

Statistical guarantees of performance for MIMO designs

Abstract—Sources of noise such as quantization, introduce randomness into Register Transfer Level (RTL) designs of Multiple Input Multiple Output (MIMO) systems. Performance of these MIMO RTL designs is typically quantified by metrics averaged over simulations. In this paper, we introduce a formal approach to compute these metrics with high confidence. We define best, bounded and average case performance metrics as properties in a probabilistic temporal logic. We then use probabilistic model checking to verify these properties for MIMO RTL and thereby guarantee the statistical performance. However, probabilistic model checking is known to encounter the problem of state space explosion. With respect to the properties of interest, we show sound and efficient reductions that significantly improve the scalability of our approach. We illustrate our approach on different non-trivial components of MIMO system designs.

I. INTRODUCTION

There is an ever growing demand to design communication and digital signal processing (DSP) systems that are area and power efficient while operating at high data-rates. *Bit Error Rate* (BER) is a commonly used performance metric for these systems. BER is an average measure of the probability with which a transmitted data bit is decoded in error. In wireless communication systems, BER requirements can be as low as 10^{-7} . MIMO systems [1] are designed to meet these requirements.

MIMO systems are complex and comprise a large number of digital components implemented at the RT Level. The process of making MIMO RTL designs meet the BER requirements is both time and resource-intensive. This is due to other criteria, such as area and power, that also need to be met. Therefore, it is desirable to have a methodology where performance estimation of MIMO RTL can be performed quickly and with a high degree of confidence.

Performance metrics are inherently probabilistic in nature due to the randomness introduced by signal corruption at the receiver and fixed-point quantization errors. Conventionally, performance estimation is done by performing Monte Carlo simulations [2] of MIMO RTL using random input vectors. Estimates that are reasonably accurate can be obtained by simulating the MIMO systems [3] over many cycles. This technique is time consuming and incomplete. FPGA implementations [4] and ASIC prototypes [5] provide accelerated simulations, thereby speeding up performance estimation. However, both these methods involve significant overheads in terms of cost.

We propose a methodology that performs efficient performance estimation for MIMO RTL by employing probabilistic model checking. Model checking exhaustively explores all

possible paths of a given length and therefore, the analysis of the design is complete and high in confidence.

MIMO RTL designs can be modeled as finite-state probabilistic systems with discrete-time transitions. Therefore, we represent them as Discrete-Time Markov Chains (DTMCs) [6].

We define BER-like performance metrics that can be expressed as properties in Probabilistic Computational Tree Logic (pCTL) [7]. In addition to average case, we also define best case and worst case performance metrics. This set of metrics can be used to rigorously analyze the error-related performance of the design, as compared to using only an average case metric.

We then use PRISM [8], a probabilistic model checking engine, to verify the pCTL properties on the DTMC models. This formally guarantees the statistical performance of MIMO RTL designs.

However, probabilistic model checking tools are known to encounter the problem of state space explosion. We address this by identifying reductions that preserve the probabilistic behaviour of the system with respect to the properties of interest. We show that these property-preserving reductions are sound by using a probabilistic bisimulation [9] argument. Although these reductions are specific to the domain of communication systems, they are not restrictive since they can be generically applied to a broad class of designs within this domain.

Markov chains have frequently been used to compute high level system performance and power [10] [11]. They have also been used at a circuit level, to design circuits with high error tolerance [12] and to analyze stability [13]. To the best of our knowledge, ours is the first work that deals with probabilistic model checking of communication systems at an RTL level for error-related performance estimation.

Therefore, our contributions in this work are as follows.

- We describe a framework in which MIMO RTL designs, including channel noise and quantization errors, are represented as DTMC models.
- We make the performance estimation quick, rigorous and high-confidence, by using probabilistic model checking over state-of-the-art simulation techniques.
- We use a more comprehensive set of performance metrics than BER.
- We introduce sound and effective property-preserving reductions and identify classes of MIMO components to which they can be generically applied.

We illustrate our technique on seminal components of a MIMO system, using a Viterbi decoder [14] and MIMO

detector [3] as case studies.

II. BACKGROUND CONCEPTS

In a communication system with digital blocks in the receiver, an Analog to Digital Converter (ADC) first translates the received analog signals into bits by *discretizing* it in time (*sampling*) as well as value (*quantization*). In this work, we confine our analysis to the digital blocks by assuming knowledge of the statistical performance of analog blocks¹.

However, imperfections such as thermal fluctuations in current and voltage and timing errors of the ADC sampler, are present in the circuitry. A large number of such small error sources are lumped together by the *Central Limit Theorem* and modeled commonly as a single random variable, called *noise*, with a zero-mean Gaussian distribution [15]. The presence of noise can lead to errors in quantization of the received sample.

Additionally, in some channels, the received sample at any time step contains components from signals transmitted in adjacent steps. This interference can be mitigated using digital blocks, such as Viterbi decoders, in the receiver. *Signal-to-Noise Ratio* (SNR) represents the level of the uncorrupted signal relative to that of the noise. For high values of SNR, the noise is insignificant compared to the signal, resulting in a low BER.

In this work, we assume that the analog blocks exhibit ideal behaviour. We also assume an *Additive White Gaussian Noise* (AWGN) model and a *Binary Phase Shift Key* (BPSK) signaling scheme [15]. However, our methodology is not limited to these assumptions.

III. OUR METHODOLOGY

We employ probabilistic model checking to formally estimate statistical performance of MIMO RTL designs. The steps involved in our methodology are:

- **DTMC modeling:** We represent the target MIMO RTL design as a finite DTMC model. We assume that every transition of the DTMC model corresponds to a single time step (modeled by an explicit clock in RTL). For a given SNR, we obtain the variance of the Gaussian distribution of noise. We use this to calculate the probability of a received sample being mapped to a particular quantization level which in turn can be used to label the transitions of the DTMC model.
- **Property specification:** We define a set of BER-like performance metrics to rigorously analyze the error-related performance of a system. We write pCTL properties corresponding to these metrics, as functions of the state variables in the DTMC model.
- **Property-preserving reduction:** We determine property-preserving reductions by analyzing certain components of a MIMO system. We show that our reductions are sound with respect to the pCTL properties. We identify that these reductions can be extended to a large class of designs for checking the same properties.

¹Analog components are used mainly for blocks like mixers, amplifiers, Phase Lock Loops (PLLs), Clock-Data Recovery units (CDRs) and ADCs

- **Probabilistic model checking:** We use PRISM to verify the specified properties by rigorously analyzing the DTMC model. PRISM is a symbolic model checking tool that uses efficient algorithms and data structures based on binary decision diagrams (BDDs).

We explore the transitions of the DTMC model until it reaches a *steady state*. A DTMC model is said to have attained a steady state when the probability of reaching a state is independent of the time step. All *finite*, *irreducible*, *aperiodic* DTMC models are guaranteed to reach a steady state [6].

IV. CASE STUDIES: MIMO SYSTEMS

A MIMO system with N_R receiver antennas and N_T transmit antennas can be modeled as,

$$\bar{y} = \mathbf{H}\bar{x} + \bar{n} \quad (1)$$

where $\bar{y} = [y_1, \dots, y_{N_R}]^T$ is the vector of received signals and $\bar{x} = [x_1, \dots, x_{N_T}]^T$ is the vector of transmitted signals. $[\]^T$ denotes the transpose of a vector. \mathbf{H} represents the $N_R \times N_T$ channel matrix and \bar{n} is an AWGN noise vector. We assume a commonly used *flat fading Rayleigh channel* model [1] and obtain the probability distribution of the elements of \mathbf{H} .

We present the following case studies.

- Estimation of error properties of a Viterbi decoder
- Estimation of error properties of a MIMO detector
- Estimation of convergence properties of a Viterbi decoder

A. Estimation of error properties of a Viterbi decoder

We briefly describe the Viterbi algorithm [14]. In this case study, we consider a transmitter whose output at time step n is obtained by adding the data bit from the current time step, $x[n]$, with the data bit from the previous time step (*i.e.*, $x[n-1]$). This system is defined to have a memory (m) equal to 1.

$q[n]$ is the quantized sample at the receiver in time step n . By itself, $q[n]$ is insufficient to determine the value of the actual data bit with a low probability of error. Therefore, the Viterbi decoder waits for the samples received in the next $L-1$ time steps before decoding the value of the data bit. Heuristically, selecting L greater than $5m$ is assumed to be sufficient for decoding the data bit with high confidence. In this example, we consider $L=6$.

The Viterbi decoder maintains two internal states (0 and 1) corresponding to the possible data bits (0 and 1, respectively) in each time step. We associate the variables *prev0* and *prev1* with the internal states 0 and 1, respectively. Since the data bit in each time step can be a 0 or a 1, a transition can occur from any one of the two internal states to another. Each transition is associated with a probability that is a function of $q[n]$. By comparing the transition probabilities, the decoder assigns values to *prev0* and *prev1* that point to the corresponding most-probable previous internal state. For example, if internal state 0 is reached with a higher probability by a transition from internal state 1 than from internal state 0, the decoder assigns a value of 1 to *prev0*.

A *trellis stage* comprises the variables $prev0$ and $prev1$ corresponding to a single time step. In each time step, the Viterbi decoder stores the variables corresponding to the previous $L-1$ trellis stages as well. Starting at one of the internal states, the decoder can traverse a path of length L through a sequence of previous internal states using the variables in the L trellis stages. This operation is called *traceback*. A *path metric* is the cost associated with a traceback path. We use $pm0$ and $pm1$ to store the path metrics associated with internal states 0 and 1, respectively.

In each time step, the Viterbi decoder uses $q[n]$ to compute the internal state transition probabilities and then assign values to $prev0$ and $prev1$ of the corresponding trellis stage. The decoder then increments $pm0$ and $pm1$ as a function of the the computed transition probabilities. The decoder chooses the internal state with the least corresponding path metric, as the starting point for traceback.

At the end of the traceback operation, the Viterbi decoder decodes the data bit. However, there is a decoding latency of $L-1$ time steps. Therefore, to verify the correctness of the decoded bit in each time step, we need to keep track of the actual data bits corresponding to the previous $L-1$ time steps.

We now describe, in detail, all the steps involved in our methodology.

1) *DTMC modeling*: A DTMC model M can be defined by a tuple (S, T_P) . S is a set of state variables of the model. $T_P: S \times S \rightarrow [0, 1]$ is the probabilistic state transition relation. A state μ of the model M is defined as a unique assignment of values to the state variables in S .

We represent the Viterbi decoder as a DTMC model M with the following state variables:

- $pm0$ and $pm1$
- $prev0_i$ and $prev1_i$: Variables used to store values of $prev0$ and $prev1$ in the i^{th} trellis stage, where $0 \leq i \leq L - 1$. $i=0$ corresponds to the trellis stage in the current time step.
- x_i : Data bit in the i^{th} trellis stage.
- $flag$: Boolean variable that is set to 1 if the decoded bit is in error.

In each time step (*i.e.*, each clock cycle in RTL), the state variables of M are assigned new values. For each variable, the set of possible values is finite. Therefore, M is a finite DTMC model. The assignment of new values denote a transition from state μ to another state μ' . The following assignments collectively define T_P .

- Data bit and path metrics: x_0 is assigned a value of 0 or 1. $q[n]$ is assigned probabilistically and is used to obtain values of $pm0$ and $pm1$, given by,

$$(pm0', pm1', x_0') = \Gamma_p(pm0, pm1, x_0) \quad (2)$$

where Γ_p is a probabilistic function with the combined probabilities of x_0 and $q[n]$. The remaining state variables are then assigned with non-probabilistic functions.

- Transmitter: F_S computes the new values of $prev0$ and $prev1$ in the current trellis stage, as functions of the new

path metrics $pm0'$ and $pm1'$. This is given by,

$$(prev0'_0, prev1'_0) = F_S(pm0', pm1') \quad (3)$$

- *Writeback*: Values of variables denoting stage i of the trellis are written to those in stage $i + 1$. This action represents the entire trellis structure being advanced by one time step.

$$(prev0'_{i+1}, prev1'_{i+1}, x'_{i+1}) = (prev0_i, prev1_i, x_i) \quad (4)$$

- *Traceback*: F_E determines the decoded bit as a function of the values of $prev0$ and $prev1$ across L trellis stages. F_E sets $flag$ to 1, if the decoded bit is not equal to the corresponding actual data bit x_{L-1} .

$$flag' = F_E(prev0'_i, prev1'_i, x_{L-1}) \quad (5)$$

States where the decoded bit is in error are of interest to us. To tag the states of interest, we use $flag$ to define a *reward* model on the DTMC. A reward is defined as a cost associated with being in various states of the DTMC. For each state, we assign a reward equal to the value of $flag$ in that state.

2) *Property specification*: We define the following BER-like metrics for T time steps and write the corresponding pCTL properties that we use in PRISM.

- **P1** (Best case error): $P=? [G \leq T (!flag)]$: Probability that no error occurs in any of the T steps.
- **P2** (Average case error): $R=? [I = T]$: Probability that an error occurs at exactly the T^{th} step.
- **P3** (Worst case error): $P=? [F \leq T (flag > 1)]$: Probability that number of errors occurring in T steps is greater than a pre-determined value (value equal to 1, in this case).

BER is not a time-bounded metric. However, in steady state, BER can be interpreted as the probability of a bit error occurring at any time step. **P2** is a reward property that computes the *expected* instantaneous value of $flag$ after exactly T transitions (time steps) of the DTMC model. We demonstrate that our systems do attain a steady state and therefore, **P2** corresponds to the BER of the system. Since we use a simple reward model that assigns rewards of 0 or 1, we do not need to express **P2** using the reward-based extension of pCTL [16].

In addition to **P2**, we analyze best and worst case error scenarios. This enables us to make stronger claims regarding the error-related performance of MIMO RTL. The properties are checked by performing an exhaustive exploration of all the possible paths of length T .

3) *Property-preserving reduction*: For error properties, it is sufficient to determine whether a bit is in error or not. Reductions can be defined for checking error properties, that compute bit errors without actually determining the values of the decoded bits. In such cases there is no comparison of values with the transmitted bits. Therefore, in designs with decoding latency, variables storing past values of transmitted bits can be discarded from the model.

We obtain the reduced Viterbi decoder model M_R by replacing the variables $prev0_i$, $prev1_i$ and x_i (excluding x_0) with the variables c_i and w_i (Figure 1). We need the variables c_i and w_i to indicate whether $prev0_i$ and $prev1_i$ point to the previous internal state corresponding to the actual data bit

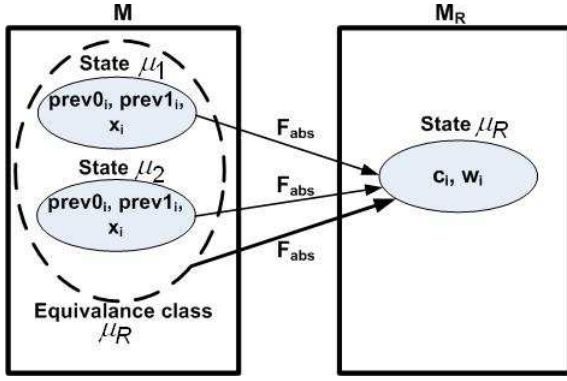


Fig. 1. Reduction from M to M_R .

x_i . This information is sufficient to check the correctness of the traceback operation and thereby, check the correctness of the decoded bit. We construct an abstraction function F_{abs} to assign values to c_i and w_i , given by,

$$(c'_i, w'_i) = F_{abs}(prev0'_i, prev1'_i, x'_i) \quad (6)$$

Multiple states in M (μ_1, μ_2, \dots) are mapped to the same state μ_R in M_R , by the function F_{abs} . This illustrates how we achieve a reduction in the state-space. Variables $pm0$, $pm1$ and x_0 from model M , are retained in the reduced model M_R . The values of these variables are same in states μ_1, μ_2 and μ_R . Therefore, the probabilistic function Γ_p is also preserved by our reduction. The non-probabilistic state transition assignments for M_R are given by,

$$(c'_0, w'_0) = F_{cw}(pm0', pm1', x'_0) \quad (7)$$

$$(c'_{i+1}, w'_{i+1}) = (c_i, w_i) \quad (8)$$

$$flag' = F_{ER}(c'_i, w'_i, x'_0) \quad (9)$$

where F_{ER} is a slightly modified version of F_E from model M .

M_R does not have information to obtain the values of the decoded bits. However, $flag$ in M_R indicates the correctness of the decoded bit, as in M . Through this reduction, the variables x_1 to x_{L-1} can be discarded. Therefore, the size of M_R is smaller than that of M .

4) *Proof of correctness*: We need to show that M_R is a probabilistic bisimulation of M . We prove this in two parts. In **Part A**, we establish that the variable based on which the error property is defined (*i.e.*, $flag$), is preserved by the reduction. In **Part B**, we show that M_R also preserves the probabilistic behavior of M . We then employ the *Strong Lumping Theorem* [17] to complete the proof.

All the states in M that are mapped to the same state in M_R through the function F_{abs} , constitute an *equivalence class* [18]. Two states are in the same equivalence class if and only if they are equivalent under a given equivalence relation. F_{abs} is the equivalence relation that establishes a one-to-one correspondence between such an equivalence class in M and the corresponding state in M_R (Figure 1). We use μ_R to refer to both a state in M_R and the corresponding equivalence class in M .

Part A: We need to prove that the value of $flag$ assigned to a state in M_R is the same as in the corresponding equivalence

class in M . We do this by verifying that Equations 5 and 9 are equivalent².

Part B: We need to prove that the equivalence classes preserve the probabilistic behaviour of the states of M . Consider a transition in M , from μ to a destination state μ' . μ' is mapped by F_{abs} to the corresponding state μ'_R in M_R according to,

$$(c'_0, w'_0) = F_{abs}(prev0'_0, prev1'_0, x'_0) \quad (10)$$

Combining Equations 4 and 6,

$$\begin{aligned} (c'_{i+1}, w'_{i+1}) &= F_{abs}(prev0'_{i+1}, prev1'_{i+1}, x'_{i+1}) \\ &= F_{abs}(prev0_i, prev1_i, x_i) \\ &= (c_i, w_i) \end{aligned} \quad (11)$$

We verify that Equation 7 and 10 are logically equivalent. This implies that if two states (μ_1, μ_2) belong to the same equivalence class in M , their respective destination states (μ'_1, μ'_2) are also equivalent under F_{abs} .

Any state transition ($\mu \rightarrow \mu'$) in M corresponds to a transition between the respective equivalence classes ($\mu_R \rightarrow \mu'_R$). In our example, for each equivalence class μ'_R , $\mu' \in \mu'_R$ is a unique destination state that $\mu \in \mu_R$ can transition to. We write the transition probability as,

$$\begin{aligned} P(\mu_R \rightarrow \mu'_R) &= P(\mu \rightarrow \mu') \\ &= \sum_{\mu' \in \mu'_R} P(\mu_R \rightarrow \mu') \end{aligned} \quad (12)$$

States in equivalent classes related by F_{abs} transition to the same set of equivalence classes related by F_{abs} and with the same probability distribution. According to the Strong Lumping Theorem, any quotient DTMC that comprises these equivalence classes as its states, is a probabilistic bisimulation of M . In fact, M_R is such a DTMC. Since Γ_p is preserved by our abstraction, the associated probabilities in Equation 12 also hold true for state transitions in M_R . Therefore, M_R is a probabilistic bisimulation of M for checking error properties.

5) *Probabilistic model checking*: We use PRISM to verify the properties **P1**, **P2** and **P3** on the reduced DTMC model M_R .

B. Estimation of error properties of a MIMO detector

For the MIMO system in Equation 1, detectors estimate the *most likely* \bar{x} , given the received vector \bar{y} . This Maximum Likelihood (ML) MIMO detection algorithm can be expressed as,

$$\hat{x} = \operatorname{argmin} |\bar{y} - \mathbf{H}\bar{s}| \quad (13)$$

where \hat{x} is the detected vector and \bar{s} is a possible value of \bar{x} .

We consider a 2x2 MIMO system with BPSK signals. Therefore, each vector element x_i can be a 0 or a 1. The ML algorithm can be implemented as in [3],

$$\begin{aligned} \hat{x} &= \operatorname{argmin} (|y_1 - h_{11}s_1 - h_{12}s_2| \\ &\quad + |y_2 - h_{21}s_1 - h_{22}s_2|) \end{aligned} \quad (14)$$

where both s_1 and s_2 are elements of \bar{s} and can have values of 0 and 1. We split Equation 14 further into real and imaginary

²Since our functions are Boolean, we use an equivalence checker [19]

parts,

$$\begin{aligned} \hat{x} &= \operatorname{argmin}(|y_{1,R} - h_{11,R}s_1 - h_{12,R}s_2| \\ &+ |y_{1,I} - h_{11,I}s_1 - h_{12,I}s_2| \\ &+ |y_{2,R} - h_{21,R}s_1 - h_{22,R}s_2| \\ &+ |y_{2,I} - h_{21,I}s_1 - h_{22,I}s_2|) \\ &= \operatorname{argmin}(M_{1,R} + M_{1,I} + M_{2,R} + M_{2,I}) \end{aligned} \quad (15)$$

where the metrics $M_{1,R}..M_{2,I}$ are computed for each of the four possible values of the vector \bar{s} . The *argmin* function determines that the most likely transmitted vector, \hat{x} , is the vector \bar{s} that corresponds to the least sum of metrics as in Equation 15.

We construct the DTMC model for the MIMO detector, as in Section IV A. We use the transmitted bit vector \bar{x} and the real and imaginary parts of the elements of both \bar{y} and \mathbf{H} , as DTMC state variables. We determine \hat{x} using Equation 15 and compare it with x to assign the value of *flag*. We use the probability distribution of the elements of \mathbf{H} and \bar{n} (based on SNR) to assign probabilities to the DTMC transitions. We use the state variable *flag* to define the DTMC reward model.

Consider a state μ_1 of the DTMC model of the MIMO detector. The variables $y_{1,R}$, $h_{11,R}$ and $h_{12,R}$ constitute the block that computes $M_{1,R}$. Let us interchange the values of these variables with those of the corresponding variables from the block that computes $M_{1,I}$ (i.e., $y_{1,I}$, $h_{11,I}$ and $h_{12,I}$ respectively). This new assignment of values corresponds to another state μ_2 of the DTMC.

From Equation 15, we observe that the computation of \hat{x} (and *flag*) is unaffected by the interchange operation between states μ_1 and μ_2 . We also observe that the probabilistic assignments to the corresponding variables in the two blocks, are symmetrical. Therefore, the states μ_1 and μ_2 exhibit symmetrical probabilistic transitions. This proves that the blocks for the metrics $M_{1,R}$ and $M_{1,I}$ are symmetric with respect to error properties that are defined based on *flag*. In fact, this is true across all the four blocks in the detector. In general, for any $N_R \times N_T$ MIMO detector, there are $2 \times N_R$ symmetric blocks.

We employ symmetry reduction [18] to reduce the size of the DTMC model, as seen in Table II. MIMO designs that exhibit such symmetries, constitute a large class of systems where symmetry reduction can be applied. In this case study, we check only the average case property **P2**.

C. Estimation of convergence properties of a Viterbi decoder

The Viterbi decoder performs the traceback operation starting from the internal state corresponding to the lowest path metric. In some cases, selecting the other internal state for traceback yields a conflicting decision of the decoded bit. When traceback paths converge, the decoded bit is independent of the internal state selected as the starting point.

A *convergent* trellis stage is defined to be a stage where both *prev0* and *prev1* are assigned the same value. All traceback paths that pass through this stage are forced to proceed through the same previous internal state. Thereafter, there is only traceback path resulting in one possible decision for the decoded

bit. If atleast one convergent stage is encountered during a traceback of length L , the traceback paths are guaranteed to converge. Heuristically, a traceback length of around $L=4m$ to $L=5m$ is chosen. However, these numbers appear to come more from empirical observations, rather than theory.

To check for convergence of traceback paths, we use the DTMC model obtained in Section IV A and introduce a new variable *count*. In each state, the assignment of values to *prev0* and *prev1* correspond to the formation of a new trellis stage. If this trellis stage is non-converging, we increment *count* by 1. We reset *count* to 0 for a convergent stage. When *count* exceeds L in a state, it implies that the previous L trellis stages are non-convergent. Therefore, the traceback paths do not converge and we set *flag* to 1. We use *flag* to define the DTMC reward model.

Similar to **P2**, we define the average case convergence property **C1**. In steady state, **C1** computes the probability that a bit decoded in any time step has non-converging traceback paths. We write **C1** as $R=? [I = T]$.

Reduction techniques are not just dependent on the type of the system, but also on the property to be checked. For the convergence property, we determine if a stage is convergent based on the values of *prev0* and *prev1*. We also require Γ_p to define the probabilistic transitions of these variables. Therefore, we need only the variables *pm0*, *pm1* and x_0 from the original model M . We discard the variables corresponding to the other $L-1$ trellis stages, obtaining a smaller DTMC for model checking.

The proof of correctness for this reduction can be explained in a manner similar to that in Section IV A 4). Instead of F_{abs} , we now consider a refining function F_{ref} that map all states with the same values assigned to *pm0*, *pm1* and x_0 , to a unique equivalence class. Since Γ_p is retained, the probabilistic behaviour of the system is preserved by the reduction. The rest of the proof follows, defining F_{ref} as the equivalence relation.

TABLE I
ERROR PROPERTIES FOR A VITERBI DECODER.

	States (Original model)	States (Reduced model)	Time (seconds)	Result
P1	53, 558, 744	8, 505, 363	90.80	3×10^{-15}
P2	53, 558, 744	8, 505, 363	184.13	0.2394
P3	107, 504, 890	16, 435, 490	365.68	≈ 1

V. EXPERIMENTAL RESULTS

We perform our experiments on a 3 GHz, 3.25 GB machine. For an SNR of 5dB, we check the error properties for the Viterbi model over $T=300$ time steps (Table I). The times listed account for both model construction and model checking. We use **P1**, **P2** and **P3** to confirm the poor error-related performance of the system for the given SNR.

Table II shows the reduction factors achieved in the MIMO detector. We consider 1x2 (SNR=8dB) and 1x4 (SNR=12dB) MIMO ML detectors. In the 1x4 detector, PRISM discards states that are reached with a probability less than 10^{-15} .

TABLE II
SYMMETRY REDUCTION OF MIMO DETECTOR.

MIMO	States (Original model M)	States (Reduced model M_R)	Reduction factor
1x2	569480	32088	18
1x4	524288	1320	400

For checking convergence ($L=8$, $\text{SNR}=8\text{dB}$) of the Viterbi decoder, the reduced DTMC has only about 61,000 states. Compared to the original model, the number of states is reduced by several orders of magnitude. We are able to check **C1** within 120 seconds. We verify from Figure 2 that the probability of non-convergence decreases with traceback length and stabilizes past $L=5m$.

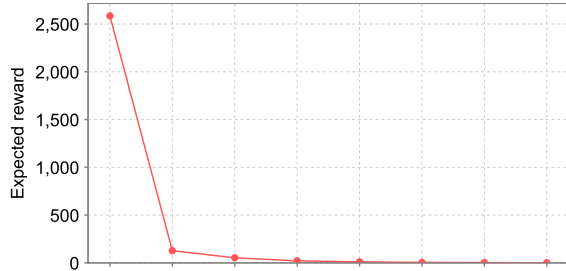


Fig. 2. **C1** as a function of L .

PRISM performs a reachability analysis first and a fixpoint is achieved. The fixpoint is referred to as *Reachability Iterations* (RI). After this fixpoint, no new states are reached in further iterations. Tables III, IV and V show the computations of **P2** and **C1** for different values of T . We observe that for values of T much greater than RI, the computed values do not change significantly. Once steady state is attained, we consider **P2** as the BER of the system.

TABLE III
P2 FOR THE VITERBI DECODER ($RI=263$).

Viterbi	T=100	T=300	T=600	T=1000
P2	0.2373	0.2394	0.2397	0.2398

The DTMC model for the Viterbi decoder is finite, irreducible and aperiodic. Therefore, the model is guaranteed to converge to a steady-state probability distribution (Section III). Although at a slower rate than for the MIMO detector, the computations for the Viterbi decoder converge reasonably quickly. To check error properties, all MIMO RTL designs will be represented as DTMC models of a similar structure. Therefore, a steady-state solution is guaranteed, although the exact time steps required to attain this may vary.

TABLE IV
CONVERGENCE OF THE VITERBI DECODER ($RI=77$).

	T=100	T=400	T=1000
C1	1.034×10^{-3}	1.042×10^{-3}	1.044×10^{-3}

The values computed in our approach closely match those obtained by performing simulations over a large number of

time steps. We simulate 10^7 time steps to estimate a BER of 1.07×10^{-5} for the 1x4 MIMO system in Table V. We observe zero bit errors in 10^5 time steps. This clearly illustrates the efficiency of our approach as compared to simulation-based techniques, particularly for very low BER requirements.

TABLE V
BER FOR MIMO DETECTORS ($RI=3$).

MIMO	T=5	T=10	T=20
1x2	0.277	0.291	0.296
1x4	1.08×10^{-5}	1.08×10^{-5}	1.08×10^{-5}

In conclusion, we have introduced a formal methodology to guarantee statistical performance of MIMO RTL designs. For larger MIMO systems, we plan to explore a compositional approach.

REFERENCES

- [1] D. Tse and P. Viswanath, *Fundamentals of wireless communication*, Cambridge University Press, 2005.
- [2] M. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE J-SAC '84*, vol. 2, no. 1, pp. 153–170, 1984.
- [3] J. H. Han, A. T. Erdogan, and T. Arslan, "A low power pipelined maximum likelihood detector for 4x4 qpsk mimo wireless communication systems," in *Proc. of ISVLSI'06*, 2006, p. 185.
- [4] D. Markovic, C. Chang, B. Richards, H. So, B. Nikolic, and R. Brodersen, "Asic design and verification in an fpga environment," Sept. 2007, pp. 737–740.
- [5] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE JSSC '05*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.
- [6] J. R. Norris, *Markov Chains*, Cambridge University Press, 1997.
- [7] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects of Computing*, vol. 6, pp. 102–111, 1994.
- [8] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 2.0: A tool for probabilistic model checking," in *Proc. of QEST '04*, 2004, pp. 322–323.
- [9] K. G. Larsen and A. Skou, "Bisimulation through probabilistic testing," *Information and Computation*, vol. 94, no. 1, pp. 1–28, 1991.
- [10] A. Nandi and R. Marculescu, "System-level power/performance analysis for embedded systems design," in *Proc. of the 38th DAC*, 2001, pp. 599–604.
- [11] G. Norman, D. Parker, M. Kwiatkowska, S. K. Shukla, and R. K. Gupta, "Formal analysis and validation of continuous-time markov chain based system level power management strategies," in *Proc. of HLDVT'02*, 2002, p. 45.
- [12] K. Nepal, R. I. Bahar, J. L. Mundy, W. R. Patterson, and A. Zaslavsky, "Designing logic circuits for probabilistic computation in the presence of noise," in *Proc. of DAC '05*, 2005, pp. 485–490.
- [13] E. Clarke, A. Donzé, and A. Legay, "Statistical model checking of mixed-analog circuits with an application to a third order $\Delta - \Sigma$ modulator," in *Proc. of HVC '08*, 2009, pp. 149–163.
- [14] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," vol. 18, no. 3, pp. 363–378, May 1972.
- [15] J. G. Proakis and M. Salehi, *Communication systems engineering*, Prentice-Hall, Inc., 1994.
- [16] S. Andova, H. Hermanns, and J.-P. Katoen, "Discrete-time rewards model-checked," in *Formal Modeling and Analysis of Timed Systems (FORMATS)*, LNCS. Springer-Verlag, 2003.
- [17] S. Derisavi, H. Hermanns, and W. H. Sanders, "Optimal state-space lumping in markov chains," *Information Processing Letters*, vol. 87, no. 6, pp. 309 – 315, 2003.
- [18] M. Kwiatkowska, G. Norman, and D. Parker, "Symmetry reduction for probabilistic model checking," in *Proc. of CAV'06*, ser. LNCS, vol. 4114, Springer, 2006, pp. 234–248.
- [19] "Synopsis formality," <http://www.synopsys.com/tools/verification/formalequivalence/pages/formality.aspx>.