
**MIT:** Douglas Allaire, Karen Willcox, Benjamin Peherstorfer, Daniel Butnaru
**UT Austin:** George Biros, Omar Ghattas
**Aurora Flight Sciences:** Jeffrey Chambers, David Kordonowy

International Conference on Computational Science
Omaha, Nebraska
June 4, 2012
Motivation and Goals

A self-aware aerospace vehicle can dynamically adapt the way it performs missions by gathering information about itself and its surroundings and responding intelligently.

Research Goal: Create a multifidelity framework for the DDDAS paradigm.

- DDDAS process draws on multiple modeling options and data sources to evolve models, sensing strategies, and predictions as the flight proceeds
- Dynamic data inform online adaptation of structural damage models and reduced-order models
- Dynamic guidance of sensing strategies
- Dynamic, online management of multifidelity structural response models and sensor data, ensuring that predictions have sufficient confidence

Leading to dynamic health-aware mission re-planning with quantifiable benefits in reliability, maneuverability and survivability.
• Resource allocation problem statement
• Global sensitivity analysis
• Global sensitivity analysis for imperfect information sources
• Incorporation/impact of information source correlation
• Next steps
**Problem Statement**

Obtain the minimum variance estimate of a quantity of interest given a set of resources, a time constraint, and a computational constraint.

**Terminology**

*Resource*: An artifact (e.g., model, sensor) that can be called upon to provide an estimate of the mean and variance of a given quantity of interest.

*Policy*: The serial employment of an ordered set of resources used to estimate the mean and variance of a given quantity of interest.
Resource Allocation – Optimization Problem

\[
\begin{align*}
\min_{P} & \quad \text{var}(Q) \\
\text{s.t.} & \quad C(P) \leq c \\
& \quad T(P) \leq \tau
\end{align*}
\]

where \( P \in \mathcal{P} \) and \( \mathcal{P} \) is the set of all policies, an element of \( \mathcal{P} \) is an ordered set of resources \( P_i = (R_{i_1}, \ldots, R_{i_s}) \), where \( R_{i_j} \in \mathcal{R} \) and \( \mathcal{R} \) is the set of available resources,

\( C(P) \) is the computational cost of policy \( P \)

\[
C(P_i) = \max_j C(R_{i_j})
\]

\( T(P) \) is the time cost of policy \( P \)

\[
T(P_i) = \sum_j T(R_{i_j})
\]
$R_1$ : Sensor data
$R_2$ : Physics-based model

Initial Estimates
$\mathbb{E}[Q], \text{var}(Q)$

$\mathbb{E}[Q \mid R_1], \text{var}(Q \mid R_1)$
$\mathbb{E}[Q \mid R_1, R_2], \text{var}(Q \mid R_1, R_2)$
$\mathbb{E}[Q \mid R_2], \text{var}(Q \mid R_2)$
$\mathbb{E}[Q \mid R_2, R_1], \text{var}(Q \mid R_2, R_1)$

Decision $| P_1$
Decision $| P_2$
Decision $| P_3$
Decision $| P_4$

Time $\tau$
$R_1$ : Sensor data

$R_2$ : Physics-based model

Policy 1: $(R_1)$

Initial Estimates

$\mathbb{E}[Q | R_1], \text{var}(Q | R_1)$

$\mathbb{E}[Q | R_2], \text{var}(Q | R_2)$

$\mathbb{E}[Q | R_1, R_2], \text{var}(Q | R_1, R_2)$

$\mathbb{E}[Q | R_2, R_1], \text{var}(Q | R_2, R_1)$

0  Time  $\tau$
Resource Allocation – Two Resources

- \( R_1 \): Sensor data
- \( R_2 \): Physics-based model

Initial Estimates:
- \( \mathbb{E}[Q], \text{var}(Q) \)

Policy 2: \((R_2)\)

\[
\begin{align*}
\mathbb{E}[Q | R_1], \text{var}(Q | R_1) & \quad \mathbb{E}[Q | R_1, R_2], \text{var}(Q | R_1, R_2) \\
\mathbb{E}[Q | R_2], \text{var}(Q | R_2) & \quad \mathbb{E}[Q | R_2, R_1], \text{var}(Q | R_2, R_1)
\end{align*}
\]
$R_1$ : Sensor data

$R_2$ : Physics-based model

Policy 3: $(R_1, R_2)$

$\mathbb{E}[Q \mid R_1], \text{var}(Q \mid R_1)$

$\mathbb{E}[Q \mid R_1, R_2], \text{var}(Q \mid R_1, R_2)$

$\mathbb{E}[Q \mid R_2], \text{var}(Q \mid R_2)$

$\mathbb{E}[Q \mid R_2, R_1], \text{var}(Q \mid R_2, R_1)$

Initial Estimates

$\mathbb{E}[Q], \text{var}(Q)$

$\tau$
$R_1$: Sensor data

$R_2$: Physics-based model

Policy 4: $(R_2, R_1)$
$R_1$: Sensor data

$R_2$: Physics-based model

Policy 1 is the only option in this case
Resource Allocation for 2\textsuperscript{nd} Order Statistics

• Solution methods
  – Brute force
  – Greedy (best first, fastest first)
  – Cast as a dynamic program?

• Requirements
  – Resources that can estimate mean and variance of the quantity of interest
  – Identification of variance reduction that results from employing a resource

• Approach
  – Adapt state of the art global sensitivity analysis methods to provide variance reduction information
  – Account for correlated information sources
Consider a square integrable function
\[ y = f(x_1, \ldots, x_n), \]
where \( x_z \in [0,1] \quad \forall z \in \{1,\ldots,n\} \)

High Dimensional Model Representation

\[
f(x) = f_0 + \sum_i f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) + \cdots + f_{12\ldots n}(x_1, x_2, \ldots, x_n)
\]

ANOVA Constraint

\[
\int_0^1 f_{i_1,\ldots,i_s}(x_{i_1}, \ldots, x_{i_s}) \, dx_p = 0 \quad \text{for} \quad p = i_1, \ldots, i_s
\]

Variance Decomposition

\[
V = \sum_i V_i + \sum_{i<j} V_{ij} + \cdots + V_{12\ldots n}
\]

Sensitivity Indices

\[
S_{i_1,\ldots,i_s} = \frac{V_{i_1,\ldots,i_s}}{V}
\]
Consider a function

\[ y = f(x_1, \ldots, x_n), \]

Law of total variance

\[ \text{var}(Y) = \mathbb{E}[\text{var}(Y | X_i)] + \text{var}(\mathbb{E}[Y | X_i]) \]

Main effect sensitivity index

\[ S_i = \frac{\text{var}(\mathbb{E}[Y | X_i])}{\text{var}(Y)} \]

Thus

\[ S_i \text{ var}(Y) = \text{var}(\mathbb{E}[Y | X_i]) \]

Define

\[ \delta = \frac{\text{var}(X'_i)}{\text{var}(X^o_i)} \]

Remaining variance after a resource is employed

Original variance

Define

\[ \text{adj}S_i(\delta) = \frac{\text{var}(Y^o)S^o_i - \mathbb{E}[\text{var}(Y'(\delta))S'_i(\delta)]}{\text{var}(Y^o)} \]
$f(X_1, X_2, X_3) = 100X_1 + 4\exp(X_2) + 350\sin(X_3)$

$X_1, X_2, X_3 \sim \mathcal{N}(0, 4)$
Once we have generated an estimate from a resource, we can fuse the new information with our previous estimates.

In the case of sensors and models, estimates are frequently assumed independent.

Two Source Update

\[
\mu = \frac{\sigma_2^2 \mu_1 + \sigma_1^2 \mu_2}{\sigma_1^2 + \sigma_2^2}
\]

\[
\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}
\]

(Adapted from March, 2010)
Model Correlation

- What if the sources are not independent?
- Owing to similar physics and underlying data, models are typically correlated
- Following Winker 1981

Two Source Update

\[
\mu = \frac{\left(\sigma_2^2 - \rho \sigma_1 \sigma_2\right) \mu_1 + \left(\sigma_1^2 - \rho \sigma_1 \sigma_2\right) \mu_2\right]}{\sigma_1^2 + \sigma_2^2 - 2 \rho \sigma_1 \sigma_2}
\]

\[
\sigma^2 = \frac{(1 - \rho^2) \sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2 - 2 \rho \sigma_1 \sigma_2}
\]

General Update

\[
\mu = e^T \Sigma^{-1} z / e^T \Sigma^{-1} e
\]

\[
\sigma^2 = 1 / e^T \Sigma^{-1} e
\]

where \(e = [1, \ldots, 1]^T\)

\[z = [\mu_1, \ldots, \mu_k]^T\]
Model Fusion

Combine similar models

Trust models with lower variance

Model 1  Model 2  Combined Estimate

Increasing correlation
Example – Effect of Correlation

\[ f(X_1, X_2, X_3) = 100X_1 + 4\exp(X_2) + 350\sin(X_3) \]

\[ X_1, X_2, X_3 \sim \mathcal{N}(0,4) \]
Conclusions/Future Work

• Developed a sensitivity analysis methodology for dynamic resource allocation optimization
  – Deals with 2\textsuperscript{nd} order statistics (Bayes Linear)
  – Incorporates information source correlation

• Next Steps
  – Efficient online solution of resource allocation problem for the case of 2\textsuperscript{nd} order statistics
  – Extension of the resource allocation problem to a more general setting
Adaptive Structural Response Model

Sensed damage adapts vehicle state model
- Local capability informs/updates airframe capability model
- Degradation models predict damaged panel residual strength and stiffness, effects of loose fasteners, fretted fastener hole, disbonded surface, etc.

Model generation aspects
- Finite Element Model (multifidelity)
- Design details at elements (e.g., bonded joint) and borders (e.g. fastener)
- Material limits for safety (e.g., strength allowable) used to define flight limits
- Includes degradation modes
References

Example Structural Response Model

- Simplified Panel
  - (18” x 18”)

- Full Aircraft Model
  - (Aurora Orion UAV)
Dynamic Data-Driven Methods for Self-Aware Aerospace Vehicles

Dynamically evolving DDDAS process: 
- Infer—Predict—Plan—Act—

Decision-making needs are informed by current quantity of interest estimates

Quantities of Interest
- Mission Plan
- Flight Limits
- Vehicle State

Models
- Adaptive Structural Response Models
- Information Fusion Models
- Planning Models

Decision-making rules determine how to allocate resources in the inference/prediction/planning process.

Models drive adaptive sensing

Environmental data inform planning models

Quantities of interest drive adaptive sensing

Sensors
- Sensors: struct. health, stress/strain, pressure
- Sensors: IMS/GPS, temperature

Vehicle data drive adaptation of multifidelity models.
Sensed damage adapts vehicle state model
- Local capability informs/updates airframe capability model
- Degradation models predict damaged panel residual strength and stiffness, effects of loose fasteners, fretted fastener hole, disbonded surface, etc.

Model generation aspects
- Finite Element Model (multifidelity)
- Design details at elements (e.g., bonded joint) and borders (e.g. fastener)
- Material limits for safety (e.g., strength allowable) used to define flight limits
- Includes degradation modes
Data Incorporation into Reduced-Order Models

- High-fidelity model runs are conducted during the offline phase
- Surrogate models are produced from high fidelity model data
- Estimates required online in regions of high uncertainty incorporate sensor data
- A fused model formed by the surrogate model and sensor data is generated
Multifidelity Models Incorporate Sensor Data

- High-fidelity model runs offline using historical sensor data to update structure based on damage/ degradation
  - High-fidelity model remeshes FEM and updates material allowables
- Medium- and low-fidelity models run online using real-time data
  - Available system resources and degradation types determine level of FEM modification
- Each level of fidelity incorporates sensor data differently to determine effects
  - **High-fidelity** = insertion of detailed damage models into remeshed structural model based on sensed material change
  - **Medium-fidelity** (shown) = course model element deletion or modification including local remeshing or user subroutines as required
  - **Low-fidelity** = fast, closed-form physics- and data-based models based on fixed model

---

**Ex: Medium-fidelity model of a wing section, no damage**

**Sensors indicate severe damage (open hole) at two locations, elements removed**

**Updated damage extent determines additional elements removed or modified**
Adaptive Multifidelity Structural Response Models

- **High Fidelity Response Model: Finite Element Method (FEM)**
  - Set of elements $e$ interconnected at nodes
  - Element stiffness matrices $k^e$ used to solve nodal displacements
  - Strain-displacement matrix $B$ to transform nodal displacements to strains
  - Elasticity matrix $E$ that transforms effective strains to stresses

- **Unique element properties**
  - Stiffness, Strength, thickness, layup (for composite materials), constituent material orientation(s), failure modes, etc.
    - Examples of failure modes: mechanical overload (maximum stress/strain criteria), buckling, fatigue, yield

- **Major advantages to using FEM approach:**
  - Use of existing solvers for complex system with external work applied
  - Finite discretization of sensor information and degradation modes allows discrete adaptation of individual elements
    - Sensor information can be applied to specific elements to modify the stiffness or strength properties or modify the mesh (e.g. remesh, element elimination) of a subset of elements to reflect the inferred change
Example Structural Response Model

- Simplified Panel
  - (18” x 18”)

- Full Aircraft Model
  - (Aurora Orion UAV)
Vehicle maneuver library defines motion primitives used to plan flight, set flight limits, and create missions.

<table>
<thead>
<tr>
<th>Motion #</th>
<th>Motion Primitive</th>
<th>Constraint</th>
<th>Parametrized by different</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Steady level</td>
<td>$\gamma = 0$</td>
<td>$v$ (constant)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\psi = constant$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\phi = 0$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Steady climb/descent</td>
<td>$\gamma = constant$</td>
<td>$L/w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\psi = constant$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\phi = 0$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Steady level turn</td>
<td>$\gamma = 0$</td>
<td>$L/w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\dot{\psi} = constant$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\phi = constant$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$v = constant$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Steady climbing/descending turn</td>
<td>$\gamma = constant$</td>
<td>$L/w$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\dot{\psi} = constant$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\phi = constant$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$v = constant$</td>
<td></td>
</tr>
</tbody>
</table>
Self-Aware Vehicle Framework

1. Initialize
   1. Read finite element model
   2. Generate nominal node map and element strength capability
   3. Pre-determine vehicle capability using maneuver library

2. Monitor (Inference)
   1. Receive measured damage information (type, size, location)
   2. Receive temperature information
   3. Modify local strength capability if necessary and determine if changes warrant update to vehicle state
   4. Modify local stiffness or remesh if necessary and determine if changes warrant update to vehicle state

3. Vehicle State
   1. Export and run updated finite element model if required/requested
   2. Import finite element model stresses
   3. Use current maneuver to modify vehicle state due to usage

4. Capability (Prediction)
   1. Calculate finite element model failure indices against load cases
   2. Notch vehicle library load cases based on reduced capability

5. Evaluation (Planning)
   1. Estimate Remaining Useful Life using simulated load spectrum/heuristic function
   2. Output cost/objective function to mission planner
Defining Sensors

- “Truth data” to replace physics (i.e. strains, pressures, temperatures, damage)
- Sensor models will sense truth data with possible uncertainties built in
  - E.g. include sensor failures, drifts, errors
- Model parameters are adjustable “in-situ”
• Online estimates of quantities of interest must rely on lower fidelity capabilities
  – Reduced-order models (ROMs)
  – Data-fit models
  – Lower fidelity physics-based models
• Sensors may detect parameter changes
• Questions
  – To what extent can a ROM be employed outside of the parameter range it was built for?
  – Can we detect when a ROM becomes obsolete?
  – Can we restore a ROM online?
ROM Restoration

• Assume sensors provide partial snapshot
• Replace Approach
  – Replace ROM result with information from sensors
  – Reconstruct ROM using iterative POD
• Gappy Approach
  – Use Gappy POD to reconstruct full snapshot with partial sensor information
  – Reconstruct ROM using iterative POD
• Enrich Approach
  – Add snapshots outside of the original parameter range
• Normal Approach
  – Do not modify the ROM
Example Problem: Plate

- Clamped, square isotropic plate with a uniform pressure load
- The deflection of the plate acting on by a pressure loading

\[ \nabla^2 (D \nabla^2 w) = p, \text{ with } D = \frac{Eh^3}{12 (1 - \nu^2)}, \]

where \( E \) is the modulus of elasticity, \( \nu \) is Poisson’s ratio, and \( h \) is the plate thickness
Example Problem: Burger’s Equation

- Unsteady Burger’s equation

$$\partial_t u + u \nabla u = \nu \Delta u$$

viscosity parameter $\nu$
Burger’s Equation Example Results