AIAA-2002-5437

Performance and Sensitivity Analysis for Large Order Linear Time-Invariant Systems

O. L. de Weck,* S. A. Uebelhart,† H. L. Gutierrez‡ and D. W. Miller,§
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Performance and sensitivity analysis are important techniques during conceptual and preliminary design of complex systems. Unfortunately, as model fidelity and size increase, excessive computation times and errors due to numerical ill-conditioning will begin to emerge. These undesired effects tend to scale with the number of system states to the third power as well as with system bandwidth. This paper presents three techniques for addressing these challenges for large linear time-invariant systems. First a diagonalized Lyapunov solver is developed by exploiting state space system sparsity. Secondly apriori error bounds are presented to facilitate balanced model reduction. Thirdly the effect of numerical ill-conditioning on analytical sensitivity analysis is discussed. A governing sensitivity equation for similarity transformed state space realizations is introduced as a potential remedy. These techniques may be used individually or in concert with each other. Their usage and the pitfalls of large ill-conditioned systems are motivated by three increasingly complex models of the Space Interferometry Mission (SIM) spacecraft.

Nomenclature

A, B, C, D	=	State space matrices
E[ ]	=	Expectation operator
G	=	Transfer function (matrix)
I	=	Identity matrix
J	=	System objective, Comp. cost
L	=	Lagrange multiplier matrix
S	=	State space system or PSD
T	=	Time,Transformation matrix
d	=	White noise disturbance input
f	=	Frequency [Hz]
m	=	Block size, # of channels
n	=	Number of states
p	=	Design parameter
q	=	State variable
t	=	Time [sec]
u	=	Actuator input
w	=	Colored noise disturbance input
y	=	Sensor signal output
z	=	Performance signal output
Σ	=	Covariance matrix, Hankel matrix
μ	=	Mean
ω	=	Frequency [rad/sec]
σ	=	Root-mean-square, Hankel s.v.

*Assistant Professor, Space Systems Laboratory, Department of Aeronautics and Astronautics, Engineering Systems Division, Member AIAA, email: deweck@mit.edu

†Graduate Research Assistant, Space Systems Laboratory, Student Member AIAA.

‡Senior Engineer, Member AIAA, currently at Ball Aerospace & Technologies Corp., Boulder, Colorado 80306.

§Associate Professor, Director Space Systems Laboratory, Department of Aeronautics and Astronautics, Senior Member AIAA.

I. Introduction

TYPICALLY new systems such as aircraft and spacecraft are developed with models of increasing fidelity as their design progresses. Initially simple approximation models are derived from first principles, see Fig. 1(a). Intermediate fidelity models such as the "stick" model in Fig. 1(b) are common during the transition from conceptual to preliminary design. High fidelity models such as the one shown in Fig. 1(c) are employed during preliminary and detailed design for design refinement and optimization.

The models in Fig. 1 represent SIM, a space interferometer, which is further described in Appendix A. Such discretized models are often based on the finite element method (FEM)[1,2] and allow representation of the structural plant dynamics in the common state space form

$$\begin{aligned} \dot{q}_p &= A_p q_p + B_w w + B_u u \\ z &= C_z q_p + D_{zw} w + D_{zu} u \\ y &= C_y q_p + D_{yw} w + D_{yu} u \end{aligned} \tag{1}$$

This may be written in compact form as

$$S_p = \underbrace{\left[\begin{array}{c|cc} A_p & B_w & B_u \\ \hline C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{array}\right]}_{(n+m_z+m_y)\times(n+m_u+m_w)} \tag{2}$$

where u, w, q_p, y, are actuator inputs, disturbance inputs, plant states and sensed outputs, respectively. The most important matrix is A_p, since it contains the eigenvalue information for the system. Most often A_p is found by solving the eigenvalue problem $[K - \lambda_i M]\phi_i = 0$, where M is the global mass and K is the global stiffness

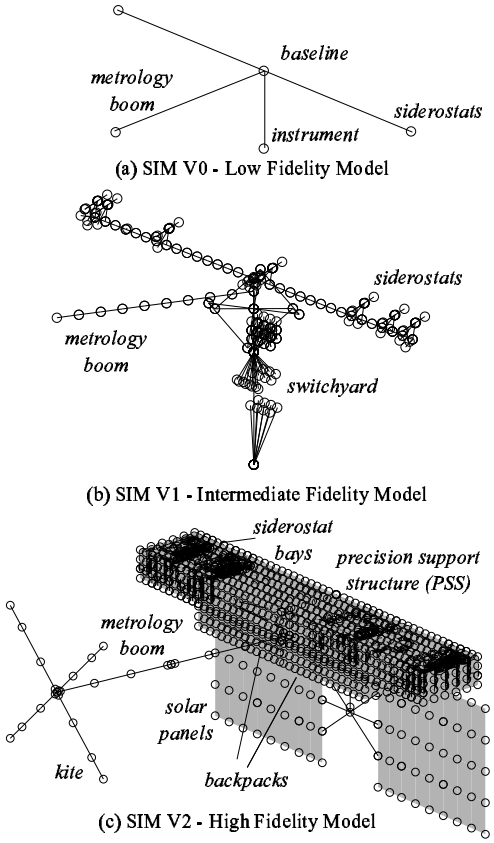

Fig. 1 Finite element models of the SIM spacecraft at three increasing levels of fidelity.

matrix of the FEM model. The modeshapes ϕ_i and natural frequencies $\omega_i = \sqrt{\lambda_i}$ are then used to obtain a state space realization in structural modal form.[3] One of the consequences of improving model fidelity is often a large increase in the size $n \times n$ of A_p. We will call this dimension, n, the "model order". This is also the number of elements in the state vector, q_p. For the purposes of this paper we will consider models with $n < 100$ as being of small order, $100 < n < 500$ as intermediate and $n > 500$ as large order. A justification for this distinction will be provided in Section IV.

To be clear, *modeling* is the process of mathematical abstraction of an existing or anticipated physical system, resulting in a state space model, S_p. *Simulation* is the process of predicting the system response $y(t)$ and/or $z(t)$ given initial conditions, $q_p(t = t_o)$, and an external (disturbance) stimulus $w(t)$. *Performance analysis* on the other hand is the process of verifying whether or not a system objective metric $J_z = f(z)$ is acceptable, i.e. whether $J_z \leq J_{z,req}$, for a given design. In this paper we will consider the root-mean-square (RMS) of the signal $z(t)$, i.e.

$$J_z = \mathrm{E}[z(t)^2]^{\frac{1}{2}} = \sigma_z \tag{3}$$

to be our performance objective. *Sensitivity analysis* is the computation of the partial derivatives of performance with respect to variable parameters, $\partial J_z / \partial p$. Sensitivity analysis is useful for identifying performance drivers and perhaps as a provider of gradient information for subsequent design optimization. The fundamental equations for performance and sensitivity analysis, given steady state disturbance inputs $w(t)$, are discussed in Sections III and VI, respectively.

Despite the fact that the governing equations for performance and sensitivity analysis remain valid for large order systems, there are two main problems that typically surface when $n > 500$. First the computational cost of solving for J_z increases proportionally to $\propto n^3$. Thus, a simulation with 1000 states will be roughly 1000 times more expensive that a simulation that uses a 100 state model. Secondly the increase in model order, n, is typically accompanied by an increase in model bandwidth $\omega_{BW} = \omega_{max} - \omega_{min}$, where ω_i is the i-th modal frequency. This leads to model ill-conditioning[1] which negatively impacts the performance and sensitivity analysis results. Ironically, in the process of trying to *improve* model fidelity one may actually inadvertently *worsen* the numerical accuracy of the results. Hence, the objective of this paper is to present techniques for dealing with larger order state space models, i.e. situations where $n > 500$, in the context of performance and sensitivity analysis.

II. Integrated Modeling

An integrated model of a multidisciplinary system incorporates all relevant disciplines that are required for meaningful performance prediction. In our case we begin with the structural (dynamics) model S_p from Eq. (2). This open loop system can also be seen in the block diagram of Fig. 2(a).

In order to represent the optical performance of the system we include an optical sensitivity matrix, C_z, which relates the translations and rotations of the vertices of optical elements[2], y, to optical performances channels, z. The ability to include linearized optical models in dynamics simulations is attributed to Redding and Breckenridge.[4] A layered control approach is usually pursued for interferometers, where coarse pointing and rigid body control is achieved by a low-bandwidth attitude control system (ACS), and fine pointing and phasing is achieved by an optical control system. These compensators are combined into the state space system denoted as, S_c. The general form of a strictly proper, linear, dynamic compensator is

$$\begin{aligned} \dot{q}_c &= A_c q_c + B_c y \\ u &= C_c q_c \end{aligned} \tag{4}$$

A detailed explanation of optical control is given by Laskin and San Martin[5] as well as Gutierrez.[3] The last

[1] As measured by the condition number $\kappa = \bar{\sigma}(A_p)/\underline{\sigma}(A_p)$ of the A_p-matrix.

[2] Optical elements include mirrors, beamsplitters, lenses and detectors among others.

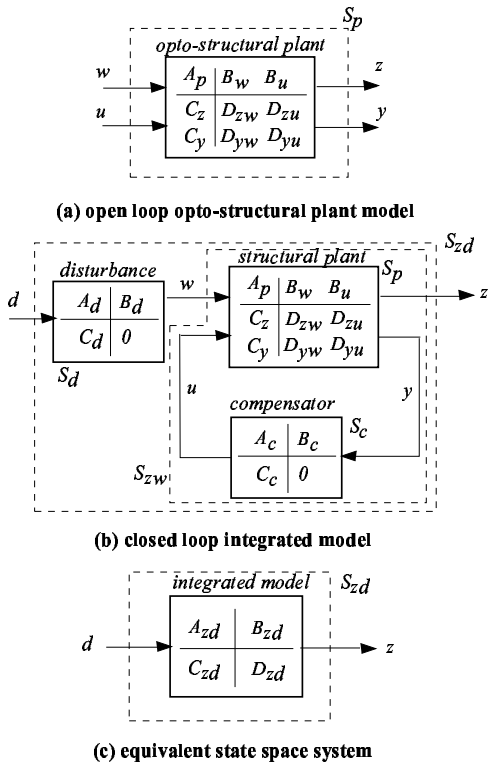

Fig. 2 Block diagrams of open loop model (a), closed loop integrated model (b) and equivalent state space system model (c).

step in the integrated modeling process is to append a state space model of the disturbance dynamics into the system. This is done via colored noise filters, S_d, as explained by Gutierrez[3] and de Weck.[6] These filters turn unit-intensity white noise, d, into a signal, w, that is shaped to contain the disturbance energy distribution of known or suspected noise sources. The disturbance included in this paper is a broadband reaction wheel noise model that injects forces F_x, F_y, F_z and torques M_x, M_y, M_z into the spacecraft. Reaction wheel disturbances have been studied extensively by Masterson,[7] Elias,[8] and Bialke[9] among others. This disturbance energy excites the flexible structure, S_p, causes motion of optical mirrors and ultimately results in an undesirable optical pathlength difference (OPD), $z(t)$. A power spectral density (PSD) of the broadband disturbance filters used in this paper is shown in Fig. 3. A derivation of the disturbance model S_d is contained in Appendix B.

This leaves us with an integrated model of the system, S_{zd}, that has a unit-intensity white noise process $d(t)$ as the input and total optical pathlength difference of the science interferometer as the performance output, i.e. $z(t) = OPD(t)$.[3] The block diagram of the integrated

[3]There are a number of other performance metrics such as internal OPD, wavefront tilt WFT, beamshear BS, and so forth. While these are all important and discussed elsewhere, they are not essential for demonstrating the problems arising from large

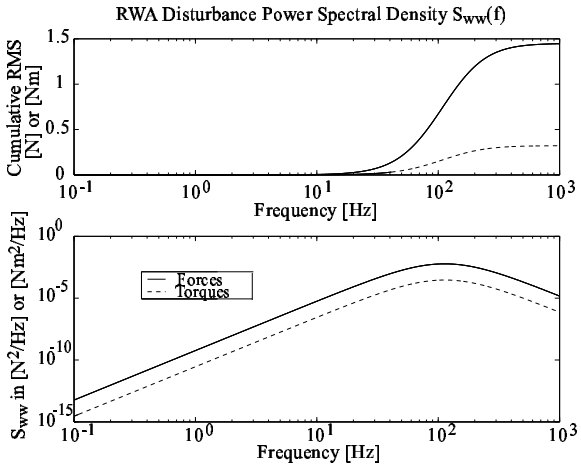

Fig. 3 Disturbance model based on Hubble Space Telescope wheel rotating in the 0-3000 [RPM] range, S_d.

system model is shown in Fig. 2(b). The closed-loop dynamics of the plant are then

$$\begin{Bmatrix} \dot{q}_p \\ \dot{q}_c \end{Bmatrix} = \begin{bmatrix} A_p & B_u C_c \\ B_c C_y & A_c + B_c D_{yu} C_c \end{bmatrix} \begin{Bmatrix} q_p \\ q_c \end{Bmatrix} + \begin{bmatrix} B_w \\ B_c D_{yw} \end{bmatrix} w$$
$$z = \begin{bmatrix} C_z & D_{zu} C_c \end{bmatrix} \begin{Bmatrix} q_p \\ q_c \end{Bmatrix} + D_{zw} w \tag{5}$$

Placing the plant in series with the disturbance filter produces the following overall state-space form.

$$\underbrace{\begin{Bmatrix} \dot{q}_d \\ \dot{q}_p \\ \dot{q}_c \end{Bmatrix}}_{\dot{q}} = \underbrace{\begin{bmatrix} A_d & 0 & 0 \\ B_w C_d & A_p & B_u C_c \\ B_c D_{yw} C_d & B_c C_y & A_c + B_c D_{yu} C_c \end{bmatrix}}_{A_{zd}} \underbrace{\begin{Bmatrix} q_d \\ q_p \\ q_c \end{Bmatrix}}_{q} + \underbrace{\begin{bmatrix} B_d \\ 0 \\ 0 \end{bmatrix}}_{B_{zd}} d \qquad z = \underbrace{\begin{bmatrix} D_{zw} C_d & C_z & D_{zu} C_c \end{bmatrix}}_{C_{zd}} \begin{Bmatrix} q_d \\ q_p \\ q_c \end{Bmatrix} \tag{6}$$

which can be written more compactly as

$$\dot{q} = A_{zd}\, q + B_{zd} d$$
$$z = C_{zd}\, q \tag{7}$$

This compact form is the equivalent state space system S_{zd}, see Fig. 2(c), and is considered the end product

order systems, since these problems are mainly due to the size and ill-conditioning of A_{zd}.

of integrated modeling. It is this system that is of concern to us for the remainder of the discussion. The vector q represents the state vector of length n and is ordered according to the convention $q = [\, q_d \; q_p \; q_c \,]^T$, where q_d are the disturbance filter states, q_p are the (structural) plant states and q_c are the compensator states. We would like to predict the value of J_z = RMS OPD $= \sqrt{\mathrm{E}[z^2]} = \sigma_z$ for a baseline design. One of the consequences of increasing model fidelity is large model order. This is summarized for the case of the SIM interferometer system in Table 1.

Table 1 Properties of integrated SIM models from Fig.1

version	ω_{BW} [Hz]	dofs	n	m_{tot} [kg]	κ
SIM V0	367	30	62	1408	$\sim 10^8$
SIM V1	7000	352	308	1962	$\sim 10^{11}$
SIM V2	1946	26946	2224	3874	$\sim 10^9$

A critical analysis reveals that the number of states, n, increases by an order of magnitude for each step increase in model fidelity. This is due in part to the increase in the number of degrees-of-freedom (dofs) of the underlying finite element model, but also due to the use of more sophisticated controllers. We observe that the model bandwidth increases from ca. 300 [Hz] to 7 [kHz] between the V0 and V1 models and that this leads to numerical ill-conditioned as expressed by κ. This is of concern to the integrated model analyst. We also notice an increase in system mass m_{tot} as more and more subsystems are included and the structural design is refined - this is of concern to the system engineer. The relative size and structure of these systems can be visualized graphically by looking at the sparsity matrices, S_{zd}, for the three systems of increasing fidelity, see Fig. 4.

While model size n increases, model sparsity clearly increases as well. This can be quantified by considering the ratio of non-zero elements (nz) to the total number of S-matrix elements, i.e. $nz/(n+1)^2$. This ratio goes from 0.14 for SIM V0, to 0.13 for SIM V1 to 0.027 for SIM V2. While model sparsity depends on the state space realization[4], it can and should be exploited when dealing with large order models.

III. Performance Analysis

Once an integrated model, S_{zd}, has been assembled performance analysis is the process of predicting the performance, $J_z = \sigma_z$ and comparing to a required level $J_{z,req}$. We consider the $\mathcal{H}_2$ performance metric according to Zhou, Doyle and Glover[10] as follows:

$$J_z = \mathrm{E}\left[z^T z\right]^{1/2} = \left(\frac{1}{T}\int_0^T z(t)^2 dt\right)^{1/2} \quad \text{RMS} \qquad (8)$$

[4]Typically a balanced realization will be fully populated.

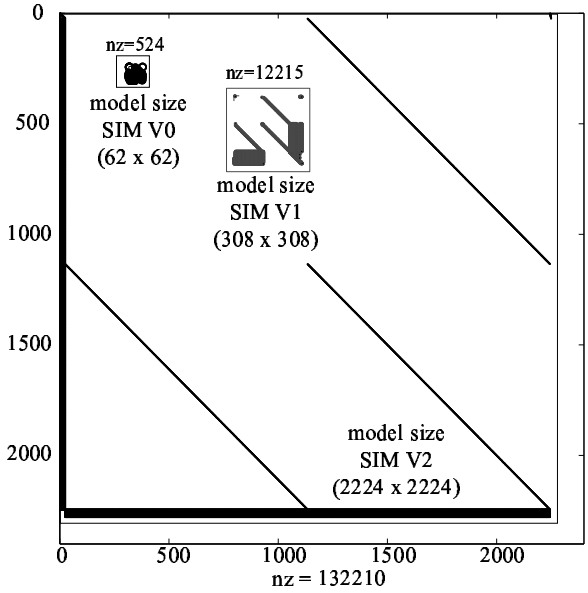

Fig. 4 Sparsity plot of integrated state space models S_{zd} for SIM V0, SIM V1 and SIM V2, respectively. Models shown in correct relative scale.

The RMS (root-mean-square) metric is typically used to describe the "on-average" optical pathlength difference (OPD) caused by dynamic disturbances in an interferometer. Three important methods for performance analysis are typically used, as discussed by Gutierrez.[3] These are time domain simulation, frequency domain analysis and Lyapunov analysis.

Time Domain Simulation

The linear time-invariant system from (colored) disturbances w to performances z is given as:

$$\begin{aligned} \dot{q} &= A_{zw} q(t) + B_{zw} w(t) \\ z(t) &= C_{zw} q(t) + D_{zw} w(t) \end{aligned} \qquad (9)$$

where q consists of structural states and compensator states. When measured time histories of the disturbances $w(t)$ exist[5], they can be used for time integration of the state space equations (9). Once the initial condition on the state vector, $q(t_o)$, is specified, numerical integration of (9) can then be performed to obtain estimates of the performance time histories $z(t)$. The solution of the state space equations is:

$$z(t) = \underbrace{C_{zd} e^{A(t-t_o)} q(t_o)}_{\text{homogenous solution}} + \underbrace{\int_{t_o}^{t} C_{zd} e^{A(t-\tau)} B_{zd} w(\tau) d\tau}_{\text{particular solution}} \qquad (10)$$

The standard difference method technique approximates the continuous first-order Eq.(9) with a difference equa-

[5]These can be obtained from reaction wheels laboratory tests.

tion such as

$$\begin{aligned} \frac{(q)_{n+1} - (q)_n}{\Delta t} &= A_{zw}(q)_n + B_{zw} w_n \\ z_n &= C_{zw}(q)_n + D_{zw} w_n \end{aligned} \tag{11}$$

The state vector $(q)_{n+1}$ at the $n+1$-th time step can be found by the forward Euler method as

$$(q)_{n+1} = [\Delta t A_{zw} + I](q)_n + \Delta t B_{zw} w_n \tag{12}$$

This integration method is simple but can diverge easily when $\Delta t \geq \Delta t_{\text{crit}}$. For time integrations in this research it was found that the Runge-Kutta solver according to Dormand and Prince[11] gave good results[6]. An advantage of the time-domain disturbance analysis is that transient effects can be observed. Generally, however, the time domain analysis is computationally more expensive than the other methods.

Time domain results $z(t)$ for SIM V0 and SIM V1 are shown in Figure 5. For V0 we obtain an RMS OPD of 43.97 [nm] in 9.3 [sec] of CPU time and for V1 we find an RMS OPD of 35.13 [nm] in 334.3 [sec][7]. These

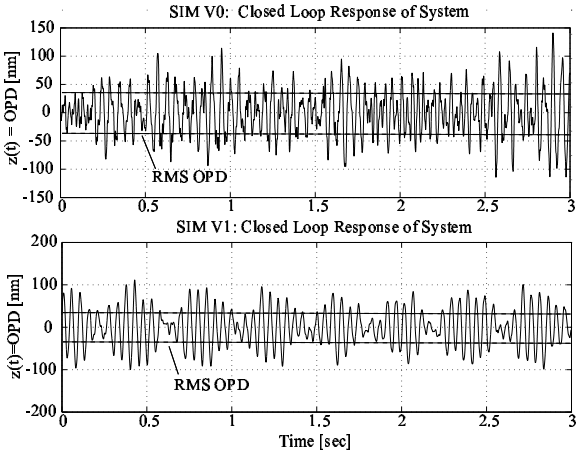

Fig. 5 Time domain simulation results $z(t)$ for SIM V0 (top) and SIM V1 (bottom). Dashed lines represent $\pm\sigma_z$. The requirement is $J_{z,req} = 10$ [nm].

results are dependent on the time horizon, T, and the time step, Δt. The horizon T should be chosen such that it captures the slowest dynamics of the system $T \sim 10 \cdot 2\pi/\omega_{min}$ while the sampling period Δt should be short enough to avoid numerical instability and aliasing. This shows that time domain simulation for large systems is difficult, particularly if the bandwidth ω_{BW} is large. It was not possible to simulate SIM V2 due to memory saturation. This is another significant problem, since typically intermediate states q_n are stored for all time steps.[8] Sou and de Weck have addressed the issue of time simulation for large order models on computers of limited random access memory.[12] This work breaks the diagonalized system, S_{zd}, into smaller subsystems according to their subsystem bandwidth and applies multi-rate sampling.

Frequency Domain Analysis

In the frequency domain (Laplace domain), the output is equal to the input multiplied by the transfer function (matrix). The system can be described by the transfer function matrix

$$G_{zw}(\omega) = C_{zw}[j\omega I - A_{zw}]^{-1} B_{zw} + D_{zw} \tag{13}$$

The disturbance spectral density matrix $S_{ww}(\omega)$ can be measured experimentally or obtained from a shaping (pre-whitening) filter as $S_{ww}(\omega) = G_d(\omega) G_d^H(\omega)$, see Fig. 3. The performance spectral density matrix S_{zz} can be obtained from[13]

$$S_{zz}(\omega) = G_{zw}(\omega) S_{ww}(\omega) G_{zw}^H(\omega) \tag{14}$$

where S_{ww} is the disturbance spectral density matrix derived in Appendix B and G_{zw} is the open or closed loop plant transfer function matrix from (13). $S_{zz}(\omega)$ provides information on the frequency content of the performances. The covariance matrix of the performances Σ_z (for zero-mean processes) is obtained as

$$\Sigma_z = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_{zz}(\omega)\, d\omega = \int_{-\infty}^{+\infty} S_{zz}(f)\, df \tag{15}$$

The variance of the i-th performance is therefore given by

$$\begin{aligned} \sigma_{z_i}^2 = [\Sigma_z]_{i,i} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} [S_{zz}(\omega)]_{i,i} d\omega = \\ \int_{-\infty}^{+\infty} [S_{zz}(f)]_{i,i} df = 2\int_0^{+\infty} [S_{zz}(f)]_{i,i} df \end{aligned} \tag{16}$$

Taking the square root of $\sigma_{z_i}^2$ produces the root-mean-square (RMS) value. One must specify whether a PSD is one or two sided and given in Hz or rad/sec.[13] In practice the upper and lower frequency integration limits are f_{min} and f_{max}, respectively.

$$\sigma_{z_i}^2 \approx 2\int_{f_{\min}}^{f_{\max}} [S_{zz}(f)]_{i,i} df \tag{17}$$

It is important to ensure that the frequency range that contributes most to the RMS value is sufficiently captured within these limits. One way to verify this is by

[6] This is implemented as ode45.m in MATLAB

[7] All computations were carried out on a personal computer with Windows XP operating system, Pentium 4 processor with 1.75 [GHz] processor speed and 512 [MB] of physical memory.

[8] A simulation with a sampling rate of 4 [kHz], i.e. $\Delta t = 1/4000$, 2000 state variables and a time horizon T of 200 [sec] would have to store a 2000 x 800000 matrix which requires memory in excess of the 1.5 [GB] memory limit which is currently practical on personal computers. MATLAB currently has a 1.5 [GB] workspace limitation, see, http://www.mathworks.com/support/technotes/1100/1106.shtml.

computing the cumulative RMS function $\sigma_{z_i,c}(f_o)$ as

$$\sigma_{z_i,c}(f_o) = \left[2\int_{f_{\min}}^{f_o} [S_{zz}(f)]_{i,i}\, df\right]^{\frac{1}{2}} \qquad (18)$$

where $f_o \in [f_{min} \ldots f_{max}]$. If most of the energy lies in this range, then $\sigma_{z_i,c}(f_{max})$ should be very close to the true value of σ_{z_i}. Generally, the frequency-domain approach is more efficient than a time-domain analysis. The method requires high frequency resolution near lightly damped modes in order to obtain correct RMS values.

Results for the three SIM models using this approach are shown in Fig. 6. The bottom plot shows $S_{zz}(f)$, whereas the top plot shows the cumulative RMS function $\sigma_{z_i,c}$. The performance can be seen as the high frequency asymptote of the cumulative RMS function. For the three models we obtain RMS values of 43.88, 27.78 and 9.73 [nm], respectively. The computation times were found to be 0.25, 17.42 and 2763.6 [sec], respectively.

As expected the computational expense increases significantly with model size. Other interesting observations are that the overall backbone of S_{zz} is globally driven by the shape of the disturbance PSD, S_{dd}, but that significant differences occur between models in terms of contributing regions and modal density. The contributing, dominant modes are identifiable by the large steps in the cumulative RMS function. The model differences are due to a combination of redesign from one stage to the next and due to the elimination of simplifying assumptions for the higher fidelity models. An example is the detailed modeling of optical elements in SIM V2. It was found that the resonances around 180 [Hz] in SIM V2 are due to local optical mount modes interacting with the underlying precision support structure (PSS). This effect is not seen in the lower fidelity models due to coarser modeling. The predicted system performance improves with each model as the structural dynamics of the plant are redesigned and more powerful, multi-stage controllers are introduced. SIM V2 is predicted to meet the pathlength performance requirement of 10 [nm]. Frequency domain analysis and transfer function computation for large order systems have been previously studied by Maghami, Kenny[14] and other researchers.

Lyapunov Analysis

The third type of performance (disturbance) analysis can be conducted if the disturbances w are modeled as the outputs of a shaping filter in the form of Eq. (94). In order to keep the disturbance w from having infinite energy, there should be no feedthrough matrix D_{wd}. If the disturbance S_d is appended into S_{zd} and this system is asymptotically stable, then the state covariance matrix, Σ_q, obeys the Lyapunov equation.[15]

$$A_{zd}\Sigma_q + \Sigma_q A_{zd}^T + B_{zd}B_{zd}^T = \dot{\Sigma}_q \qquad (19)$$

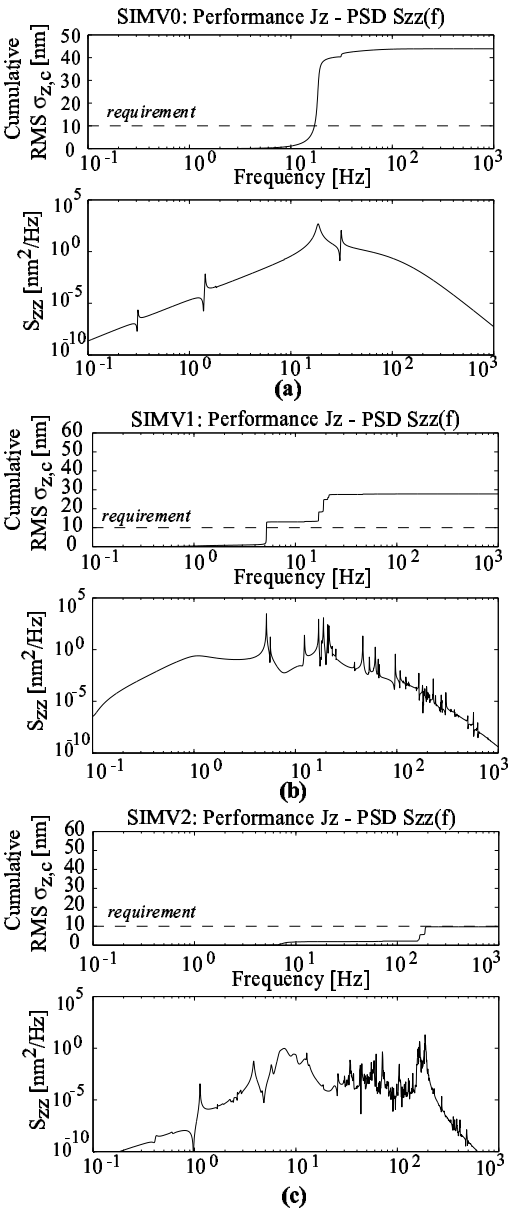

Fig. 6 Frequency domain results $S_{zz}(f)$ for three SIM models: (a) SIM V0, (b) SIM V1 and (c) SIM V2. Dashed line in cumulative RMS plot represents the performance requirement $\sigma_{z,req} = 10$ [nm]

In order to do time integration of the above dynamics, the initial state covariance, Σ_{q_o}, would have to be specified. Since the white noise disturbance processes d are assumed to be stationary, the statistics of the state vector are also stationary and $\dot{\Sigma}_q = 0$. One may then solve the steady-state Lyapunov equation of order n for the state covariance matrix Σ_q of the system (7).

$$A_{zd}\Sigma_q + \Sigma_q A_{zd}^T + B_{zd}B_{zd}^T = 0 \qquad (20)$$

For $i = 1, 2, \ldots, n_z$, where n_z is the number of performance metrics, one solves for each RMS value by extracting the i-th row from the C_{zd} matrix, pre- and post-multiplying Σ_q and by taking the square root. The

RMS of the i-th performance metric is then given as:

$$\sigma_{z_i} = \left(C_{zd,i} \Sigma_q C_{zd,i}^T \right)^{1/2} \tag{21}$$

where $C_{zd,i}$ is the vector formed by the i-th row of the C_{zd} matrix. The Lyapunov method provides a relatively direct way of arriving at the RMS (in the sense of statistical steady state) by solving one matrix equation (20) and computing a matrix triple product (21).

Solving for $J_z = \sigma_z$ using the Lyapunov approach results in predicted RMS values of 43.88, 27.52 and 9.80 [nm], respectively. The CPU time required was 0.14, 12.36 and 2547.1 seconds. The `lyap.m` solver in MATLAB was used, which requires a Schur decomposition of the A matrix.[16] Advantages of the Lyapunov method are that it generally is the fastest of the three methods and that is immune to frequency and time resolution issues. The main drawback is that it does not provide insight into the frequency content of the outputs.

Summary of Results

Table 2 summarizes the performance analysis results for the three models SIM V0, V1 and V2 for the three methods: Time domain, Frequency domain and Lyapunov analysis.

Table 2 Summary of performance analysis results.

method	result	SIM V0	SIM V1	SIM V2
Time Dom.	σ_z [nm]	43.97	35.13	NC
Freq Dom.	σ_z [nm]	43.88	27.78	9.73
Lyapunov	σ_z [nm]	43.88	27.52	9.80
Time Dom.	CPU [sec]	9.3	334.3	NC
Freq Dom.	CPU [sec]	0.3	17.4	2763.6
Lyapunov	CPU [sec]	0.125	12.36	2547.1

Some trends can be noticed. (1) The computational cost increases significantly with model size. (2) The time domain simulation poses memory challenges for large order systems. (3) The Lyapunov analysis is most time-efficient under the assumption of stationary disturbances. (4) The results match well across methods for low order models, but tend to diverge for large order models. Viewed in isolation the computational times for the Lyapunov method for model SIM V2 (ca. 42 minutes per run) do not seem unreasonable. If, however, we want to consider sensitivity analysis, isoperformance analysis or even optimization, this simulation will have to be executed hundreds and perhaps thousands of times. Now we face the dilemma that the low order models execute fast, but the results are suspect due to low model fidelity. The high fidelity models on the other hand are more accurate but significantly more expensive to run. The next section discusses two possible strategies for increasing computational efficiency, while retaining the information of the high order model.

IV. Fast Lyapunov Equation Solver

Based on the above results the Lyapunov method will be investigated further. The authors showed previously that the cost of solving Lyapunov equations such as (20) is roughly $50n^3$ floating point operations [FLOPS].[17] Efforts were undertaken to find a solution more efficiently when $n > 500$. This can be achieved in two different ways. The first way is to diagonalize the integrated state space system, S_{zd}, and to apply a fast Lyapunov solver which exploits sparseness. This drops the exponent in the Lyapunov solution cost from 3 to roughly 2. The second approach is to reduce the number of states, n, while retaining the important information in the model. The next section derives an apriori error bound for performance analysis of large order, reduced systems that relates directly to σ_z.

Computational Cost Estimate

The computational cost of solving the Lyapunov Eq. (20) is illustrated with the following numerical experiment. A random, stable state space system, S_{zd}, of size n is created and Eq. (20) is solved with computational time, J_{lyap}. The system size is increased from $n = 4$ to $n = 2048$ in factor two steps and a power law $J_{lyap} = \alpha \cdot n^\beta$ is fit to the data ($\square$) shown in Fig. 7. The computational cost of the standard Lyapunov solver (`lyap.m`) with Schur decomposition obeys the following power law.

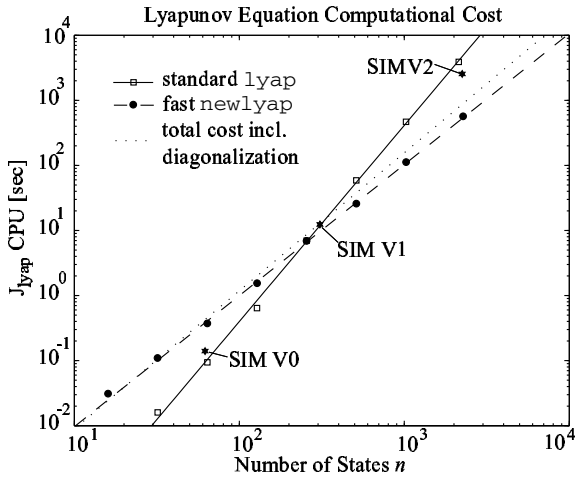

Fig. 7 Lyapunov solution cost in CPU [sec] versus number of states n. Solid line and squares shows standard algorithm `lyap.m`, dashed line and circles shows fast `newlyap.m`.

$$J_{lyap} \cong \alpha \cdot (n)^\beta \tag{22}$$

While the factor α depends on the speed and load of the particular processor used, the exponent β depends on the problem itself and the algorithm. [9] A least-squares

[9]The exact operation count and time for solving a Lyapunov equation cannot be determined apriori. This is similar to LU-

fit to the computational cost can be found as follows. The mean-square error for the computational costs J_{lyap} is expressed as the difference of the logarithms of the data and the empirical fit:

$$\varepsilon = \frac{\sum_{i=1}^{N} \left(\log(J_{lyap,i}) - \log(\alpha\,(n_i)^{\beta}) \right)^2}{N} = \frac{\sum_{i=1}^{N} \left(\log(J_{lyap,i}) - \hat{\alpha} - \beta \cdot \log n_i \right)^2}{N} \tag{23}$$

Note that N is the number of Lyapunov runs and $\hat{\alpha} = log(\alpha)$. Next, minimize ε by setting the partial derivatives with respect to the unknown coefficients to zero:

$$\frac{\partial \varepsilon}{\partial \hat{\alpha}} = \frac{-2}{N} \sum_{i=1}^{N} \left(\log\left(J_{lyap,i}\right) - \hat{\alpha} - \beta \log n_i \right) = 0 \tag{24}$$

and

$$\frac{\partial \varepsilon}{\partial \beta} = \frac{-2}{N} \sum_{i=1}^{N} \log n_i \left(\log\left(J_{lyap,i}\right) - \hat{\alpha} - \beta \log n_i \right) = 0 \tag{25}$$

Solving the above system leads to the following result for the standard Lyapunov solver (`lyap`):

$$\alpha = 3.74 \cdot 10^{-7} \qquad \beta = 3.0142 \approx 3 \tag{26}$$

We were able to confirm empirically that the operation depends on the number of states to the third power. The computational times for the three SIM models from Table 2 follow this law as depicted in Fig. 7.

Fast Solver Development

The general form of the steady-state Lyapunov Eq. (20) can be written as

$$AX + XA^T + Q = 0 \tag{27}$$

Diagonalize A using its eigenvectors[10], $A = S\Lambda S^{-1}$. Substitute this matrix product for A in Eq. (27), pre-multiply by S^{-1} and post-multiply by S^{-T}.

$$\begin{aligned} S^{-1}\left((S\Lambda S^{-1})X + X(S^{-T}\Lambda S^T) + Q \right) S^{-T} &= 0 \\ \underbrace{S^{-1}S}_{=I}\Lambda S^{-1}XS^{-T} + S^{-1}XS^{-T}\Lambda \underbrace{S^TS^{-T}}_{=I} + S^{-1}QS^{-T} &= 0 \end{aligned}$$

This can be rewritten as

$$\Lambda \underbrace{S^{-1}XS^{-T}}_{\equiv \tilde{X}} + \underbrace{S^{-1}XS^{-T}}_{\equiv \tilde{X}}\Lambda + \underbrace{S^{-1}QS^{-T}}_{\equiv \tilde{Q}} = 0 \tag{28}$$

This is a new Lyapunov equation with the matrix definitions given above. The important property is that Λ is diagonal. For a simple 2×2 case we obtain

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} \\ \tilde{x}_{21} & \tilde{x}_{22} \end{bmatrix} + \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} \\ \tilde{x}_{21} & \tilde{x}_{22} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} + \begin{bmatrix} \tilde{q}_{11} & \tilde{q}_{12} \\ \tilde{q}_{21} & \tilde{q}_{22} \end{bmatrix} = 0 \tag{29}$$

λ_i are the diagonal values of Λ, and the $\tilde{x}_{ij}$ are the scalar entries in each (i,j) element of $\tilde{X}$. We also know that $\tilde{x}_{21} = \tilde{x}_{12}$ and that $\tilde{q}_{21} = \tilde{q}_{12}$. By multiplying the matrices out, the diagonal nature of Λ succeeds in decoupling the set of equations.

$$\begin{bmatrix} \lambda_1\tilde{x}_{11} + \tilde{x}_{11}\lambda_1 + \tilde{x}_{11} = 0 & \lambda_1\tilde{x}_{12} + \tilde{x}_{12}\lambda_2 + \tilde{q}_{12} = 0 & \cdots \\ \lambda_2\tilde{x}_{21} + \tilde{x}_{21}\lambda_1 + \tilde{q}_{21} = 0 & & \\ \vdots & & \end{bmatrix}$$

Thus, solving for each $\tilde{x}_{ij}$ requires solving a simple algebraic equation $\lambda_i\tilde{x}_{ij} + \tilde{x}_{ij}\lambda_j + \tilde{q}_{ij} = 0$.

$$\tilde{x}_{ij} = -\frac{\tilde{q}_{ij}}{(\lambda_i + \lambda_j)}. \tag{30}$$

After solving for all the $\tilde{x}_{ij}$, they can be incorporated into $\tilde{X}$, which in turn can be substituted into the original matrix product $X = S\tilde{X}S^T$. This produces the Lyapunov solution X we were seeking.

A potentially faster algorithm can be constructed, which exploits the diagonal properties described above. The same idea can be applied when Λ is block diagonal instead of diagonal. Using the variable A in place of Λ, consider the general Lyapunov form described by Equation (27), but for the case where A is 2×2 matrix-block diagonal.

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} + \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \cdot \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}^T + \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} = 0 \tag{31}$$

Four matrix equations result.

$$\begin{aligned} &1)\ A_1X_{11} + X_{11}A_1^T + Q_{11} = 0 \\ &2)\ A_1X_{12} + X_{12}A_2^T + Q_{12} = 0 \\ &3)\ A_2X_{12}^T + X_{12}^TA_1^T + Q_{12}^T = 0 \\ &4)\ A_2X_{22} + X_{22}A_2^T + Q_{22} = 0 \end{aligned} \tag{32}$$

factorization which goes as $\sim (2/3)n^3$ [FLOPS], but depends on the intricacies of the Gaussian elimination algorithm, e.g. row swapping if a zero-pivot is encountered.

[10] In practice, the eigenvalue method described here may be unstable if the matrices are too ill-conditioned. Problems can also arise from repeated eigenvalues.

Notice that Eq. 1) and 4) in Eq. (32) are just new Lyapunov equations. Equations 2) and 3) are also Lyapunov equations, though in a more general form $AX + XB + C = 0$, termed the *Sylvester* equation. Because of their smaller size and the n^3 law these solutions individually take less time than the full $n \times n$ system. Also note that Eq. 3) is just the transpose of Eq. 2), so of the four equations only three must be solved.

The requirement on A is that it can be made into a block diagonal form. The decoupled modal canonical form resulting from a normal modes analysis is frequently written in such a form. If the system is no longer in a modal form, the eigenvalues of most A matrices can be written in a diagonal *Jordan* form.[10]

$$A_i = \begin{bmatrix} -\zeta_i\omega_i & \omega_i\sqrt{1-\zeta_i^2} \\ -\omega_i\sqrt{1-\zeta_i^2} & -\zeta_i\omega_i \end{bmatrix} \tag{33}$$

Using the 2×2 modal system, there are now $n/2$ blocks along the diagonal of A. Keeping in mind the symmetry of X, this means

$$N_{lyap} = \frac{\frac{n}{2}\left(\frac{n}{2}+1\right)}{2} \tag{34}$$

separate 2×2 Lyapunov solutions X_{ij} must be solved. One concern may be that although the computational time for each solution X_{ij} is fast, the total number of computations and the inefficiencies of the required *for*-loops may not yield an overall time savings. As will be seen, the time to solve all 2×2 equations is less than the time required to solve the entire $n \times n$ problem; however this is not generally the most efficient block size, m, to use.

There is no reason that larger blocks can not be selected. Using a block size of m, where m is an even factor of n, a general relation for the number of Lyapunov equations can be written.

$$N_{lyap} = \frac{\frac{n}{m}\left(\frac{n}{m}+1\right)}{2} \tag{35}$$

In order to find efficient block sizes, m, three random systems of sizes (n=128, 256 and 512) were solved using various block sizes (m=2,4,8,...). The results are shown graphically in Fig. 8 and seem to suggest that small block sizes are inefficient due to looping overhead and perhaps the fact that pipelining is not exploited. Large system sizes are inefficient due to the aforementioned n^3 scaling. It appears that block sizes in the range $20 < m < 40$ are most efficient and this conclusion holds true regardless of system size, n.

Lyapunov Efficiency Improvements

To demonstrate the efficiency that can be gained a second numerical experiment was performed using this new algorithm (`newlyap.m`). Random state space systems S_{zd} with sizes between $n = 4$ and $n = 2048$ were created, diagonalized and the state covariance matrix $X = \Sigma_q$ computed. The resulting computation cost is shown in Fig. 7 with filled circles and a dashed line. The total cost of computation including diagonalization is shown with a dotted line. A number of observations can be made.

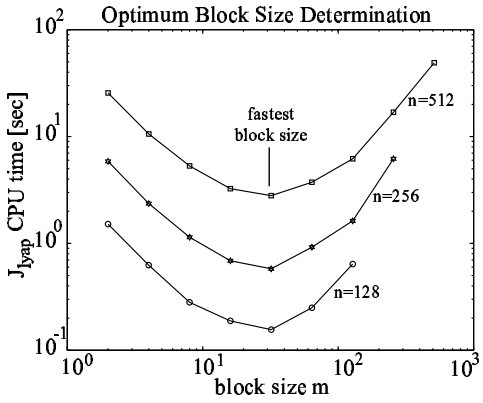

Fig. 8 Lyapunov solution cost in CPU [sec] versus different block sizes m using fast Lyapunov solver `newlyap.m`.

First, the computational cost scaling law for the fast Lyapunov solver is $J_{newlyap} = 9.38 \cdot 10^{-5} \cdot n^{2.0135}$ [sec]. Thus, the exponent has been reduced from third to second order. Even when the cost of diagonalization is included, the benefit remains: $J_{tot} = 3.73 \cdot 10^{-5} \cdot n^{2.2456}$. Thus, the computational cost is comprised of the overhead of diagonalization (essentially an eigenproblem solution) and the solution of the decoupled Lyapunov solution. Secondly, the proposed solution only improves efficiency for large order systems. The crossover is found to be around $n \sim 300$ states. This rationalizes the distinction of systems below $n < 100$ as small (diagonalization doesn't pay off), intermediate $100 < n < 500$ (both methods are comparable), and large $n > 500$ (diagonalization pays off). This crossover is expected to be fairly independent of processor speed (α).

The quality of the solution is checked by placing the answer back into the Lyapunov Eq. (27). The resultant matrix $R = A\hat{X} + \hat{X}A^T + Q$ should equal zero. Due to numerical inaccuracies, the maximum value of the resultant was actually on the order of 10^{-13} for the cases under investigation, whereby the floating point relative accuracy of the computer is on the order of 10^{-16}. What is important is that the resultant for each block solution is not worse than the resultant for the full $n \times n$ solution. It was found that the solutions suffer from no additional inaccuracies. Inaccurate solutions most frequently result when 2×2 modal blocks are inadvertently split in two, which is illegal in this algorithm. Bartels[16] suggests that the number of required operations for solving a Lyapunov equation is probably overestimated by

$$N_{FLOPS} < (5.5 + 4\sigma)n^3$$

where σ is an average number of steps in the factorization, dependent on the matrix A. It was empirically found that the number of FLOPS goes as $50n^3$, which would give a value of $\sigma = 11.125$. Using the fast Lyapunov method, a single $50n^3$ operation is replaced by many $50m^3$ operations, with $n \gg m$. If the number of Lyapunov equations is described by Equation (35), then the estimated number of FLOPS is

$$J_{newlyap} = 25(mn^2 + m^2 n) \tag{36}$$

Uebelhart[18] shows that this estimate is very accurate. The fast Lyapunov method presented here succeeds in significantly reducing the FLOPS count and computation times for large order systems. For the 2244 SIM V2 state space model, the full $n \times n$ system requires over 4×10^{11} FLOPS compared to 4.9×10^8 FLOPS when a block size of $m = 2$ is used. Applying this technique to all three SIM models yields the results in Table 3.

Table 3 Comparison of SIM results with conventional solver (`lyap` - top) and fast diagonal solver (`newlyap` - bottom).

method	result	SIM V0	SIM V1	SIM V2
`lyap`	σ_z [nm]	43.8804	27.5193	9.80
`lyap`	CPU [sec]	0.125	12.36	2547.1
blocksize	m	2	28	16
`newlyap`	σ_z [nm]	43.8804	27.5145	9.7984
resultant	$\|R\|_\infty$	$5E{-}10$	$4E{-}9$	$2E{-}19$
`diag`	CPU [sec]	0.141	1.797	927.672
`newlyap`	CPU [sec]	0.438	1.015	119.938
total	CPU [sec]	0.579	2.812	1047.61

While the `newlyap` algorithm is more expensive for small systems it significantly speeds up large system Lyapunov computations with no degradation of accuracy. The burden is mainly shifted to the diagonalization step $A = S\Lambda S^{-1}$, an area with significant ongoing research and existing fast eigenvalue solvers.

V. Model Reduction apriori Error Bound

The second strategy for reducing the burden of computing the performance, J_z, is to reduce the number of states from n to n_k, while preserving the important dynamics in the integrated model. The first step in model reduction is typically a similarity transformation, such as internal balancing. While the procedure for balanced model reduction is very well established, it is less clear what amount of reduction is appropriate for an acceptable tradeoff between speed and loss of accuracy. This section explores this tradeoff for large order systems and proposes a useful apriori error bound directly on the RMS performance, σ_z.

Internal Balancing Transformation

The internal balancing transformation $\tilde{q} = Tq$ of a state space system was first described by Moore[19] as well as Laub[20,21] and co-workers. Gregory[22] successfully applied internal balancing to the model reduction of lightly damped space structures. Essentially the internal balancing operation is a similarity transformation, which does not affect the input-output relationship of the state space system, i.e. the transfer function matrix $G_{zd}(j\omega)$ from Eq. (13). The goal of the operation is to obtain a (unique) state space realization in which the transformed controllability and observability gramians $\tilde{W}_c$ and $\tilde{W}_o$ are equal to each other, thus the term "balanced".

$$\tilde{W}_c = \tilde{W}_o \tag{37}$$

This allows ranking the states in decreasing order of disturbability/performability or controllability/observability. The diagonal elements of the transformed gramians are the *Hankel singular values* of the system.

$$\begin{gathered}\tilde{W}_c = TW_cT^T = \tilde{W}_o = \left(T^{-1}\right)^T W_o T^{-1} = \\ \Sigma^H = \operatorname{diag}\left\{\sigma_1^H, \ldots, \sigma_n^H\right\}\end{gathered} \tag{38}$$

To obtain the transformation T, we first compute the controllability and observability gramians of the original system W_c, W_o by solving the Lyapunov equations

$$\begin{aligned} A_{zd}W_c + W_cA_{zd}^T + B_{zd}B_{zd}^T &= 0 \\ A_{zd}^TW_o + W_oA_{zd} + C_{zd}^TC_{zd} &= 0 \end{aligned} \tag{39}$$

A number of different algorithms for finding T, starting with Moore's method[19] in 1981, have been suggested over the years. The second method uses a singular value decomposition (SVD) and was developed by Laub.[21] The resulting properties of T are such that it is a square and non-singular similarity transformation matrix. Once T and its inverse are known, the internal balancing operation transforms the states of the original system q into the states of the internally balanced system $\tilde{q} = Tq$. The original state vector is recovered as:

$$q = T^{-1}\tilde{q} \tag{40}$$

Substituting the above equation into the original state space system (7), noting that T is independent of time and pre-multiplying with the transformation matrix T we obtain the internally balanced state space system as

$$\begin{aligned} \dot{\tilde{q}} &= TA_{zd}T^{-1}\tilde{q} + TB_{zd}d = \tilde{A}_{zd}\tilde{q} + \tilde{B}_{zd}d \\ z &= C_{zd}T^{-1}\tilde{q} = \tilde{C}_{zd}\tilde{q} \end{aligned} \tag{41}$$

The tilde superscript denotes the balanced realization. We can use the S-matrix notation and write:

$$S_{zd} = \begin{bmatrix} A_{zd} & B_{zd} \\ C_{zd} & D_{zd} \end{bmatrix} \Rightarrow \tilde{S}_{zd} = \begin{bmatrix} TA_{zd}T^{-1} & TB_{zd} \\ C_{zd}T^{-1} & D_{zd} \end{bmatrix} \quad (42)$$

The RMS of the i-th performance metric of the internally balanced system is obtained as

$$\tilde{\sigma}_{z_i} = \left(\tilde{C}_{zd,i}\Sigma_{\tilde{q}}\tilde{C}_{zd,i}^T\right)^{1/2} = \left(C_{zd,i}T^{-1}\Sigma_{\tilde{q}}T^{-T}C_{zd,i}^T\right)^{1/2} \quad (43)$$

Since the internal balancing operation does not change the input-output relationship, but only transforms the internal states of the system, the RMS of the i-th performance of the original and the transformed system are identical in theory. In practice it is observed that the answers do not match exactly, if T is computed for a large-order nearly unobservable/uncontrollable system. Robustness issues in balanced model reduction were addressed by Mallory and Miller.[23] Nevertheless we can write:

$$\sigma_{z_i} = \tilde{\sigma}_{z_i} \quad (44)$$

By comparing the inner terms of Equation (43) with (21), we see that the following identities are true:

$$\Sigma_q = T^{-1}\Sigma_{\tilde{q}}T^{-T}, \ \Sigma_{\tilde{q}} = T\Sigma_q T^T \quad (45)$$

Here $\Sigma_{\tilde{q}}$ is the state covariance matrix of the internally balanced system and is obtained as a solution to the steady-state Lyapunov equation:

$$\tilde{A}_{zd}\Sigma_{\tilde{q}} + \Sigma_{\tilde{q}}\tilde{A}_{zd}^T + \tilde{B}_{zd}\tilde{B}_{zd}^T = 0 \quad (46)$$

This can be written as a function of the original system matrices as:

$$TA_{zd}T^{-1}\Sigma_{\tilde{q}} + \Sigma_{\tilde{q}}T^{-T}A_{zd}^T T^T + TB_{zd}B_{zd}^T T^T = 0 \quad (47)$$

At this point we recognize that the *state covariance matrix* of the balanced system $\Sigma_{\tilde{q}}$ is diagonal and contains the Hankel singular values as diagonal elements, thus $\Sigma_{\tilde{q}} = \Sigma^H$. The proof of this statement is straightforward, since for the internally balanced system the observability and controllability gramians $\tilde{W}_o$ and $\tilde{W}_c$ are equal to each other and equal to the Hankel singular value matrix [11], see,[19].[20] We write

$$\tilde{W}_o = \tilde{W}_c = \Sigma^H \quad (48)$$

Furthermore the controllability gramian of the internally balanced system obeys the same Lyapunov equation (46) as the state covariance matrix $\Sigma_{\tilde{q}}$.

$$\tilde{A}_{zd}\tilde{W}_c + \tilde{W}_c\tilde{A}_{zd}^T + \tilde{B}_{zd}\tilde{B}_{zd}^T = 0 \quad (49)$$

[11] The squares of the Hankel singular values are sometimes referred to as the "second order modes" of the system.[19]

From (49), (48) and (46) we conclude that

$$\tilde{W}_c = \Sigma_{\tilde{q}} = \Sigma^H \quad (50)$$

This is a very useful property, since it means that the Lyapunov equation for the balanced state covariance matrix does not have to be solved, since the entries (same as Σ^H) are already known. Fig. 9 shows the gramians, i.e. the Hankel singular values σ_i^H for the three models SIM V0, V1 and V2.

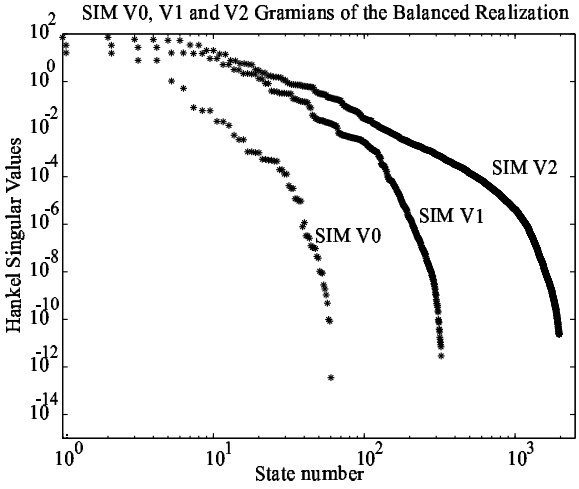

Fig. 9 Gramians - Hankel Singular Values for three models of increasing size: SIM V0, V1 and V2. Unobservable/uncontrollable states removed.

While the number of singular values is equal to n, the overall trend is similar in the sense that there are a few dominant singular values and a rapid decrease in gramian magnitude. The very low gramian values are indicative of poor numerical conditioning. The gramian plot, Fig. 9, indicates that all three systems have model reduction potential.

Model Reduction

States that are both highly disturbable (= controllable) and contribute significantly to the performance (= observable) correspond to large Hankel singular values σ_i^H. This property can be invoked for model reduction in the sense that only states with large Hankel singular values are kept in the model. Once the system has been internally balanced we can partition the state space system from Eq. (41) into states that are going to be "kept" and states that are going to be "removed" as follows:

$$\dot{\tilde{q}} = \tilde{A}_{zd}\tilde{q} + \tilde{B}_{zd}d = \begin{bmatrix} \tilde{A}_{kk} & \tilde{A}_{kr} \\ \tilde{A}_{rk} & \tilde{A}_{rr} \end{bmatrix} \begin{bmatrix} \tilde{q}_k \\ \tilde{q}_r \end{bmatrix} + \begin{bmatrix} \tilde{B}_k \\ \tilde{B}_r \end{bmatrix} d$$
$$z = \tilde{C}_{zd}\tilde{q} = \begin{bmatrix} \tilde{C}_k & \tilde{C}_r \end{bmatrix} \begin{bmatrix} \tilde{q}_k \\ \tilde{q}_r \end{bmatrix} \quad (51)$$

Here the subscript "k" indicates states that are "kept" and subscript "r" refers to states that are "removed"

which together add up to the original number of states, n.

The reduced state space system can then be written as:

$$\begin{aligned} \dot{\bar{q}} &= \bar{A}_{zd}\bar{q} + \bar{B}_{zd}d = \tilde{A}_{kk}\tilde{q}_k + \tilde{B}_k d \\ z &= \bar{C}_{zd}\bar{q} = \tilde{C}_k\tilde{q}_k \end{aligned} \tag{52}$$

Here the superscript bar refers to the reduced (= truncated and previously internally balanced) system. The reduction operation to go from the balanced system (51) to the reduced system (52) can also be represented by a multiplication with a selection matrix P, defined as:

$$P = \left[\begin{array}{cc} I_{n_k x n_k} & 0_{n_k x n_r} \end{array} \right] \tag{53}$$

Then the reduced system can be written as:

$$\begin{aligned} &\dot{\bar{q}} = \bar{A}_{zd}\bar{q} + \bar{B}_{zd}d = P\tilde{A}_{zd}P^T\bar{q} + P\tilde{B}_{zd}d = \\ &PTA_{zd}T^{-1}P^T\bar{q} + PTB_{zd}d \\ &z = \bar{C}_{zd}\bar{q} = \tilde{C}_{zd}P^T\bar{q} = C_{zd}T^{-1}P^T\bar{q} \end{aligned} \tag{54}$$

where the feedthrough matrix is $\bar{D}_{zd} = 0$. The operator P is a rectangular matrix, which selects the states to be kept. This truncation method is sometimes casually referred to as "brutal" truncation. The reason for this designation is that the effect of the removed states on the DC-gain is lost and the reduced and original system do - in general - not have matching DC-gains. Static condensation is a more sophisticated reduction method, which is not treated here, but which can achieve matching DC-gains followed by removal of the resulting D-term as shown by Gutierrez [3, p.171]. In general, however, the disturbance and sensitivity analysis results obtained for a truncated model are very good, since "DC" contributes only negligibly to the RMS values for zero-mean disturbance processes.

The effect of balancing the original system on the matrix, A_{zd}, and the subsequent truncation operation are illustrated for the SIM V1 model in Fig. 10. A significant improvement in numerical conditioning, κ, is also noted.

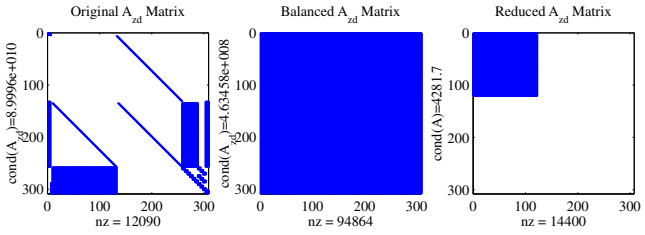

Fig. 10 Sparsity structure of (a) the original SIM V1 A-matrix (308 states, sparse), (b) balanced A-matrix (308 states, fully populated) and (c) the reduced A-matrix (120 states, fully populated).

The RMS can be computed using the reduced system matrices directly by substituting Eq. (52) into Eq. (21). The RMS performance of the i-th performance metric is computed as:

$$\bar{\sigma}_{z_i} = \left(\bar{C}_{zd,i}\Sigma_{\bar{q}}\bar{C}^T_{zd,i}\right)^{1/2} \tag{55}$$

Now we have introduced an approximation, since the matrix P selects only the n_k states that have Hankel singular values σ_i^H above a threshold value that must be defined before the reduction step. Thus, it is evident that $\bar{\sigma}_{z_i} \neq \sigma_{z_i} = \tilde{\sigma}_{z_i}$, since states have been removed and so has their contribution to the resulting RMS.

Derivation of apriori Error Bound

We are interested in finding the error in the RMS value, $\bar{\sigma}_{z_i}$, introduced by model reduction. The absolute error for the RMS is the difference between the RMS of the original (balanced) and the reduced system. The difference in the variance of the performance is:

$$\begin{aligned} &\Delta\sigma^2_{z_i} = \sigma^2_{z_i} - \bar{\sigma}^2_{z_i} = \tilde{\sigma}^2_{z_i} - \bar{\sigma}^2_{z_i} = \\ &\tilde{C}_{zd,i}\Sigma_{\tilde{q}}\tilde{C}^T_{zd,i} - \bar{C}_{zd,i}\Sigma_{\bar{q}}\bar{C}^T_{zd,i} = \\ &\tilde{C}_{zd,i}\Sigma_{\tilde{q}}\tilde{C}^T_{zd,i} - \tilde{C}_{zd,i}P^T\Sigma_{\bar{q}}P\tilde{C}^T_{zd,i} = \\ &\tilde{C}_{zd,i}\left(\Sigma_{\tilde{q}} - P^T\Sigma_{\bar{q}}P\right)\tilde{C}^T_{zd,i} \end{aligned} \tag{56}$$

The expression in round brackets is interesting enough to be analyzed in more depth. We have previously established that the balanced state covariance matrix $\Sigma_{\tilde{q}}$ is equal to the Hankel singular value matrix Σ^H. We use this fact to rewrite (56) as

$$\begin{aligned} &\Delta\sigma^2_{z_i} = \tilde{\sigma}^2_{z_i} - \bar{\sigma}^2_{z_i} = \tilde{C}_{zd,i}\left(\Sigma_{\tilde{q}} - P^T\Sigma_{\bar{q}}P\right)\tilde{C}^T_{zd,i} = \\ &\tilde{C}_{zd,i}\left(\Sigma_{\tilde{q}} - \left[\begin{array}{cc} \Sigma_{\bar{q}} & 0^{n_k x n_r} \\ 0^{n_r x n_k} & 0^{n_r x n_r} \end{array}\right]\right)\tilde{C}^T_{zd,i} = \\ &\tilde{C}_{zd,i}\left(\Sigma^H - \left[\begin{array}{cc} \Sigma_k & 0^{n_k x n_r} \\ 0^{n_r x n_k} & 0^{n_r x n_r} \end{array}\right]\right)\tilde{C}^T_{zd,i} = \\ &\tilde{C}_{zd,i}diag\left(0^{1 x n_k}, \underbrace{\sigma^H_{n_k+1}, \ldots, \sigma^H_n}_{\text{removed singular values}}\right)\tilde{C}^T_{zd,i} = \\ &\tilde{C}_{zd,i}\Sigma_R\tilde{C}^T_{zd,i} \end{aligned} \tag{57}$$

The matrix Σ_R is a diagonal matrix of the same size as the Hankel singular value matrix Σ^H, but with only the removed singular values retained and zeroes everywhere else.

$$\Sigma_R = \left[\begin{array}{cc} 0^{n_k x n_k} & 0^{n_k x n_r} \\ 0^{n_r x n_k} & \Sigma^H_{n_r x n_r} \end{array}\right] \tag{58}$$

The full RMS performance can then be written as:

$$\tilde{\sigma}_{z_i} = \left(\bar{\sigma}^2_{z_i} + \left(\tilde{\sigma}^2_{z_i} - \bar{\sigma}^2_{z_i}\right)\right)^{1/2} = \left(\bar{\sigma}^2_{z_i} + \tilde{C}_{zd,i}\Sigma_R\tilde{C}^T_{zd,i}\right)^{1/2} \tag{59}$$

This is used to obtain a closed form expression for the RMS prediction error due to balanced model reduction.

$$\varepsilon_{RMS} = \Delta\sigma_{z_i} = \tilde{\sigma}_{z_i} - \bar{\sigma}_{z_i} = \left(\bar{\sigma}_{z_i}^2 + \tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T\right)^{1/2} - \bar{\sigma}_{z_i} \tag{60}$$

The *relative error* for the RMS σ_{z_i} due to model reduction is obtained by dividing Eq. (60) by the nominal RMS value for the i-th performance metric σ_{z_i}:

$$\frac{\Delta\sigma_{z_i}}{\sigma_{z_i}} = \frac{\Delta\sigma_{z_i}}{\tilde{\sigma}_{z_i}} = 1 - \frac{\bar{\sigma}_{z_i}}{\left(\bar{\sigma}_{z_i}^2 + \tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T\right)^{1/2}} \tag{61}$$

It is noteworthy that the reduced model performance $\bar{\sigma}_{z_i}$ must first be computed before the relative RMS error according to (61) can be obtained. It would be useful to have an upper bound for the relative RMS error $\Delta\sigma_{z_i}/\sigma_{z_i}$ *before* the model reduction is actually performed. In order to develop this error bound we first consider the algebraic quantities $a, b \in \mathbb{R}$, where $a > b > 0$. If we know the difference of the squares of a and b we can write

$$a^2 - b^2 = (a+b)(a-b) \tag{62}$$

Then the following inequality is true, since $a > b > 0$:

$$2a(a-b) > \underbrace{(a+b)(a-b)}_{a^2-b^2} > 2b(a-b) \tag{63}$$

Dividing inequality (63) by $2b$ and only focusing on the two last terms of the inequality we obtain

$$\frac{a^2 - b^2}{2b} > a - b \tag{64}$$

Now we substitute for $a = \tilde{\sigma}_{z_i}$ and $b = \bar{\sigma}_{z_i}$ and knowing that the inequality $\tilde{\sigma}_{z_i} > \bar{\sigma}_{z_i} > 0$ holds, we get an error bound on the RMS for $\Delta\sigma_{z_i} = \tilde{\sigma}_{z_i} - \bar{\sigma}_{z_i}$ such that

$$\frac{\tilde{\sigma}_{z_i}^2 - \bar{\sigma}_{z_i}^2}{2\bar{\sigma}_{z_i}} > \Delta\sigma_{z_i} \tag{65}$$

Dividing by the reduced model performance RMS $\bar{\sigma}_{z_i}$ yields the desired relative error bound

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{\tilde{\sigma}_{z_i}^2 - \bar{\sigma}_{z_i}^2}{2\bar{\sigma}_{z_i}^2} \tag{66}$$

We now attempt to express the inequality in terms of the singular values of the system. Substituting (57) in the numerator and (55) in the denominator of (66) we can write

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{\tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T}{2\cdot\bar{C}_{zd,i}\Sigma_{\bar{q}}\bar{C}_{zd,i}^T} = \frac{1}{2}\frac{\tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T}{\tilde{C}_{zd,i}P^T\Sigma_{\bar{q}}P\tilde{C}_{zd,i}^T} = \frac{1}{2}\frac{\tilde{C}_{zd,i}diag\left(0^{1xn_k}, \sigma_{n_k+1}^H, \ldots, \sigma_{n_s}^H\right)\tilde{C}_{zd,i}^T}{\tilde{C}_{zd,i}diag\left(\sigma_1^H, \ldots, \sigma_{n_k}^H, 0^{1xn_r}\right)\tilde{C}_{zd,i}^T} \tag{67}$$

Taking the trace operator of both sides of (67) is allowed, since they are scalars. A tight error bound on the relative RMS error is then given by:

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{1}{2}\frac{trace\left[\tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T\right]}{trace\left[\tilde{C}_{zd,i}\Sigma_K\tilde{C}_{zd,i}^T\right]} \tag{68}$$

All the information in (68) is available before the RMS of the reduced system $\bar{\sigma}_{z_i}$ is actually computed. A more elegant, but unfortunately not absolute, bound is achieved when we simplify to[12].

$$\frac{trace\left[\tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T\right]}{trace\left[\tilde{C}_{zd,i}\Sigma_K\tilde{C}_{zd,i}^T\right]} < \frac{trace\left[\Sigma_R\right]}{trace\left[\Sigma_K\right]} \tag{69}$$

This yields a useful error bound for the relative RMS error due to balanced model reduction:

$$\boxed{\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{1}{2}\cdot\frac{\sum_{i=n_k+1}^{n}\sigma_i^H}{\sum_{i=1}^{n_k}\sigma_i^H}} \tag{70}$$

This apriori error bound is useful for determining the number of states n_k to be kept during model reduction in order to achieve a desired accuracy of the RMS prediction. The inequality is half the ratio of the sum of removed Hankel singular values over the sum of kept singular values. Gutierrez[3] for example has previously stated that *"In actuality, the model reduction process should be iterative in nature, and states should be removed until performance predictions begin to deviate by a predetermined amount."* This suggests that the model should be run several times until the correct level of reduction is found. This time consuming procedure can be avoided by applying the error bound in (70). If for example an accuracy of 1 % is desired on the RMS performance prediction, the left side of the inequality becomes 0.01. Then the expression on the right hand side can easily be evaluated for all values of n_k between 1 (all states except one truncated) and $n-1$ (only one state truncated), since the Hankel singular values σ_i^H are known[13]. The correct value for n_k is then the smallest number of states, which still meets the above inequality. Also the error bound is consistent with the balanced model reduction bound typically used in the literature, see Eq. (71) from Reference [10, p. 159]. The advantage here is that (70) applies directly to the RMS performance quantity that we care to consider.

[12]The magnitude of the entries in $\tilde{C}_{zd,i}$ are generally (but not always) decreasing in a way such that the inequality holds true.

[13]In practice we remove 2 states at a time, since 2 states are required to represent an open-loop structural mode or a complex conjugate pair of poles.

$$\left\|G(s) - \bar{G}(s)\right\|_\infty \leq 2 \sum_{i=n_k+1}^{n_s} \sigma_i^H \qquad (71)$$

To illustrate the validity of the proposed apriori error bound the three Models SIM V0, V1 and V2 are successively reduced by increasing amounts, i.e. fewer n_k are kept, and the RMS performance $\bar{\sigma}_z(n_k)$ is computed. The actual relative % error due to model reduction is then plotted against the error bound for that particular reduction level. This is shown in Fig. 11 for all three models, whereby the solid lines represent the actual relative RMS error and the dashed lines are given by the error bound. We can see that the error generally remains below the bound and that the bound is not overly conservative. The bound is then put to use in determining an appropriate reduction level such that the allowable relative RMS error is $< 0.01\%$. This leads to reduced system sizes of n_k=16, 102, and 326 for SIM V0, V1 and V2, respectively. The truncation levels are indicated by the dashed vertical line.

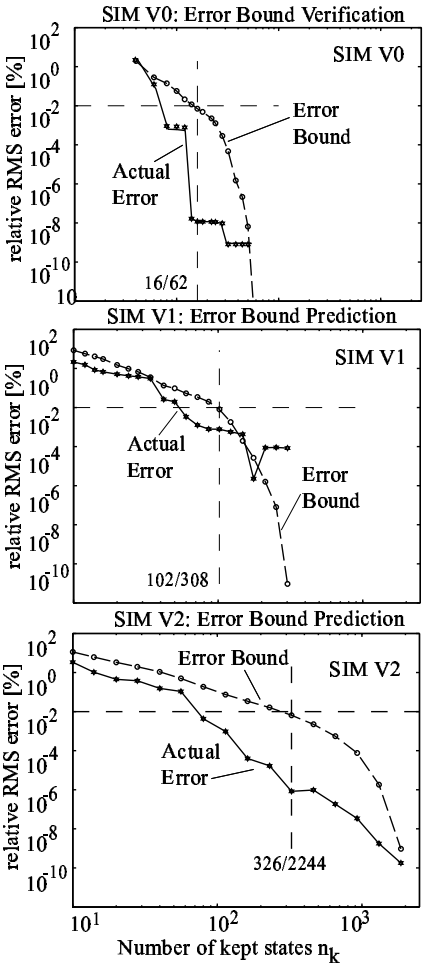

Fig. 11 Reduction level n_k versus predicted (maximum) and actual RMS errors.

Another interesting result is the explicit tradeoff between computation time (speed) and relative RMS error (accuracy). This is shown in Fig. 12. This curve will obviously shift to the left as computational speed increases in the future, but the tradeoff characteristic will remain.

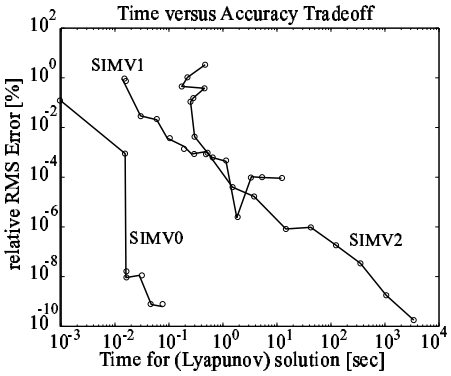

Fig. 12 Time versus RMS error tradeoff curves for SIM V0, SIM V1 and SIMV2

The accelerated Lyapunov solver and the apriori RMS error bound could be used in conjunction with each other. An example of this is the use of `newlyap` for gramian calculations. This is further discussed by Uebelhart.[18]

VI. Sensitivity Analysis

Determining the sensitivity of the performance $J_z = \sigma_z$ with respect to system parameters p, i.e. $\partial\sigma_z/\partial p$, can provide useful information. This information is used for model tuning, uncertainty analysis, isoperformance analysis and optimization among others. Gutierrez[3] proposed a Lagrangian approach for obtaining the sensitivities of LTI systems based on earlier work by Jacques.[24] Note that p can be a vector of modal or physical parameters of the system. In this paper we consider primarily the triplet $p = [\omega_i, \zeta_i, m_i]$ of modal frequency, modal damping and modal mass for a subset of system modes.

The first step, for each performance metric σ_{z_i}, $i = 1, 2, \ldots, n_z$, is to solve for the corresponding Lagrange multiplier matrix L_i. A steady state Lyapunov equation of order n has to be solved for each performance.

$$L_i A_{zd} + A_{zd}^T L_i + C_{zd,i}^T C_{zd,i} = 0 \qquad (72)$$

Next the governing sensitivity equation (GSE) is solved by substituting the results from Eq. (20) and (72). into Eq. (73). Additionally the matrix derivatives with respect to the parameters of interest p_j , $j = 1, 2, \ldots, n_p$, need to be computed. The result is the partial derivative of the variance of the i-th performance, $\sigma_{z_i}^2$, with

respect to the j-th parameter p_j.

$$\frac{\partial \sigma_{z_i}^2}{\partial p_j} = trace\left[\Sigma_q \frac{\partial\left(C_{zd,i}^T C_{zd,i}\right)}{\partial p_j}\right] + trace\left[L_i\left\{\frac{\partial A_{zd}}{\partial p_j}\Sigma_q + \Sigma_q \frac{\partial A_{zd}^T}{\partial p_j} + \frac{\partial\left(B_{zd}B_{zd}^T\right)}{\partial p_j}\right\}\right] \tag{73}$$

A detailed derivation of this equation is provided by Gutierrez[3] and Jacques.[24] The above equation gives the sensitivity of the variance $\sigma_{z_i}^2$, but usually the sensitivity with respect to the RMS is desired. The results from (21) and (73) are substituted here

$$\frac{\partial \sigma_{z_i}}{\partial p_j} = \frac{1}{2\sigma_{z_i}} \cdot \frac{\partial \sigma_{z_i}^2}{\partial p_j} \tag{74}$$

in order to obtain the desired sensitivity. Normalization with a factor $p_{j,nom}/\sigma_{z_i,nom}$ allows comparing sensitivities with respect to parameters of different units:

$$\frac{p_{j,nom}}{(\sigma_{z_i})_{nom}}\frac{\partial \sigma_{z_i}}{\partial p_j} = \frac{\dfrac{\partial \sigma_{z_i}}{(\sigma_{z_i})_{nom}}}{\dfrac{\partial p}{p_{j,nom}}} \approx \frac{\dfrac{\Delta \sigma_{z_i}}{(\sigma_{z_i})_{nom}}}{\dfrac{\Delta p}{p_{j,nom}}} \approx \frac{\%\text{ change in } \sigma_{z_i}}{\%\text{ change in } p} \tag{75}$$

The matrix partial derivatives such as $\partial A_{zd}/\partial p_j$ in (73) represent the main difficulty in finding the sensitivities for large order systems. The calculation of these sensitivities requires mode shape and frequency derivatives, which fall under the category of eigenvalue and eigenvector derivatives. A good survey of various eigenderivative methods is provided by Murthy and Haftka.[25] When the parameters are element mass and stiffness properties of a finite-element model, these derivatives can be computed exactly using methods developed by Fox and Kapoor[26] and Nelson.[27] Practical implementation of these methods is done by Kenny,[28] and this work is extended by Gutierrez.[3] Previous work by Hou and Koganti[29] in the context of integrated controls-structure design also makes extensive use of sensitivities. While these techniques work well for reasonably sized, well conditioned systems, there are a number of difficulties to address for large order ill-conditioned systems.

Transformed Governing Sensitivity Equation

The assumed structure of the A_{zd} matrix is shown in Equation (6). Any kind of similarity transformation, $\tilde{q} = Tq$, removes the explicit dependence of the state space matrices on the parameters of interest. This is the case both for the diagonalization used in Section IV and the balanced reduction in Section V. In that case the matrix partial derivatives $\partial A_{zd}/\partial p_j$, $\partial B_{zd}/\partial p_j$, $\partial C_{zd}/\partial p_j$ needed for determining the sensitivity of the root-mean-square (RMS) of the i-th performance metric $J_{z,i}$, with respect to the j-th parameter p_j , i.e. $\partial J_{z,i}/\partial p_j = \partial \sigma_{z_i}/\partial p_j$ cannot be easily computed.

To overcome this hurdle a *transformed governing sensitivity equation* (TGSE) was derived which allows sensitivity computations for similarity transformed systems - while using the matrix derivatives $\partial A_{zd}/\partial p_j$, $\partial B_{zd}/\partial p_j$, $\partial C_{zd}/\partial p_j$ in the non-transformed realization. This was prompted by Gutierrez's[3] call for future work:

> The numerical conditioning problem discovered during the finite-difference computations in the SIM Classic example needs to be resolved. Balancing and model reduction techniques might alleviate this problem; however, the sensitivity framework currently does not support alternative system realizations. Efforts to accommodate other state-space realizations should be made.

. The governing sensitivity equation for a similarity transformed system was determined to be:

$$\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} = trace\left[T^{-1}\Sigma_{\tilde{q}}\left(T^{-1}\right)^T \frac{\partial\left(C_{zd,i}^T C_{zd,i}\right)}{\partial p_j}\right] + trace\left[\tilde{L}_i\left\{T\frac{\partial A_{zd}}{\partial p_j}T^{-1}\Sigma_{\tilde{q}} + \Sigma_{\tilde{q}}\left(T^{-1}\right)^T \frac{\partial A_{zd}^T}{\partial p_j}T^T\right\}\right] + trace\left[\tilde{L}_i\left\{T\frac{\partial\left(B_{zd}B_{zd}^T\right)}{\partial p_j}T^T\right\}\right] \tag{76}$$

The TGSE (76) allows computing the partial derivative of the variance of the i-th performance $J_{z,i} = \sigma_{z,i}$ with respect to the j-th parameter p_j using the transformed quantities, including the Lagrange multiplier matrices for the transformed system. The matrix derivatives in (76) may still be computed using the original (non-transformed) system matrices, where the parameters appear in known locations. At first the simplicity of the governing sensitivity equation (76) is surprising, since we expect to find derivative terms of the transformation matrix T in this equation. It turns out, however, that all $\partial T/\partial p$ terms can be made to disappear. A short derivation of the TGSE is contained in Appendix C, while a longer derivation is shown elsewhere.[17]

The derivation of the sensitivity for the reduced, balanced system is straightforward with Equation (76) as a starting point. This is due to the fact that the selection matrix P is not a function of any system parameters p_j, since it contains only ones and zeros. The sensitivity of a reduced, internally balanced state space system is given as:

$$\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j} = \frac{1}{2\bar{\sigma}_{z_i}} \cdot \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} \tag{77}$$

Here the partial derivative with respect to the variance is obtained from the transformed governing sensitivity equation (TGSE) with the reduced system matrices

from (52) substituted in.

$$\frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} = trace\left[\Sigma_{\bar{q}} \frac{\partial\left(\bar{C}_{zd,i}^T \bar{C}_{zd,i}\right)}{\partial p_j}\right] + trace\left[\bar{L}_i\left\{\frac{\partial \bar{A}_{zd}}{\partial p_j}\Sigma_{\bar{q}} + \Sigma_{\bar{q}}\frac{\partial \bar{A}_{zd}^T}{\partial p_j} + \frac{\partial\left(\bar{B}_{zd}\bar{B}_{zd}^T\right)}{\partial p_j}\right\}\right] \tag{78}$$

After taking into account (76) and some matrix algebra we obtain

$$\frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} = trace[T^{-1}P^T\Sigma_{\bar{q}}P\left(T^{-1}\right)^T \frac{\partial\left(C_{zd,i}^T C_{zd,i}\right)}{\partial p_j}] + trace\bar{L}_i\{PT\frac{\partial A_{zd}}{\partial p_j}T^{-1}P^T\Sigma_{\bar{q}} + \Sigma_{\bar{q}}P\left(T^{-1}\right)^T\frac{\partial A_{zd}^T}{\partial p_j}T^TP^T + PT\frac{\partial\left(B_{zd}B_{zd}^T\right)}{\partial p_j}T^TP^T\}]$$

Eq. (79) shows that the sensitivity of a reduced, internally balanced system can be calculated, if the matrix partial derivatives $\partial A_{zd}/\partial p_j$, $\partial B_{zd}/\partial p_j$, $\partial C_{zd}/\partial p_j$ of the original system, the balancing transformation matrix T and the selection matrix P are known. This is an important result, since it was previously not possible to compute this quantity due to the fact that the balancing transformation and model reduction removes the explicit dependency of the parameters p_j in the system matrices $\bar{A}_{zd}$, $\bar{B}_{zd}$ and $\bar{C}_{zd}$, as mentioned by Gutierrez [3, p.176].

The Lagrange multiplier matrix for the reduced system obeys:

$$\bar{A}_{zd}^T\bar{L}_i + \bar{L}_i\bar{A}_{zd} + \bar{C}_{zd,i}^T\bar{C}_{zd,i} = 0 \tag{80}$$

In the case where there is only one performance metric $i = 1$, i.e. $\bar{C}_{zd} \in \mathbb{R}^{1xn_k}$, the Lagrange multiplier matrix $\bar{L}_i$, is equal to the diagonal matrix of Hankel singular values $\Sigma_{n_k xn_k}^H$ that are kept in the system, $\bar{L}_i = \Sigma_{n_k xn_k}^H$, where

$$\Sigma_{n_k xn_k}^H = \begin{bmatrix} \sigma_1^H & & \\ & \ddots & \\ & & \sigma_{n_k}^H \end{bmatrix} \tag{81}$$

Consider that in this particular case the solution to the Lyapunov equation (80) is already known, since there is only one performance metric. In this instance it is actually computationally cheaper to compute the sensitivity for a balanced system (reduced or unreduced) than for an unbalanced system, provided the transformation matrix T is already known. A computational benefit also exists in the general case (of several performances), since we can now solve for the Lagrange multiplier matrices for the reduced system according to (80). One can establish an inequality[17] such that $J_r < J_o$:

$$\left(\frac{n_k}{n}\right)^3 < \frac{n_z - 2}{n_z} \tag{82}$$

The above inequality is found for the crossover point, i.e. when J_r (balanced-reduced system cost) $< J_o$ (original system cost). The above inequality is useful since it can provide the required state reduction ratio n_k/n_s that will lead to a time savings, given the number of performances n_z. For $n_z = 4$ a reduction of slightly more than 21 % of the number of states of the original system is sufficient to speed up computations. This percentage becomes lower as more performance metrics of interest are present in the system. The SIM V1 model, originally had 15 performances of interest. The reason for the high "upfront" cost is that the system balancing operation requires the solution of two Lyapunov equations for the gramians, two Cholesky factorizations of the gramians, a singular value decomposition (SVD) of the product of the Cholesky factors[14] and matrix multiplications to obtain T , T^{-1} and the balanced system matrices. This results in an approximate cost of $150n^3$ [FLOPS].

Sensitivity apriori Error Bound

As was the case for the computation of the reduced system RMS performance $\bar{\sigma}_{z_i}$, we can expect that the reduced system sensitivity $\partial\bar{\sigma}_{z_i}/\partial p_j$ will be in error due to the removal of system states. A sensitivity error bound similar to the error bound for the performance was found:

$$\left|\frac{\Delta\frac{\partial\sigma_{z_i}}{\partial p_j}}{\frac{\partial\bar{\sigma}_{z_i}}{\partial p_j}}\right| < \frac{\sum_{i=1}^{n}\left(\sigma_i^H\right)^2}{\sum_{i=1}^{n_k}\left(\sigma_i^H\right)^2} - 1 + \frac{1}{2}\cdot\frac{\sum_{i=n_k+1}^{n_s}\sigma_i^H}{\sum_{i=1}^{n_k}\sigma_i^H} \tag{83}$$

The ratio involving the squared singular values on the right side is generally very close to 1, since the Hankel singular values that are kept in the system are often larger by several orders of magnitude than the singular values that have been removed. In practice it is then observed that the relative error bound on the sensitivity (83) is nearly identical to the relative error bound on the RMS itself according to (70). This bound involves the same ratio of Hankel singular values as for the relative RMS error in addition to a ratio of the Hankel singular values squared. It shall be noted that in this case the error bound is on the absolute value of the relative sensitivity error, since the sensitivity $\partial\bar{\sigma}_{z_i}/\partial p_j$ could be positive or negative.

Application to SIM V1

In order to demonstrate the usefulness of the sensitivity analysis results, a modal sensitivity analysis was conducted. The results for the original SIM V1 (308 states) are shown in Figure 13.

The normalized modal parameter sensitivity results for 10 modes of the system can be compared between the

[14] According to Laub's method[21]

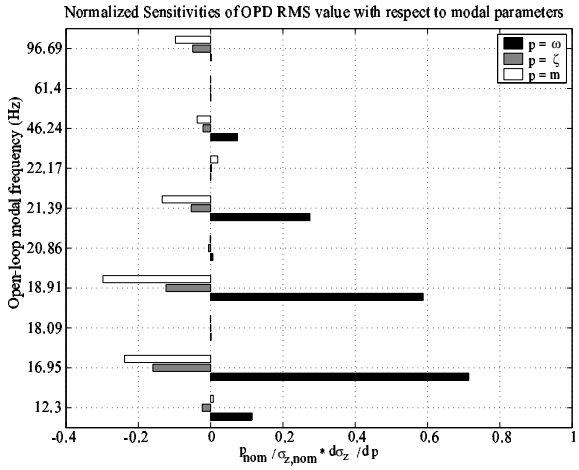

Fig. 13 Normalized SIM V1 modal parameter sensitivities for 10 modes and the full-order system (308 states, 78.2 seconds CPU time)

original system (308 states) and the reduced system (110 states) of SIM V1 in Fig. 13 and Fig. 14, respectively.

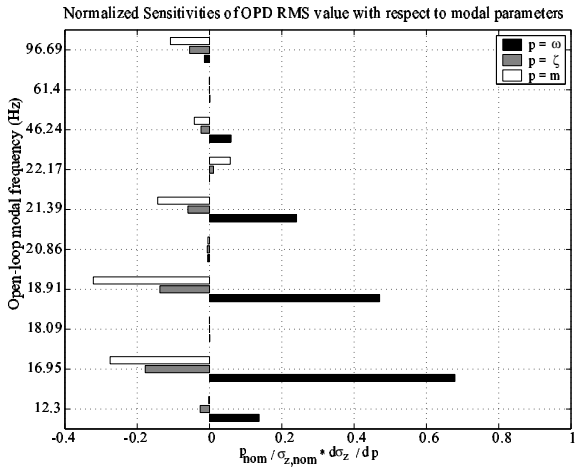

Fig. 14 Modal sensitivity results for balanced and reduced SIM V1 model (110 states, 3.3 seconds CPU time)

The results are in good agreement with the exception of some of the less dominant modes. It is noticeable that the agreement is better for "critical" modes of the system compared to "non-critical" modes (i.e. modes that do not significantly contribute to the performance RMS).

Numerical Conditioning Problems

So far we have mainly addressed system order n and computational cost as a problem. However, significant and persistent numerical problems were initially encountered, when trying to produce and verify the results in Fig. 13. The reason has to do with the numerical conditioning of the system. The results in Fig. 14 were obtained by successfully applying Mallory's pre-balancing method for model reduction.[30] The key to the pre-balancing method is that the system is brought into 2x2 block-diagonal form and each of these 2x2 blocks is balanced individually. Then nearly uncontrollable states are removed based on a user-defined tolerance before the overall T-matrix is computed.

Another noteworthy result is that finite difference validation of the modal sensitivities became possible with the balanced reduced system. Gutierrez has previously reported the inability of the finite difference approximation to validate the analytical sensitivities for large order ill-conditioned systems [3, p.124] . This ill-conditioning caused problem is removed with the method presented in this paper. The resulting finite differences for the balanced reduced system closely match the analytical answers as shown by the bar chart in Fig. 15. The figure shows the logarithm of the relative percent difference between the analytical sensitivities and the finite difference approximations of the balanced, reduced system. This quantity was computed as $\log\{100 \cdot (\partial\sigma_z/\partial p - \Delta\sigma_z/\Delta p) / (\partial\sigma_z/\partial p)\}$, where the term with the partials comes from the analytical sensitivity calculations and the Δ-term comes from the finite difference approximation. Thus, a value of one for a horizontal bar indicates a 10% difference between the analytical sensitivity and the finite difference approximation.

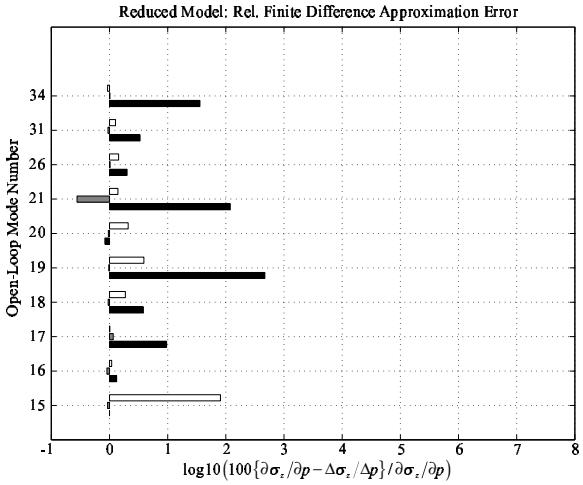

Fig. 15 Good approximation with finite differences (1%) in the reduced model case. Mode numbers correspond to modes in Figure 13

An interesting observation was made when the RMS for SIM V1 (308 states) was initially computed for the original and the balanced (unreduced) system and the answers were compared. Even though the RMS resulting from the Lyapunov approach should be identical for both cases (similarity transformation), the answers differed by roughly 10% even though the system was merely balanced and no states were removed. The cause of this disturbing result was linked to the presence of unobservable/uncontrollable states in the original system. The erroneous results for the modal parameter sensi-

tivities in this case are shown in Fig. 16. The correct answers are shown in Fig. 13 and Fig. 14. The erroneous answers in Fig. 16 are produced without any warning, which suggests that numerical conditioning checks and safeguards as well as engineering judgement are indispensable for large order systems analysis.

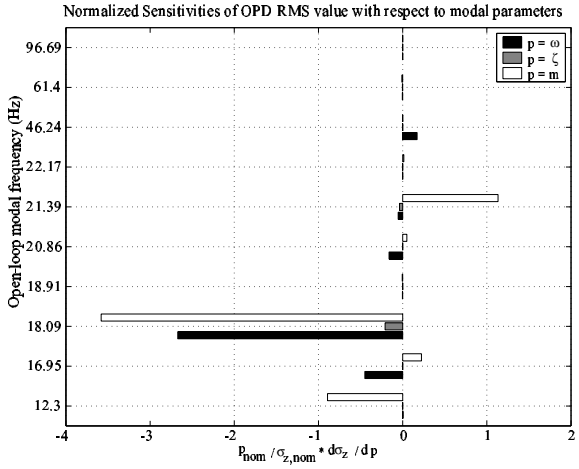

Fig. 16 Erroneous Sensitivity analysis results for balanced, but unreduced SIM V1 model, $\tilde{S}_{zd}$, due to ill-conditioned T matrix.

The numerical reason for the erroneous result is that very small (on the order of $\texttt{eps} = 2.2204 \cdot 10^{-16}$) Hankel singular values are associated with the (nearly) unobservable/uncontrollable states. These very small numbers are inverted in the computation of T or T^{-1}, where they become very big numbers, even though they correspond to insignificant states. Consider the last term in the following equation, which inverts the Hankel singular values in Moore's method:[19]

$$T^{-1} = \Sigma_c^{-\frac{1}{2}} U_c^T U_b \left(\Sigma^H\right)^{-\frac{1}{2}} \tag{84}$$

This leads to ill-conditioning of the T-matrix itself and to numerical errors in the TGSE, Eq. (76). Mallory's pre-balancing method[30] was successfully applied to solve this problem, since the pre-balancing algorithm removes the unobservable/uncontrollable states before the inversion. This leads to a balanced, reduced, minimal and well-conditioned system and accurate answers for the RMS and sensitivities.

Summary and Recommendations

The challenges of large order LTI system models are addressed in this paper by presenting a fast diagonal Lyapunov solver, apriori error bounds for model reduction as well as a governing sensitivity equation for similarity transformed state space realizations. These contributions taken together enable several analyses for large order integrated models, with an increase in efficiency and known tradeoff between speed and accuracy. A standard recommended process could be as follows:

1. Assembly of integrated model, S_{zd}, Eq.(6)
2. System diagonalization and conditioning, Eq.(28)
3. Model reduction with error bounds, Eq.(70,83)
4. Performance analysis using `newlyap` (Sect. IV)
5. Sensitivity analysis using TGSE, Eq.(79)

Some numerical conditions must be met in order to successfully apply these techniques. First the system, S_{zd}, must be asymptotically stable. This means that no poles (of the appended, closed-loop system) are allowed on the $j\omega$-axis or in the right half-plane (RHP). Secondly the system must be in a minimal realization. This requirement is harder to meet since larger order models invariably all feature pole-zero cancellations after initial model assembly, which are indicative of uncontrollable or unobservable states. Even though these states do not generally prevent a solution for Σ_q to be found (by definition they don't contribute to the RMS), they cause the system to be ill-conditioned. This causes problems when computing the balancing matrix, T. This calls for a comprehensive framework of model quality management such as the one proposed by Uebelhart.[18]

Suggestions for future work include a more versatile and robust subsystem planner for the fast Lyapunov solver, analysis of physical parameter sensitivity for large systems and linking these techniques to fast eigenvalue solvers. The diagonalization approach might lend itself to parallel computation of the Lyapunov solution. Additional research is needed in the areas of time domain simulation for large order, hybrid systems. Finally, for sensitivity analysis and optimization we want to avoid expensive computations of the transformation T for small perturbations. The range of validity of T for small perturbations should therefore be investigated.

This paper focused mainly on the numerical consequences of increasing model fidelity and size, n. A perhaps more important question is the issue of tracking design changes, performance improvements and uncertainty predictions for evolving design iterations like the ones shown in Fig. 1.

Acknowledgements

This research was supported by the NASA Goddard Space Flight Center under contracts No. NAG5-6079 and No. NAG5-7839 and by the Jet Propulsion Laboratory under the SIM research contract No. JPL 961123. The above research contracts were monitored by Mr. Gary Mosier (GSFC), Dr. Sanjay Joshi (JPL), Mr. Robert Grogan (JPL) and Dr. Ipek Basdogan (JPL).

Appendix A: SIM Description

The Space Interferometry Mission (SIM) is designed as a space-based 10-m baseline optical Michelson interferometer operating in the visible waveband. This

mission will open up many areas of astrophysics, via astrometry with unprecedented accuracy. Over a narrow field of view SIM is expected to achieve a mission accuracy of 1 μas. In this mode, SIM will search for planetary companions to nearby stars, by detecting the astrometric 'wobble' relative to a nearby (1^o) reference star. In its wide-angle mode, SIM will be capable of providing 4 μas precision absolute position measurements of stars, with parallaxes to comparable accuracy, at the end of a 5-year mission. One of the key conditions for obtaining interference fringes is tight control of optical pathlength differences (OPD). Source: NASA Jet Propulsion Laboratory.

Appendix B: Disturbance Model

The disturbances $w(t)$ can be undesired forces, torques, base motion, sensor and actuator noise, among others. The correlation function of the random process $w(t)$ is defined as

$$R_{ww}(t_1, t_2) = E\left[w(t_1)\, w^T(t_2)\right] \tag{85}$$

where $E[\,]$ is the expectation operator and $w(t)$ is a vector of stochastic random processes. If $w(t)$ is stationary the values of $w(t)$ will change over time, but the *statistics* of $w(t)$ will not and R_{ww} is a function of the single time-lag τ.

$$R_{ww}(\tau) = E\left[w(t)\, w^T(t+\tau)\right] \tag{86}$$

Assuming that all $w(t)$ are zero-mean, the covariance matrix Σ_w of the disturbance signals is the value of the correlation matrix R_{ww} for $\tau = 0$. The simplification in the covariance matrix Σ_w can be made for zero-mean processes, since for a typical term $\sigma_{w_i w_j}$ in the covariance matrix we can write

$$\begin{aligned}&\sigma_{w_i w_j} = E\left[(w_i - \mu_{w_i})\,(w_j - \mu_{w_j})\right] = \\ &E[w_i w_j] - \underbrace{E\left[w_i \mu_{w_j}\right]}_{=0} - \underbrace{E\left[\mu_{w_i} w_j\right]}_{=0} + \underbrace{E\left[\mu_{w_i}\mu_{w_j}\right]}_{=0} \\ &= E[w_i w_j] = \sigma_{w_i w_j}\end{aligned} \tag{87}$$

where $\mu_{w_i} = E[w_i]$ is the mean (expected value) of the i-th random process. The mean-square values of the elements of w are simply the diagonal entries in the covariance matrix.

$$(\Sigma_w)_{i,i} = E\left[w_i^2(t)\right] \tag{88}$$

where $w_i(t)$ is the i-th element in w. If w is zero-mean, then the mean-square values and the variances are identical.

$$\sigma_{w_i}^2 = E\left[w_i^2(t)\right] - \underbrace{\left(E[w_i]\right)^2}_{=0} = \mathrm{RMS}^2 \tag{89}$$

The (power) spectral density function $S_{ww}(\omega)$ can be obtained by taking the Fourier transform of equation (86)

$$S_{ww}(\omega) = \mathcal{F}\left[R_{ww}(\tau)\right] = \int_{-\infty}^{+\infty} R_{ww}(\tau)\, e^{-j\omega\tau} d\tau \tag{90}$$

Note that the $1/2\pi$ factor is not included in the definition of the Fourier transform. Other authors[13] place it in the Fourier transform formula. Either definition will produce the same result in the end as long as the definition is used consistently. The inverse Fourier transform of $S_{ww}(\omega)$ recovers the correlation function.

$$R_{ww}(\tau) = \mathcal{F}^{-1}\left[S_{ww}(\omega)\right] = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_{ww}(\omega)\, e^{+j\omega\tau} d\omega \tag{91}$$

Evaluating (91) at $\tau = 0$ will produce the covariance matrix of w.

$$R_{ww}(0) = \Sigma_w = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_{ww}(\omega)\, d\omega \tag{92}$$

Equation (92) suggests an alternative way of calculating the mean-square values of w by integrating under the spectral density functions, namely

$$\sigma_{w_i}^2 = [\Sigma_w]_{i,i} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} [S_{ww}(\omega)]_{i,i}\, d\omega \tag{93}$$

The diagonal elements of the spectral density function matrix $S_{ww}(\omega)$ are usually referred to as power spectral densities (PSD's), whereas the off-diagonal elements are the cross spectral densities. Depending on the shape of the PSD's representing w it is possible to approximate the shape of the functions $S_{ww}(\omega) = G_{wd} G_{wd}^H$ by prewhitening filters in state space form, where the inputs to the state space system are unit intensity white noise processes d:

$$\begin{aligned}\dot{q}_d &= A_{wd} q_d + B_{wd} d \\ w &= C_{wd} q_d + D_{wd} d\end{aligned} \tag{94}$$

Note that the feedthrough matrix D_{wd} is generally zero. A plot of the disturbance PSD's and the cumulative RMS functions in this paper is shown in Figure 3.

Appendix C: Derivation of TGSE

A simple derivation can be carried out by invoking the similarity transformation property. By substituting the inverse transformation of the balanced state covariance matrix

$$\Sigma_q = T^{-1}\Sigma_{\tilde{q}} T^{-T} \tag{95}$$

and the Lagrange multiplier

$$L_i = T^{-1}\tilde{L}_i T^{-T} \tag{96}$$

into the original GSE (73), we obtain

$$\frac{\partial \sigma_{z_i}^2}{\partial p_j} = trace\left[\Sigma_q \frac{\partial\left(C_{zd,i}^T C_{zd,i}\right)}{\partial p_j}\right] + trace\left[L_i\left\{\frac{\partial A_{zd}}{\partial p_j}\Sigma_q + \Sigma_q \frac{\partial A_{zd}^T}{\partial p_j} + \frac{\partial\left(B_{zd}B_{zd}^T\right)}{\partial p_j}\right\}\right] = trace\left[T^{-1}\Sigma_{\tilde{q}}T^{-T}\frac{\partial\left(C_{zd,i}^T C_{zd,i}\right)}{\partial p_j}\right] + trace\left[T^{-1}\tilde{L}_i T^{-T}\left\{\frac{\partial A_{zd}}{\partial p_j}\Sigma_q + \Sigma_q \frac{\partial A_{zd}^T}{\partial p_j} + \frac{\partial\left(B_{zd}B_{zd}^T\right)}{\partial p_j}\right\}\right] \quad (97)$$

The transformed GSE, Eq.(76) is then obtained by pre-multiplying the second trace term with T^{-T} and post-multiplying it with T^T.

$$\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} = trace\left[T^{-1}\Sigma_{\tilde{q}}\left(T^{-1}\right)^T\frac{\partial\left(C_{zd,i}^T C_{zd,i}\right)}{\partial p_j}\right] + trace\left[\tilde{L}_i\left\{T\frac{\partial A_{zd}}{\partial p_j}T^{-1}\Sigma_{\tilde{q}} + \Sigma_{\tilde{q}}\left(T^{-1}\right)^T\frac{\partial A_{zd}^T}{\partial p_j}T^T\right\}\right] + trace\left[\tilde{L}_i\left\{T\frac{\partial\left(B_{zd}B_{zd}^T\right)}{\partial p_j}T^T\right\}\right] \quad (98)$$

References

[1] K. J. Bathe. *Finite Element Procedures.* Prentice-Hall, Inc., 1996.

[2] Jet Propulsion Laboratory. *Integrated Modeling of Optical Systems User's Manual - Version 3.0*, January 1998. JPL D-13040.

[3] Homero L. Gutierrez. *Performance Assessment and Enhancement of Precision Controlled Structures During Conceptual Design.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 1999.

[4] D. C. Redding and W. G. Breckenridge. Optical modeling for dynamics and control analysis. *Journal of Guidance, Control, and Dynamics*, 14(5):1021–1032, Sept.-Oct. 1991.

[5] R. A. Laskin and M. San Martin. Control/structure system design of a spaceborne optical interferometer. In *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Stowe, VT, August 1989. AAS Paper No. 89-424.

[6] Olivier L. de Weck. Integrated Modeling and Dynamics Simulation for the Next Generation Space Telescope. Master's thesis, Massachusetts Institute of Technology, June 1999.

[7] R. A. Masterson, D. W. Miller, and R. L. Grogan. Development and validation of reaction wheel disturbance models: Empirical model. *Journal of Sound and Vibration*, 249(3):575–598, March 2002.

[8] L. M. Elias and D. W. Miller. A coupled disturbance analysis method using dynamic mass measurement techniques. In *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Denver, CO, April 2002. AIAA-2002-1252.

[9] B. Bialke. A compilation of reaction wheel induced spacecraft disturbances. In *Proceedings of the 20th Annual AAS Guidance and Control Conference*, Breckenridge, CO, February 5–9, 1997. AAS Paper No. 97-038.

[10] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice-Hall, Inc., 1996.

[11] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *J. Comp. Appl. Math.*, 6:19–26, 1980.

[12] Kin Cheong Sou. Fast time domain simulation for large order hybrid systems. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA 02139, May 2002. SSL Report # 11-02.

[13] P. H. Wirsching, T. L. Paez, and H. Ortiz. *Random Vibrations: Theory and Practice.* John Wiley & Sons, Inc., 1995.

[14] Peiman G. Maghami and Sean P. Kenny. Algorithms for efficient computation of transfer functions for large order flexible systems. Technical Report 206949, NASA Langley Research Center, March 1998.

[15] A. Gelb, editor. *Applied Optimal Estimation.* The M.I.T. Press, 1974.

[16] R. H. Bartels and G.W. Stewart. Solution of the matrix equation ax + xb = c. *Communications of the ACM*, 15(9), 1972.

[17] Olivier de Weck. *Multivariable Isoperformance Methodology for Precision Opto-Mechanical Systems.* PhD thesis, Massachusetts Institute of Technology, 77 Mass Ave, Cambridge, MA 01760, June 2001.

[18] Scott A. Uebelhart. Conditioning, Reduction, and Disturbance Analysis of Large Order Integrated Models of Space-Based Telescopes. Master's thesis, Massachusetts Institute of Technology, February 2001.

[19] B. C. Moore. Principal component analysis of linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26:17–32, February 1981.

[20] A. Laub. Computation of balancing transformations. *Proceedings of JACC*, 1(FA8-E), 1980.

[21] A.J. Laub, M.T. Heath, C.C Paige, and R.C Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Trans. Automatic Control*, (AC-32):115–122, 1987.

[22] C. Z. Gregory. Reduction of large flexible spacecraft models using internal balancing theory. *Journal of Guidance, Control, and Dynamics*, 7(6):17–32, December 1984.

[23] Gregory Mallory and David W. Miller. Increasing the numerical robustness of balanced model reduction. *Journal of Guidance, Control, and Dynamics*, 25(3):596–598, May-June 2002. Engineering Notes.

[24] R. N. Jacques. An approach to the preliminary design of controlled structures. Master's thesis, Massachusetts Institute of Technology, February 1991. SERC Report #1-91.

[25] D. V. Murthy and R. T. Haftka. Survey of methods for calculating sensitivity of general eigenproblems. In *Sensitivity Analysis in Engineering*, pages 177–196, Langley Research Center, September 1986. NASA CP 2457.

[26] R. L. Fox and M. P. Kapoor. Rates of change of eigenvalues and eigenvectors. *AIAA Journal*, 6:2426–2429, December 1968.

[27] R. B. Nelson. Simplified calculations of eigenvector derivatives. *AIAA Journal*, 14:1201–1205, September 1976.

[28] S. P. Kenny. Eigenvalue and eigenvector derivatives for structures. Final Report, MIT Course 2.093: Computer Methods in Dynamics, April 29,1997.

[29] G. J.-W. Hou and G. Koganti. Sensitivity analysis of Lyapunov and Ricatti equations with application to controls-structures integrated design. In *Proceedings of the 34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pages 1906–1915, LaJolla, CA, April 1993. AIAA Paper No. 93-1529.

[30] Gregory Mallory and David W. Miller. Decentralized state estimation for flexible space structures. *40th AIAA Structural Dynamics and Materials Conference*, 1999.