Accelerating Large-Scale Non-Linear Models for Monitoring and Control using Spatial and Temporal Correlations

Robert Bos, Xavier Bombois and Paul Van den Hof

Abstract— **For non-linear state space models, model reduction alone does not decrease the time required to compute the state update. This paper suggests methods for generating models that approximate the original reduced order models by faster equivalents. Where updating reduced order models normally requires the computation of the original large-scale model, we only compute the original large-scale model for a subset of its states. The new state of the reduced order model can then no longer be computed exactly, but has to be estimated from subset of states that have been computed using the large-scale model. It is shown that the new state of the reduced order model can be estimated accurately using spatial and temporal correlations. This acceleration method can be viewed as a partial linearization of the system equations. The methods in this paper are illustrated using a simulation example of a physical system.**

I. Introduction

First principal models of complex systems often consist of a set of complex non-linear partial differential equations (PDE). The solutions of these equations are often approximated numerically by transforming the PDEs into a set of ordinary differential equations (ODE) on a fine discrete grid. These ODEs can finally be rewritten in the familiar state space form. An advantage of using state space models that have been derived in this manner is that the state vector often has a clear physical meaning.

State space models derived using first principle are generally very high order non-linear models. In order to use these models for control or monitoring purposes, it can be necessary to use non-linear control or monitoring algorithms. Using these algorithms can often be difficult for two reasons. The first problem is that the state dimension is very high. This generally makes it very difficult to do the manipulations required by a chosen algorithm. The second problem is that these techniques often require a high number of model evaluations. This can be a problem with first principle models, because computing the time update step often takes a relatively long time.

Given the mentioned problems, our goal is to find a general method for reducing both the model order and computational complexity of complex first principles models, but without losing the physical interpretation of the state vector.

Both goals can be obtained by using a coarser grid to approximate the PDEs with ODEs. A coarser grid will decrease both the state dimension and the number of computations required to evaluate the state update. A drawback of course is that increasing the average cell size will also decrease model accuracy.

Another method is to linearize the high order model. After a model reduction step, both model order and computation time will be reduced. A drawback of this method is that the linearized model will only be accurate in a certain operating region.

In this paper we will take another approach to reducing the order and computational complexity of first principe models. First by using a set of basis functions for the solution of the ODE, the model order can be reduced. In literature, different methods for choosing a set of basis functions can be identified. Examples of popular methods are Proper Orthogonal Decomposition [1] and Empirical Balancing [2]. Experience with both techniques shows that often very high order systems (order > 1000) can be accurately approximated with much lower order models (order < 25) [3][4].

After a suitable set of basis functions has been found, a lower dimensional approximation of the model can be obtained using Galerkin projection [1][5]. If the original model was linear, this procedure also reduces the number of computations required to update the system state. For non-linear models, this procedure does not reduce the number of computations required for the evaluation of the model. The reason is that in order to update a reduced-order model, we still need to compute the state update for the original large-scale model. The state update of the original model generally is the most time consuming step in updating the reduced order model. This means that potentially much could be gained if it would not be necessary to update the original model for all states, but only for a subset of its states.

If we only compute the state update for a subset of the original model's states, the new reduced state in the reduced order model can no longer be calculated exactly, but has to be estimated. This paper describes two techniques for estimating the new reduced order state from a subset of the original state. The first technique is a linear least squares estimation. The accuracy of this estimation is improved in the second technique by using spatial and temporal correlations that can be gathered in off-line simulations with the original large-scale model. The new information is used in the estimation using the Best Linear Unbiased (BLU) estimator. It will be shown that using this second technique, the estimation results can be much more accurate than the

The work of Robert Bos is supported by TNO-TPD (Dutch research organization for applied physics research).

All authors are with the Delft Center for Systems and Control (DCSC), Delft University of Technology, 2628 CD Delft, The Netherlands r.bos@dcsc.tudelft.nl.

least squares estimate.

The outline of this paper is as follows. In section 2 a brief introduction of model reduction techniques is given. Section 3 describes methods that can be used to speed up computations. Section 4 illustrates the methods of sections 2 and 3 using an example.

II. Model reduction

Suppose a physical model can be represented by the following high order state space system:

$$x_{k+1} = f(x_k, u_k) \quad (1)$$
$$y_k = h(x_k, u_k) \quad (2)$$

where x_k is a n-dimensional state vector, y_k is a vector of measurements, u_k is a vector containing the inputs to the system and f and h are non-linear functions. We will assume that the state dimension n is very high ($n > 1000$).

Since x_k is an n-dimensional vector, all possible vectors x_k can be written as a linear combination of a set of n independent basis vectors $b^{(1)}$ through $b^{(n)}$. In practice however the solutions of (1) largely remain in a lower dimensional subspace. Therefore, in good approximation it is often possible to write:

$$x_k \approx \sum_{i=1}^{m} a_k^{(i)} b^{(i)} \quad (3)$$

with $a_k^{(i)}$ a time dependant scalar value, $b^{(i)}$ a constant basis vector, $m << n$, provided that the vectors $b^{(i)}$ have been chosen appropriately. Since the basis vectors $b^{(i)}$ are constant, all information about the currently approximated state vector is contained in time varying the coefficients $a_k^{(i)}$. We can therefore introduce a new state vector as:

$$x_k^{red} = [a_k^{(1)} \; a_k^{(2)} \; \dots \; a_k^{(m)}]^T. \quad (4)$$

Define B as the matrix whose columns consist of the m chosen basis vectors $b^{(1)}$ through $b^{(m)}$. The approximated state at sample instant k can then be written as:

$$x_k \approx B x_k^{red}. \quad (5)$$

The optimal reduced state can be calculated by:

$$x_k^{red} = B^{\dagger} x_k, \quad (6)$$

where $B^{\dagger}$ is the pseudo inverse of matrix B:

$$B^{\dagger} = (B^T B)^{-1} B^T. \quad (7)$$

Using Galerkin projection, a reduced order approximate of the state update model (1) can now be constructed as:

$$x_{k+1}^{red} = B^{\dagger} f(B x_k^{red}, u_k). \quad (8)$$

Problem remains of course, how the basis vectors $b^{(i)}$ should be chosen. Two generally applicable methods for chosing B are Proper Orthogonal Decomposition (POD) [1] and Empirical Balancing [2].

III. Accelerating model computations

For non-linear models, the computation of (8) consists of three steps:

1) Expand the reduced state x_k^{red} to x_k via (5),
2) compute new state x_{k+1} via (1),
3) compute reduced state x_{k+1}^{red} from x_{k+1} via (6).

So in order to update the reduced order model, we not only need to compute the original high order model, but we also have to do two linear projections. As a result the number of calculations required to compute (8) is actually larger than the number of calculations required to update the original high order model. In this section we focus on methods that reduce the number of computations required for the reduced model (8).

Generally, by far the most time consuming step in updating (8) is the update of the original state vector via (1). Therefore much could potentially be gained if it would not be necessary to compute (1) for all of the original state elements, but only for a subset of the original state elements. For this purpose, we will create an approximate model that only requires a part of the complete full state to be calculated. Later in this section, it will be shown that the information in the new state that is lost by only using a part of the state update function, can be recovered using spatial and temporal correlations.

In order for this acceleration to be possible, we thus require that the original state can be partitioned into at least two parts for which the update can be calculated separately. Mathematically this means that we require that (1) can be rewritten as:

$$\begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} f_1(x_k, u_k) \\ f_2(x_k, u_k) \end{bmatrix}. \quad (9)$$

A second requirement is that after partitioning the original model as in (9), the time required to update $x^{(1)}$ is less than the time required to calculate the full state. Although this is a rather obvious requirement, it can limit the allowed number of elements of $x^{(1)}$.

It is obvious that if we only calculate certain elements of the original state vector, we can no longer exactly calculate (8). The new reduced state x_{k+1}^{red}, now has to be estimated using those elements of the full state vector that are calculated exactly.

In the remainder of this section two methods of estimating the reduced order state x_{k+1}^{red} will be discussed. The first method is based on a least squares collocation scheme. Then, a second method is introduced that incorporates more knowledge of the system to reduce the approximation error. In both methods we will assume that when updating (8), only the elements in $x_{k+1}^{(1)}$ will be calculated using (9). Both estimators of x_{k+1}^{red}in this section will be linear, since these allow a rapid computation of the estimate for the new reduced state.

Even though we only use linear estimators for the new reduced state, the accelerated model will remain non-linear,

since we update selected elements of the original state vector with the original non-linear model. Therefore the methods discussed in this section can also be interpreted as a partial linearization of the original system.

A. Least squares

The first method for estimating x_{k+1}^{red} that we will describe is a least squares based method. This method is based on the missing data problem in [6]. It can be shown that a property of the reduced order state x_{k+1}^{red} computed using Galerkin projection (8) is that it is the solution to the following least squares problem:

$$x_{k+1}^{red} = \arg\min_{\tilde{x}_{k+1}^{red}} \|f(Bx_k^{red}, u_k) - B\tilde{x}_{k+1}^{red}\|^2. \quad (10)$$

So the new reduced order state is that state which minimizes a least squares criterion in which all states are weighted equally. If we choose to compute only a subset of the original state, a natural method to obtain an approximation of the new reduced state vector would be to replace the least squares criterion over all elements with a criterion that only takes into to account those elements of the state which we have actually computed. If we partition the long original state vector as in (9) and we only choose to compute the elements in $x_{k+1}^{(1)}$, the new criterion will thus be:

$$\hat{x}_{k+1}^{red} = \min_{\tilde{x}_{k+1}^{red}} \|f_1(Bx_k^{red}, u_k) - B_1\tilde{x}_{k+1}^{red}\|^2, \quad (11)$$

where B_1 is a matrix consisting of the rows of B that correspond to the elements in $x^{(1)}$. The solution of this least squares problem is given by:

$$\hat{x}_{k+1}^{red} = B_1^{\dagger} f_1(Bx_k^{red}, u_k), \quad (12)$$

where $B_1^{\dagger}$ is the pseudo inverse of B_1. By doing so, our procedure to update x_k^{red} becomes:

1) Expand the reduced state: $\hat{x}_k = B\hat{x}_k^{red}$ via (5),
2) compute new partial state via $\hat{x}_{k+1}^{(1)} = f_1(\hat{x}_k, u_k)$,
3) estimate new reduced state $\hat{x}_{k+1}^{red} = B_1^{\dagger}\hat{x}_{k+1}^{(1)}$.

This new procedure involves considerably less computation time, because the computation time of step 2 (the most time consuming) and 3 are drastically decreased, since the operators $f(\cdot)$ and $B^{\dagger}$ are replaced by operators $f_1(\cdot)$ and $B^{\dagger}$ of lower order.

The estimate for x_{k+1}^{red} is the best possible linear estimate using only $x_{k+1}^{(1)}$. Experience with this technique shows that quite often it is possible to generate a reasonable approximation even if $x_{k+1}^{(1)}$ contains only a limited fraction of the total number of states.

B. Improved estimation

When we use (12) to estimate the reduced state vector x_{k+1}^{red} at time $k+1$, we lose the information on x_{k+1}^{red} that was contained in the neglected vector $x_{k+1}^{(2)}$. In this subsection we will show how a part of this lost information can be recovered without losing the advantage of a shorter computation time.

A better estimate of x_{k+1}^{red} can be computed by taking into account the correlation between $x_{k+1}^{(1)}$ and $x_{k+1}^{(2)}$, but also, as will be shown later, the correlation between x_{k+1}^{red} and the same vector one time step earlier (ie. x_k^{red}) and the correlation between x_{k+1}^{red} and u_k. The correlation between $x_{k+1}^{(1)}$ and $x_{k+1}^{(2)}$ gives us spatial information, because it relates states at the same time. The correlation between x_{k+1}^{red} and x_k^{red} and the correlation between x_{k+1}^{red} and u_k gives us temporal information, since it relates quantities at different times.

These correlation matrices are not known and have to be estimated in a separate experiment. After they have been estimated, they are assumed to be constant for all k. Note that if POD is used in the reduction step, the correlation between $x_{k+1}^{(1)}$ and $x_{k+1}^{(2)}$ has already been estimated for the reduction step.

Our method for improving the estimate x_{k+1}^{red} is based on the notion of the BLU estimator [7]. If two random variables P and Q have means μ_P and μ_Q and covariance:

$$E\begin{bmatrix} P \\ Q \end{bmatrix}[P\ Q] = \begin{bmatrix} R_{PP} & R_{PQ} \\ R_{QP} & R_{QQ} \end{bmatrix} \quad (13)$$

the best linear unbiased estimate of P given $Q = q$ is:

$$\hat{p} = \mu_P + R_{PQ}R_{QQ}^{-1}(q - \mu_Q). \quad (14)$$

The covariance matrix of the estimate is:

$$E(p - \hat{p})(p - \hat{p})^T = R_{PP} - R_{PQ}R_{QQ}^{-1}R_{QP}. \quad (15)$$

Simply applying the BLU estimator for the estimation of x_{k+1}^{red} from $x_{k+1}^{(1)}$ yields:

$$\hat{x}_{k+1}^{red} = \mu_{x_{k+1}^{red}} + R_{x_{k+1}^{red}x_{k+1}^{(1)}} R^{-1}_{x_{k+1}^{(1)}x_{k+1}^{(1)}} (x_{k+1}^{(1)} - \mu_{x_{k+1}^{(1)}}) \quad (16)$$

in which $\mu_{x_{k+1}^{red}}$ and $\mu_{x_{k+1}^{(1)}}$ are the means of x_{red}^{k+1} and $x_{k+1}^{(1)}$. Like the covariance matrices, the means have the be estimated in separate experiments, and are assumed to be constant. In the previous expression, covariance matrix between the x_{k+1}^{red} and $x_{k+1}^{(1)}$ is equal to:

$$R_{x_{k+1}^{red}x_{k+1}^{(1)}} = E(x_{k+1}^{red} - \mu_{x_{k+1}^{red}})(x_{k+1}^{(1)} - \mu_{x_{k+1}^{(1)}})^T \quad (17)$$

$$\begin{aligned} &= E\left\{ [B_1^{\dagger}\ B_2^{\dagger}] \begin{bmatrix} x_{k+1}^{(1)} - \mu_{x_{k+1}^{(1)}} \\ x_{k+1}^{(2)} - \mu_{x_{k+1}^{(2)}} \end{bmatrix} (x_{k+1}^{(1)} - \mu_{x_{k+1}^{(1)}})^T \right\} \\ &= B_1^{\dagger}R_{x_{k+1}^{(1)}x_{k+1}^{(1)}} + B_2^{\dagger}R_{x_{k+1}^{(2)}x_{k+1}^{(1)}}, \end{aligned} \quad (18)$$

where $B_1^{\dagger}$ and $B_2^{\dagger}$ consist of the columns of $B^{\dagger}$ corresponding to $x_{k+1}^{(1)}$ and $x_{k+1}^{(2)}$. Using this relation, (16) becomes:

$$\begin{aligned} &\hat{x}_{k+1}^{red} = \mu_{x_{k+1}^{red}} + \\ &\left(B_1^{\dagger} + B_2^{\dagger}R_{x_{k+1}^{(2)}x_{k+1}^{(1)}} R^{-1}_{x_{k+1}^{(1)}x_{k+1}^{(1)}}\right)(x_{k+1}^{(1)} - \mu_{x_{k+1}^{(1)}}) \end{aligned} \quad (19)$$

From this last equation it can be seen that in order to obtain the BLU estimate for x_{k+1}^{red} from $x_{k+1}^{(1)}$ we require

information about the average value of the complete state and the covariance matrix of the complete state x_{k+1}.

By applying (15) the error covariance matrix of the BLU estimate can be shown to be equal to:

$$E\left\{(\hat{x}_{k+1}^{red} - x_{k+1}^{red})(\hat{x}_{k+1}^{red} - x_{k+1}^{red})^T\right\} = B_2^{\dagger}\left[R_{x^{(2)}x^{(2)}} - R_{x^{(2)}x^{(1)}}R_{x^{(1)}x^{(1)}}^{-1}R_{x^{(1)}x^{(2)}}\right]B_2^{\dagger T}. \quad (20)$$

The total MSE can be calculated by computing the trace of this matrix. It can be used to determine how many elements of the original state vector need to be exactly calculated in order to achieve a certain estimation accuracy. It can also be used to determine which elements of the original state vector need to be computed for an optimal result.

The BLU estimate is generally more accurate than the least squares estimate. The accuracy is improved more if there is more correlation between the sets of states $x^{(1)}$ and $x^{(2)}$.

A drawback of using the BLU estimate is that the results will depend on how accurately the covariance matrix of the complete state, $R_{x_k x_k}$, is known. As mentioned earlier, this matrix can be estimated in separate off-line experiments. A problem with this covariance matrix is that it is dependent on the chosen input u_k, so in order to estimate $R_{x_k x_k}$ we already need to know which input signals are likely to be used. If the estimated covariance matrix that is used in (16) deviates from the true covariance matrix, results will be less accurate. If the required means are not equal to the true means, (16) no longer results in an unbiased estimate, and the true error will be higher than the expected error (20).

So far we have used the BLU to incorporate only spatial information, i.e. we have only used the relation between state elements at the same time instant. We can thus refine the estimate even further by also including temporal information, i.e. the relation between the current state with the previous state and inputs. Since the system is in state space form all information about the past is contained in these two quantities.

The BLU estimate $\hat{x}_{k+1}^{red}$ from $x_{k+1}^{(1)}$, x_k^{red} and u_k is:

$$\hat{x}_{k+1}^{red} = \mu_{x_{k+1}^{red}} + VW^{-1}\left(\begin{bmatrix} x_{k+1}^{(1)} - \mu_{x_{k+1}^{(1)}} \\ x_k^{red} - \mu_{x_k^{red}} \\ u_k - \mu_{u_k} \end{bmatrix}\right) \quad (21)$$

with

$$V = \begin{bmatrix} R_{x_{k+1}^{red}x_{k+1}^{(1)}} & R_{x_{k+1}^{red}x_k^{red}} & R_{x_{k+1}^{red}u_k} \end{bmatrix} \quad (22)$$

and

$$W = \begin{bmatrix} R_{x_{k+1}^{(1)}x_{k+1}^{(1)}} & R_{x_{k+1}^{(1)}x_k^{red}} & R_{x_{k+1}^{(1)}u_k} \\ R_{x_k^{red}x_{k+1}^{(1)}} & R_{x_k^{red}x_k^{red}} & R_{x_k^{red}u_k} \\ R_{u_k x_{k+1}^{(1)}} & R_{u_k x_k^{red}} & R_{u_k u_k} \end{bmatrix}. \quad (23)$$

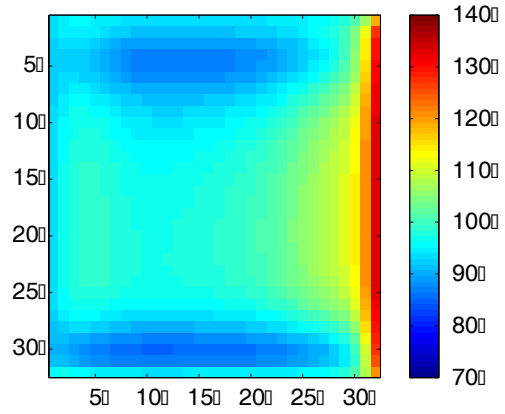

Fig. 1. Simulated heated plate example. The plate is heated or cooled along the complete edges on all sides. The physical equation are solved on a 32 by 32 grid in 5 Euler integration steps.

The expected estimation error is given by:

$$E\left\{(\hat{x}_{k+1}^{red} - x_{k+1}^{red})(\hat{x}_{k+1}^{red} - x_{k+1}^{red})^T\right\} = R_{x_{k+1}^{red}x_{k+1}^{red}} - VW^{-1}V^T. \quad (24)$$

The new BLU estimator that also includes the temporal information will in general be more accurate then the previous BLU estimate. Accuracy is especially improved if there is significant correlation between the current reduced state and the previous state and input (e.g. slowly varying systems).

Although the accuracy is again improved, more information is required that has to be estimated in advance.

So far, we have not mentioned how the partition (9) should be done. The chosen partition can seriously influence the the accuracy of the accelerated model. An optimal partition for the BLU techniques can be found by searching for that partition that minimizes the expected error, which can be computed by taking the trace of (20) or (24), depending on the chosen BLU estimator. This optimization problem quickly gets impossible to solve, because the number of possible partitions is very large. Sometimes however, it may be possible to create a good partition using physical insight in the process model.

IV. Simulation example

In this section the presented methods of the previous sections will be illustrated by a simulation example. The chosen example is that of a fictive solid square plate that is heated and cooled at the edges, see Figure 1. Each side of the solid plate is connected to a surface of which the temperature can be controlled. This means the model has four inputs.

For an infinitesimal element in the interior of the plate, the temperature of this element is given by:

$$\frac{\partial T(x,y,t)}{\partial t} = \kappa\left(\frac{\partial^2 T(x,y,t)}{\partial x^2} + \frac{\partial^2 T(x,y,t)}{\partial y^2}\right). \quad (25)$$

where $T(x, y, t)$ is the temperature of the plate at location x, y at time instant t. The κ factor is often chosen as a constant depending on material properties in a given operating region. If κ is constant the resulting model in state space form would be linear. In our example we have chosen to make κ temperature dependent. The physical interpretation is that we now create a model that can describe the behavior of the system in a larger operating region. The resulting state-space model will be non-linear.

The PDE can be translated into a state space model by imposing a grid on the plate. We have chosen to use a grid of 32 by 32 elements. In each element the temperature and all other material properties are assumed to be constant. The PDE can now be translated into an ODE consisting of 1024 coupled equations. To compute the solution of the ODE for $T = t + \Delta t$, all that is required is the solution at time t and all inputs in the interval t and $t + \Delta t$. It is therefore logical to choose the vector containing all the temperatures of the grid elements as the system's state vector. The resulting state space system therefore has order 1024.

The new state is solved on the grid by dividing each sampling interval into 5 steps, and using Euler integration for each step. Using this integration method, the new temperature of an element is dependent on up to 40 neighboring elements.

The sampling interval for this model was chosen to be 1 second. This rather short sampling interval causes the system to evolve rather slowly. This also causes high correlation between successive state vectors. The short sampling interval also causes a high correlation between adjacent grid elements.

The temperature input signals were constructed by giving each signal a mean and then adding a signal that changed every 2s with a standard deviation of 25 degrees. During the 2 seconds between temperature changes, the signals are kept constant.

The model order was reduced from 1024 to just 15 using POD model reduction. For the model reduction, the covariance of the state vector was estimated using 500 state vectors obtained in a separate simulation run. The first 15 eigenvalues of the covariance matrix accounted for 96% of the power of the system.

After the model reduction step we still want to accelerate model computations. In order the compare the different accelerated models, we need to choose an error criterium. The criterium we shall use in this paper is the one step ahead prediction criterium:

$$Err = \|x_{k+1}^{red} - \hat{x}_{k+1}^{red}\|^2 \tag{26}$$

2000 Test inputs were applied to the reduced order model. This data will serve as test data for the accelerated models.

Because of the simple structure of the model it is very easy to create partitions of the full state in the required form of (9). As mentioned at the end of the previous section, the accuracy of the accelerated models is dependent on the choice of the partition (9). The state partition we have used in the experiment was created as follows. First the order of elements in the state vectors was randomly shuffled. Afterwards the elements of $x^{(1)}$ were simply chosen as the first l states. Given the fact that this partition was randomly generated, it will likely not be optimal for any of the estimators.

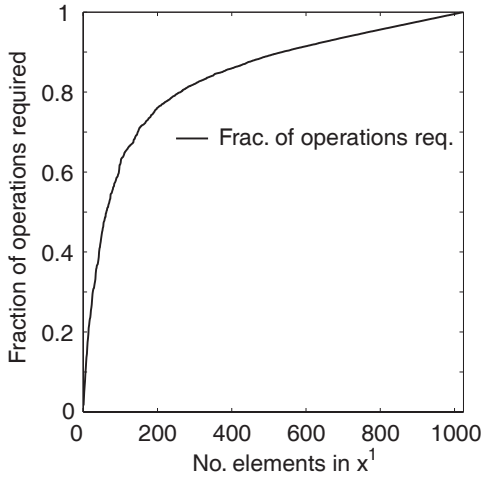

Fig. 2. Time required to compute a partial state vector as a function of the number of elements in the partial vector. Time required is normalized using the time required to update the complete system.

A requirement on the partition is that computing the partial state vector requires less time than updating the complete model. Figure 2 shows the fraction of operations required to update $x^{(1)}$ as a function of the number of elements in $x^{(1)}$. The figure shows that the fraction of operations required rises quickly for this example. In order to reduce the computation time by a factor 2, $x^{(1)}$ cannot contain more than 70 elements. It is therefore important that a chosen approximation method gives good results, especially if $x^{(1)}$ contains few elements.

Figure 3 gives the prediction error of the least squares approach (12) as a function of the number of elements in $x^{(1)}$. As can be seen, the error drops slowly as the number of elements in $x^{(1)}$ increases. This is a problem for accelerating the system using the least squares approach since it is not possible to get accurate results if $x^{(1)}$ has only a few elements. Thus using least squares approximation, we are forced to compute close to the entire state so that no significant speed up is possible.

In Figure 4 the performance of the BLU based estimators is tested. Both the BLU using only spatial information (19) as well as the BLU estimate that also uses temporal information (21) is given. Immediately clear is that both estimates perform much better than the least squares estimate. Apparently the extra information contained in the covariance between states and the state averages allows significant improvements. This was to be expected in this simulation example, because correlation between elements is very high. The error of both estimates quickly approaches zeros if more than 45 elements are calculated.

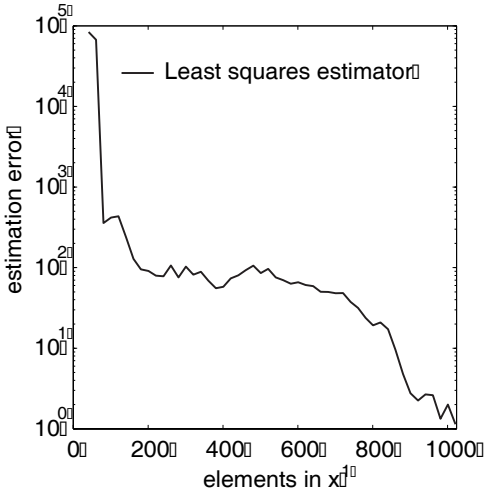

Fig. 3. Approximation error of the least squares based estimate (12) as a function of the number of elements of $x^{(1)}$

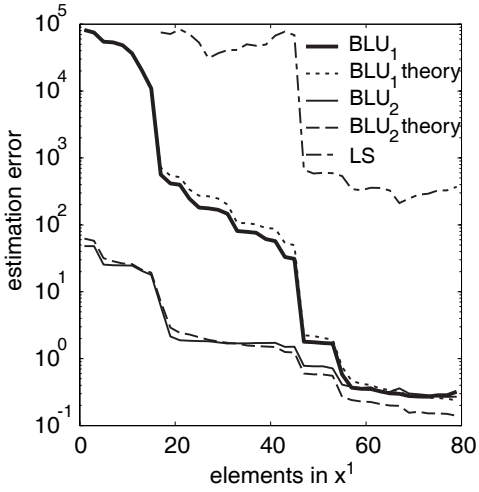

Fig. 4. Approximation error of both BLU based estimates as a function of the number of elements of $x^{(1)}$. BLU_1 is the estimated based on spatial information, corresponding to (19), BLU_2 also uses temporal information, see (21). Apart from the found one step ahead prediction errors, the predicted errors using (20) and (24) are also plotted.

Although the BLU estimate with temporal information performs better than the spatial only BLU estimate, the main difference between the two estimators is the approximation error if $x^{(1)}$ contains less than 45 elements. If $x^{(1)}$ contains less than 45 elements, the approximation error of the spatial only BLU method increases rapidly while the temporal BLU methods error remains small. As mentioned before, since the system evolves relatively slowly, there is a high correlation between successive state vectors. This correlation can be exploited if it is included for estimation.

Finally, Figure 4 also compares the found prediction errors with the theoretically calculated errors using (20) and (24). The actually found errors are well predicted by their respective equations. This shows that these equations can be used for determining how many elements $x^{(1)}$ should have to obtain a certain maximal error, or to choose that subset of state elements that minimizes the approximation error.

The results in this section are all for the one step ahead prediction error. If the accelerated model is used for longer prediction horizons, the approximation errors might accumulate. This is not a problem however if an accelerated model is used in conjunction with a state filter. Using the feedback from a state estimation filter the prediction errors will remain limited.

V. Conclusion

State space models creating using first principles modelling are often non-linear, high dimensional and require a lot of time to evaluate. The order of systems can be reduced using model reduction techniques, but this does not solve the computational complexity of the model. This paper introduces and compares methods for acceleration of model evaluations.

Using a least squares based collocation method a reasonable approximation of the state can be obtained. To improve performance, extra knowledge can be included using a BLU based estimate. The BLU estimate uses spatial and temporal correlations to improve results. If there is high spatial or temporal correlation between states, this can severely improve the accuracy of the accelerated model.

References

[1] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 1996.

[2] S. Lall, J. E. Marsden, and S. Glavaski, "A subspace approach to balanced truncation for model reduction of nonlinear control systems," *International Journal on Robust and Nonlinear Control*, vol. 12, pp. 519–535, 2002.

[3] P. Astrid, L. Huisman, S. Weiland, and A. Backx, "Reduction and predictive control design of a computational fluid dynamics model," in *Proceedings of 41st IEEE Conference in Decision and Control*, Las Vegas, 2002.

[4] H. Banks, C. del Rosario, and R. C. Smith, "Reduced-order model feedback control design: Numerical implementation in a thin shell model," *IEEE Trans. Autom. Control*, vol. 45, no. 7, pp. 1312–1324, 2000.

[5] C. A. J. Fletcher, *Computational Galerkin methods*. New York: Springer-Verlag, 1984.

[6] M. Kirby, *Geometric Data Analysis*. New York: Wiley and Sons, 2001.

[7] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, N.J.: Prentice-Hall, 1979.