A Statistical Approach to V&V for Embedded Systems

Pam Binns
Honeywell Laboratories
Minneapolis, MN 55418
Pam.Binns@honeywell.com*

Abstract

We outline how a versatile statistical methodology can be used in the verification and validation (V&V) process. This methodology is illustrated on an example of a computation time property for a software implementation of a non-linear real-time controller defined as a function of controller state variable values. We compare our approach to verification with some alternative statistical techniques used for estimating execution times and other measures of performance. We close with some topics for future work.

1. Problem Overview and Motivation

Verification of embedded systems today is confronting a crisis. Systems are becoming so complex that they cannot be reliably verified by traditional methods of testing (*e.g.* requirements and structure based testing). At the same time they are beyond our current ability to model and analyze using first principles. This paper explores an approach intermediate between these two, a statistically sound and well-structured testing methodology. The use of empirical data directly verifies the implementation, by-passing the need to verify that the implementation complies with an analytic model. Our methods provide usefully high and quantifiable levels of confidence for real-world problems, with reasonable amounts of test data and statistical computation. Our methods can also provide insight into test design. Collectively, this may lead to both reduced testing effort and increased levels of assurance for systems that otherwise could only be verified using relatively ad-hoc methods.

2. A Statistical V&V Framework

Our statistical V&V framework is based on Statistical Learning Theory (SLT) pioneered by Vapnik [4, 5]. Brief descriptions of the essential mathematical ingredients for our statistical verification framework are compactly summarized in Table 1.

Roughly, our verification methodology uses empirical data to select a "best-fitting"(to be defined) region from a set of predefined regions that ensures near optimal performance, relative to a designated property, to within a user-designated confidence level. Each region can often be described by a mathematical function, ϕ_α defined on the function input space Ω_X. Collectively the index set and its associated "region functions" are called a hypothesis space, denoted by Λ. In Figure 1, the hypothesis index set is $\Lambda = [0, M]$, with each function ϕ_α a circle on $\Omega_X \subseteq \mathcal{R}^2$ of radius α (only one function is shown). Sampled data are labelled or classified. Each of the 20 sampled data values in classified by a binary label, with 0 (-) as a negative (or unsafe) data and 1 (+) for a positive (or safe) data. A labelled sample is on Ω_Z, with $z = (x, w)$ for $x \in \Omega_X$ and $w \in \Omega_W$. A "+" classification indicates a desired condition was observed to hold - for example, execution times less than T or maximum amplitude less than A.

Other familiar examples of functions that define "simple" regions in $\mathcal{R}^3$ are all cubes, spheres, and pyramids. Indices for cubes and spheres might be expressed using a center-point and radius pair, respectively. Pyramids are compactly represented by their vertices (or defining hyperplanes). In statistical terms, we seek to find a hypothesis that "best" explains (*i.e.* fits) the data. The circle shown in Figure 1 might best describe the data, depending on the loss function. Application developers specify hypothesis index

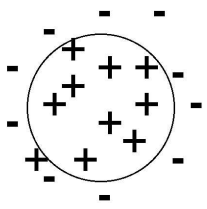

Figure 1. Data Classification Example

set Λ, sample space Ω_Z and a set of loss function Q defined on Λ and Ω_Z. The symmetric difference binary loss,

$$Q(\alpha, z) = 1 \text{ if } \phi_\alpha(x) = w \text{ and } Q(\alpha, z) = 0 \text{ otherwise,}$$

is very common. In Figure 1, only the data point with the "+" label "outside" the circle has a loss of 1.

Making good choices of Λ and Q can be difficult and will depend on both the application and its requirements. Given Λ and Q, there is the relationship between η, ϵ and s that once two are chosen the third is determined. See Table 1 for definitions. In turn, Λ and Q affect the sample size s. We attempt to highlight these tradeoffs.

A measure of complexity, called VC dimension (or VCD) [4], is associated with the number of sample data

*This work was supported in part by the DARPA/AFRL 'Software Enabled Control' program under contract number F33615-01-C-1848.

Var	Description
Ω_Z	a sample space from which data is drawn. $\Omega_Z = \Omega_X \times \Omega_W$. For $z \in \Omega_Z$, $z = (x, w)$ where x is the application domain and w is a "training label". F_Z is an unknown joint probability distribution on Ω_Z. The statistical guarantees are "distribution free" - only mild regularity conditions are assumed about F_Z.
Λ	an index set to a predefined set of hypotheses (or region functions). Each $\alpha \in \Lambda$ identifies a function ϕ_α with domain Ω_X.
Q	a non-negative loss function on $\Lambda \times \Omega_Z$ assigns the loss when selecting z for hypothesis α. $Q(\alpha, z) = 0$ when there is "no loss". For binary Q its range is $\{0, 1\}$.
R	a risk function defined over Λ. The expected loss for each hypothesis α and a measure of the "fit" of α. In symbols, $R(\alpha) = \int_{\Omega_Z} Q(\alpha, z) dF_Z(z)$.
α^*	an optimal hypothesis in Λ, one that minimizes the risk. $R(\alpha^*) = \text{argmin}_{\alpha \in \Lambda} R(\alpha)$. $R(\alpha^*) \neq 0$, necessarily. For large $R(\alpha^*)$, no hypothesis in Λ fits the underlying probability space defined on Ω_Z well.
Z^s	a random sample from Ω_Z of size s. $Z^s = z_1, z_2, ..., z_s$ where $z_j = (x_j, w_j)$, $1 \leq j \leq s$. Z^s is the sample data that drives hypothesis selection in the learning process. Z^s is assumed independent and identically (*iid*) distributed. Relaxation of the iid assumption is discussed at the end of the paper.
$\hat{R}_s(\alpha)$	the empirical risk. $\hat{R}(\alpha)$ estimates the true risk evaluated at hypothesis α, $R(\alpha)$. For sample Z^s $$\hat{R}_s(\alpha) \equiv s^{-1} \sum_{j=1}^{s} Q(\alpha, z_j).$$
α_s^*	a hypothesis that minimizes the empirical risk. s is a reminder of the dependence on the sample Z^s. Finding a value for α_s^* is the computational part of the verification problem. For any sample Z^s, $\alpha_s^* \equiv \text{argmin}_{\alpha \in \Lambda} \hat{R}_s(\alpha)$.
ϵ	a bound on the distance between the minimum true risk and the minimum empirical risk, given a sufficiently large sample. ϵ is sometimes called the accuracy of the estimated optimal risk, $\hat{R}_s(\alpha_s^*)$.
$1 - \eta$	a measure of confidence that the estimated optimal risk is within ϵ of the optimal obtainable risk.

Table 1. Verification Setup

points that can be "shattered" (*i.e.* in some sense uniquely identified by) a loss function over a hypothesis space. The VC dimension is defined in terms of the maximal number of subsets of a sample Z^s that can be generated by (binary) loss function values over hypothesis a space.

We illustrate the concept of VCD via example. Consider $\Omega_X = [0, 10]$ and $\Lambda = \{[x, y] \mid 0 \leq x < y \leq 10\}$. The VCD of Λ on Ω_X is $d = 2$ (which is the same as the VCD of Λ and Q on Ω_Z for binary loss Q). This follows since for any sample (we need only one sample though) of size 2, we can use elements in Λ to generate all 4 possible subsets. Now consider a sample of size 3. Suppose $X^3 = \{x_1, x_2, x_3\}$, with $x_1 < x_2 < x_3$. No element in Λ can generate the subset $\{x_1, x_3\}$ because $\{x_1, x_2, x_3\} \cap [a, b]$ must include x_2 if it includes both x_1 and x_3. (All other 7 subsets can be generated). Another example we will use later is that the VCD of a set of simplexes in $\Omega_X = \mathcal{R}^n$ is $n(n + 1)$. Recall, a simplex in $\mathcal{R}^n$ is the region contained in $n + 1$ intersecting hyperplanes. The reader might try this for triangles in $\mathcal{R}^2$.

The VCD was illustrated as the largest number for which all 2^s subsets of a sample Z^s could be generated and for which not all 2^{s+1} subsets of any sample Z^{s+1} could be generated, where $N^\Lambda(s)$ is the maximum number of subsets that can be generated for any sample of size s. Vapnik [4, 5] relates $N^\Lambda(s)$ to the VCD of a parameter space and loss function in Equation 1.

$$N^\Lambda(s) \begin{cases} = 2^s & \text{for } s \leq d \\ \leq \left(\frac{es}{d}\right)^d & \text{for } s > d, d \leq \infty. \end{cases} . \quad (1)$$

Equation 1 says that the growth rate of different possible samples increases only with sample size s and VCD d of Λ with loss Q, not with the dimensionality of the sample space Ω_Z.

When selecting a hypothesis space and loss function, two objectives often compete. The set of functions $\{\phi_\alpha | \alpha \in \Lambda\}$ should be as simple as possible so that VC dimension is small. The loss function Q on Λ and Ω_Z should fit the data well, that is $R(\alpha^*) \approx 0$.

For finite VCD, d, [5] shows that the empirical risk minimization method, which chooses α_s^* to minimize the empirical risk $\hat{R}_s(\alpha_s^*)$ on the basis of sample Z^s, will converge to the "right" classifier α^* in the sense that $R(\alpha^*) - \hat{R}_s(\alpha_s^*)$ is small. This convergence result is a form of (statistical) consistency, and is a generalization to the Weak Law of Large Numbers (WLLN). Finding a verifiably safe region using Equation 3 reduces the verification problem to one of finding a minimizing function α_s^* using data Z^s.

Vapnik [5] also showed the more specific and very important result in Equation 2 which transforms the imprecise "large number" (yes, but exactly how large?) result to a practical result, quantifiable for a precisely defined finite sample size. The final equality in Equation 2 is a defining property for the relationship between η, ϵ and sample size s.

$$\begin{aligned} & P(\sup_{\alpha \in \Lambda} |R(\alpha) - \hat{R}_s(\alpha)| > \epsilon) \\ & \leq \quad 4 \exp\left(s \left[\left(d \left[1 + \ln\left(\tfrac{2s}{d}\right)\right] / s\right) - (\epsilon - s^{-1})^2\right]\right) \\ & \approx \quad 4 \cdot \exp\left(d \left[1 + \ln\left(\tfrac{2s}{d}\right)\right] - s \cdot \epsilon^2\right) = \eta \end{aligned} \quad (2)$$

Equation 3 describes an asymmetric loss that is "safe". Loss is incurred only when a negative label appears in the positive region of the indicator set for α. The effect is to shrink the size of sets containing negative labels until an α that contains few to no points with negative labels is found. This makes Equation 3 "safe" because an α that omits many positive points will be selected in favor of an α with even a handful of negative points. A more detailed explanation of the rationale and mathematical justification is available in [1]. For the data in Figure 1, the empirical risk (see Table 1), $\hat{R}_{20}(\alpha) = 0$.[1]

[1] For comparison, the symmetric binary loss gives $\hat{R}_{20}(\alpha) = 0.05$.

$$Q(\alpha, z) = \begin{cases} 0 & \text{if } \phi_\alpha(x) = 0 \\ 0 & \text{if } \phi_\alpha(x) = w = 1 \\ 1 & \text{if } \phi_\alpha(x) = 1 \text{ and } w = 0. \end{cases} \quad (3)$$

3 OAV Controller V&V Example

To provide a greater sense of the V&V procedure, we include a sketch of a previous application to the iteration counts of the control law for the "Organic Air Vehicle" (See Figure 2).[2] Space precludes the control law description, only the mapping into our V&V framework is presented. A detailed mathematical description of the controller's trim computation is available in [2].

Figure 2. Organic Air Vehicle (OAV)

The OAV has a ducted fan propulsion unit, with control provided by movable vanes in the propwash. The fact that the vanes are situated in the propulsion airflow results in significant nonlinear interactions between the propulsion and the control surfaces. The real-time trim calculation for the OAV is an iterative algorithm whose computational time depends on several state variables, like vehicle velocity and rotation rates. The computation time must be predictable for reliable control. Accurate assessment of the range of these conditions leads to a greater operational envelope for the vehicle.

For simplicity computation time is equated with iteration count. The number of iterations must be below some threshold in order for the controller to meet its deadlines. We seek to estimate the largest flight envelope within which we have a quantifiable statistical confidence that the iteration count falls within the permissible threshold. Variables used when formulating the verification problem are introduced and described primarily in Table 2.

Consider the set of hypothesis Λ defined by the simplexes (intersected with Ω_X) described in Table 2, which has VC dimension $4 \cdot 5 = 20$. Figure 3 shows a 2-dimensional slice on $(v_x, \dot{\psi})$. The dark shaded (blue) asterisks show the points at which convergence occurred in two or fewer iterations (*i.e.* when $w \leq 2$, or the "+" examples). The lightly shaded (green) asterisks show the points at which convergence occurred but with more than two iterations. The (red) pluses show regions of divergence (when iteration count > 50).

[2]The term "organic" refers to the soldier-portable nature of the vehicle.

Var	Description
Ω_X	For $x \in \Omega_X$, $x = (v, \dot{\psi})$, where $v \in [-20, 200] \times [-20, 20] \times [-10, 40] \subseteq \mathcal{R}^3$ is the body axis velocity vector and $\dot{\psi} \in [-2, 2]$ is the vertical angular rate.
Ω_W	$= \{1, 2, > 2\}$, the interation count labels.
w	For $z = (x, w) = ((v, \dot{\psi}), w)$, $w = \min$ (iterations to convergence when computing controller trim values, 50).
Λ	the parameter space index set into the set of all simplexes contained in Ω_X. A simplex in $\mathcal{R}^4$ is described by 5 vertices.
$Q(\alpha, z)$	is the asymmetric loss function of Equation 3: $Q(\alpha, z) = \begin{cases} 1 & \text{if } \phi_\alpha(x) = 1 \text{ and } w > 2 \\ 0 & \text{otherwise} \end{cases}$

Table 2. OAV Control Variables

Figure 3 shows a candidate simplex for the flight envelope with computions requiring two or fewer iterations. The 2 simplex lines run diagonally. The horizontal and vertical boundaries are the lines that result when intersecting the simplex with Ω_X. For Z^s where $s = 34,000$ and $\epsilon = 0.05$, Equation 2 gives $\eta = 1$, which gives us no confidence in our selected classifier. Frequently η is a requirement, so s is computed once η and ϵ are known. For example, when $\eta \leq 0.0342$, using simplexes, we must have a sample size $s \geq 82,000$.

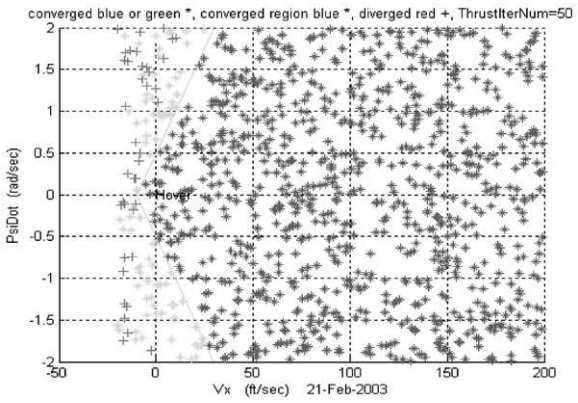

Figure 3. OAV Controller Trim Convergence

We previously verified the controller specification using parallel axis simplexes in $\mathcal{R}^4$ which has VC dimension 8. In that analysis $\epsilon = 0.05$ and $s = 34,000$. An application of Equation 2 gives $\eta = 0.0395$.

Choosing a hypothesis space with smaller VC dimension always provides greater confidence for a fixed sample size s and fixed accuracy ϵ. Often, hypotheses with increased VC dimension can be found to better fit the data. The cost is an increased number of samples, and also the possibility of overfitting the data.

4 Some Other Verification Techniques

When an application's design is sufficiently complex, a tractable mathematical analysis for its required properties becomes infeasible and simulation becomes the tried-and-true verification method. Virtually all control engineers are familiar with Monte Carlo techniques as a tool to "test" their designs, yet the question "how much testing is needed" rarely seems to be asked, let alone answered. There does seem to be an acknowledgement that when not enough simulations are run, the results might not be reliable.

The time and cost for systems integration and field testing of any non-trivial vehicle are well known to dwarf the sum total of all design verification activities. More times than not, the simulation test environment is rarely the same as the target environment. Our statistical verification approach not only applies on real world data and targets implementation code, but has the potential to work well in those settings, factoring in hardware effects (*e.g.* caching, pipelining, out-of-order execution, faults) that contribute to execution time variability. Our study used iteration count and an assumed WCET per iteration, but our method could be used with actual controller execution time measurements, had they been available. This is in contrast to traditional formal methods approaches (*e.g.* model checking), where only a model can be verified.

An alternative to estimating worst case values is to use extremal statistics, developed specifically to describe the tails (*vs.* the mid-range) of a distribution. Burns et. al [3] report some preliminary success using an extremal distribution to predict WCETs. More generally, they propose a probabilistic framework for schedulability analysis. Within their framework, they propose modeling response times as random variables because schedulability analyses for systems using only WCETs is far too pessimistic. Within this framework they are exploring the use of copulas (a sort of normalized joint distribution representation) to capture the dependencies of the tasks' random execution times when summing them. Because copulas are partially ordered, there is a "worst case" copula that could be used to provide a pessimistic estimate of the sum. If the bound is not overly pessimistic, this might be useful in practice.

5. Some Future Directions

The assumption of an independent and identically distributed *iid* sample is common for statistical tests. In practice, assumptions of stationarity and independence are rarely met. Consider the task execution times that are dependent on the time varying hardware configuration affected by applications' uses of caches, schedulers, etc. The Law of Large Numbers (LLN) over an abstract parameter space Λ lies at the basis of SLT, and we have been assuming *iid* samples. Many variants of the LLN exist, including the use of dependent and time varying random samples. Understanding of how application requirements and implementations map to the underlying probability space to appropriately apply theoretical variations can be challenging.

The number of samples needed for moderately high levels of assurance is large (10's to 100's of thousands). This is because the SLT forumlation makes almost no distributional assumptions on F_Z - in essence, a most pessimistic distribution is assumed. Sharper bounds on required sample size can and have been found when additional distributional assumptions on F_Z are made (*e.g.* when data labels are known to be noise-free). Nonetheless, estimating only the properties of interest may require fewer samples than first estimating a joint distribution which is known to be combinatoric in the number of variables.

The risk function is an average, and the convergence of the empirical risk to the true risk is based on the LLN. This raises questions about the suitability of SLT for predicting extremal (*i.e.* worst case) statistics. Estimating fixed percentiles (*e.g.* 98^{th}) may be viable. By design, many worst case estimates are not critical. For example, most control algorithms can sustain an occaisonal lost input value. We advocate the codesign principle of coordinated development of control algorithms with fault and resource management infrastructures as a viable mechanism for reducing the significance of the requirement to "know" extremal values.

Reporting real-valued losses may benefit the further design of critical tests as well as provide a more realistic risk evaluation. Under the assumption that "*not all faults are equally bad*", we may want to recognize the really severe faults and assign greater loss to them.

6 Acknowledgements

Many thanks to collaborators Mike Elgersma, Vu Ha and Tariq Samad in the OAV flight envelope analysis and verification. Thanks also to Vladimir Cherkassky of University of Minnesota for helpful discussions on VC Theory.

References

[1] P. Binns, "An Introduction to Aspects of Statistical Learning Theory," Honeywell Laboratories Technical Report AES-R03-001, July 2003

[2] P. Binns, M. Elgersma, S. Ganguli, V. Ha, and T. Samad, "Statistical Verifi cation of Two Non-linear Real-Time UAV Controllers," *Proceedings of the Tenth IEEE Real-Time and Embedded Technology and Applications Symposium* (RTAS04), May 2004

[3] Alan Burns, Guillem Bernat, Ian Broster, "A Probabilistic Framework for Schedulability Analysis," *The ACM Third International Conference on Embedded Software* (EMSOFT), October 2003

[4] V. N. Vapnik and A. Chervonenkis, "On the Uniform Convergence of Relative Frequencies of Events to their Probabilities," *Theory of Probability and its Applications,* 1971

[5] V. N. Vapnik, "Statistical Learning Theory," John Wiley & Sons, 1998