1}) \gamma^{\nu\mu}(t_n - t_{i+1}) h^\mu(\hat{u}_0) \right)
\]

(49)

where \( j = 1, 2, 3 \) for the three resolved variables \( \omega_1, \omega_2, \alpha_2 \) and \( t_{n+1} = n\Delta t \).

We did not observe large differences between the results of the two numerical schemes. We will present results only for the first scheme (forward Euler scheme for Markovian and memory terms).

4.2.5. Numerical results for variable memory length. In this section, we present results of the reduced model for various values of the memory length including the case without a memory term, which uses only the Markovian term. The range of integration in the memory term of the system (47) extends from 0 to \( t \). Note that due to the convolutional nature of the integral that represents the memory term, for \( s = 0 \) the memory kernel integrand corresponds to the current time \( t \). For \( s = t \),
it corresponds to the initial time 0. For system (47), the memory includes all the history of the system. We call this case the infinite memory case.

For the case of finite memory length, say $t_{\text{memory}}$, we must truncate the range of integration from 0 to $t_{\text{memory}}$, i.e., we take into account only the recent $t_{\text{memory}}$ units of the history. The system (47) is rewritten as

$$
\frac{d}{dt} \hat{u}(t) = \hat{R}(\hat{u}(t)) + \int_0^{t_{\text{memory}}} B(s) \Gamma(t - s) h(\hat{u}_0) ds, \quad \hat{u}(0) = \hat{u}_0.
$$

We conducted numerical experiments for several values of the memory length $t_{\text{memory}}$. We present results that show the sensitive dependence of the accuracy of the reduced model on the length of the memory. Figure 5 compares results for the reduced model with infinite memory and the reduced model without memory, which includes only the Markovian term in (47). The model with memory uses only basis functions up to order $p = 1$, namely linear basis functions, to represent the memory. For both the models with and without memory, we use $\Delta t = 5 \times 10^{-5}$.

First, it is obvious that the complete absence of memory is detrimental to the accuracy of the reduced model. Second, it is clear that the accuracy degradation rate of the memoryless reduced model is not the same for all the resolved variables. In particular, while the predictions of the memoryless model for $\omega_1$ and $\omega_2$ quickly become highly inaccurate, the prediction for $\alpha_2$ loses the phase but retains the order of magnitude of the exact solution. The reason for this can be found in the RHS of equation (25) for the evolution of $\alpha_2$. The RHS depends only on the resolved variables $\omega_1$ and $\omega_2$. Thus, the reduced model equation for this variable does not require the explicit presence of any memory term (see also (31)). Of course, because the other two resolved variables $\omega_1$ and $\omega_2$ are not accurately evolved by the memoryless model, the evolution of $\alpha_2$ is affected, albeit only in terms of losing the phase but not the magnitude of the exact solution.

The third observation concerns the cause of the accuracy degradation of the memoryless model. Note that here we have examined a case where the initial conditions of the unresolved variables were not allowed to have any fluctuations. So, the projection operator that we have chosen, which also does not allow fluctuations in the initial conditions of the unresolved variables, is not the cause of the degradation of the memoryless model. Rather, the memory effects are due to the absence of timescale separation between the resolved and unresolved variables, which makes it mandatory to account for the history of the resolved variables through a memory term.

Because we have to account for the history of the resolved variables, we investigate the actual length of the history (memory) that is necessary to guarantee accurate predictions of the reduced model for long times. Figure 6 shows the predictions of the reduced model for various memory lengths $t_{\text{memory}}$ (see also (50)). We observe the following: first, the prediction accuracy of the reduced model with memory length $t_{\text{memory}}$ already starts degrading for times that are slightly larger than $t_{\text{memory}}$. The second observation is that even if we truncate the memory length, the reduced model does not become unstable. On the contrary, the predictions for $\omega_1$ and $\omega_2$ (not shown) appear to asymptote toward the same long time value.

The high accuracy of the reduced model for $\Delta t = 5 \times 10^{-5}$ raises the question of whether we can increase the timestep without suffering a severe accuracy degradation. To this purpose, we conducted numerical experiments with $\Delta t = 10^{-4}$ and allowed the order $p$ of the finite-rank projection of the memory to vary between 0.
Figure 5. Reduced model for 3-bus system. Comparison of reduced models with infinite memory and without memory. (a) Evolution of the resolved variable $\alpha_2$, (b) Evolution of the resolved variable $\omega_1$, and (c) Evolution of the resolved variable $\omega_2$.

and 5. In other words, we want to see if we can compensate for a larger timestep with a higher-order finite-rank projection. Figure 7 shows the evolution of the resolved variables $\alpha_2$ and $\omega_1$ (the evolution of $\omega_2$ is similar, and we omit it). The results reveal that while the $p = 0$ finite-rank projection of the memory slightly underestimates the solution of the exact model, all the higher-order approximations of the memory overestimate it and in fact become unstable for long times. This means that at least for this example, increasing the timestep cannot be compensated for by a higher-order finite-rank projection for the memory.

4.2.6. The linear representation of the memory. We make one final remark about the representation of the memory. The results in Figs. 5 and 6 for $\Delta t = 5 \times 10^{-5}$, as well as those in Fig. 7 for $\Delta t = 10^{-4}$, show that the memory can be essentially captured by the linear basis functions in the finite-rank projection onto the resolved variables, i.e., the functions of order $p = 1$. The higher-order basis functions do not seem to contribute substantially to the representation of the memory. We can provide a qualitative explanation of this occurrence by more closely examining the evolution of the resolved and unresolved variables in the full system as well as the
structure of the RHS of the equations of the DeMarco-Zheng model. We will focus on the case of the memory of the resolved variable $\omega_1$. A similar analysis can be done for $\omega_2$. Finally, note that the reduced equation for $\alpha_2$ has no memory because the RHS depends only on the resolved variables $\omega_1$ and $\omega_2$.

We begin with three observations: (1) from Figs. 2(c) and 3(a), we see that the resolved variable $\alpha_2$ and the unresolved variable $\alpha_3$ evolve at very similar timescales and also at similar magnitudes (they are also in phase due to initial conditions); (2) from Fig. 3(b), we see that the unresolved variable $V_3$ does not change its value much during the time interval of simulation and remains close to 1, so its rate of change given by $R_5$ in (27) is small. Also from Fig. 3(a), we see that the unresolved variable $\alpha_3$ changes rapidly, and its rate of change given by $R_4$ in (26)
is therefore large; and (3) from the definition of the parameters in our example
and the definition of the projection operator, we see that \( F_1(u_0, 0) = QL\omega_{10} = a_1.4b_2V_1[\nu_0^3 \sin(\alpha_{30}) - V_3^4 \sin(\alpha_{30})] \).

We remind the reader that the memory kernels \( b_j^\nu(s) = (LF_j(u_0, s), h^\nu(\tilde{u}_0)) \) in (37) require the orthogonal dynamics to compute \( \tilde{F}_j(u_0, s) \), which in turn leads to the solution of Volterra equation system in (39). However, for \( s = 0 \) we can perform an analysis to gain insights about the behavior of the memory kernels \( b_j^\nu(s) \). We can acquire additional insight about \( b_j^\nu(s) \) from the quantities \( f_j^\nu(s) = (Le^sL\tilde{F}_j(u_0, 0), h^\nu(\tilde{u}_0)) \), which also appear in the Volterra equation (39) and for which we can obtain an analytical expression using the identity in (40). The quantities \( f_j^\nu(s) \) are the analogs of the memory kernels, but instead use the full dynamics and not just the orthogonal dynamics (see [3, 4] for a more extended discussion).

To proceed, we must estimate the action of \( L \) on \( F_1(u_0, 0) \). From observation (3), we have that

\[
LF_1(u_0, 0) = a_1.4b_2V_1LV_{30} \sin(\alpha_{30}) + a_1.4b_2V_1\nu_{30}L \sin(\alpha_{30}) = a_1.4b_2V_1R_5(u_0) \sin(\alpha_{30}) + a_1.4b_2V_1\nu_{30}R_4(u_0) \cos(\alpha_{30}),
\]

where \( R_5(u_0) \) and \( R_4(u_0) \) are given by (27) and (26), respectively. From observation (2), the expression \( R_5(u_0) \) is small while \( R_4(u_0) \) is large. Therefore, the expression \( a_1.4b_2V_1R_5(u_0) \sin(\alpha_{30}) \) is small compared to \( a_1.4b_2V_1\nu_{30}R_4(u_0) \cos(\alpha_{30}) \) (recall that \( V_{30} \) is close to 1). From this, we find that the expression \( LF_1(u_0, 0) \approx a_1.4b_2V_1\nu_{30}R_4(u_0) \cos(\alpha_{30}) \). Using (26) for \( R_4(u_0) \), we find

\[
LF_1(u_0, 0) \approx a_1.4b_2V_1\nu_{30}[a_4.1M_1\omega_{10} + a_4.4b_2V_1\nu_{30} \sin(\alpha_{30}) + b_3\nu_{30} \sin(\alpha_{30} - \alpha_{20}) + P_3] \cos(\alpha_{30}).
\]

From observation (1) we see that \( \alpha_3 - \alpha_2 \) is small, and thus \( \sin(\alpha_{30} - \alpha_{20}) \approx \alpha_3 - \alpha_2 \). Therefore, we have for \( LF_1(u_0, 0) \) that

\[
LF_1(u_0, 0) \approx a_1.4b_2V_1\nu_{30}[a_4.1M_1\omega_{10} + a_4.4b_2V_1\nu_{30} \sin(\alpha_{30}) + b_3\nu_{30} \sin(\alpha_{30} - \alpha_{20}) + P_3] \cos(\alpha_{30}).
\]

From (53), we see that \( LF_1(u_0, 0) \) can be reasonably approximated by a linear function of the resolved variables \( \omega_{10} \) and \( \alpha_{20} \). Finally, if we inspect Figs. 2(a) and 2(b), we see that \( \omega_1 \) and \( \omega_2 \) are in phase, of comparable magnitude, and of opposite signs. Thus, \( LF_1(u_0, 0) \) will also depend approximately linearly on \( \omega_{20} \). From these considerations, we expect that the finite-rank projection of the memory for \( \omega_1 \) on the resolved variables will essentially contain only linear basis functions.

Figs. 8(a)–8(c) corroborate our brief analysis. Fig. 8(a) shows the evolution of the memory kernel \( b_4^{(1,0,0)}(s) \), which is the projection coefficient of \( LF_1(u_0, s) \), on the degree 1 Hermite polynomial \( h^{(1,0,0)}(\tilde{u}_0) = 2\omega_{10} \). The memory kernel \( b_4^{(1,0,0)}(s) \) evolves quickly to the asymptotic value -0.0526. The memory kernel \( b_4^{(0,1,0)}(s) \) shown in Fig. 8(b) is the projection coefficient of \( LF_1(u_0, s) \) on the degree 1 Hermite polynomial \( h^{(0,1,0)}(\tilde{u}_0) = 2\omega_{20} \) and asymptotes to the value 0.0526. This is not accidental because as we have mentioned above, \( \omega_1 \) and \( \omega_2 \) are related. Finally, Fig. 8(c) shows the memory kernel \( b_4^{(0,0,1)}(s) \), which is the projection coefficient of \( LF_1(u_0, s) \) on the degree 1 Hermite polynomial \( h^{(0,0,1)}(\tilde{u}_0) = 2\alpha_{20} \) and decays quickly to the value 0.0048. Again, this is to be expected because the dependence of \( LF_1(u_0, 0) \) on \( \alpha_{20} \) is weak. This is because \( \alpha_{20} \) only enters \( LF_1(u_0, 0) \) through the difference \( \alpha_{30} - \alpha_{20} \) and \( \alpha_{30} \) has similar magnitude to \( \alpha_{20} \).
Figures 8(a)–8(c) show the evolution of the memory kernels $b_1^\mu(s)$ of the resolved variable $\omega_1$ on linear functions of the resolved variables. (a) Projection on the 1 degree Hermite polynomial $h_{(1,0,0)}$, (b) Projection on the 1 degree Hermite polynomial $h_{(0,1,0)}$, and (c) Projection on the 1 degree Hermite polynomial $h_{(0,0,1)}$. We have also included in the insets the evolution near the time origin (see text for details).

Figs. 9(a)–9(c) show the evolution of $f_1^\mu(s) = (Le^sL F_1(u_0, 0), h^\mu(\hat{u}_0))$, which are used to set up the Volterra equation (39). As we have mentioned, these quantities are the analogs of the memory kernels $b_1^\mu(s)$ but are computed using full dynamics instead of orthogonal dynamics. Comparing Figs. 8(a)–8(c) with Figs. 9(a)–9(c), we see that even though $b_1^\mu(s)$ and $f_1^\mu(s)$ are similar for small $s$ (they have to be because of their definitions), they quickly start deviating from one another. Even though the $f_1^\mu(s)$ follow decaying oscillations, the memory kernels $b_1^\mu(s)$ settle relatively fast to a small but nonzero value. This nonzero value is the reason that we need to keep track of the history of the resolved variables for so long into the distant past. We note that such a behavior is peculiar for systems with long memory where one expects the memory kernels to keep oscillating for long times (see, e.g., the analytically tractable example in Section 4.2.7). This behavior must be related to the special structure of the DeMarco-Zheng model, and we plan to investigate further in a future publication.
In summary, despite the peculiarity of the behavior of the memory kernels, our qualitative analysis shows that the adequate representation of the memory using only linear basis functions is not surprising. It will be interesting to see what the behavior of the memory kernels is for a DeMarco-Zheng model with significantly more buses or for even more complicated power grid models.

4.2.7. Exact reduced model for a linear oscillator system. Motivated by the pattern we observed for the behavior of the reduced model, we study in this section results for the reduced model of a single particle coupled linearly to a harmonic oscillator heat bath [23]. This system is much simpler than the DeMarco-Zheng model, but our motivation is twofold: (1) examine a system with very long memory where an exact reduced model for a part of it can be derived analytically and (2) show that the accuracy of the exact reduced model depends crucially on the length of the retained history in the memory term. Through this, we want to show that given
the approximations we employed to compute the reduced model and the complexity of the DeMarco-Zheng model, the results we obtained for the reduced model are rather encouraging and optimal.

The results are optimal in the sense that the same qualitative behavior exhibited for the reduced model of the DeMarco-Zheng model appears also in the case of the exact reduced model of a single particle coupled linearly to a heat bath. In particular, if we truncate the memory of the reduced model, then the prediction of the evolution of the single particle quickly loses accuracy for times longer than the memory length. In addition, the truncated memory model loses only accuracy but not stability. Finally, even the inclusion of a short memory is much better than not including any memory at all.

The particle is described by a coordinate \( x \) and its conjugate momentum \( p \). The heat bath is described by a set of coordinates \( q_j \) and their conjugate momenta \( p_j \). For simplicity, all oscillator masses are set equal to 1. The particle Hamiltonian \( H_s \) is

\[
H_s = \frac{p^2}{2m} + U(x)
\]

(54)

where \( U(x) \) is a potential. We have taken \( U(x) = \cos(2x) \). The heat bath Hamiltonian \( H_B \) is given by

\[
H_B = \sum_j \left( \frac{p_j^2}{2} + \frac{1}{2} \omega_j^2 \left( q_j - \frac{\gamma_j}{\omega_j^2} x \right)^2 \right).
\]

(55)

where \( \omega_j \) is the frequency of the \( j \)th oscillator and \( \gamma_j \) measures the strength of coupling of the particle to the \( j \)th oscillator. The equations of evolution of the system of the particle and the heat bath are given by

\[
\frac{dx}{dt} = \frac{p}{m}, \quad \frac{dp}{dt} = -U'(x) + \sum_j \gamma_j \left( q_j - \frac{\gamma_j}{\omega_j^2} x \right)
\]

(56)

\[
\frac{dq_j}{dt} = p_j, \quad \frac{dp_j}{dt} = -\omega_j^2 q_j + \gamma_j x.
\]

(57)

Assuming that \( x(t) \) is known, the evolution of each \( q_j \) can be found from (57),

\[
q_j(t) = q_j(0) \cos(\omega_j t) + p_j(0) \frac{\sin \omega_j t}{\omega_j} + \gamma_j \int_0^t ds x(s) \frac{\sin \omega_j(t-s)}{\omega_j}.
\]

(58)

Integration by parts in (58) allows us to obtain an expression that involves \( \frac{p(s)}{m} \) instead of \( x(s) \) and that will yield a closed equation for \( p(t) \). In particular,

\[
q_j(t) - \frac{\gamma_j}{\omega_j^2} x(t) = \left( q_j(0) - \frac{\gamma_j}{\omega_j^2} x(0) \right) \cos(\omega_j t) + p_j(0) \frac{\sin \omega_j t}{\omega_j}
\]

\[
-\gamma_j \int_0^t ds \frac{p(s)}{m} \cos \omega_j(t-s) \frac{\omega_j(t-s)}{\omega_j^2}.
\]

(59)

We use (59) in (56), and we find

\[
\frac{dp}{dt} = -U'(x) - \int_0^t ds K(s) \frac{p(t-s)}{m} + F_p(t),
\]

(60)
where the memory kernel $K(t)$ is given by

$$K(t) = \sum_j \frac{\gamma_j^2}{\omega_j} \cos(\omega_j t)$$  \hspace{1cm} (61)

and the noise $F_p(t)$ is given by

$$F_p(t) = \sum_j \gamma_j p_j(0) \frac{\sin \omega_j t}{\omega_j} + \sum_j \gamma_j \left( q_j(0) - \frac{\gamma_j}{\omega_j} x(0) \right) \cos(\omega_j t).$$  \hspace{1cm} (62)

Different choices of frequencies $\omega_j$ and coupling constants $\gamma_j$ lead to different behaviors of the memory kernel $K(t)$ and the noise $F_p(t)$. In our numerical simulations, the heat bath consists of five oscillators. The parameters are $\gamma_j = 1/(j/3 + 1)$ and $\omega_j = j$ for $j = 1, \ldots, 5$. Also, we chose the initial positions $q_j(0)$ and momenta $p_j(0)$ of the heat bath oscillators equal to 1. For the particle, we chose $x(0) = 0$ and $p(0) = 1$, and we also set the mass $m = 1$.

In Figures 10 and 11, we compare the evolution of the particle’s position and momentum as predicted by the full system, the exact reduced model (60), the reduced model (60) without the memory term, and reduced models with memory of various lengths $t_{\text{memory}}$. We note that when the memory length is only $t_{\text{memory}}$, the reduced model for the particle becomes

$$\frac{dp}{dt} = -U'(x) - \int_0^{t_{\text{memory}}} ds K(s) \frac{p(t-s)}{m} F_p(t).$$  \hspace{1cm} (63)

![Figure 10. Reduced model for particle coupled to heat bath - Evolution of the position $x(t)$ of the particle (a) For $t_{\text{memory}} = 1$, (b) For $t_{\text{memory}} = 2$, and (c) For $t_{\text{memory}} = 3$.](image)
It is clear from Figs. 10 and 11 that the truncation of the memory length drastically affects the accuracy of the reduced model. A reduced model that does not include a memory term loses accuracy very fast. In addition, for reduced models with finite memory $t_{\text{memory}}$, the accuracy of the reduced model predictions for both the position and the momentum degrade rapidly for time intervals larger than $t_{\text{memory}}$. These results show that even for simple systems with long memory, extending the temporal interval of accurate predictions of a reduced model is a non-trivial task. They also show that the fast increase of the error for the finite memory reduced model for the DeMarco-Zheng model for times longer than $t_{\text{memory}}$ is not due to the inadequacy of the finite-rank projection of the memory but rather to the failure to include all the necessary history of the resolved variables.

5. Discussion and future work. We have presented results from the Mori-Zwanzig formalism for the construction of reduced order models for the DeMarco-Zheng power grid model. Even though the DeMarco-Zheng model can be considered an idealization, it exhibits important qualitative features of more realistic models. In particular, there is an absence of timescale separation between generators and load buses, which can complicate the construction of an accurate reduced order model for subsets of the state variables.

Our results corroborate the expectation that in systems without timescale separation between resolved and unresolved variables, it is imperative to account for
long memory effects in order to construct an accurate reduced order model. Truncating the memory length can lead to reduced accuracy for integration times that are longer than the memory length. However, even the inclusion of a short memory in the reduced order model results in significant improvement over a memoryless reduced model. This is an important result because in power grid applications, one may be more interested in short time dynamics, e.g., for planning purposes, where one could benefit by having a reduced order model. In such scenarios, there may be the impression that short time predictions with a memoryless model can have acceptable accuracy. Our results present strong evidence that this is not the case and that memory must be included. There is a silver lining to the presented results, namely that, at least for the model investigated, truncating the length of the memory may lead to reduced accuracy but preserved stability. This is also important because stability is a major concern of reduced order models.

An interesting research direction that is relevant to power grid applications concerns allowing fluctuations for the initial conditions of the unresolved variables. As we have explained in Section 2.2, our choice of projection operator commutes with a nonlinear function exactly because it does not allow any fluctuations in the unresolved variables. This makes Eq. (6) of the MZ formalism valid pathwise. However, as we saw in Section 4.2.3, to obtain an analytical expression for the memory kernel we had to approximate our projection operator by a finite-rank projection operator. The finite-rank projection operator requires the existence of fluctuations for the resolved variables but also allows fluctuations for the unresolved variables, even though we did not utilize this in the current work. In power grid applications, if we treat the loads as unresolved variables, then we can envision scenarios where their initial conditions are allowed to fluctuate around some operational point. If we allow these fluctuations to be incorporated in the finite-rank projection operator used in the estimation of the memory, then we can obtain a representation of the memory (the usual bottleneck of reduced order models), which has built-in information about the fluctuations of the unresolved variables. This will be achieved while maintaining the pathwise validity of the equations for the reduced order model without needing a noise term as is usually required when fluctuations are allowed (see, e.g., [2]). We note that while such a construction may incorporate basic aspects of the behavior of power grid systems, it is also interesting on its own and can be applied to other systems as well.

Another interesting avenue to pursue is investigating how the properties of the memory depend on the number of unresolved variables. For example, suppose we consider a power grid network with a few generators and a large number of loads. If we construct a reduced order model for the generators while treating the loads as unresolved variables, an obvious question is whether the properties of the memory scale in some easily predictable fashion with the number of loads. This would allow us to perform expensive computations to determine the memory for a small number of loads and then use the scaling relation to acquire the memory for a larger number of loads. Such a reduction in the computational cost of constructing the reduced model can bring the concept of model reduction closer to realistic applications. The existence of such scaling would most likely need to assume a certain degree of homogeneity on the part of the loads. However, it will be interesting to see if an approximate scaling exists even for inhomogeneous loads.

Finally, we note that because the reduced models need to include a long memory in order to be accurate, we want to examine the possible computational savings
from more efficient implementations of the memory, including extended Markovian formulations (see, e.g., [1, 9]). Such implementations become more relevant for the case of power grid systems involving large numbers of buses. Moreover, such implementations are well-suited for the case where the memory can be represented only with linear functions of the resolved variables, as we have seen to be the case in our admittedly restricted example.

These ideas are under investigation, and we will report on them in a future publication.

Acknowledgments. We want to thank M. Anitescu and D. Barajas-Solano for useful discussions and comments.

REFERENCES

[1] M. Ceriotti, G. Bussi and M. Parrinello, Colored-noise thermostats ` a la carte, J. Chem. Theory Comput., 6 (2010), 1170–1180.
[2] A. J. Chorin, O. H. Hald and R. Kupferman, Optimal prediction and the Mori-Zwanzig representation of irreversible processes, Proc. Natl. Acad. Sci. USA, 97 (2000), 2968–2973.
[3] A. J. Chorin, O. H. Hald and R. Kupferman, Optimal prediction with memory, Phys. D, 166 (2002), 239–257.
[4] A. J. Chorin and P. Stinis, Problem reduction, renormalization, and memory, Commun. Appl. Math. Comput. Sci., 1 (2006), 1–27.
[5] J. H. Chow, Power System Coherency and Model Reduction, Power Electronics and Power Systems, 94, Springer, New York, 2013.
[6] I. Dobson, B. A. Carreras, V. E. Lynch and D. E. Newman, Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization, Chaos, 17 (2007).
[7] D. Givon, R. Kupferman and A. Stuart, Extracting macroscopic dynamics: Model problems and algorithms, Nonlinearity, 17 (2004), R55–R127.
[8] O. H. Hald and P. Stinis, Optimal prediction and the rate of decay for solutions of the Euler equations in two and three dimensions, Proc. Natl. Acad. Sci. USA, 104 (2007), 6527–6532.
[9] H. S. Lee, S.-H. Ahn and E. F. Darve, The multi-dimensional generalized Langevin equation for conformational motion of proteins, J. Chem. Phys., 150 (2019).
[10] J. Li and P. Stinis, A unified framework for mesh refinement in random and physical space, J. Comput. Phys., 323 (2016), 243–264.
[11] J. Li and P. Stinis, Mori-Zwanzig reduced models for uncertainty quantification, J. Comput. Dyn., 6 (2019), 39–68.
[12] A. E. Motter, S. A. Myers, M. Anghel and T. Nishikawa, Spontaneous synchrony in power-grid networks, Nature Physics, 9 (2013), 191–197.
[13] D. Osipov and K. Sun, Adaptive nonlinear model reduction for fast power system simulation, IEEE Trans. Power Syst., 33 (2018), 6746–6754.
[14] J. Qi, J. Wang, H. Liu and A. D. Dimitrovski, Nonlinear model reduction in power systems by balancing of empirical controllability and observability covariances, IEEE Trans. Power Syst., 32 (2017), 114–126.
[15] I. Simonsen, L. Buzna, K. Peters, S. Bornholdt and D. Helbing, Transient dynamics increasing network vulnerability to cascading failures, Phys. Rev. Lett., 100 (2008).
[16] F. Sloothaak, S. C. Borst and B. Zwart, Robustness of power-law behavior in cascading line failure models, Stoch. Models, 34 (2018), 45–72.
[17] P. Stinis, A phase transition approach to detecting singularities of partial differential equations, Commun. Appl. Math. Comput. Sci., 4 (2009), 217–239.
[18] P. Stinis, Renormalized Mori-Zwanzig-reduced models for systems without scale separation, Proc. A., 471 (2015), 13pp.
[19] S. Tamrakar, M. Conrath and S. Kettemann, Propagation of disturbances in AC electricity grids, Scientific Reports, 8 (2018).
[20] D. Wittnaut and M. Timme, Braess’s paradox in oscillator networks, desynchronization and power outage, New J. Phys., 14 (2012).
[21] D. Xiu and J. S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput., 27 (2005), 1118–1139.
[22] H. Zheng and C. L. DeMarco, *A bi-stable branch model for energy-based cascading failure analysis in power systems*, North American Power Symposium 2010, Arlington, TX, 2010.

[23] R. Zwanzig, *Nonequilibrium Statistical Mechanics*, Oxford University Press, New York, 2001.

Received November 2020; 1st revision August 2021; 2nd revision November 2021; early access December 2021.

*E-mail address:* jingli1023@hotmail.com
*E-mail address:* panos.stinis@pnnl.gov