

Systems That Sense and Respond: Modeling, Analysis, and Control of Buildings



Athanasios Antoulas, Danny C. Sorensen, Kyle Gallivan, Paul van Dooren,
Ananth Grama, and Ahmed Sameh

Abstract Complex systems are increasingly designed with data sensing and control built into their functional specification, with the goal of modeling, understanding, and managing dynamic conditions in their operations. Such systems range from large civil infrastructures (e.g., buildings, bridges, etc.) to manufacturing environments (e.g., blast furnaces, Computer Numerical Control (CNC) manufacturing processes, etc.). Design and operation of such systems requires effective and scalable computation and communication infrastructure for sensing system state,

Christoph Hoffman died before publication of this work was completed.

A. Antoulas

Professor of Electrical and Computer Engineering, Rice University, West Lafayette, IN, USA
e-mail: aca@rice.edu

D. C. Sorensen

Noah Harding Professor of Computational and Applied Mathematics, Rice University, West Lafayette, IN, USA
e-mail: sorensen@rice.edu

K. Gallivan

Department of Mathematics, Florida State University, Tallahassee, FL, USA
www.math.fsu.edu/~gallivan

P. Van Dooren

Professor of Mathematical Engineering, Catholic University, Louvain, Belgium
e-mail: paul.vandooren@uclouvain.be

A. Grama (✉)

Samuel D Conte Distinguished Professor of Computer Science, Purdue University, West Lafayette, IN, USA
e-mail: ayg@cs.purdue.edu

A. Sameh

Samuel Conte Professor of Computer Science, Purdue University, West Lafayette, IN, USA
e-mail: sameh@cs.purdue.edu

modeling and prediction of system response to stimuli, and control through suitable actuation mechanisms; these systems approaches align with and benefit from the DDDAS paradigm and can be used to predict and mitigate effects of adverse structural and operational conditions, such as instabilities or other breach of system integrity, allowing decision-making and response. This chapter describes the computational elements of the design process, including simulation of complex structures through model reduction techniques, instrumentation for data sensing and integration, and systems infrastructure for software development and deployment in real-world settings. We demonstrate these techniques in the context of control of large buildings, with a retrospective case study of the catastrophic 9/11 events, such as the collapse of the World Trade Center (WTC), and discuss how the methods presented, placed in the context of DDDAS, could have been used to predict the catastrophic collapse. Underpinning the advances and dynamically integrated modeling with instrumentation, DDDAS-based approaches are agile computation and communication infrastructures to support the associated heterogeneous and multi-level multi-sensor support infrastructure, and they are also discussed in this chapter, together with a discussion of lessons learned and future directions for development of new methods to address outstanding challenges.

Keywords Modeling of structures · World Trade Center · Real-time control · Model reduction · Sensing and actuation

1 Introduction and Overview

Emerging paradigms for large engineered structures are classic examples of DDDAS-based systems. Conventionally built structures such as buildings, bridges, and dams are designed to withstand a predefined external load—typically in terms of ground movement during earthquakes, high winds, and hurricanes. However, as these structures age, or when external events cause the load to exceed design limits, catastrophic events can ensue. As will be discussed in this chapter, DDDAS-based methods for dynamic modeling, coupled with instrumentation, can be used to predict and/or mitigate structural damage and to also provide opportunities for informed response, including mitigating loss of lives. For example, the Seattle earthquake of February 28, 2001, resulted in over 272 injuries and over \$2 billion (estimated) in infrastructure loss. It is believed that the extent of infrastructure loss for a quake that measured 6.8 on the Richter scale was excessive. Other incidences (in India and Turkey) have underscored the need for designing infrastructure that can withstand higher loads. For example, a 2 km viaduct at Kaynasli, Turkey, completed in 1999 was damaged severely by the November 12, 1999, earthquake, which followed on the heels of another one on August 17, 1999. The viaduct was of current design (American Association State Highway and Transportation Officials '96) and was equipped with custom-built passive damping devices. The surface trace of the November 12 event intersected the trajectory of the viaduct. On the basis of

visual evidence alone, it has become very difficult to understand the primary cause of failure (distortion of ground, shaking of the bridge, malfunction of the dampers) and, therefore, very difficult to decide on a course of action (fix the bridge as is, redesign, or move to another site). The availability of information even from as few as a handful of sensors on the viaduct and on the ground around it would have made it possible to make the right decision quickly and avoid the current state of uninformed inaction. More importantly, had data been acquired during the August 17 event, the catastrophe in November might have been predicted, if not avoided, and informed emergency response preparations could have been planned.

The past few decades have seen significant advances in the design and construction of active structures—structures that damp response to external stressors. Early work on active structures relied on the use of active mass dampers (AMDs). AMDs are weights (typically less than 1% of building mass) attached via an actuator to the structure to counterbalance the effect of high winds and earthquakes. AMDs use a control system that relies on sensed inputs (typically wind vanes and anemometers) and computed responses to affect appropriate actuation. More modern structures frequently use a hybrid mass damper (HMD), which relies on the motion of a tuned mass damper (such as a pendulum) or magnetorheological (MR) fluids, along with a control strategy to maximize its effectiveness and robustness. The essential characteristic of MR fluids is their ability to reversibly change from a free-flowing, linear viscous fluid to semi-solid with controllable yield strength in milliseconds when exposed to a magnetic field. These fluids are finding increasing application in semi-active vibration control systems.

The true potential of structural control lies in the ability to affect actuation using a large number of distributed actuators (as opposed to a small number of semi-active actuators). Similarly, the impact of improved sensor technology is motivated by the shortcomings of the current state of the art. As an early example, during the El Centro earthquake of 1979, a six-story reinforced concrete building, instrumented with thirteen accelerometers, suffered severe damage. The accelerometers produced data, which were studied in detail by researchers in various laboratories. Each lab provided a different scenario for the events that preceded and caused the observed failure. The data obtained was not sufficient to identify decisively the causes of failure, which also meant that the experience had little impact on design practice.

These developments motivated three major lines of research and development, which find increasing use in current structural design: (i) accurate design and modeling techniques that predict structural response to a variety of external stimulus; (ii) control strategies that sense and respond to external stimulus to mitigate damage and loss of life; and (iii) computational systems infrastructure for sensing, networking, computation, and real-time control. All these developments put in the context of DDDAS, where ubiquitous instrumentation data (archival and real-time) is integrated with the modeling methods described here (in a feedback control loop), can be used to provide real-time assessment and prediction and control to mitigate damage or take other response actions to mitigate harm. In this chapter, we summarize work on all three aspects—we exemplify structural simulations (based on “model reduction” approaches) through work, which modeled

the collapse of the World Trade Center (WTC) and the damage to the Pentagon due to the 9/11 terrorist attack of planes crashing into these buildings. We then describe challenges in model reduction and control of complex structures, and we describe work on building sensor networks and systems infrastructure for real-time assessment and control, to take mitigating actions for managing adverse situations, including emergency response. Referring to the 9/11 attacks on the WTC buildings, our simulations have shown that the initial impact did compromise a portion of the buildings (structural columns and floors), but it was the ensuing fire that accentuated the damage; namely, the fire-induced heat caused buckling of support columns and beams, and this was combined with the momentum carried by the floors collapsing on the floors underneath that precipitated the entire buildings' collapse. Another point of interest shown by the simulations we performed for the attack to WTC and the Pentagon was that the initial damage of the planes crashing onto the buildings was in part due to the impact of the plane fuselage, but the major part of the initial damage was due to the momentum and imparted force from the fuel weight carried by the planes.

2 Structural Modeling and Simulations

There has been significant recent work on development of simulation methodologies and validation techniques for understanding the behavior of large structures—buildings, bridges, dams, etc. These efforts are aimed at designing structures that are robust to natural excitations (earthquakes, floods, high winds) as well as human factors (explosions, fire). The eventual goal of these efforts is to understand why structures fail and how to predict and mitigate these failures. We describe efforts aimed at developing complex structural models, using them to understand structural failure (the 9/11 collapse of the WTC Twin Towers and the Pentagon), and experimental efforts aimed at calibrating and validating computational models. In particular, we demonstrate that using DDDAS-based methods, together with high-performance computing platforms, powerful simulation engines, high-resolution geometric models, and accurate materials models, we can gain considerable understanding of structural response to excitations. We also describe the complexity of experiment design and data acquisition from experiments. Note that the work described here is not only a retrospective assessment, but many data used are estimates, as the buildings did not have the full instrumentation needed to apply the full power of DDDAS.

2.1 Simulation of Large-Scale Structures

The objective of computational simulation of structures is to develop models capable of describing structural response to a high degree of accuracy. These models are used to predict failure modes, as well as to guide techniques for model reduction.

Reduced-order models (models with fewer degrees of freedom) can be used for real-time control of structures. In this section, we describe work on model construction, simulation, and validation. We also present some insights on the interplay between the numerical and geometric aspects of the problem.

The initial model for the simulation has two parts: a description of the geometry of the elements and a description of the physical properties. The former is translated into node coordinates and connectivity information to specify the elements (beam, shell, and volume elements); the latter reduces to a system of equations from which a numerical problem is constructed. Since high-fidelity models of structures have large numbers of elements, it is best to use computational models where available and to discretize them using an automatic mesh generator, instead of building finite-element models from scratch. Many choices for mesh generation exist, with varying degrees of suitability for classes of objects. We describe each of these phases of modeling using a case study of the aforementioned 9/11 events.

2.1.1 Modeling the North Tower of the World Trade Center and the Airframe

The discussion in this subsection provides a background on the considerations for modeling the event of the aircraft crash on the WTC.

In the case of the World Trade Center simulation, it was possible to use a partially meshed model of the exterior of the North Tower (WTC-1), which was then completed by adding the interior structure, consisting of the buildings core, the spandrel beams supporting the floor, and the elements for the concrete flooring as well (Fig. 1, lower panels). Staircases were not modeled in detail. This task of completing the interior of WTC-1 used public domain blueprints. The model had 640,000 nodes, 530,000 beam elements, and 360,000 shell elements. Aside from the model size, meshing the building structure of modern high-rises is not difficult [5]. However, much of the detailed structure of the joints gets abstracted. By this, we mean that the geometric structure is reduced to a shared node among several beam elements, for example. The corresponding material response is then captured by the governing equations and material properties. Also, the beam shapes are often abstracted as well, represented simply by line elements with an intrinsic orientation and a specification of the type of beam. This geometric information is lost in translation and has to be reconstructed for visualizing the simulation results. Fortunately, obtaining reliable information about the WTC-1 building was not difficult, and the validity of the model is established through experiments, described later in this section. Modeling the airplane posed challenges however. Here, it turned out to be difficult to obtain reliable information about the Boeing 767 airframe structure. Overall dimensions are readily available, but except for an accurate reconstruction of the planes exterior, little information is available about the interior structure, the shape, and dimensions of the beams, ribs, and stringers. Understandably, Boeing considers a detailed model of its planes proprietary, so we needed to find a different way to create a reasonable model and test its accuracy as best as possible [10]. This task begins with a graphical model of the plane's exterior.

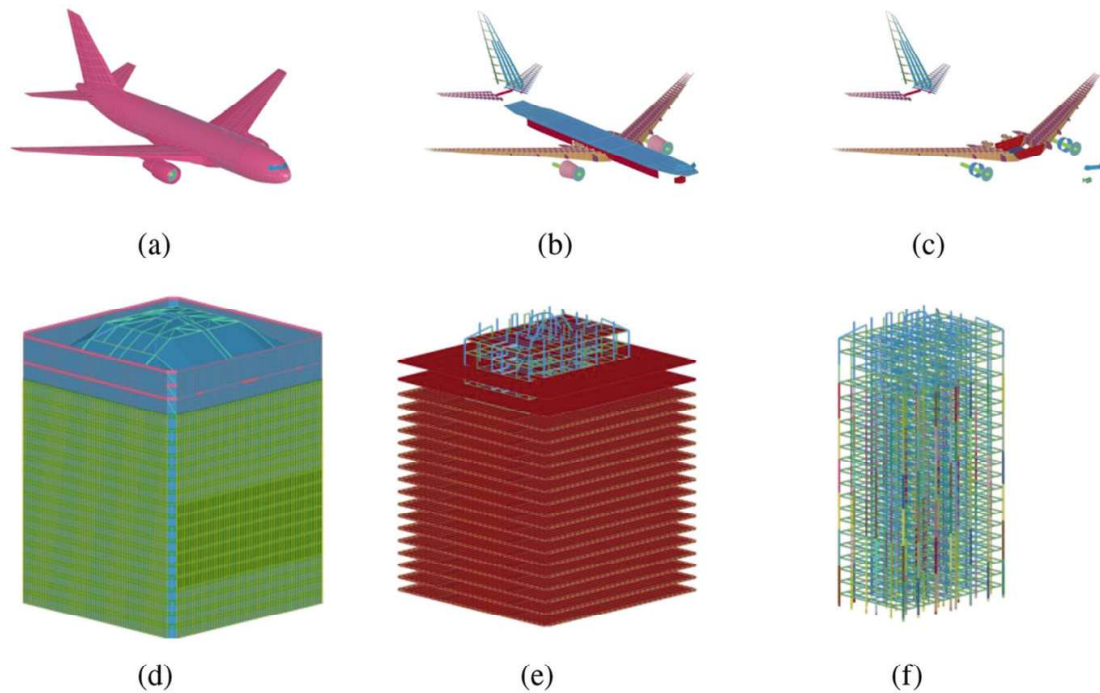


Fig. 1 Constructing models of the aircraft and the World Trade Center Tower. (a) Simulation setup: modeling the aircraft. (b) Structural and skin elements. (c) Engines and wings. (d) Simulation setup: modeling the building. (e) Floors. (f) Structural skeleton

A graphical model is wholly unsuitable for finite element analysis (FEA). Its surfaces are often composed of very long and slender triangles; a minimum angle of less than 10 degrees is not uncommon. Moreover, relying on the graphics hardware to cull invisible regions of the surface, parts such as the skin of a wing will extend part-way into the interior of the fuselage without any explicit representation of the intersection curve. Thus, the first step is to rectify the airplane's skin and to suitably mesh it, a task that requires substantial manual labor even using the meshing capabilities of the LS-DYNA structural analysis—multi-physics simulation software package [4]. The keel beam, the wing anchor beams, floor supports, and the floor are added to the fuselage ribs and stringers. The spars and ribs are added to the wings, aileron, and rudder. Aircraft engines and landing gear contain substantial parts. The landing gear typically is a titanium forging, and the engine shaft is another substantial part. Both structures were modeled and meshed appropriately. Sections of the resulting mesh are shown in Fig. 1 (top panels).

2.1.2 Model Calibration

Since there is little hard data on the structural components of the aircraft, experts in aerospace engineering were consulted, and the Riera approach was used to calibrate the aircraft model [31, 34]. The Riera approach is an upshot of experiments at Sandia National Labs measuring reaction forces and damage from aircraft impact.



Fig. 2 Visualizations showing SPH model of the fuel and combustion in the building

The impact forces, measured from the time of impact to its completion, relate intuitively to the ability of the impacting structures to absorb energy by deformation. Accordingly, the forces are highest when the most massive and rigid structural elements hit. Figure 2 shows both a moment in our impact simulation and the Riera curve used to calibrate the aircraft model.

The fuel of the impacting aircraft carries a large fraction of the kinetic energy and effect on the impact. The fuel is modeled using smoothed particle hydrodynamics (SPH) method, which benefited from a separate calibration step. The SPH calibration was done by civil engineers modeling the liquid of a beverage can impacting a rigid metal plate. Led by Santiago Pujol, the simulation results were compared with experimental measurements. These are shown in Fig. 2.

2.1.3 Simulation and Results

After constructing models and calibrating them, the impact of the loaded airframe into the top floors of the WTC-1 building is simulated. Initial conditions were chosen to approximate the known attitude and position of the North Tower impact.

The difficulty obtaining accurate dimensions and properties of the air frame make quantitative conclusions from the simulations challenging. In fact, there are widely diverging estimates of the core damage of the North tower [12, 26, 29, 35]. Nevertheless, it is possible to conclude with considerable confidence that the collapse mechanism of the North Tower was initiated by the ensuing fires, which compromised the integrity of the metal structural support columns, beams, and joints [25]. Those fires easily elevated the temperature of the metal core structure

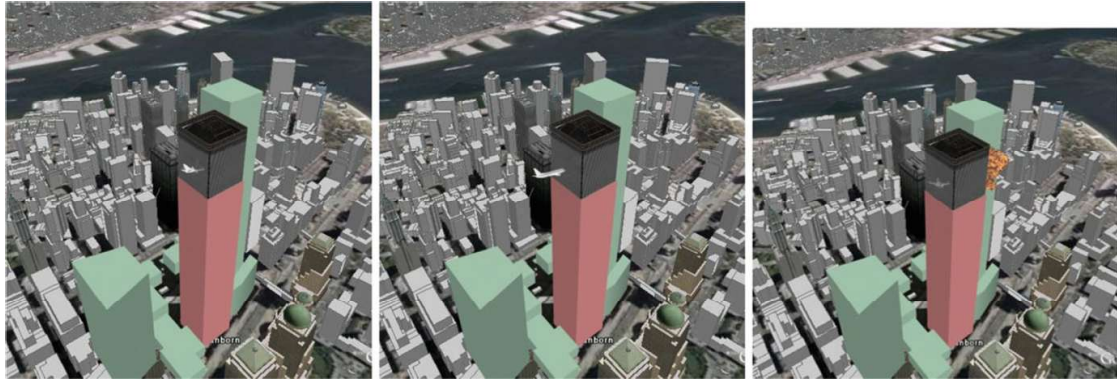


Fig. 3 World Trade Center along with its surrounding structure, showing point of impact

sufficiently so that it could not carry the building load—essentially, the metal load-bearing columns and beams buckled and/or snapped due to the fire-induced heat and weight of the floors, which collapsed (pancaked) on top of the floors underneath. That load was increased by the extra load that rested on the core due to the perimeter structure being cut in a wide swath. Further corroboration that these simulations captured the essential elements of the event, despite the uncertainty surrounding the exact properties of the air frame, can be seen in Fig. 6, which compares the simulated damage to the facade, at the entry of the colliding aircraft, with a contemporary image reproduced from [3]. This work was first published in [24].

2.1.4 Visualization of Simulation Results

The final part of the 9/11 simulation effort, led by Drs. Voicu Popescu, Paul Rosen, and Chris Hoffman, was to render and animate the results of the simulation using state-of-the-art techniques from computer graphics. Downloaded to date several million times, the simulation animation [2] has generated tremendous debate and continues to be used in documentaries in the USA and abroad. Here, the key step was to export the simulation data, as computed by the FEA model, and import it into 3dsMax, an animation system that incorporates leading-edge graphics algorithms. In this part of the effort, a major difficulty is the aforementioned abstraction of geometric shape, particularly the shape of beams and how different structural elements are joined. Some still images from the visualization are shown in Figs. 3, 4, and 5.

A similar physically grounded simulation of the Pentagon crash has also been performed (Fig. 6). Simulation results from this crash are illustrated in Fig. 7.

2.1.4.1 Geometric Simplification

Geometry simplification is quite common in finite element analysis: symmetry is exploited where possible; in the case of WTC-1, details of the lower floors of the building are eliminated; detail shapes such as beam interconnections and cross sections are not represented geometrically. As we have seen, such simplifications of

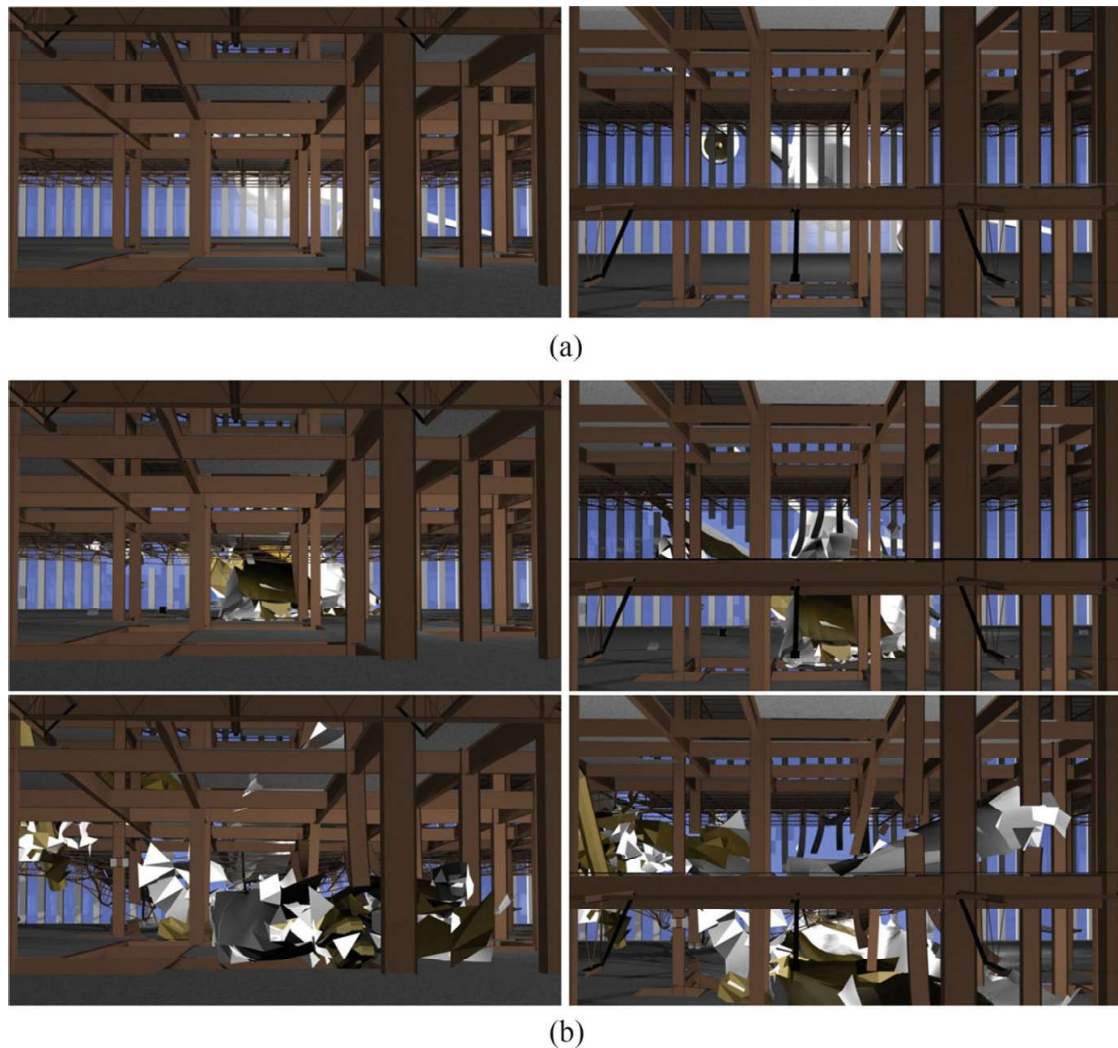


Fig. 4 High-fidelity simulations of aircraft impact with structural elements of the building. (a) Building model. (b) Model of aircraft crashing in building

the geometry lose important detail that will be missed when creating subsequently sophisticated simulations and visualizations. In the larger scientific workflow, therefore, it is helpful to have a detailed geometric model from which to derive, as suggested by the numerical properties, a low-order shape approximation that can be lifted (traced back) to the original, detailed geometry model. Both the reduced-order physics and the corresponding shape model, therefore, become information structures derived from a master model, a concept familiar to the research community from the area of discrete manufacturing simulations.

3 Model Reduction and Control of Large Dynamical Systems

Many physical processes in science and engineering are modeled accurately using finite dimensional dynamical systems. Examples include weather simulation, molecular dynamic simulations (e.g., modeling of bio-molecules and their

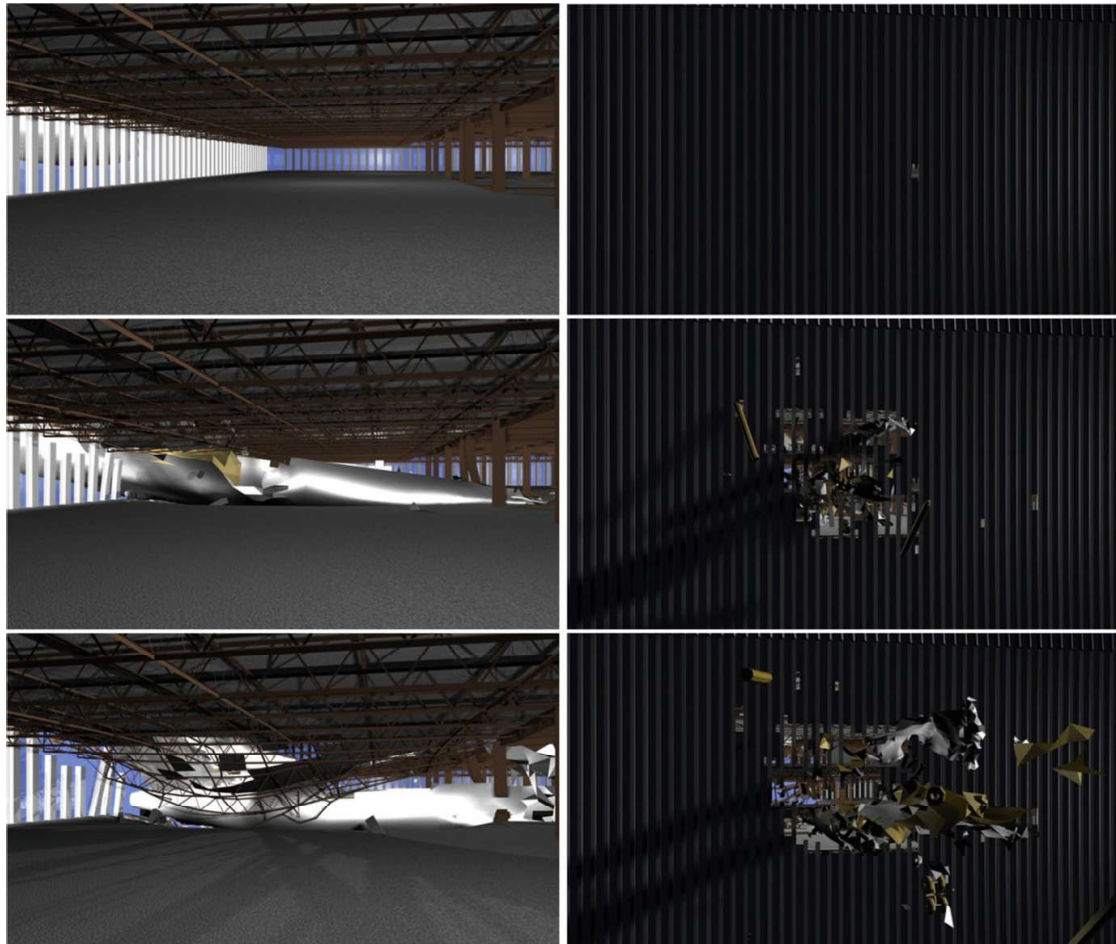


Fig. 5 Side view of the impact area (left panels) and debris exiting the building (right panels)

identification), structural dynamics (e.g., flex models of the international space station, and structural response of high-rise buildings to wind and earthquakes), electronic circuit simulation, semiconductor device manufacturing (e.g., chemical vapor deposition reactors), and simulation and control of micro-electro-mechanical (MEMS) devices. An important subclass of these processes can be effectively monitored to gather data in support of simulation, diagnosis, prognosis, prediction, and control. This class of dynamic data-driven application systems (DDDAS) pose challenging problems ranging from data gathering, assimilation, effective incorporation into simulations, to real-time control.

In the vast majority of applications where control can be affected, the original system is augmented with a second dynamical system called a controller. In general, the controller has the same complexity as the system to be controlled. Since in large-scale applications of interest one often aims at real-time control, reduced complexity controllers are required. One possible solution to building low-complexity controllers is to design such controllers based on reduced order models of the original dynamical systems. In yet other systems in which real-time prognostics and prediction are required, constraints on compute power, memory, communication bandwidth, and available data might necessitate the use of such

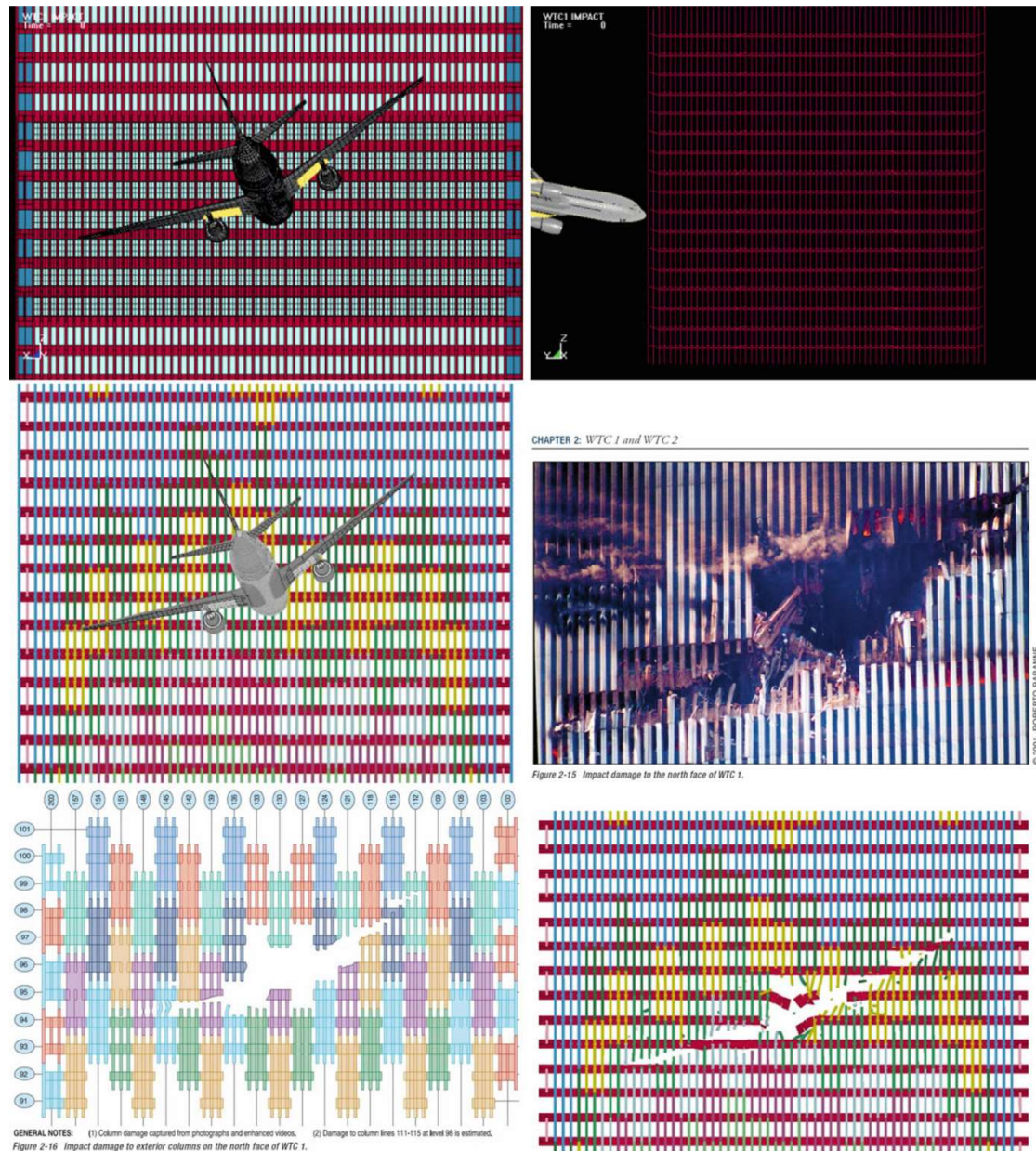


Fig. 6 Angle of entry of aircraft, simulated damage, and comparison to documented damage to the building

reduced-order models. This chapter outlines the state of the art in model reduction, discusses outstanding challenges, and presents possible solution strategies for addressing these challenges.

Model reduction seeks to replace a large-scale system of differential or difference equations by a system of substantially lower dimensions that has nearly the same response characteristics. Two main themes can be identified among several methodologies: (a) *balancing based methods* and (b) *moment matching methods*. Balanced approximation methods are built upon a family of ideas with close connection to the singular value decomposition. These methods preserve stability and allow for global error bounds. Available algorithms for these methods, however, are not

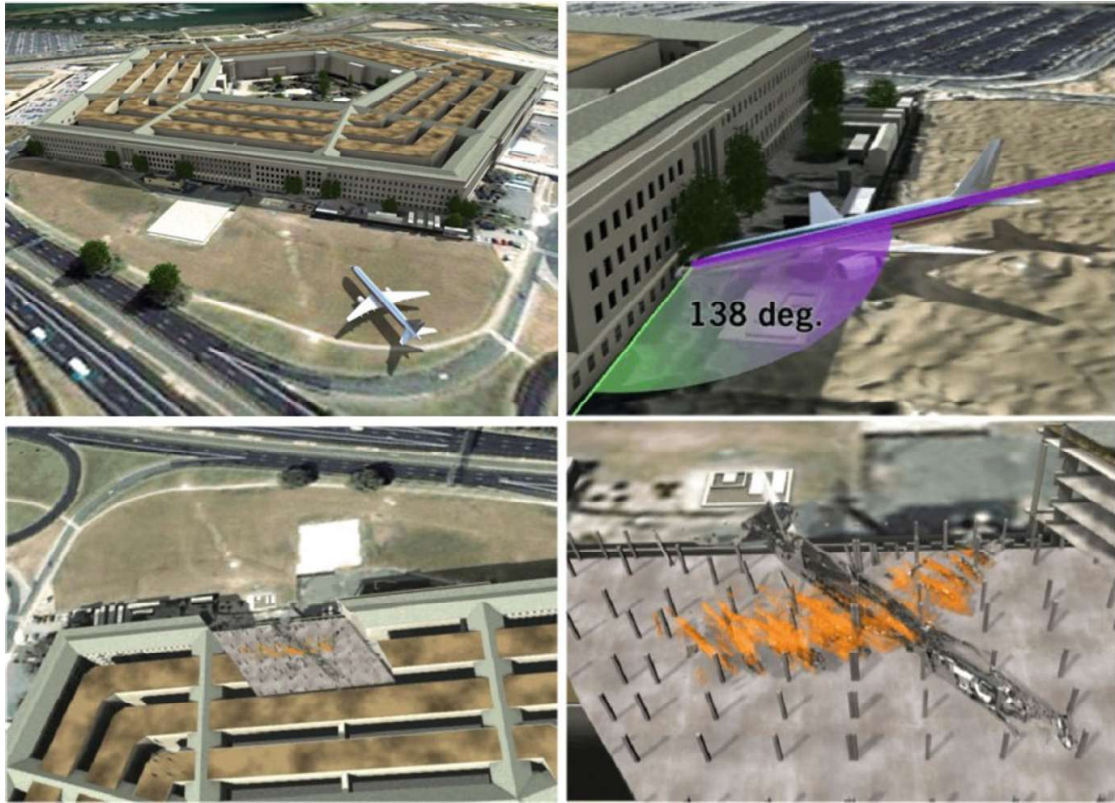


Fig. 7 Model of the Pentagon and aircraft, showing impact with pillars and beams and fuel

suitable for large-scale problems since they have been developed mainly for dense matrix computations. Moment matching methods are based primarily on Padé-like approximations, which, for large-scale problems, have led naturally to the use of Krylov and rational Krylov subspace projection methods. While these moment matching schemes enjoy greater efficiency for large-scale problems, maintaining stability in the reduced-order model cannot be guaranteed. Consequently, their use can be problematic at times. Moreover, no a priori global error bounds exist for moment matching schemes.

A current research trend aims at combining these two approaches by deriving iterative methods that incorporate the desirable properties of both of these classes. This chapter addresses several important unresolved issues in model reduction of large-scale linear time-invariant (LTI) systems, extending them to a new class of problems that require adaptive models. It also addresses large-scale structured problems that are either time varying or that require iterative updating of the initial reduced models to obtain better approximation properties.

3.1 Motivating Data-Driven Control Applications

The problems addressed in this chapter focus on reduced models for dynamic data-driven systems, which are characterized by model adaptivity and need for

maintaining or exploiting a specific type of structure. This can be either due to time variations in the model itself or due to the fact that the model to be approximated varies at each iterative step of some design loop.

Large-Scale Mechanical Systems Consider simulating a structure that is discretely modeled by a large-scale finite element formulation such as a long-span bridge, a tall building, or even a car windshield, which is subject to various external forces. In tall buildings, for example, these forces could be the result of strong ground motion or wind. Typically, the model is given by second-order systems of equations of the form:

$$\mathbf{M}(\omega, t)\ddot{\mathbf{x}}(t) + \mathbf{D}(\omega, t)\dot{\mathbf{x}}(t) + \mathbf{K}(\omega, t)\mathbf{x}(t) = \mathbf{f}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \quad (1)$$

Here, $\mathbf{M}(\omega, t)$ is the mass matrix, $\mathbf{D}(\omega, t)$ is the damping matrix, and $\mathbf{K}(\omega, t)$ is the stiffness matrix of the system. All three matrices are assumed to be frequency dependent and possibly time dependent. For large-scale structures, the state vector $\mathbf{x}(t)$ is of order N , which can reach in the tens of millions. The forcing function $\mathbf{f}(t)$ represents wind and earthquake effects and is of the same dimension. We assume that the structure under consideration is densely instrumented by networked sensing and actuator elements. Further, assuming that a sensor-actuator complex (SAC) is monolithically integrated with a strut system containing controllable dampers that can change the stiffness characteristics of the structure in milliseconds, the control function $\mathbf{u}(t)$, an m -dimensional vector, represents the action that can be affected through the smart strut system. Finally, the p -dimensional vector $\mathbf{y}(t)$ represents the signals collected via the sensor network. The dimension N of the state vector $\mathbf{x}(t)$ is much larger than m and p . The objective of the model reduction problem is to produce a model of the structure that possesses the “essential” properties of the full-order model. Such a reduced-order model can then be used for the design of a reduced-order controller to affect real-time control. The reduced-order model may be used to predict the onset of failure, and the reduced-order controller can be used for real-time control of the structure so as to mitigate failure.

There are several preliminary problems that must be solved before we realize an appropriate reduced-order model. These include (i) computing the optimal location of the sensors and actuators, where we could start from a nominal model with $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ constant; (ii) improving the dynamical model once the sensor/actuator placements are determined, using validated simulations (via physical laboratory models of the structure) or via measurements performed on the actual structure, if possible; (iii) designing a closed-loop controller that is derived from the reduced-order model; and finally, (iv) adapting the reduced-order model to the closed-loop configuration of the system (this will change its dynamics and, therefore, also its reduced-order model). In each of these steps, it is useful to have simplified (i.e., reduced-order) models on which the most time-consuming tasks can be performed in reasonable time.

3.2 *Balanced Approximation Methods*

Projection methods have provided effective model reduction algorithms for constructing reduced-order models in the large-scale setting. Given a continuous-time state space system [8]:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \quad (2)$$

with input $\mathbf{u}(t) \in \mathbb{R}^m$, state $\mathbf{x}(t) \in \mathbb{R}^N$, and output $\mathbf{y}(t) \in \mathbb{R}^p$, one defines a *projected* model of reduced-order $n \ll N$ for a reduced-state $\hat{\mathbf{x}}(t) \in \mathbb{R}^n$:

$$\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}\mathbf{u}(t), \quad \hat{\mathbf{y}}(t) = \hat{\mathbf{C}}\hat{\mathbf{x}}(t), \quad (3)$$

where

$$\mathbf{I}_n = \mathbf{W}^T \mathbf{V}, \quad \hat{\mathbf{A}} = \mathbf{W}^T \mathbf{A} \mathbf{V}, \quad \hat{\mathbf{B}} = \mathbf{W}^T \mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{C} \mathbf{V}. \quad (4)$$

The widely used balanced truncation technique constructs the “dominant” spaces $\mathbf{W}$ and $\mathbf{V}$ such that

$$\mathcal{P}\mathbf{W} = \mathbf{V}\Sigma_+, \quad \mathbf{Q}\mathbf{V} = \mathbf{W}\Sigma_+, \quad \mathbf{W}^T \mathbf{V} = \mathbf{I}_n \quad (5)$$

where Σ_+ is symmetric, positive definite, and usually diagonal and $\mathcal{P}$ and $\mathbf{Q}$ are the controllability and observability gramians, respectively. It follows that $\mathcal{P}\mathbf{Q}\mathbf{V} = \mathbf{V}\Sigma_+^2$ and $\mathbf{Q}\mathcal{P}\mathbf{W} = \mathbf{W}\Sigma_+^2$ and hence Σ_+^2 should contain the largest eigenvalues of $\mathcal{P}\mathbf{Q}$ or $\mathbf{Q}\mathcal{P}$. The gramians are non-singular if and only if the system $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ is controllable and observable. However, the eigenspaces of (5) essentially yields a projected system that keeps those states that are the most observable and controllable simultaneously. This is related to the fact that for a given state $\hat{\mathbf{x}}$, $\hat{\mathbf{x}}^T \mathcal{P}^{-1} \hat{\mathbf{x}}$ can be viewed as the input energy needed to steer the system from the zero state to $\hat{\mathbf{x}}$, and $\hat{\mathbf{x}}^T \mathbf{Q} \hat{\mathbf{x}}$ can be viewed as the energy of the output resulting from the initial state $\hat{\mathbf{x}}$ (see [38] for a more rigorous discussion).

Efficient algorithms for the large-scale continuous-time case have been derived by exploiting the fact that the gramians can be obtained from the Lyapunov equations

$$\mathbf{A}\mathcal{P} + \mathcal{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{0}, \quad \mathbf{A}^T \mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^T \mathbf{C} = \mathbf{0}.$$

and in the discrete-time case from the solution of the Stein equations:

$$\mathbf{A}\mathcal{P}\mathbf{A}^T - \mathcal{P} + \mathbf{B}\mathbf{B}^T = \mathbf{0}, \quad \mathbf{A}^T \mathbf{Q}\mathbf{A} - \mathbf{Q} + \mathbf{C}^T \mathbf{C} = \mathbf{0}.$$

For example, in [37, 39], an iteration called AISIAD was developed. The algorithm alternates between two coupled Sylvester equations that are approximations to the projected equations

$$\begin{aligned}\mathbf{A}(\mathcal{P}\mathbf{W}) + (\mathcal{P}\mathbf{W})\mathbf{H}^T + \mathbf{B}(\mathbf{B}^T\mathbf{W}) &= \mathcal{P}\mathbf{F}_w, \\ \mathbf{A}^T(\mathcal{Q}\mathbf{V}) + (\mathcal{Q}\mathbf{V})\mathbf{H} + \mathbf{C}^T(\mathbf{C}\mathbf{V}) &= \mathcal{Q}\mathbf{F}_v,\end{aligned}$$

where $\mathbf{F}_w = (\mathbf{I} - \mathbf{W}\mathbf{V}^T)\mathbf{A}^T\mathbf{W}$ and $\mathbf{F}_v = (\mathbf{I} - \mathbf{V}\mathbf{W}^T)\mathbf{A}\mathbf{V}$. Given a current guess $\mathbf{W}$, $\mathbf{V}$, the algorithm proceeds by determining $\mathbf{Z}$ by solving $\mathbf{A}\mathbf{Z} + \mathbf{Z}\mathbf{H}^T + \mathbf{B}(\mathbf{B}^T\mathbf{W}) = \mathbf{0}$, where $\mathbf{H} = \mathbf{W}^T\mathbf{A}\mathbf{V}$ and $\mathbf{V} \leftarrow \mathbf{V}_+$ is obtained from $\mathbf{Z}$. $\mathbf{Y}$ is then determined by solving $\mathbf{A}^T\mathbf{Y} + \mathbf{Y}\mathbf{H}_+ + \mathbf{C}^T(\mathbf{C}\mathbf{V}) = \mathbf{0}$ where $\mathbf{H}_+ = \mathbf{W}^T\mathbf{A}\mathbf{V}_+$, and $\mathbf{W} \leftarrow \mathbf{W}_+$ is obtained from $\mathbf{Y}$ to maintain relations (5).

The AISIAD iteration is effective, but no convergence results are known. These results may be potentially derived from the fact that one can show (see [32, 33]) that $\mathbf{V}$ and $\mathbf{W}$ will span the dominant invariant subspace of the *positive definite generalized eigenvalue problems* $(\lambda\mathcal{P}^{-1} - \mathcal{Q})$ and $(\lambda\mathcal{Q}^{-1} - \mathcal{P})$ if and only if they satisfy the respective “Tracemin” conditions:

$$\begin{aligned}\min \text{trace } \mathbf{V}^T\mathcal{P}^{-1}\mathbf{V}, \quad & \text{s.t. } \mathbf{V}^T\mathcal{Q}\mathbf{V} = \mathbf{I}_n, \\ \min \text{trace } \mathbf{W}^T\mathcal{Q}^{-1}\mathbf{W}, \quad & \text{s.t. } \mathbf{W}^T\mathcal{P}\mathbf{W} = \mathbf{I}_n.\end{aligned}$$

Identifying such subspaces associated with these generalized eigenvalue problems was one of the motivations for our work on Riemannian manifold optimization methods [6, 7, 11] that has, in turn, led to several efficient and robust Riemannian optimization methods applicable to a wide range of problems.

3.3 Model Reduction for Second-Order Systems

Models of mechanical systems are often of the type (1), where $\mathbf{M} = \mathbf{M}^T$, $\mathbf{D} = \mathbf{D}^T$, and $\mathbf{K} = \mathbf{K}^T$ are, respectively, the mass, damping, and stiffness matrices, since this represents the equation of motion of the system. In the Laplace domain, the *characteristic polynomial matrix* $\mathbf{P}(s) := \mathbf{M}s^2 + \mathbf{D}s + \mathbf{K}$ and the transfer function $\mathbf{G}(s) = \mathbf{C}\mathbf{P}^{-1}(s)\mathbf{B}$ appear, and the zeros of $\det(\mathbf{P}(s))$ are also known as the *characteristic frequencies or poles* of the system. Using an extended state $\xi = [\mathbf{x}^T \dot{\mathbf{x}}^T]^T$, this can be linearized to a generalized state-space model $\{\mathcal{E}, \mathcal{A}, \mathcal{B}, \mathcal{C}\}$:

$$\underbrace{\begin{bmatrix} \mathbf{D} & \mathbf{M} \\ \mathbf{M} & \mathbf{0} \end{bmatrix}}_{\mathcal{E}} \dot{\xi} = \underbrace{\begin{bmatrix} -\mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}}_{\mathcal{A}} \xi + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}}_{\mathcal{B}} \mathbf{u}, \quad \mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix}}_{\mathcal{C}} \xi. \quad (6)$$

Since $\mathbf{M}$ is invertible, we can also transform this model to the standard state-space form $\{\mathcal{E}^{-1}\mathcal{A}, \mathcal{E}^{-1}\mathcal{B}, C\}$, where $\mathcal{E}^{-1}\mathcal{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}$ and $\mathcal{E}^{-1}\mathcal{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{B} \end{bmatrix}$. The controllability gramian $\mathcal{P}$ and observability gramian $\mathcal{Q}$ of the state-space model $\{\mathcal{E}^{-1}\mathcal{A}, \mathcal{E}^{-1}\mathcal{B}, C\}$ can be computed via the solution of the Lyapunov equations:

$$\mathcal{E}^{-1}\mathcal{A}\mathcal{P} + \mathcal{P}\mathcal{A}^T\mathcal{E}^{-T} + \mathcal{E}^{-1}\mathcal{B}\mathcal{B}^T\mathcal{E}^{-T} = \mathbf{0}, \quad \mathcal{Q}\mathcal{E}^{-1}\mathcal{A} + \mathcal{A}^T\mathcal{E}^{-T}\mathcal{Q} + C^TC = \mathbf{0}. \quad (7)$$

Moreover, if $\mathbf{C} = \mathbf{B}^T$, the transfer function and model (6) are symmetric and the gramians of the state-space model are defined in terms of the solution $\mathcal{G}$ of a single generalized Lyapunov equation:

$$\mathcal{A}\mathcal{G}\mathcal{E} + \mathcal{E}\mathcal{G}\mathcal{A} + \mathcal{B}\mathcal{B}^T = \mathbf{0}, \quad \mathcal{P} = \mathcal{G}, \quad \mathcal{Q} = \mathcal{E}\mathcal{G}\mathcal{E}. \quad (8)$$

Most model reduction methods use a projection to build the reduced-order model: given a generalized state-space model $\{\mathcal{E}, \mathcal{A}, \mathcal{B}, C\}$, the reduced-order model is given by $\{\mathbf{Z}^T\mathcal{E}\mathbf{V}, \mathbf{Z}^T\mathcal{A}\mathbf{V}, \mathbf{Z}^T\mathcal{B}, C\mathbf{V}\}$, where $\mathbf{Z}$ and $\mathbf{V}$ are matrices of dimension $2N \times k$, with k the order of the reduced system. The widely used balanced truncation technique constructs $\mathbf{V}$ and $\mathbf{Z} := \mathcal{E}^{-1}\mathbf{W}$ such that (5) is satisfied with Σ_+^2 the $k \times k$ matrix containing the largest eigenvalues of $\mathcal{P}\mathcal{Q}$ [15, 16].

This technique cannot be applied directly to a second-order system since, in general, the resulting reduced-order system is no longer a second-order system. There is a need for *adaptive* techniques to find reduced-order models of second-order type. The idea is to choose $k = 2n$ and to restrict the projection matrices $\mathbf{V}$ and $\mathbf{W}$ to have a block diagonal form, where each block is $N \times n$:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{22} \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22} \end{bmatrix}. \quad (9)$$

which turns out to be a *sufficient* condition for obtaining an appropriate reduced-order model. Notice that for the symmetric case, we have automatically $\mathbf{V} = \mathbf{W}$. If we want to impose a block diagonal form, we can, for example, relax the equality conditions in (6).

In this context, the following technical challenges must be addressed: (i) use a relaxation that can be dealt with using Tracemin-like optimization methods, such as

$$\begin{aligned} \min \text{trace } \mathbf{V}_{11}^T[\mathcal{P}^{-1}]_{11}\mathbf{V}_{11} + \mathbf{V}_{22}^T[\mathcal{P}^{-1}]_{22}\mathbf{V}_{22}; \\ \text{s.t. } \mathbf{V}_{11}^T[\mathcal{Q}]_{11}\mathbf{V}_{11} = \mathbf{I}_n, \quad \mathbf{V}_{22}^T[\mathcal{Q}]_{22}\mathbf{V}_{22} = \mathbf{I}_n \end{aligned} \quad (10)$$

(ii) define alternative gramians of dimension $N \times N$ for second-order systems, and analyze their use for model reduction; and (iii) analyze modal approximation based on the polynomial matrix $\mathbf{P}(s)$. One then computes the right and left eigenvectors

$\mathbf{P}(\lambda_i)\mathbf{x}_i = \mathbf{0}$ and $\mathbf{y}_i^T \mathbf{P}(\lambda_i) = \mathbf{0}$ via the linearized eigenvalue problem [13, 14, 17, 18, 20, 21]:

$$\begin{bmatrix} \lambda_i \mathbf{D} + \mathbf{K} & \lambda_i \mathbf{M} \\ \lambda_i \mathbf{M} & -\mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \lambda_i \mathbf{x}_i \end{bmatrix} = \mathbf{0},$$

$$\begin{bmatrix} \mathbf{y}_i^T & \lambda_i \mathbf{y}_i^T \end{bmatrix} \begin{bmatrix} \lambda_i \mathbf{D} + \mathbf{K} & \lambda_i \mathbf{M} \\ \lambda_i \mathbf{M} & -\mathbf{M} \end{bmatrix} = \mathbf{0},$$

and selects n of them to form the projection matrices $\mathbf{V}_{11}$ and $\mathbf{W}_{11}$.

3.4 Time Variance and Adaptivity

Time variance and adaptivity can be either due to actual time variation of the parameters of the system or due to an external iteration used to adaptively tune the reduced-order system. We first explain how to cope with general time-varying systems and show how these methods can also be used for time invariant or for periodic systems. We then discuss systems that are iteratively adapted within an outer tuning loop.

3.4.1 Iterative Schemes

Reduced-order stabilization of a large plant $\mathbf{G}(s)$ amounts to constructing a small order controller $\hat{\mathbf{K}}(s)$ such that the closed-loop transfer function $\phi(\mathbf{G}, \hat{\mathbf{K}}) = \mathbf{G}(s)(\mathbf{I} + \hat{\mathbf{K}}(s)\mathbf{G}(s))^{-1}$ is stable and such that at the same time some performance criterion is optimized.

Most often, a crude low-order (PI) controller is available initially. In order to come up with a simple controller that minimizes an appropriate norm of the transfer function between the inputs and the tracking/focusing errors, respectively, an *iterative procedure* is used. The point of the iteration is to successively reduce the order of the high-order system until the corresponding optimal controller has low enough order as well as the specified performance properties. Thus, at each reduction step, performance is checked by means of estimating a certain relative error. The rationale of this approach is that the controller should be designed for a small order plant $\hat{\mathbf{G}}_i(s)$ using dense matrix techniques. However, this controller will only have a good performance for the original plant $\mathbf{G}(s)$ if the approximation $\hat{\mathbf{G}}_i(s)$ is obtained using the closed-loop error $\|\phi(\mathbf{G}, \hat{\mathbf{K}}_i) - \phi(\hat{\mathbf{G}}_i, \hat{\mathbf{K}}_i)\|$, where $\phi(\mathbf{G}, \mathbf{K}) := \mathbf{G}(s)(\mathbf{I} + \mathbf{K}(s)\mathbf{G}(s))^{-1}$, hence the need for an outer iteration. An important ingredient for making such an outer iteration tractable is contained in our previous work on the construction of stabilizing controllers using invariant subspace ideas and Riccati iterations for large sparse system models [19, 30].

In this context, one needs to analyze the convergence of the closed-loop iteration and find global/local convergence strategies. These ideas can be applied to finding reduced-order models of plants where certain properties are imposed, such as stability, passivity, or second-order model constraints. Finally, this problem can be embedded in a model reduction problem with time-varying coefficients. Therefore, it is possible to take advantage of this fact in order to compute the new plant/controller by *updating* the previous ones.

4 Calibration and Validation of Structural Modeling

Two important aspects of computational simulations are calibration of underlying models (material properties, joint stiffness, etc.) and validation of models using measured excitations and structural response. We have referred to some of these earlier—namely, the Riera curve approach and the SPH calibration through fluid-structure impaction. Experimental validation requires instrumentation of structures, real-time techniques for data acquisition, and data analysis. Each of these represent significant computational challenges associated with implementation of the complex distributed software that senses, communicates, and stores data. This distributed programming infrastructure has exacting requirements on performance (real time), constraints on resources (communication rates, buffers), and guarantees on correctness. In heterogeneous fault-prone environments such as structures subject to external stresses, implementing such distributed programs is a major undertaking. In the remainder of this chapter, we describe efforts aimed at implementing real-time sensing and control.

As applications of sensor networks mature, there is increasing realization of the complexity associated with programming large numbers of heterogeneous devices operating in highly dynamic environments. Much of this complexity stems from the need to account for network state, failures, time and resource constraints, heterogeneity and scalability issues, and dynamic data-driven in-network processing. Even tasks that are conceptually straightforward from the point of view of integrated system specification, such as data acquisition, processing, and aggregation require significant programming effort to support real-time requirements.

Traditional approaches to programming sensor networks rely on “network-enabled” applications for individual nodes that code distributed behavior through explicit messaging. Application development in this framework often requires implementation of system-level primitives or reliance on inflexible or inefficient underlying platforms. Consequently, application development is time and effort intensive and error prone and has limitations with respect to scalability, heterogeneity, and performance. Indeed, software development cycles for complex applications at current state of the art do not keep pace with updates in sensing hardware. In contrast, a macroprogram that codes the behavior of an entire ensemble, as opposed to individual devices, directly specifies the data acquisition and control function.

Through suitable abstractions, it eases programmer burden in dealing with resource constraints, performance optimization, and scalability and adaptability with respect to network and load dynamics. Macroprograms can often be statically verified to enhance robustness and to enforce time and resource constraints.

The presence of a small number of low-cost and low-power yet relatively powerful devices can significantly enhance sensor network performance [36]. Existing development platforms target development for specific device capabilities. This lack of vertical integration across heterogeneous devices results in increased development complexity and programmer effort. Consequently, there is a need for a unified macroprogramming approach that realizes benefits from heterogeneous environments. The dynamic nature of such networks necessitates a runtime environment that can execute on resource-constrained sensor nodes (motes) [22, 23, 27, 36] while potentially scaling to resource-rich nodes.

The goal of supporting easy-to-use high-level abstractions that hide system-level details is often at odds with requirements of low overhead and flexibility. This trade-off is the primary determinant of the design choices associated with a realizable macroprogramming architecture. The underlying architecture must provide a low overhead execution environment for fundamental macroprogram primitives while naturally supporting rich extensions based on the system's operational requirements or application domain. Among other design objectives for a macroprogramming architecture, the reuse of software components across applications is an important engineering parameter, especially as sensor systems become more commonplace [28].

With these motivating objectives, the next section describes COSMOS, an architecture for macroprogramming heterogeneous sensor networks that we have developed [9]. COSMOS is comprised of a lean operating system, mOS, and an associated programming language, mPL. COSMOS supports complex applications, built upon reusable components. In the COSMOS framework, aggregate system behavior is specified in terms of the underlying distributed data processing. This specification consists of functional components (FCs), which provide computing primitives, and an interaction assignment (IA), which specifies distributed dataflow in terms of the FCs. An FC specifies a typed declaration in mPL and the corresponding function, specified in a suitably constrained subset of C. Node capability constraints associated with each FC allow the programmer to effectively and safely utilize heterogeneous devices in the network.

mPL supports the expression of IAs as a fundamental primitive with support for static program verification. Synergistically, mOS provides a low-footprint runtime environment providing dynamic instantiation of FCs and realization of IAs through these instances, along with other requisite OS subsystems. Building on this simple notion of expressing behavior through composition of FCs, the COSMOS architecture supports the ability to easily append rich high-level macroprogramming abstractions in mPL, without modification to the OS. The following two subsections overview the COSMOS runtime environment and the associated mOS system.

4.1 *COSMOS Runtime Environment*

The fundamental design objective for our runtime system is to minimize its processing and memory footprint and to allow performance scaling on resource-rich nodes. A macroprogram specifies connections between FCs and device port instances using abstract asynchronous data channels. Abstract data channels are realized as data queues, whose nodes encapsulate data. Each device port of an FC instance has an individual queue associated with each of its outputs. During application initialization, these queues are attached to the inputs of an instance of the next component, as specified by the IA. The runtime system also handles network data flow transparently. If the communicating instances are not on the same node, output queues connect to the network service FC of the source nodes. An output queue from the network service of the destination node to the succeeding component completes the virtual data channel. The asynchronous semantics ideally suit the network data channel and impose few restrictions on the semantics of the underlying network communication, allowing the use of simple low overhead protocols.

4.2 *mOS Operating System Architecture*

The mOS operating system provides a low overhead implementation of the runtime system for COSMOS-based macroprogramming. Architecturally, mOS consists of a core kernel, which is logically divided into a platform-independent core and hardware specific drivers and routines. The key subsystems of the platform-independent core include the scheduler, timer, dynamic memory manager, dynamic component loading and dataflow setup manager, dataflow API, device abstraction layer, and a system information and control API. We have implemented the mOS operating system on POSIX platforms. On motes, mOS is a full-fledged operating system directly running atop the underlying hardware. On resource-rich nodes (e.g., in our case, Stargate SBC devices and PCs running Linux), mOS sits atop the POSIX layer and provides a transparent environment for executing macroprogram applications, which are, by design, platform independent. To execute a macroprogram on POSIX devices, the mOS management thread loads FCs (which may be a part of the binary of mOS or compiled as individual loadable libraries) and executes the FCs (waiting for inputs) each as an individual thread. Both on POSIX and on the motes, mOS is accessible to the dynamically loaded components through a system call pointer table.

COSMOS is currently in use on a real-world three-story building testbed to validate structural response to ground motion, at the Bowen Labs at Purdue University [1].

5 Concluding Remarks

In this chapter, we have presented methods for DDDAS-based approaches that allow modeling (simulation) for analysis, understanding complex systems' design, and predicting and controlling behaviors during their operational cycle, including actions to mitigate adverse effects and increase the system's tolerance or to provide heads-up for decision support in potential catastrophic events.

Lessons Learned and Future Efforts

Our work on structural modeling and control reveals a number of important lessons and considerations for future design and construction. First, it was demonstrated by our simulations and is now generally accepted that the primary driver of structural failure in WTC was the high temperature that the structural elements were subjected to—eventually leading to their failure. Second, the structural elements had been thermally insulated, but the impact compromised this thermal insulation, contributing to the failure. Third, failure of certain structural elements led to the cascading failure of impact loads of higher floors collapsing on to lower floors. Finally, through inline simulations, such as those enabled by DDDAS, it would have been possible to predict imminent structural failure, potentially reducing loss of life.

More generally, our work demonstrates that simulation and validation of structural models are critical components of understanding and designing robust structures, predicting their response in real time and incorporating passive or active control. Simulations require complex geometric models, material property databases, high-performance simulation engines, and massively parallel computers. Tests on real structures serve two important purposes—they provide material properties and model calibration for simulations and validate simulation results through experiments. Comprehensive validation requires complex instrumentation, multimodal sensing, and real-time data acquisition, typically through wireless sensor networks. Significant progress has been made in all of these computational aspects—leading to real-world deployment of many of these technologies in design and operation. The potential of ubiquitous instrumentation together with efficient modeling computational methods and DDDAS-based approaches increases the effectiveness of such infrastructures for more robust designs and real-time system cognizant optimization operational control and decision support.

Acknowledgments This work is supported by the US National Science Foundation through grants ITR 0325227, OAC 1908691, ACI 0324944, and CCF 2019263.

The authors acknowledge the significant contributions of Profs. Voicu Popescu and Paul Rosen in simulations and animations of the WTC and Pentagon attacks.

This work is a tribute to the many contributions to geometric modeling, design, and simulations of our dear colleague, Prof. Christoph Hoffman.

References

1. Bowen labs. <https://engineering.purdue.edu/CE/BOWEN/>.
2. Purdue university creates animation of September 11, 2001 attack. <https://www.youtube.com/watch?v=NOKJ4ZXgK4Q>.
3. National Institute of Standards and Technology (NIST). Final report of the national construction safety team on the collapses of the world trade center towers, NIST/ NCSTAR 1, 2005.
4. Livermore software corporation. LS-DYNA, 2009.
5. Sandia National Laboratories, international meshing roundtable, 2009.
6. P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region algorithms for the generalized symmetric eigenvalue problem. *Journal of Computational and Applied Mathematics*, 189:274–285, 2006.
7. P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.
8. Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control. SIAM, 2005.
9. Asad Awan, Suresh Jagannathan, and Ananth Grama. Macroprogramming heterogeneous sensor networks using cosmos. *SIGOPS Oper. Syst. Rev.*, 41(3):159–172, mar 2007.
10. M. Badrocke and B. Gunston. *Boeing Aircraft Cutaways*. Osprey Publishing Ltd. Oxford, 1998.
11. C. Baker, P.A. Absil, and K. A. Gallivan. An implicit trust-region method on Riemannian manifolds. *IMA Journal of Numerical Analysis*, 28(4):665–689, 2008.
12. Z. P. Bazant and Y. Zhou. Why did the world trade center collapse? *Journal of Engineering Mechanics*, 128(1) 2–6, 2002.
13. Y. Chahlaoui, K. A. Gallivan, and P. Van Dooren. The H-infty-norm Calculation for Large Sparse Systems. In *International Symposium on Mathematical Theory of Networks and Systems*, 2004.
14. Y. Chahlaoui, K. A. Gallivan, and P. Van Dooren. Calculating the H-infty-norm of Large Sparse Systems via Chandrasekhar Iterations and Extrapolation. In *European Series in Applied and Industrial Mathematics Proceedings*, volume 20, pages 83–92, 2007.
15. Y. Chahlaoui, K. A. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of second order systems. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-scale Systems, Lecture Notes in Computational Science and Engineering*, volume 45, pages 149–172. Springer-Verlag, 2005.
16. Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, and P. Van Dooren. Second-order balanced truncation. *Linear Algebra and its Applications*, 415(2):373–384, 2006. Special Issue on Order Reduction of Large-Scale Systems.
17. P. Van Dooren, K. A. Gallivan, and P.-A. Absil. H2-optimal Model Reduction with Higher Order Poles. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2738–2753, 2010.
18. P. Van Dooren, K. A. Gallivan, and P.-A. Absil. H2-optimal Model Reduction of MIMO Systems. *Applied Mathematics Letters*, 21(12):1267–1273, December 2008.
19. K. Gallivan, X. Rao, and P. Van Dooren. Singular Riccati equations stabilizing large-scale systems. *Linear Algebra and Its Applications*, 415:359–372, 2006.
20. K. A. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction and Sylvester equations. proceedings. In *International Symposium on the Mathematical Theory of Networks and Systems*, pages 1652–1664, 2006.
21. Kyle Gallivan, A. Vandendorpe, and Paul Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Analysis Applications*, 26:328–349, 01 2004.
22. Chih-Chieh Han, Ram Kumar Rengaswamy, Roy Shea, Eddie Kohler, and Mani Srivastava. Sos: A dynamic operating system for sensor networks. In *Proc. of MobiSys*, June 2005.
23. Jason Hill, Robert Szweczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proc. of ASPLOS-IX*, November 2000.
24. Christoph M. Hoffmann, Voicu Popescu, Sami Kilic, and Mete Sozen. Modeling, simulation and visualization: the pentagon on September 11th. *Comput. Sci. Eng.*, 6(1):52–60, 2004.

25. A. Irfanoglu and C. Hoffmann. An engineering perspective of the collapse of WTC-1. *ASCE Journal on Performance of Constructed Facilities*, 22,62, 2008.
26. M. Karim and M. Fatt. Impact of the Boeing 767 aircraft into the world trade center. *Journal of Engineering Mechanics*, 131(10) 1066–1072, 2005.
27. Ting Liu and Margaret Martonosi. Impala: a middleware system for managing autonomic, parallel sensor systems. In *Proc. of PPOPP*, June 2003.
28. Bertrand Meyer. Applying design by contract. *IEEE Computer*, 25(10):40–51, October 1992.
29. Y. Omika, E. Fukuzawa, N. Koshika, H. Morikawa, and R. Fukuda. Structural responses of world trade center under aircraft attacks. *Journal of Structural Engineering*, 131(1), 6–15, 2005.
30. X. Rao, K. Gallivan, and P. Van Dooren. Efficient stabilization of large scale dynamical systems. In *Proceedings IEEE Conference on Computer Aided Control Systems Design*, 2000.
31. J. D. Riera. On the stress analysis of structures subjected to aircraft impact forces. *Nuclear Engineering and Design*, 8:415–426, 1968.
32. A. Sameh and Z. Tong. The Trace Minimization method for the symmetric generalized eigenvalue problem. *Journal of Computational and Applied Mathematics*, 2000.
33. A. H. Sameh and J. A. Wisniewski. A trace minimization algorithm for the generalized eigenvalue problem. *SIAM Journal of Numerical Analysis*, 1982.
34. T. Sugano, H. Tsubota, Y. Kasai, N. Koshika, H. Ohnuma, W. von Riesenmann, D. Bickel, and M. Parks. Full-scale aircraft impact test for evaluation of impact force. *Nuclear Engineering and Design*, 140:373–385, 1993.
35. T. Wierzbicki and X. Teng. How the airplane wing cut through the exterior columns of the world trade center. *International Journal of Impact Engineering*, 2003.
36. Mark Yarvis, Nandakishore Kushalnagar, Harkirat Singh, Anand Rangarajan, York Liu, and Suresh Singh. Exploiting heterogeneity in sensor networks. In *Proc. of INFOCOM*, March 2005.
37. Y. K. Zhou. *Numerical Methods for Large Scale Matrix Equations with Applications in LTI System Model Reduction*. PhD thesis, Department of Computational and Applied Mathematics, Rice University, 2002.
38. Y. K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1995.
39. Y. K. Zhou and D. C. Sorensen. Approximate implicit subspace iteration with alternating directions for LTI system model reduction. Technical report, Rice University, 2002.