Fast-Accurate Non-Polynomial Metamodeling for nano-CMOS PLL Design Optimization

O. M. Garitselov\textsuperscript{1}, S. P. Mohanty\textsuperscript{2} and E. Kougianos\textsuperscript{3}

NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu) University of North Texas, Denton, TX 76203, USA.\textsuperscript{1,2,3}

Email: omg0006@my.unt.edu\textsuperscript{1}, saraju.mohanty@unt.edu\textsuperscript{2} and eliask@unt.edu\textsuperscript{3}

Acknowledgment: This research is supported in part by SRC award P10883 and NSF awards CNS-0854182 and DUE-0942629.
Outline of the Talk

- Introduction
- Proposed Design Flow
- Phase Locked Loop Circuit
- Polynomial and Neural Network Metamodeling
- Bee Colony Optimization Algorithm
- Results and Comparison
- Conclusions and Future Research Directions
Introduction

• Goal: Reduce the complexity of computations for analog circuits to include parasitics.
• Physical layout and simulation analysis is very time consuming processes in the design flow.
• Metamodels use mathematical formulas that represent circuit behavior within a given range using small amount of sampling points.
• This paper targets sampling techniques which are technology independent and the amount that is needed to create an accurate metamodel
Design Flow

- Verification data set is 30% < Sample (training) data set.
- Most of the work is done outside of SPICE simulator.
Phase Locked Loop

Reference Clock In

Phase Detector (PD)

Charge Pump Loop Filter (CP)

LC-VCO

Clock Out

Frequency Divider

\[ \frac{\cdot}{2} \]
Phase Locked Loop

PLL physical layout
Parameters Considered in PLL

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Parameter</th>
<th>Min (m)</th>
<th>Max (m)</th>
<th>Optimal Value (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase Detector</td>
<td>$W_{ppd1}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.66$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{npd1}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.11$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{ppd2}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>784n</td>
</tr>
<tr>
<td></td>
<td>$W_{npd2}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>689n</td>
</tr>
<tr>
<td></td>
<td>$W_{ppd3}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.54$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{npd3}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>737n</td>
</tr>
<tr>
<td>Charge Pump</td>
<td>$W_{nCP1}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.24$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{pCP1}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.35$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{nCP2}$</td>
<td>1$\mu$</td>
<td>4$\mu$</td>
<td>1.35$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{pCP2}$</td>
<td>1$\mu$</td>
<td>4$\mu$</td>
<td>2.88$\mu$</td>
</tr>
<tr>
<td>LC-VCO</td>
<td>$W_{nLC}$</td>
<td>3$\mu$</td>
<td>20$\mu$</td>
<td>18.62$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{pLC}$</td>
<td>6$\mu$</td>
<td>40$\mu$</td>
<td>37.48$\mu$</td>
</tr>
<tr>
<td>Divider</td>
<td>$W_{p1Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.65$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{p2Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.54$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{p3Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.38$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{p4Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.96$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{n1Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.09$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{n2Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.17$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{n3Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.29$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{n4Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>1.95$\mu$</td>
</tr>
<tr>
<td></td>
<td>$W_{n5Div}$</td>
<td>400n</td>
<td>$2\mu$</td>
<td>536n</td>
</tr>
</tbody>
</table>
Polynomial Metamodels

- 21 design parameters used.
- 100/30 samples points used.
- Polynomial models are created using the stepwise regression to minimize the amount of coefficients.
- Order 1-6 models are picked for each FoM.
- Best models from each FoM, based on RMSE and $r^2$ are picked for both training and verification sets.
Polynomial Regression

Fig. 7. Generated $R^2$ and $R^2_{adj}$ for every order of the polynomial metamodel for settling time

Fig. 8. Number of coefficients corresponding to the order of the generated metamodel for settling time
Neural Network Models

- 100/30 sample points used.
- Feed-forward dual layer NNs (FFDL) and radial NNs are considered in this work.
- Multiple FFDL networks for each FoM:
  - Two types of non-linear hidden layer functions are considered each varying hidden neurons 1-20:
    - A) \( b_j(v_j) = \left( \frac{1}{1 + e^{-\lambda v_j}} \right) \)
    - B) \( b_j(v_j) = \tanh(\lambda v_j) \)
### TABLE I
**FREQUENCY NON-POLYNOMIAL METAMODEL COMPARISON OF THE PLL.**

<table>
<thead>
<tr>
<th>Function</th>
<th>Data Filtering</th>
<th>R²-test</th>
<th>R²-verification</th>
<th>RMSE</th>
<th>Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>logsig→purelin</td>
<td>none</td>
<td>0.802</td>
<td>0.725</td>
<td>52.74 MHz</td>
<td>4</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>none</td>
<td>0.839</td>
<td>0.713</td>
<td>51.24 MHz</td>
<td>3</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>none</td>
<td>0.020</td>
<td>0.490</td>
<td>81.51 MHz</td>
<td>9</td>
</tr>
<tr>
<td>logsig→purelin</td>
<td>minmax</td>
<td>0.917</td>
<td>0.664</td>
<td>48.39 MHz</td>
<td>1</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>minmax</td>
<td>0.855</td>
<td>0.699</td>
<td>53.65 MHz</td>
<td>1</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>minmax</td>
<td>0.844</td>
<td>0.712</td>
<td>50.88 MHz</td>
<td>1</td>
</tr>
<tr>
<td>logsig→purelin</td>
<td>meanstd</td>
<td>0.845</td>
<td>0.733</td>
<td>53.60 MHz</td>
<td>1</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>meanstd</td>
<td>0.793</td>
<td>0.762</td>
<td>51.64 MHz</td>
<td>5</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>meanstd</td>
<td>0.848</td>
<td>0.749</td>
<td>48.97 MHz</td>
<td>5</td>
</tr>
<tr>
<td>polynomial</td>
<td>none</td>
<td>0.930</td>
<td>4</td>
<td>77.96 MHz</td>
<td>48</td>
</tr>
</tbody>
</table>

### TABLE II
**LOCKING TIME NON-POLYNOMIAL METAMODEL COMPARISON OF THE PLL.**

<table>
<thead>
<tr>
<th>Function</th>
<th>Data Filtering</th>
<th>R²-Test</th>
<th>R²-Verification</th>
<th>RMSE</th>
<th>Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>logsig→purelin</td>
<td>none</td>
<td>0.828</td>
<td>0.873</td>
<td>1.30 μs</td>
<td>1</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>none</td>
<td>0.850</td>
<td>0.723</td>
<td>1.44 μs</td>
<td>9</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>none</td>
<td>0.078</td>
<td>0.830</td>
<td>2.26 μs</td>
<td>10</td>
</tr>
<tr>
<td>logsig→purelin</td>
<td>minmax</td>
<td>0.826</td>
<td>0.870</td>
<td>1.29 μs</td>
<td>1</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>minmax</td>
<td>0.839</td>
<td>0.942</td>
<td>1.12 μs</td>
<td>10</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>minmax</td>
<td>0.931</td>
<td>0.508</td>
<td>1.65 μs</td>
<td>10</td>
</tr>
<tr>
<td>logsig→purelin</td>
<td>meanstd</td>
<td>0.826</td>
<td>0.906</td>
<td>1.22 μs</td>
<td>2</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>meanstd</td>
<td>0.737</td>
<td>0.939</td>
<td>1.12 μs</td>
<td>3</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>meanstd</td>
<td>0.963</td>
<td>0.691</td>
<td>1.23 μs</td>
<td>3</td>
</tr>
<tr>
<td>polynomial</td>
<td>none</td>
<td>0.877</td>
<td>4</td>
<td>1.91 μs</td>
<td>56</td>
</tr>
</tbody>
</table>

### TABLE III
**POWER MODELS COMPARISON**

<table>
<thead>
<tr>
<th>Function</th>
<th>Data Filtering</th>
<th>R²-test</th>
<th>R²-verification</th>
<th>RMSE</th>
<th>Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>logsig→purelin</td>
<td>none</td>
<td>0.235</td>
<td>0.314</td>
<td>3.2 mW</td>
<td>10</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>none</td>
<td>0.217</td>
<td>0.233</td>
<td>2.8 mW</td>
<td>1</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>none</td>
<td>0.533</td>
<td>0.582</td>
<td>2.6 mW</td>
<td>1</td>
</tr>
<tr>
<td>logsig→purelin</td>
<td>minmax</td>
<td>0.995</td>
<td>0.990</td>
<td>325.30 μW</td>
<td>1</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>minmax</td>
<td>0.996</td>
<td>0.991</td>
<td>316.20 μW</td>
<td>1</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>minmax</td>
<td>0.996</td>
<td>0.992</td>
<td>300.55 μW</td>
<td>1</td>
</tr>
<tr>
<td>logsig→purelin</td>
<td>meanstd</td>
<td>0.995</td>
<td>0.992</td>
<td>300.63 μW</td>
<td>1</td>
</tr>
<tr>
<td>tanσ→purelin</td>
<td>meanstd</td>
<td>0.996</td>
<td>0.993</td>
<td>281.29 μW</td>
<td>1</td>
</tr>
<tr>
<td>radial→purelin</td>
<td>meanstd</td>
<td>0.996</td>
<td>0.992</td>
<td>290.26 μW</td>
<td>1</td>
</tr>
<tr>
<td>polynomial</td>
<td>none</td>
<td>0.998</td>
<td>4</td>
<td>2.66 mW</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Filtering</th>
<th>R²</th>
<th>Order</th>
<th>RMSE</th>
<th>Number of Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>polynomial</td>
<td>none</td>
<td>0.877</td>
<td>4</td>
<td>1.91 μs</td>
</tr>
</tbody>
</table>
Bee Colony Optimization

State diagram for bee transition

Algorithm 1 Proposed Bee Colony Optimization Algorithm.
1. Initialize maximum iterations $i = max$.
2. Set the boundaries for each parameter $P(i) \leftarrow [\text{min}, \text{max}]$.
3. NumberBees $\leftarrow$ Define the number of bees.
4. buffer $\leftarrow$ Number of close worker bees dispersal.
5. Initialize a matrix as follows: $bees_{matrix}(3, \text{NumberBees}) \leftarrow \{\text{workers, onlookers, scouts}\}$.
6. Set $bees_{matrix}$ first half to be workers and other onlookers.
7. Initialize food sources.
8. while ($Counter \leq max$) do
9. for each $i$ from 1 to NumberBees do
10. if ($bees_{matrix}(1, i) \leftarrow 1$) then
11. (1) Send worker bee to a random known food source.
12. Calculate $P_{\text{on}}$ and $P_{\text{off}}$ using metamodels.
13. Calculate the proposed FoM of the PLL.
14. if ($\text{current FoM}$ is better than the previous FoM) then
15. Update result and location.
16. else
17. Convert bee to onlooker.
18. end if
19. else
20. if ($bees_{matrix}(1, i) \leftarrow 1$) then
21. (2) Send onlooker bee.
22. Calculate probability that the food source is good
23. if (probability is high) then
24. $P \leftarrow (P_{\text{on}} + \text{random}(1) \times P_{\text{max}}) \times \text{buffer}$.
25. $P_{\text{on}}$ and $P_{\text{off}}$ to random location for each design parameter $P$.
26. Calculate the FoM.
27. if ($\text{current FoM}$ is better than the previous FoM) then
28. Update result and location.
29. Convert bee to worker.
30. else
31. Convert bee to scout.
32. end if
33. else
34. (3) Send scout bee.
35. Pick the best result as $\text{best}_i$.
36. $P \leftarrow P_{\text{on}} + \text{random}(1) \times P_{\text{max}}$.
37. Send the scout to random location for each $P$.
38. if ($\text{current FoM}$ is better than previous FoM) then
39. Update result.
40. Convert bee to worker.
41. end if
42. end if
43. end if
44. end if
45. end if
46. end while
47. $Counter \leftarrow Counter + 1$.
48. end while
49. Return result and location.
Bee Colony Optimization

- Output frequency of PLL is used as a constraint function.

\[ FoM = \left( \frac{1}{\text{power} \times \text{lockingTime}} \right) \]
## Optimization Results

<table>
<thead>
<tr>
<th>FoM</th>
<th>Baseline Power</th>
<th>Power optimality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Power</td>
<td>3.9 mW</td>
<td>3.9 mW</td>
</tr>
<tr>
<td>Locking Time</td>
<td>8.476 μs</td>
<td>3.3147 μs</td>
</tr>
<tr>
<td>Frequency</td>
<td>2.6909 GHz</td>
<td>2.7026 GHz</td>
</tr>
</tbody>
</table>
Conclusions

- A design flow for metamodeling using the neural networks is proposed.
- A 180nm PLL was subjected to the proposed design flow.
- Feed-forward NNs show better accuracy than polynomial models.
- Bee colony optimization algorithm is implemented for circuit transistor sizing optimization.
- It is observed that there is no amount of hidden neurons that provide best result.
- 56% increase in accuracy is observed using NN over polynomial metamodels.
- On average 3.2% error is observed using NN.
Thank you !!!