Postprint of Research on Bird Strike Resistance of Civil Aircraft Sidewall Panels Based on Rivet Dynamic Model

Authors: Chen Xiaofeng

Date: 2025-02-13T00:00:00+00:00

Abstract

To investigate the bird strike resistance performance of civil aircraft sidewall panel structures, a dynamic elastoplastic damage model for aerospace rivets was established based on Abaqus. The accuracy of the dynamic model for rivet fasteners and the bird strike dynamic response of civil aircraft sidewall panels were studied through simulations of rivet dynamic loading tests, civil aircraft sidewall panel bird strike tests, and simulation analysis. The results indicate that: under bird strike impact, civil aircraft sidewall panel structures primarily exhibit extensive failure of rivet fasteners leading to stiffener detachment. The structural longitudinal frames, stiffeners, and skin did not experience tearing failure. The Abaqus-based dynamic elastoplastic damage model for rivets can effectively describe the strain rate strengthening effect under dynamic loading, can relatively accurately simulate the damage and failure behavior under tension-shear coupling loading, and the application of this model can accurately simulate the failure modes of rivet fasteners under bird strike impact, thereby obtaining strain results that are relatively consistent with experimental results. It is evident that this rivet model can provide an Abaqus-based modeling method for rivet fasteners for research on bird strike resistance problems in civil aircraft structures.

Full Text

Preamble

Sequence-to-sequence (S2S) models represent a foundational architecture in modern deep learning, particularly for tasks involving sequential data transformation. The standard encoder-decoder framework compresses variable-length input sequences into fixed-dimensional context vectors, which are then used by the decoder to generate output sequences. However, this compression inevitably leads to information loss, especially for long sequences, motivating the development of attention mechanisms that enable dynamic access to encoder states during decoding.

Early S2S implementations relied on recurrent neural networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures. The attention mechanism in these models computes alignment scores between the current decoder hidden state and all encoder hidden states, producing a weighted context vector. Two primary formulations emerged: additive attention, as proposed by Bahdanau et al., which employs a feed-forward network for score computation, and multiplicative attention, introduced by Luong et al., which offers improved computational efficiency through matrix multiplication.

The mathematical formulation of attention can be expressed as follows. Given encoder hidden states $\mathbf{h}_i$ and decoder hidden state $\mathbf{s}_t$, the alignment scores are computed as:

$$e_{ij} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_i + \mathbf{U}_a \mathbf{h}_j)$$

The attention weights are obtained through softmax normalization:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})}$$

The context vector is then calculated as the weighted sum of encoder states:

$$\mathbf{c}_i = \sum_{j=1}^{n} \alpha_{ij} \mathbf{h}_j$$

For multi-head attention mechanisms, the model projects queries, keys, and values into multiple subspaces:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}^V$$

Each head computes attention independently:

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

The outputs are concatenated and linearly transformed:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)\mathbf{W}^O$$

The Transformer architecture revolutionized this paradigm by replacing recurrence entirely with self-attention. Each layer computes attention between all positions in the sequence, capturing long-range dependencies through parallelizable operations. The self-attention mechanism allows each token to attend to every other token, generating context-aware representations without sequential processing constraints.

Computational complexity remains a critical consideration. Standard attention mechanisms exhibit quadratic complexity $\mathcal{O}(n^2)$ with respect to sequence length n, as each token attends to all others. Various optimizations have been proposed, including sparse attention patterns, low-rank approximations, and linear attention mechanisms that reformulate the softmax operation to achieve sub-quadratic complexity.

Recent advances extend attention mechanisms beyond natural language processing to computer vision, speech recognition, and multi-modal applications. Cross-attention enables information flow between different modalities, while specialized positional encoding schemes preserve sequential information in non-recursive architectures. The evolution from RNN-based attention to pure attention models demonstrates the power of selective information access in deep learning systems.

Attention Mechanisms in Sequence-to-Sequence Models

The attention mechanism fundamentally addresses the bottleneck problem of compressing entire sequences into single vectors. By allowing decoder states to attend differentially to encoder states, the model learns to focus on relevant input segments during each generation step. This soft-alignment approach has proven particularly effective for long sequences where traditional fixed-context methods fail.

Formally, the attention score computation involves several key transformations. The energy function e_{ij} measures compatibility between decoder state $\mathbf{s}_i$ and encoder state $\mathbf{h}_j$:

$$e_{ij} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a \mathbf{h}_j)$$

where $\mathbf{W}_a$, $\mathbf{U}_a$, and $\mathbf{v}_a$ are learnable parameters. The attention weights α_{ij} are derived via softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

producing a probability distribution over source positions. The context vector $\mathbf{c}_i$ becomes:

$$\mathbf{c}_i = \sum_{j=1}^{T} \alpha_{ij} \mathbf{h}_j$$

In practice, multi-layer networks often employ different attention variants per layer. Some layers may use dot-product attention:

$$\text{score}(\mathbf{s}_i, \mathbf{h}_j) = \frac{\mathbf{s}_i^\top \mathbf{h}_j}{\sqrt{d_k}}$$

while others implement location-based attention that considers previous attention weights:

$$e_{ij} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a \mathbf{h}_j + \mathbf{V}_a \mathbf{f}_{i,j})$$

The versatility of attention mechanisms extends to handling different input and output lengths, enabling applications in machine translation, text summarization, and dialogue systems. Modern implementations frequently incorporate coverage mechanisms to prevent repeated attention to the same input positions, addressing the "over-attention" problem in generation tasks.

Transformer-based architectures further generalize attention through self-attention layers where queries, keys, and values originate from the same sequence. The multi-head formulation captures diverse dependency patterns:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)\mathbf{W}^O$$

with each head using different projection matrices:

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

The final representation combines all heads through layer normalization and residual connections:

$$\mathbf{h}' = \text{LayerNorm}(\mathbf{h} + \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}))$$

Layer normalization and residual connections stabilize training, while position-wise feed-forward networks process the attended representations. This architecture achieves state-of-the-art performance across numerous tasks, establishing attention as the dominant mechanism in contemporary deep learning.

Computational optimizations continue to emerge, including kernel-based attention, reversible layers, and gradient checkpointing to manage memory requirements. The theoretical underpinnings of attention mechanisms reveal connections to kernel methods and probabilistic graphical models, suggesting further avenues for improvement in efficiency and interpretability.

Note: Figure translations are in progress. See original paper for figures.