Behavioral-level performance modeling of analog and mixed-signal system using support vector machines

Xianqiang Ren and Tom J Kazmierski
University of Southampton, UK
{xr03r,tjk}@ecs.soton.ac.uk
Introduction

Selection of the performance modeling method is one of the key elements in AMS synthesis

- Fuzzy logic and neural network method
- Symbolic method
- Simulation-based methods
Introduction

An SVM-based approach has been introduced recently

- An alternative to fuzzy-logic and neural networks
- Limited to classical 'good-bad' analysis
- Full space regression model construction
Support vector machines

• Structured risk minimization: considers both training errors and separation hyperplane

• Right choice of the kernel function can simplify the computational cost of SVM.

1: Linear Kernel: \( k(x, x') = \bar{x} \cdot \bar{x}' \)
2: Polynomial Kernel: \( k(x, x') = (\gamma \bar{x} \cdot \bar{x}' + r)^d \)
3: Radial Basis Function Kernel (RBF):
   \( k(x, x') = \exp(-\gamma ||\bar{x} - \bar{x}'||^2), \gamma > 0 \)
4: Sigmoid Kernel: \( k(x, x') = \tanh(\gamma \bar{x} \cdot \bar{x}' + r) \)
Support vector machines

Compared to traditional neural network models, SVM models have:

• Superior generalization capability
• Higher execution speed

However, suitable modeling techniques are essential to utilise the potential offered by SVMs.
**Linearly graded performance models**

To provide effective and accurate modeling of AMS systems

- Partition the entire performance space into sub-spaces
- Construct regression models of each sub-space

Reduces the complexity of performance space exploration.
**Data**
- Design space
- Performance space

**Operations**
- Simulation
- Grading
- Classification training
- Regression training

---

**Linearity of performance models**

- m-dimensional design space
- Design parameter
  - Performance parameter

- Simulation

- Grading

- Classification training
- Regression training

- Grouped SVM models for each of the performance parameter
  - $P_1$ models:
    - Classification model
    - Regression model for each of the class
  - $P_2$ models:
    - Classification model
    - Regression model for each of the class
  - $P_m$ models:
    - Classification model
    - Regression model for each of the class

---

School of Electronics and Computer Science

University of Southampton, UK
Linearly graded performance models

• **Design space**
  each design parameter represents a dimension

• **Performance space**
  each performance parameter represents a dimension

• **Grading**
  Each performance parameter is automatically graded into sub-ranges.
  Combined sub-ranges form hypercubes in the performance space and are linked with corresponding subspaces in the design space.
Interim results

- Class matrix related with each performance parameter
- Classification models
- Regression models

Results

- Grouped classification and regression models for each parameter.
Linearly graded performance models

- **Classification training**
  uses points in the design space and their corresponding classes

- **Regression training**
  uses points in the design and performance spaces and their corresponding classes

- **Grouped models**
  each performance parameter has a set of models including the classification models for classes and the regression models for each of the class
Model construction process

• Major operations
  simulation
  grading
  training
  testing

• Balanced Data
  Grading (BDG)
  algorithm

• Training algorithm
Model construction process

Grading with the BDG algorithm

• Grading along a parameter dimension with approximately similar amount of data material in the performance space

• Algorithm avoids sparse or over-concentrated distributions of data

The process starts with the user provided grading requirements and calculates balanced grading vectors;
Performance model construction process

SVM training: construction of the models

- Grid search
  - coarse grid search (CGS) with low grid resolution
  - refined grid search (RGS) with high grid resolution
    - to enhance the model construction efficiency

- Cross-validation
  - to enhance accuracy and generality (occurs also in neural networks)

Testing: validation of the models
Case study – a 2nd order Sigma Delta Modulator

- Design space
  - amplifier gains, etc.

- Performance space
  - SNR
  - Output dynamic range (DR)
  - Stability
  - Integrator DR (INT1, INT2)
## Results - kernel comparison

<table>
<thead>
<tr>
<th>SVM method</th>
<th>Sample parameters</th>
<th>Linear kernel</th>
<th>RBF kernel</th>
<th>Sigmoid kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification</strong></td>
<td>Stability - accuracy</td>
<td>82.98%</td>
<td>99.58%</td>
<td>99.05%</td>
</tr>
<tr>
<td></td>
<td>Stability - CPU time</td>
<td>16:48:09</td>
<td>00:34:17</td>
<td>08:22:00</td>
</tr>
<tr>
<td></td>
<td>SNR - accuracy</td>
<td>82.01%</td>
<td>98.77%</td>
<td>98.22%</td>
</tr>
<tr>
<td></td>
<td>SNR - CPU time</td>
<td>22:30:40</td>
<td>00:53:54</td>
<td>17:13:37</td>
</tr>
<tr>
<td><strong>Regression</strong></td>
<td>SNR - CPU time</td>
<td>02:48:31</td>
<td>31:49:49</td>
<td>17:13:37</td>
</tr>
</tbody>
</table>
Results - BDG

- SNR: 58.3 dB vs. 55 dB
- Dynamic Range (DR): 55.2 dB vs. 55 dB
- Amplitude distributions for INT1 and INT2
  - INT1: 0.29, 0.35, 0.41
  - INT2: 0.56, 0.71, 0.84

School of Electronics and Computer Science
University of Southampton, UK
Results - training

stability

INT1_0.35
Results - training

SNR

82.5  82  81.5  81  80.5  80  79.5  79  78.5  78  77.5

85  84.5  84  83.5  83  82.5  82  81.5  81  80.5  80  79.5  79  78.5  78  77.5

DR

97.5  97  96.5  96  95.5  95  94.5  94  93.5  93  92.5  92  91.5  91  90.5  90  89.5  89  88.5  88  87.5  87  86.5  86  85.5  85  84.5  84  83.5  83  82.5  82  81.5  81  80.5  80  79.5  79  78.5  78  77.5

INT1

97.5  97  96.5  96  95.5  95  94.5  94  93.5  93  92.5  92  91.5  91  90.5  90  89.5  89  88.5  88  87.5  87  86.5  86  85.5  85  84.5  84  83.5  83  82.5  82  81.5  81  80.5  80  79.5  79  78.5  78  77.5

INT2

100  99.5  99  98.5  98  97.5  97  96.5  96  95.5  95  94.5  94  93.5  93  92.5  92  91.5  91  90.5  90  89.5  89  88.5  88  87.5  87  86.5  86  85.5  85  84.5  84  83.5  83  82.5  82  81.5  81  80.5  80  79.5  79  78.5  78  77.5

Stability


CGS  RGS

| CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  | RGS  | CGS  |

Total time  47:50:46  02:18:04

School of Electronics and Computer Science

University of Southampton, UK
Results - training

<table>
<thead>
<tr>
<th>Approach</th>
<th>Computational cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linearly graded</td>
<td>50:08:50 (classification)</td>
</tr>
<tr>
<td></td>
<td>49:27:57 (regression CGS)</td>
</tr>
<tr>
<td>Full-space analysis</td>
<td>190:44:11 (CGS only)</td>
</tr>
</tbody>
</table>
# Results - testing

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Testing1 (Classification accuracy)</th>
<th>Testing2 (Classification accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>66.72%</td>
<td>78.73%</td>
</tr>
<tr>
<td>DR</td>
<td>74.5%</td>
<td>62.7%</td>
</tr>
<tr>
<td>INT1</td>
<td>87.7%</td>
<td>97.1%</td>
</tr>
<tr>
<td>INT2</td>
<td>78%</td>
<td>99.6%</td>
</tr>
<tr>
<td>Stability</td>
<td>98.4%</td>
<td>100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Testing1 (Regression accuracy (mean squared error dB))</th>
<th>Testing2 (Regression accuracy (mean squared error dB))</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR_55.3</td>
<td>-5</td>
<td>-26</td>
</tr>
<tr>
<td>SNR_45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DR_55.2</td>
<td>-11</td>
<td>-6</td>
</tr>
<tr>
<td>DR_45</td>
<td>-5</td>
<td>-5</td>
</tr>
<tr>
<td>INT1_0.29</td>
<td>-43</td>
<td>-47</td>
</tr>
<tr>
<td>INT1_0.35</td>
<td>-37</td>
<td>-50</td>
</tr>
<tr>
<td>INT1_0.41</td>
<td>-38</td>
<td>-</td>
</tr>
<tr>
<td>INT1_0.5</td>
<td>-28</td>
<td>-20</td>
</tr>
<tr>
<td>INT2_0.56</td>
<td>-28</td>
<td>-29</td>
</tr>
<tr>
<td>INT2_0.71</td>
<td>-26</td>
<td>-</td>
</tr>
<tr>
<td>INT2_0.84</td>
<td>-14</td>
<td>-</td>
</tr>
<tr>
<td>INT2_1.5</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Conclusion and further work

• A new concept for performance analysis has been introduced: linearly graded performance models
• A suitable modeling process has been developed
• Demonstrated by a case study of a difficult AMS system: 2\textsuperscript{nd} order SDM.
• AMS performance optimization is possible – next step towards a general AMS synthesis system.
• More work needed on grading algorithms
• More work to explore the trade-off between model construction computational cost and prediction accuracy