

Efficient Estimation of Stochastic Power Supply Noise Induced Jitter (SPSIJ) in CMOS Inverters via Knowledge-Based Neural Networks

A. Javaid¹, R. Achar¹, and J. N. Tripathi²

¹Carleton University, Ottawa, Canada

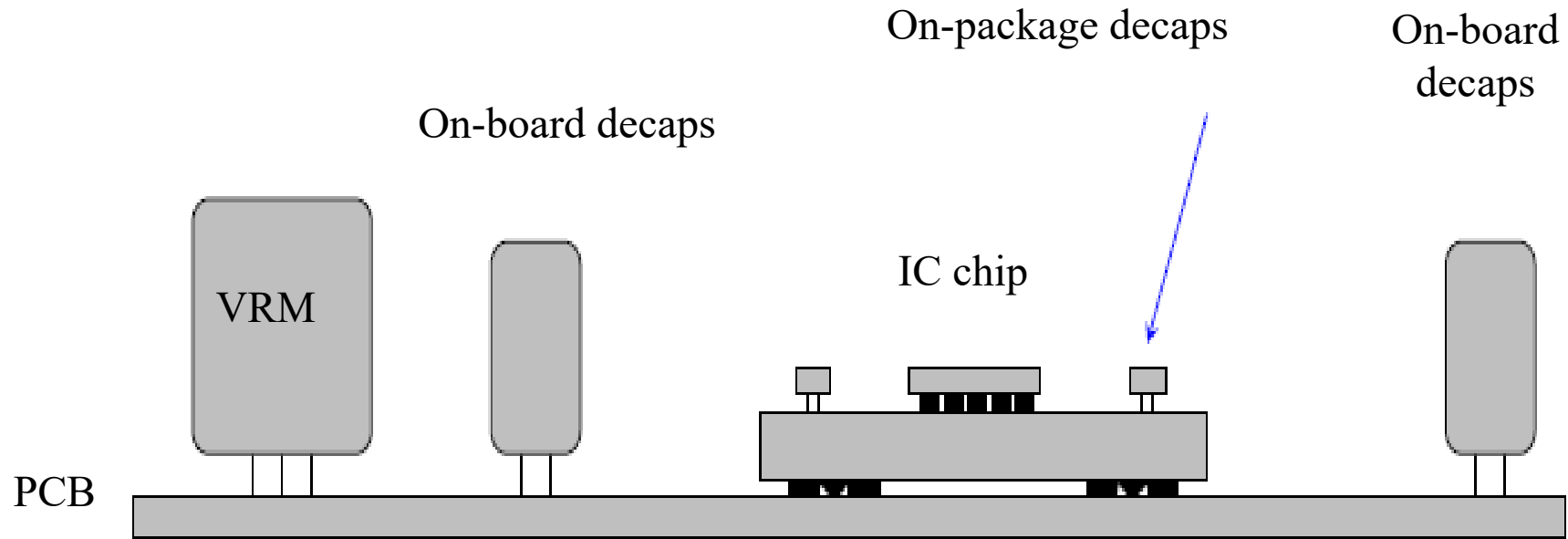
²IIT Jodhpur, Jodhpur, India.

Agenda

- ☐ Introduction
- ☐ Contributions
- ☐ Development of the Proposed Method
- ☐ Validating examples
- ☐ Conclusions

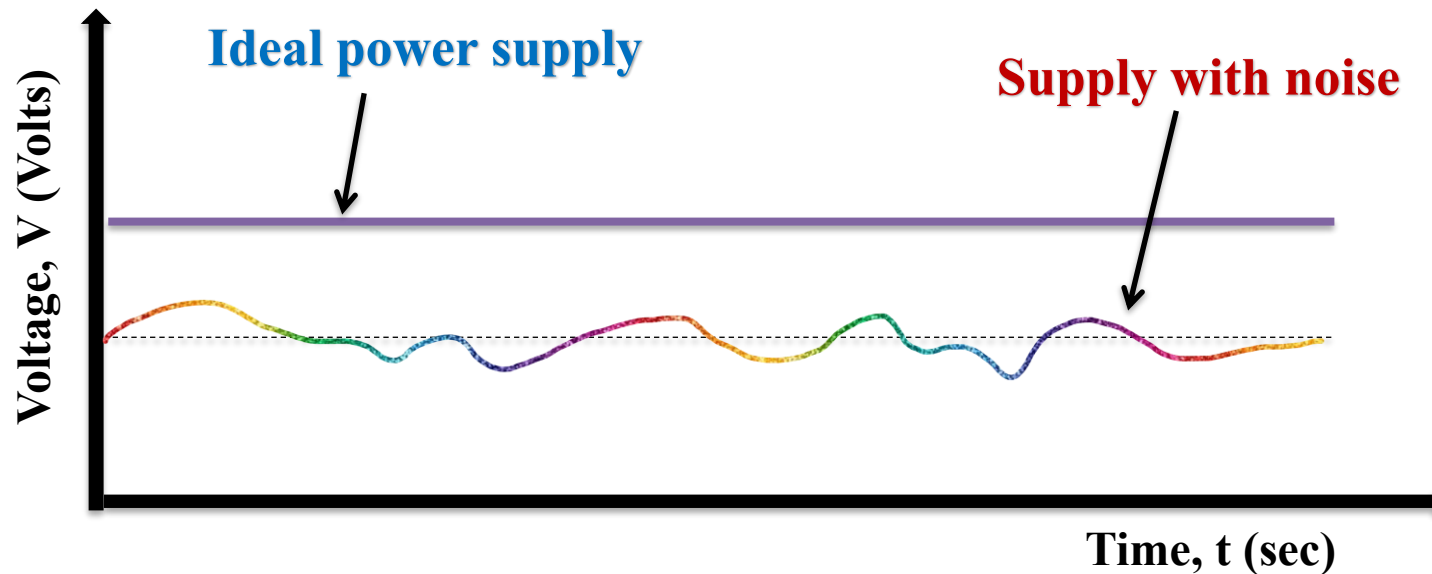
- Power delivery has become a major challenge in high-density high-speed low-power modern circuit VLSI designs.

Power Distributed Network



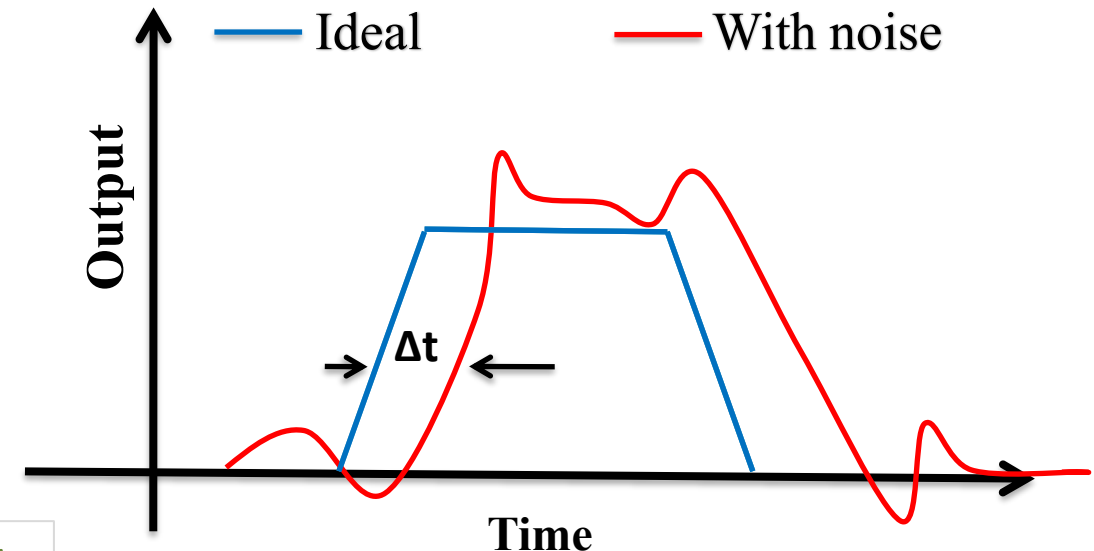
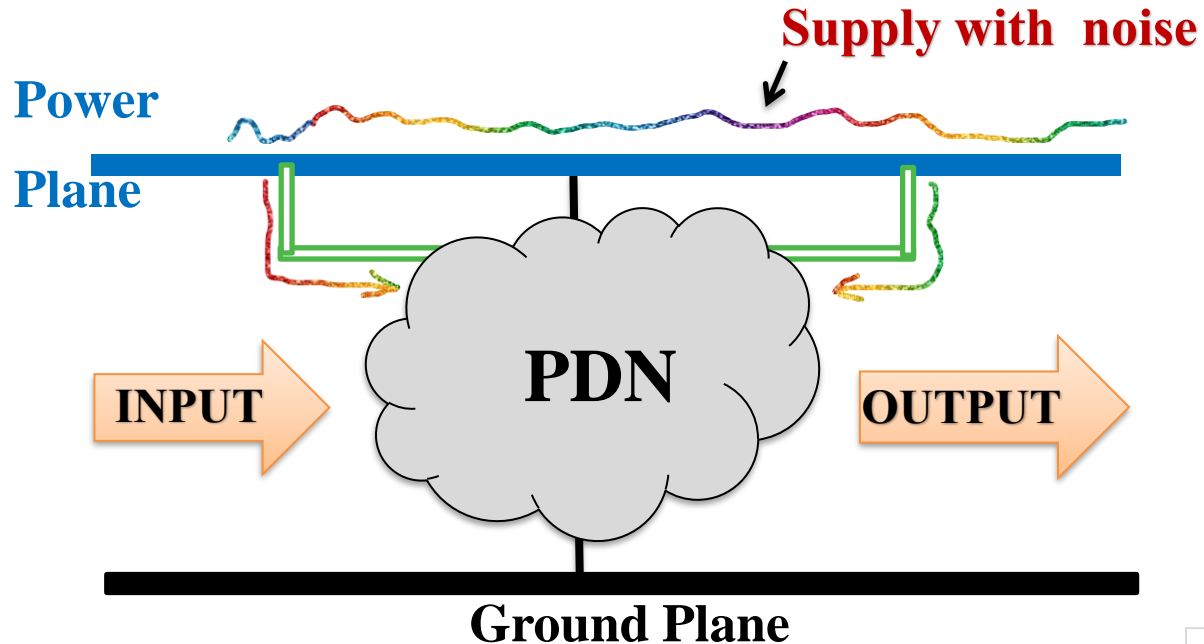
- Power delivery network (PDN) is greatly effected by the power supply noise and may cause a system failure.

Power Supply Noise



- PSN produces instantaneous fluctuations in the system hence alters the timing behavior from its ideal timing.
- These variations are commonly known as jitter.

PDN with PSN



Jitter

- Jitter becomes very critical when power supply is shared with on-chip modules, making it harder to achieve the desired timing budget.

Causes - examples

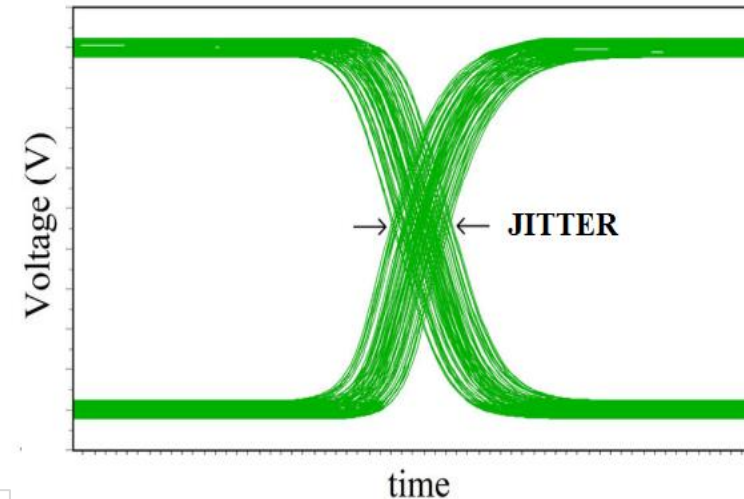
ISI, Crosstalk

Ground noise

Power Supply with stochastic noise

Measurements

Eye diagram



Prior work

Tail fitting algorithm based on the Gaussian tail model is used to find the probability distribution of the total jitter.

** M. P. Li et. al, “A new method for jitter decomposition through its distribution tail fitting,” in *Proc. IEEE Int. Test Conf.*, 1999

Frequency domain conversion method decomposes jitter using power spectral density.

** J. Kho and T. Y. Ling, “Fast and accurate technique to decompose jitter for very long pattern length waveform,” in *Proc. IEEE Elect. Des. Adv. Packag. Syst. Symp.*, 2014

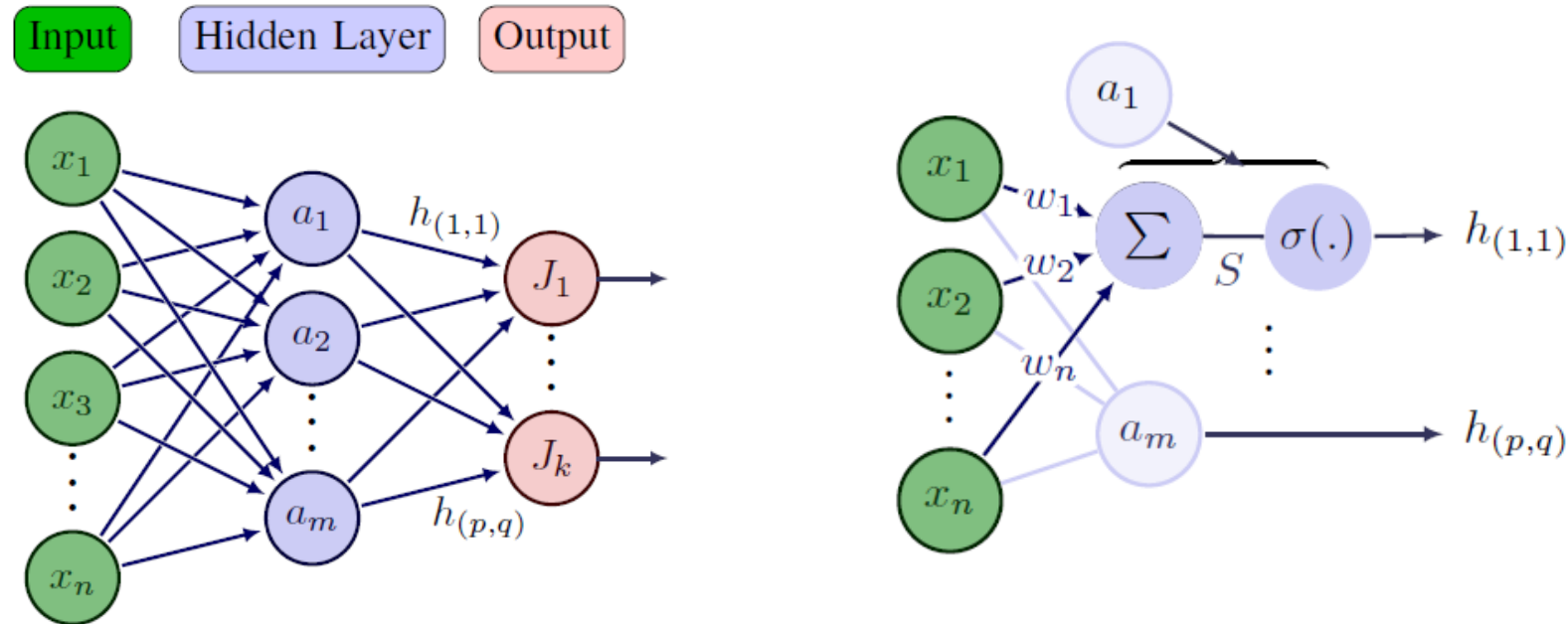
Using a convolutional neural network (CNN) technique, training samples from the ADS are obtained in order to develop jitter histograms, simulating a high-speed serial link.

** N. Ren, Z. Fu, D. Zhou, H. Liu, Z. Wu, and S. Tian, “Jitter decomposition by Convolutional Neural Networks,” *IEEE Trans. Electromagn. Compat.*, Oct. 2021

Development of the Proposed Method

- In the proposed hybrid approach, knowledge-based neural network (**KBNN**) is developed to estimate jitter (SPSIJ) in the presence of **stochastic power supply noise** using training data obtained from both analytical expressions as well as commercial simulator.
- For efficient training, **large number** data is obtained using fast-to-evaluate analytical models and **limited data** is extracted using HSPICE.

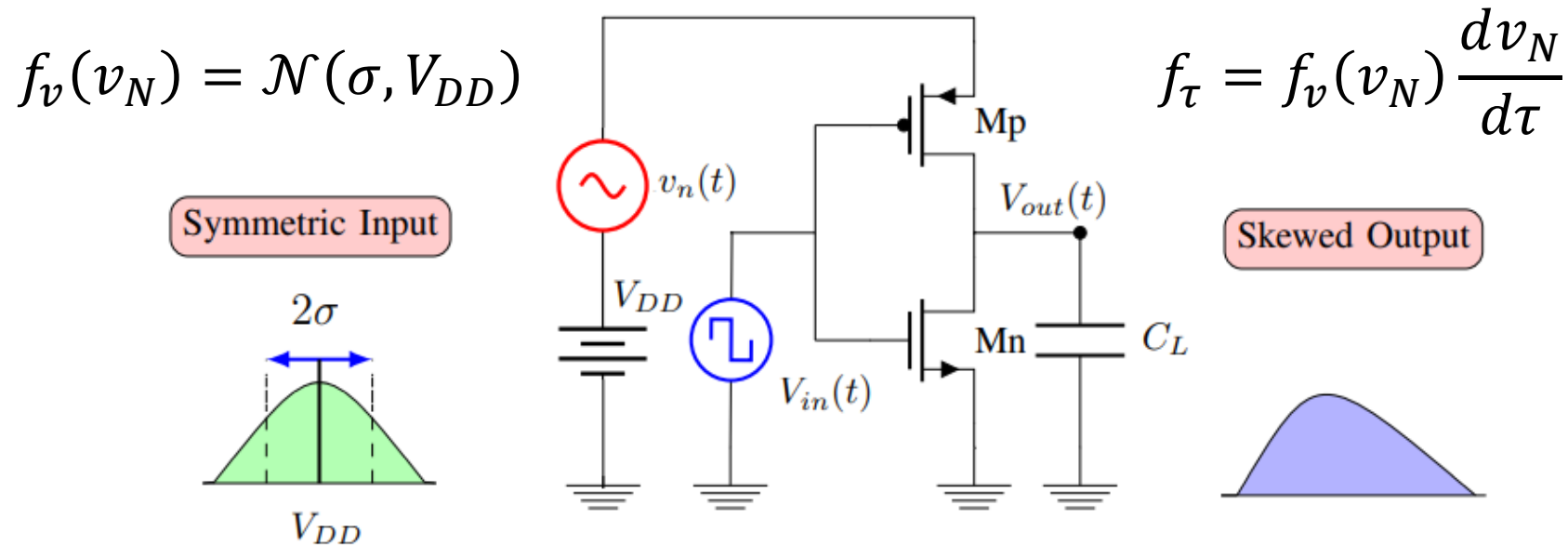
Review of Artificial Neural Network



- **Output of the neurons** $\rightarrow h = \sigma(S) = \sigma\left(w_o + \sum_{i=0}^n x_i w_i\right)$
- Training algorithms (such as Backward propagation method, secant method, Levenberg algorithm etc.) are considered for ANN development.

Review of Analytical Model

- Analytical expressions for jitter use probability density functions of both, the input noise and the time delay for CMOS inverters.



Jitter response: $\rightarrow \text{SPSIJ} = \sigma \left. \frac{d\tau}{dv} \right|_{v=v_H}$

Review of Analytical Model

Proposed Model for SPSIJ:
$$v_H = v_o \frac{a}{(c + C_L)^b}$$

The unknown parameters associated with v_H are given as:

Using Common term of Taylor series:
$$a = -\frac{2}{v_{th}}$$

Using 1st and 2nd order terms of Taylor series:
$$c = 1$$

and
$$b = 0.5 \left[-1 - \sqrt{1 + 12 \left(\frac{1+\alpha}{v_{th}} \right)} \right]$$

Initial guess using $d\tau/dv = 0$:
$$v_o = \frac{2 v_{th}}{1 - \alpha}$$

Development of the Proposed Approach

- **Step-1:** In order to develop a knowledge-based neural network, training data for SPSIJ is generated using analytical model and a circuit simulator.

Analytical Model*

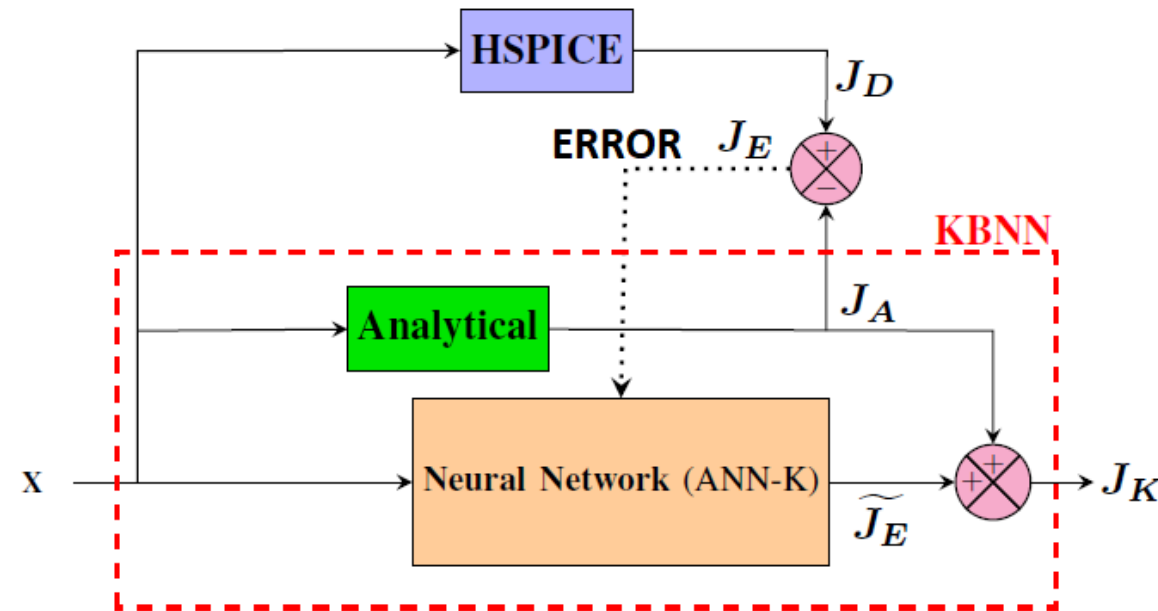
- Fast to evaluate
- Large number of data can be obtained.
- Not accurate (due to the approximate nature)

Circuit simulator (HSPICE)

- Computationally expensive
- Limited number of data can be collected
- Produces accurate results

Knowledge-based Neural Network

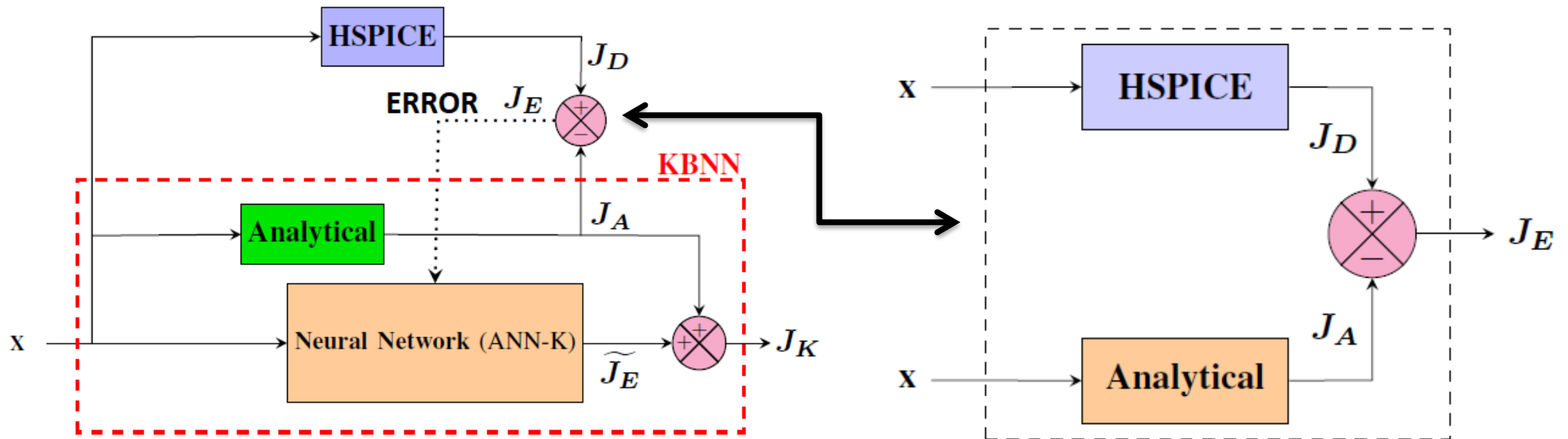
- Step-2:** KBNN uses analytical relations as an additional layer of knowledge for an efficient training.



- KBNN model or Step-2 is divided into three phases.

Development of the Proposed Method

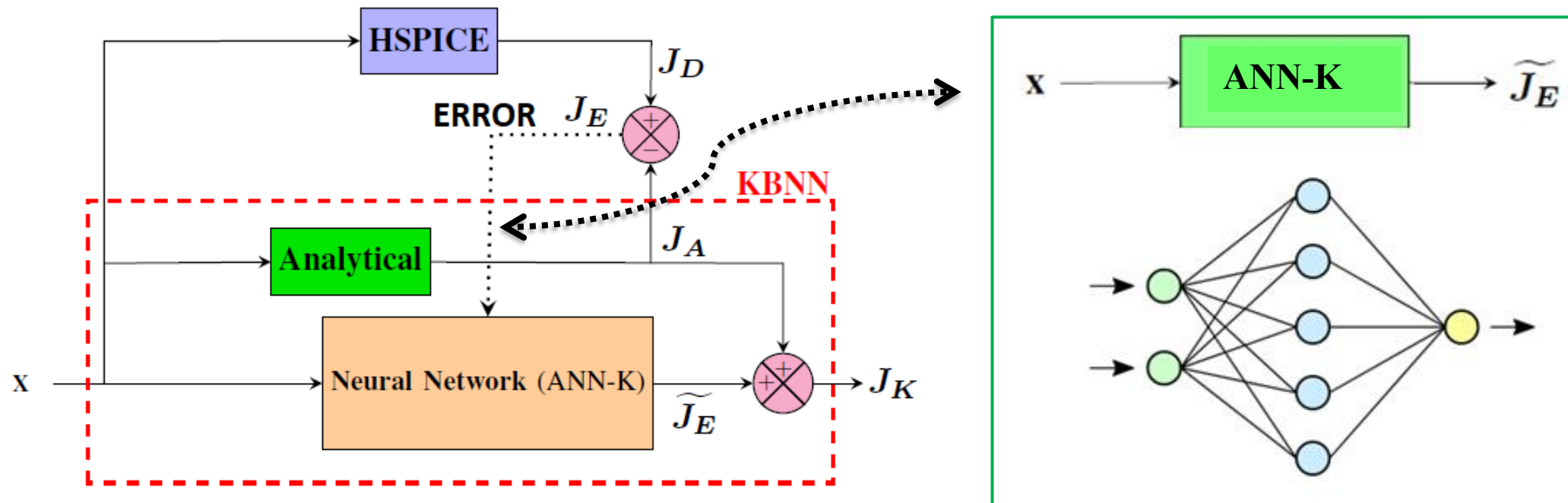
- **Phase 1:** Using source difference method, functional dependence of jitter on the input vector (\mathbf{x}) is captured in terms of an error vector.



Error vector: $\rightarrow J_E(\mathbf{x}) = J_A(\mathbf{x}) - J_D(\mathbf{x})$

Development of the Proposed Method

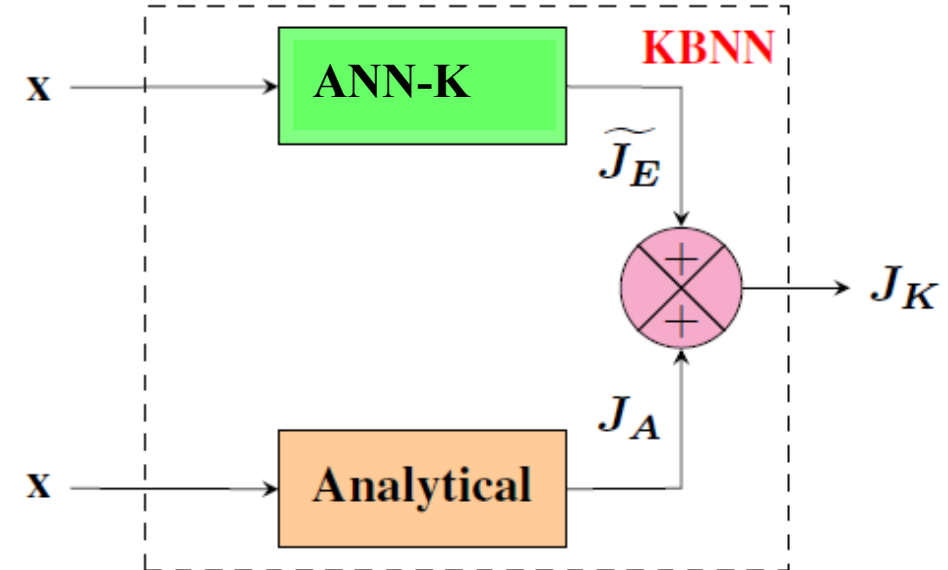
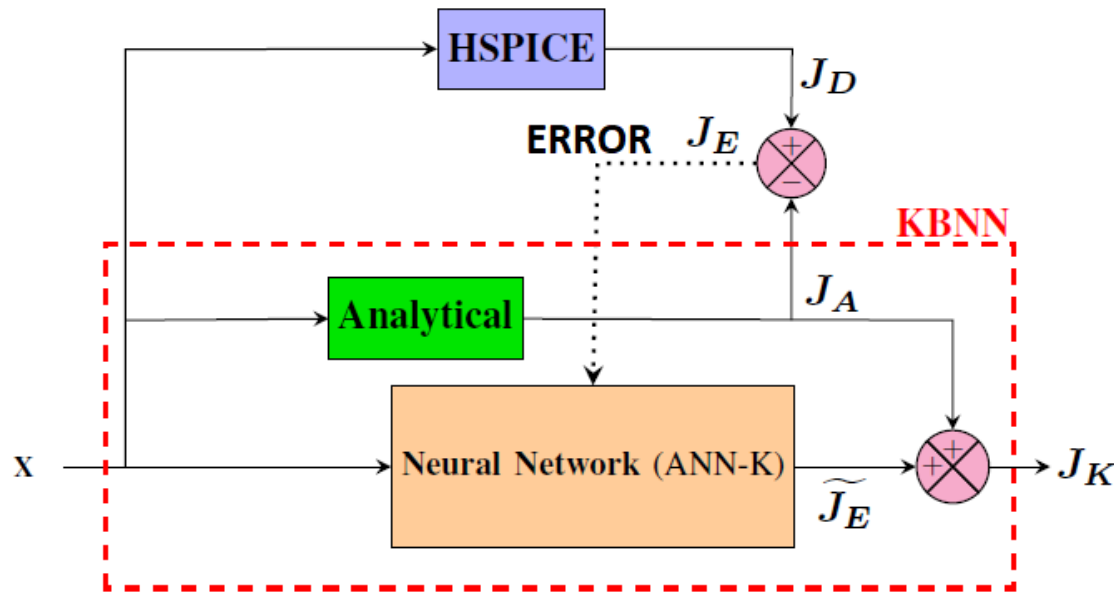
- **Phase 2:** An artificial neural network (ANN-K) is developed using error vector as a training data.



- **Levenberg–Marquardt algorithm is used for ANN training.**
- Gauss-Newton and gradient descent

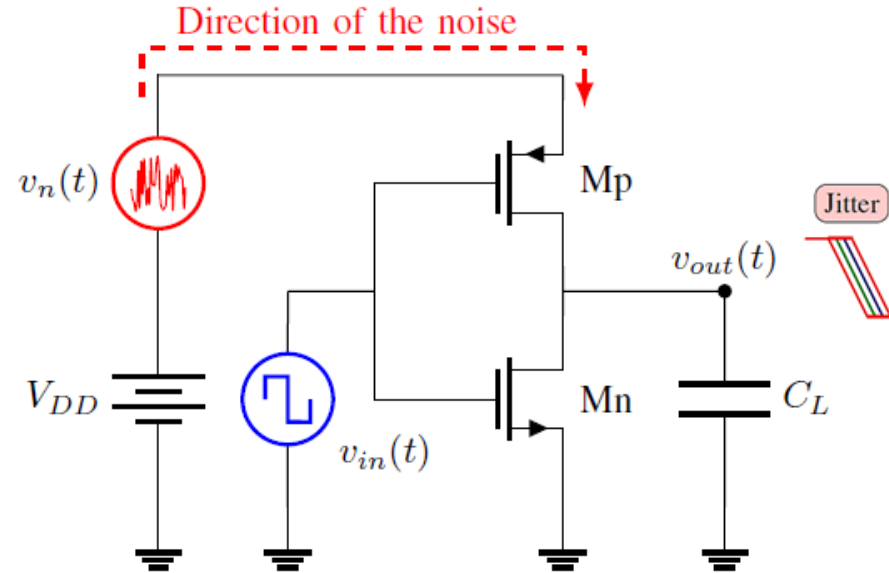
Development of the Proposed Method

- **Phase 3:** Output of ANN-K (J_K) is combined with the analytical response to obtain the final jitter (SPSIJ) response.



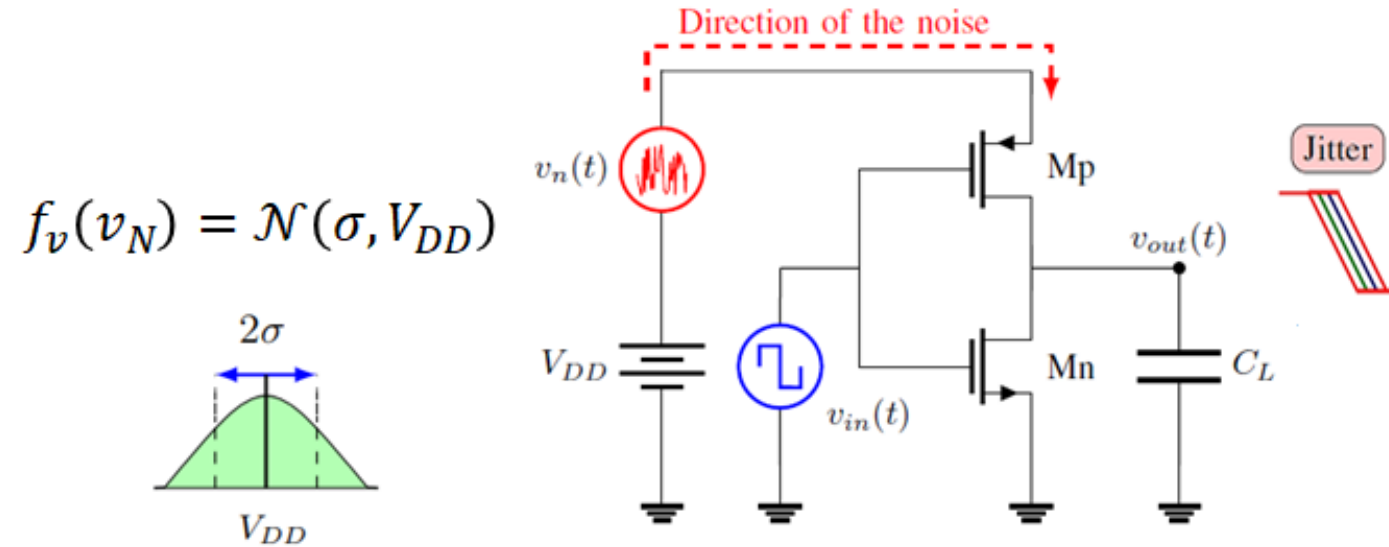
Output of the KBNN: $\rightarrow J_K(\mathbf{x}) = J_A(\mathbf{x}) + \tilde{J}_E(\mathbf{x})$

Numerical Examples



- CMOS inverter module is operating at 125 Mbps.
- Jitter is evaluated at the midpoint of the rising edge.
- 45 nm CMOS technology is used to simulate NMOS and PMOS transistors.

Example 1



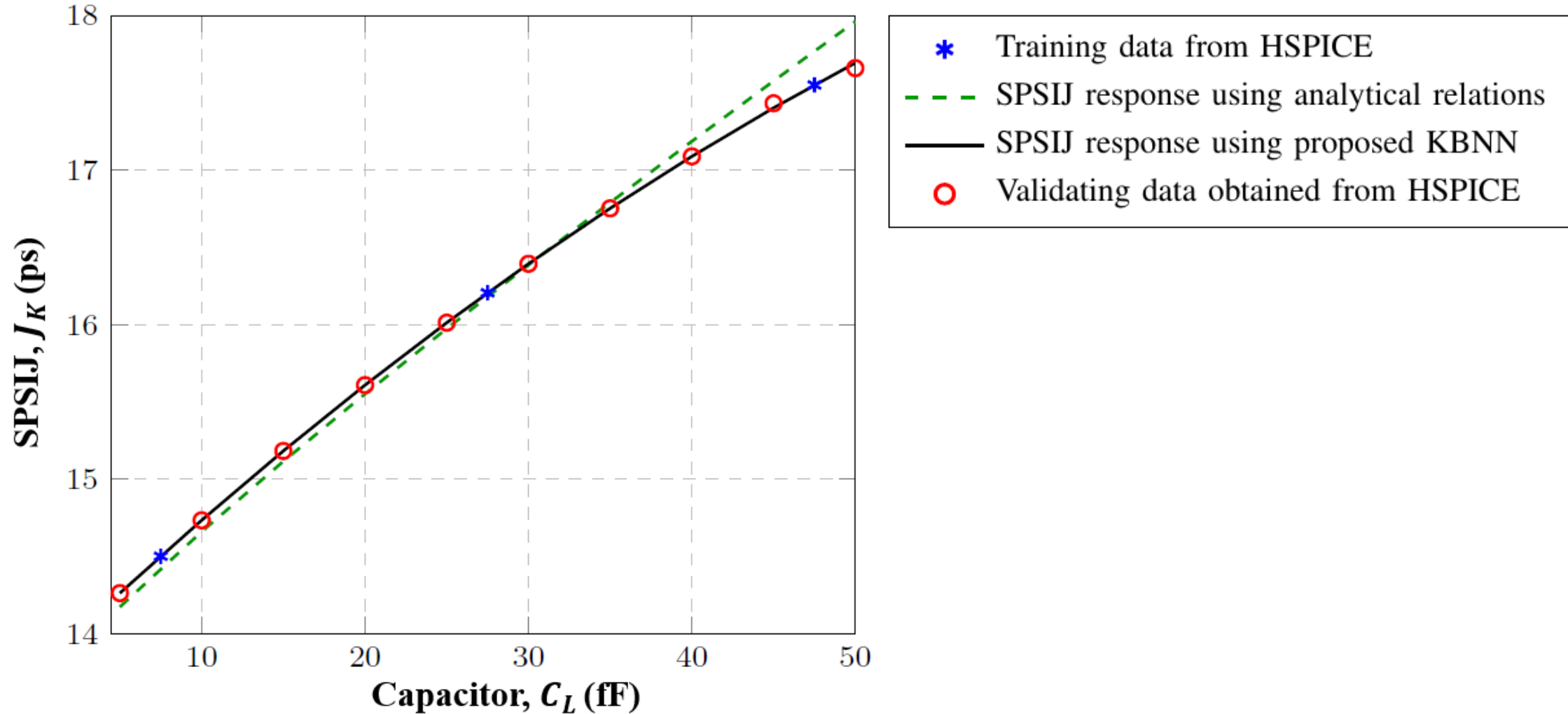
- Normally distributed stochastic power supply noise is considered.
 - RMS voltage: $\sigma = 50 \text{ mV}$
 - Nominal voltage: $V_{DD} = 1.2 \text{ V}$.
- Load capacitor is varied from 5fF to 50fF.

Example 1 – Details of Training Data

Data generation	HSPICE	Analytical relations	Error Vector
Number of points	3	25	25
CPU time	818 sec	0.15 sec	0.21 sec

Training Process of ANN-K	
Number of hidden layers	1
Number of hidden neurons	9
Number of epochs	45
Set training error	9.2×10^{-11}
Training time	0.99 sec

Example 1 – SPSIJ Responses



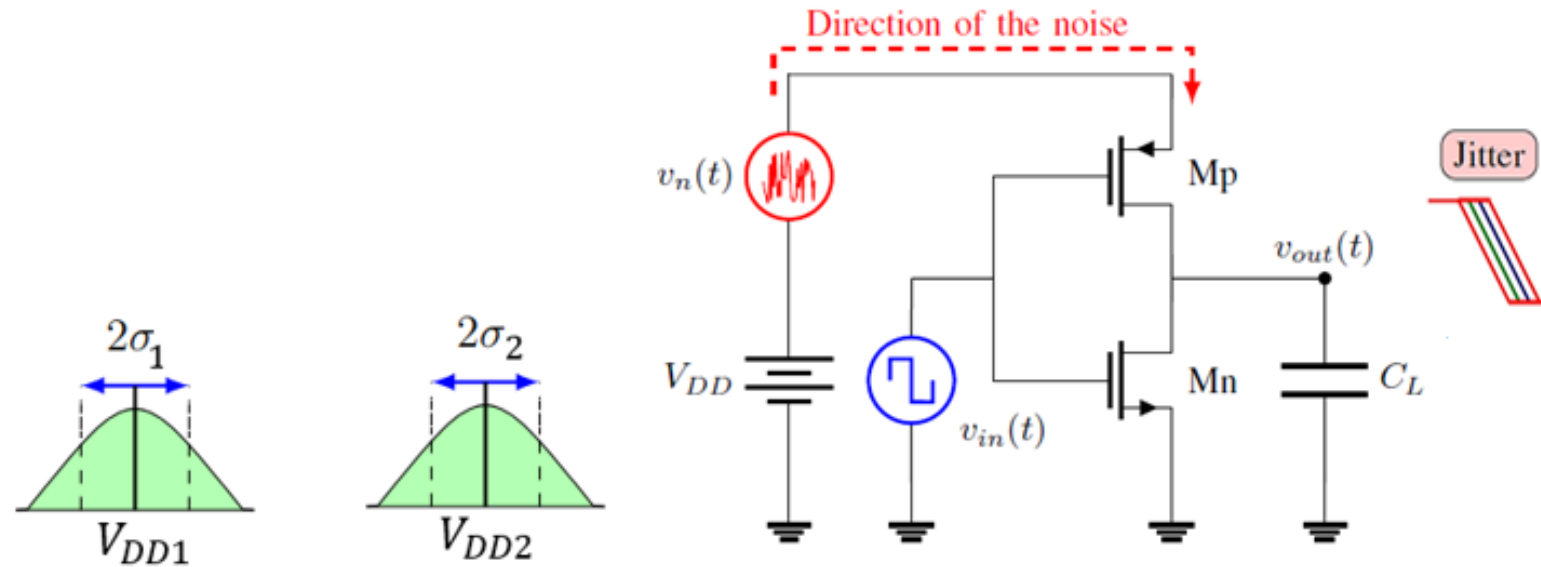
Example 1 – CPU Time Comparison

- SPSIJ is estimated at **10** data points using both, the proposed model and the commercial simulator HSPICE.

Approaches		CPU time (sec)	Speed up	Errors (%)
Conventional (10 HSPICE data points)		2726	-	-
Analytical relations (Evaluated at 10 data points)		0.0515	52,932	11.99
Proposed KBNN	ANN-K	0.0521	-	-
	Evaluation time for analytical model	0.0515	-	-
	Total for KBNN	0.1036	26,313	0.671

- The proposed approach achieves a speed up of more than **26k** as compared to using HSPICE.

Example - 2



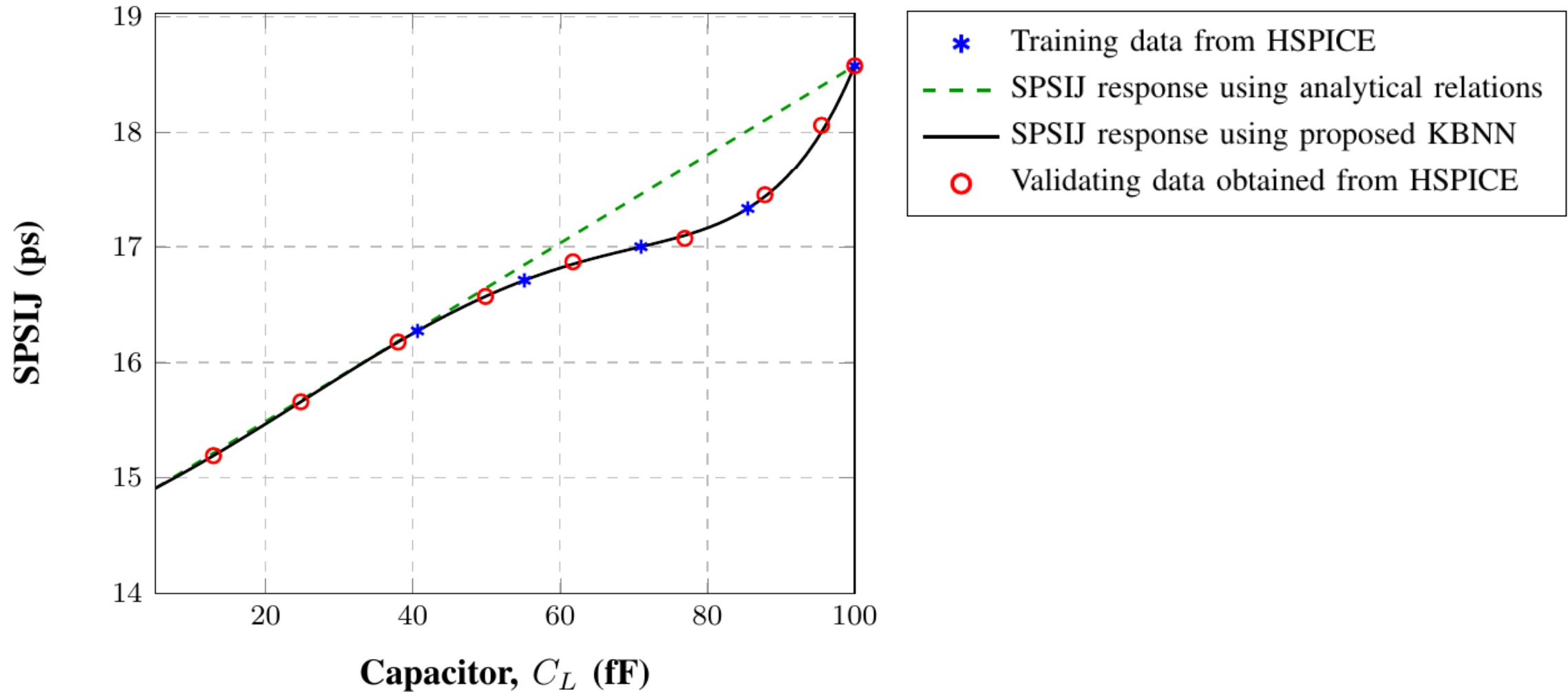
- Two normally distributed stochastic power supply noises are considered.
 - RMS voltage: $\sigma_1 = 50 \text{ mV}$ and $\sigma_2 = 30 \text{ mV}$
 - Nominal voltage: $V_{DD1} = 0.4 \text{ V}$ and $V_{DD2} = 0.8 \text{ V}$
- Load capacitor is varied from 5fF to 100fF.

Example 2 – Details of Training Data

Data generation	HSPICE	Analytical relations	Error Vector
Number of points	5	25	25
CPU time	1363 sec	0.145 sec	0.22 sec

Training Process of ANN-K	
Number of hidden layers	1
Number of hidden neurons	10
Number of epochs	45
Set training error	9.2×10^{-11}
Training time	0.95 sec

Example 2 – SPSIJ Responses



Example 2 – CPU Time Comparison

- SPSIJ is estimated at **9** data points using both, the proposed model and the commercial simulator HSPICE.

Approaches		CPU time (sec)	Speed up	Errors (%)
Conventional (9 HSPICE data points)		2454	-	-
Analytical relations (Evaluated at 9 data points)		0.051	48,117	19
Proposed KBNN	ANN-K	0.054	-	-
	Evaluation time for analytical model	0.052	-	-
	Total for KBNN	0.106	23,151	0.51

- The proposed approach achieves a speed up of more than **23k** as compared to using HSPICE.

Conclusions

- Knowledge based neural network is developed using training data from both: from a computationally expensive circuit simulator (HSPICE) as well as from an efficient analytical model.
- Small number of data is obtained using HSPICE and large number of data is extracted using analytical expressions for efficient and accurate modeling.
- Validating examples demonstrate the accuracy and efficiency achieved using the proposed method.

