Proceedings - 5th IMA Conference on Mathematics in Defence
23 November 2017, The Royal Military Academy Sandhurst, Camberley, UK

CASSANDRA: A Tool for Mission System Performance Prediction

By Dr David R. Parker

BAE Systems Military Air and Information, Warton Aerodrome, PR4 1AX, UK.

Abstract

A military aircraft mission system comprises: one or more sensors, sensor data processing and sensor fusion algorithms. These are used to provide tactical information to the weapon system and its operator, in a timely manner. Development of a fast jet mission system has traditionally been a challenging task to complete to time and cost. Part of the challenge arises because overall system performance only becomes apparent when the system components are integrated to form the full system, late in the development process, when changes to the system architecture, or component specifications, are expensive and time-consuming to implement.

BAE Systems are acting to reduce the risk associated with mission system development by creating a performance modelling tool that allows performance models, for individual system components, to be combined to synthesise overall mission system performance early in the design life-cycle. This tool allows system requirements to be validated, it supports trade-off studies between competing system design options and allows the sensitivity of system performance to component specification tolerance to be quantified.

The philosophy of mission system performance model development is that the process of sensing the environment results in erroneous and uncertain measurements that propagate through the sensor data and fusion processing. These result in erroneous and uncertain reports to the weapon system and operator. Characterisation of the gross and minor sensor measurement errors and propagation of this uncertainty through the data and fusion algorithmic processes, either analytically or using Monte Carlo techniques, allows the system output error to be characterised and quantified.

The BAE Systems mission system performance modelling tool is named Cassandra, after the heroine of Greek mythology who was granted the power to predict the future. The current state of development of the Cassandra tool is described in the paper and examples of its application to topical mission system design problems are presented. Finally, the proposed next stages of Cassandra development are discussed.

1. Introduction

A major difficulty in the development of a military aircraft mission system is that overall system performance does not become apparent until the constituent system components are assembled to form the system and it is subject to flight trials, in a representative environment. At this stage of development, any change to the system architecture, or to system components; to address performance problems, is extremely expensive. Further, significant flight test effort, at significant cost, is often required to isolate problems through large scale empirical investigation.

Future military aircraft mission systems are likely to be increasingly complex: to provide a semi-autonomous operating capability for reduced aircrew workload in manned operations, or to provide an autonomous operating capability for unmanned aircraft operating under strict communications constraints. The increased complexity of future aircraft mission systems will require an enhanced development approach that goes beyond component integration, system test and iteration over a number of cycles until performance is satisfactory. This problem was

Proceedings - 5th IMA Conference on Mathematics in Defence
23 November 2017, The Royal Military Academy Sandhurst, Camberley, UK

recognised a number of years ago, by the image processing and computer vision communities and the proposed solution involved creation of forward models for system performance. Such models allow requirements validation and system optimisation, without exhaustive experimentation; see Ratches et al (1997) and Thacker et al (1998) for example. BAE Systems MAI are applying a similar forward model based approach, coupled with rigorous application of a standard engineering process, to manage system performance throughout the design life-cycle. Hence, the BAE Systems Cassandra mission system performance modelling tool has been developed to:

- Allow the predictions from disparate component performance models to be combined to provide a prediction of overall system performance.
- Enforce a strict application programme interface (API) to allow re-use of existing component models in multiple system models.
- Encourage development of performance models for system components previously thought to be unsuitable for performance prediction: automatic target recognition algorithms for instance.
- Enforce formal documentation and testing of system component models to encourage re-use and to provide a repository for retention of corporate knowledge.
- Support the formal mission systems engineering design process.

The background to the military aircraft mission system design problem is described in section 2 and the Cassandra tool and its contribution to addressing the mission system design problem is discussed in section 3. Three example applications are described in section 4 and conclusions regarding the work to date are discussed in section 5 together with a summary of planned future work.

2. Background

The mission system in a military aircraft is responsible for providing tactical information to the aircrew or remote operator, in order to allow optimum achievement of the mission objectives. A generic mission system comprises a number of sensors and associated detection, track processing and classification functions: to detect objects of interest and to maintain an estimate of their current position, velocity and identity. A mission system also includes a fusion stage: to combine reports from multiple sensors, to form a single “best” set of tactical information from the available sensor data.

Mission system performance in a particular scenario, in terms of timeliness, validity and accuracy of the tactical information it provides, is only optimised if the performance of each component is optimised. The development process illustrated in Figure 1 exploits system forward performance models, in order to manage and optimise overall system performance throughout design and development. In this process, a series of increasingly sophisticated system performance models are developed, in parallel with the main system development programme, to support: requirements capture and validation, design trade-offs, sensitivity analysis, component specification, sub-system and system testing. Whenever possible, component performance models are re-used from previous projects, to exploit prior model validation. As the project progresses to detailed design, supplier performance models are used, when available, to leverage supplier domain knowledge.

BAE Systems MAI have invested significant effort in developing performance models for complex algorithmic processes over a number of years, see Noonan & Orford (1996), Parker (2012) and Willis (2015) for descriptions of performance models for track fusion processing, detection processing in electro-optic imagery and target de-lineation in SAR imagery.

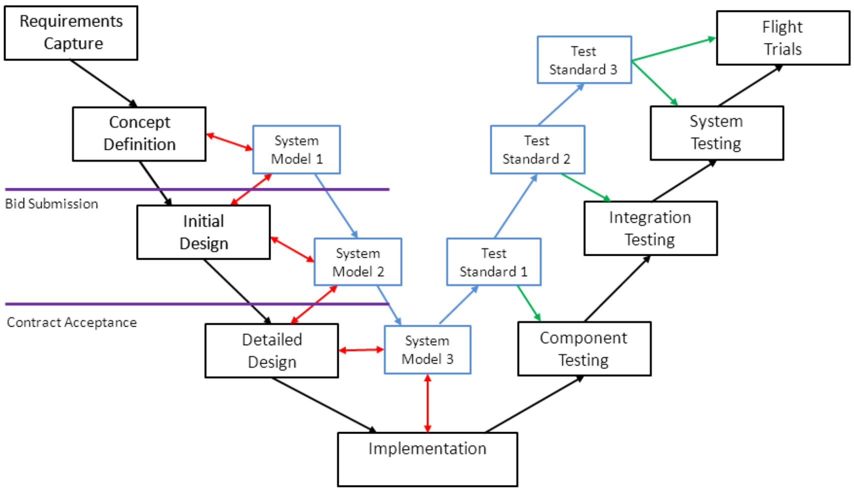


Figure 1 : Generic system design process

3. Cassandra

The Cassandra framework is written in Matlab and uses Simulink to provide a friendly graphical user interface. The component models are largely written in Matlab; however, models produced in other languages can be accommodated by virtue of Matlab MEX functionality. Model requirements are formally documented, as is all verification and validation testing, thus providing robust provenance for all models. Each component module is divided into code: written to a proprietary standard, that represents the physics of the module and text files in which configuration parameters are specified to represent a particular sub-system. The combination of code and configuration files allows performance prediction for a specific example of a module. Thus the module code can be re-used on multiple projects and only the configuration files need to be classified to protect national security and or project confidentiality.

Easy integration of disparate component models is achieved by Cassandra, through application of a clear interface standard and rigorous error checking requirements imposed on the component models. Information is passed between component performance models via the Cassandra system databus, allowing any Cassandra compliant module to be connected to any other. The data checking on module input and output ensures that no module executes without a valid set of input parameters and that the output parameters generated are within pre-defined acceptable ranges, thus helping model de-bugging and avoidance of erroneous input and output. The friendly graphical user interface supports non-specialist use of the tool.

The current module library includes over thirty modules for: infra-red search and track (IRST) and synthetic aperture radar (SAR) sensors, detection, tracking and classification algorithms for various sensing modalities and modules to predict target geo-location accuracy.

4. Example Applications

Three examples are presented below to demonstrate the application of the Cassandra system to real world problems and to illustrate some of the mathematical techniques used.

4.1. Synthetic Aperture Radar Target Recognition Performance

An advanced SAR system comprises not only the sensor head and low level image formation processing, but, also higher level detection, classification and recognition algorithms; to allow

Proceedings - 5th IMA Conference on Mathematics in Defence
23 November 2017, The Royal Military Academy Sandhurst, Camberley, UK

the system to present a user with candidate targets for confirmation, avoiding the need for exhaustive manual inspection of all images.

In a SAR system, image formation is a coherent process, over many pulses, which creates an image corrupted by distinctive “speckle” noise. Whilst SAR images are superficially similar to monochrome optical images, their characteristics are in fact very different, requiring significantly different algorithmic approaches for object detection and recognition. A major feature of detection and recognition processing in SAR imagery is that assumptions of Gaussian image variation: used routinely when processing optical images, are poorly suited to SAR imagery. Instead SAR image variation is usually represented by a distribution from the gamma family: the K-distribution being particularly popular. This distribution allows representation of the average terrain radar cross section (RCS): using the distribution’s μ parameter, and the characteristic “long-tailed” SAR image speckle noise: using the distribution’s v parameter. The distribution’s third parameter L represents pixel averaging in the image formation process. Spatial correlation of the random field, with correlation length τ, is introduced to represent image structure. See Oliver & Quegan (2004) for further details.

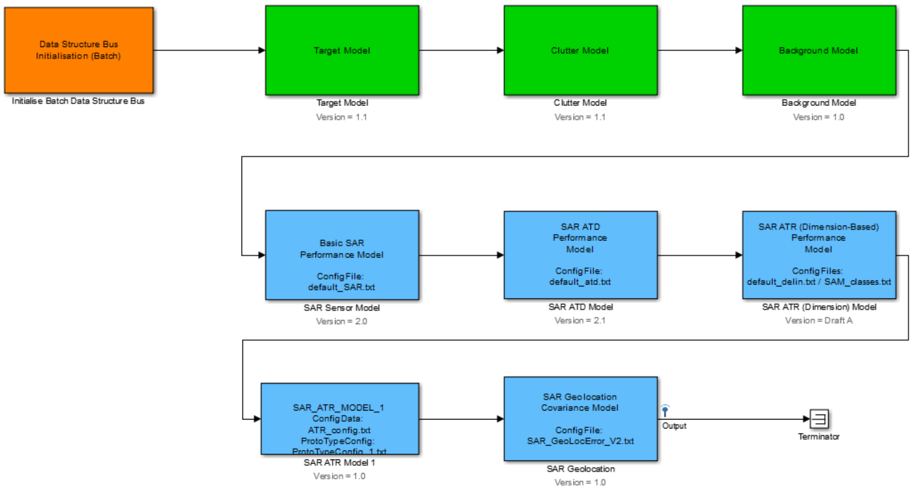


Figure 2 : SAR Functional Chain Performance Model

Key questions for system designers, integrators and operators are:

- What are the optimum radar operating parameter values to maximise target detection, classification and recognition performance for the target of interest in a specified operating environment? In particular what SAR image resolution is required to provide acceptable performance?
- How does system performance vary as a function of perturbations in the characteristics of the operating environment? In particular, how does variation in object contrast effect performance?
- What false positive rate can be expected for a specified true positive rate and how sensitive is this parameter to environmental variation?

In order to address these design questions, the model shown in Figure 2 has been developed. This model includes a number of components to predict the performance of the detection, classification and recognition elements in the processing chain and hence the overall system performance. The recognition component model, developed at BAE Systems AI Labs Great Baddow, assumes a template matching algorithm and predicts performance in terms of the true and false positive probabilities, given a definition of: the imaged object, the object and background image statistics and the target library.

The assumed recognition algorithm compares a region of the SAR image: an image chip, detected by a previous stage of processing, with a library of target templates at all possible orientations. The identity of the template that achieves the best comparison is assigned to the detected SAR image object. A minimum level of comparison is specified to allow an "unknown" classification decision. Comparison between the SAR image chip and the target template is typically undertaken using the well-known Pearson correlation metric, equation (4.1.1), though other metrics can easily be implemented for comparison.

$$r = \frac{\Sigma\left[\left(I_{(x,y)} - \bar{I}\right)\left(T_{(x,y)} - \bar{T}\right)\right]}{\sqrt{\Sigma\left(I_{(x,y)} - \bar{I}\right)^2}\sqrt{\left(\Sigma\left(T_{(x,y)} - \bar{T}\right)^2\right)}} \tag{4.1.1}$$

Where:

- $I_{(x,y)}$ is the SAR image chip value at position (x, y).
- $T_{(x,y)}$ is the target template value at position (x, y).
- $\bar{I}$ and $\bar{T}$ are the mean values of $I_{(x,y)}$ and $T_{(x,y)}$ respectively.

Target templates are assumed to be generated off-line, by SAR imaging of targets of interest under controlled conditions on a test range.

The recognition model is thus required to predict the central value and variation of the correlation metric for a specified object of interest, target template library and image grey level statistics. Analytic techniques are available to predict the distribution of correlation metric values in the very specific case of Gaussian input image variation and a linear correlation function: zero mean cross correlation for instance: see Oakley (1998) for a detailed discussion. However, for the SAR ATR problem, the correlation metric is not a linear function of its inputs and the input image variation is highly non-Gaussian. In this situation, a Monte Carlo simulation is used. The Monte Carlo simulation relies on repeated synthesis of a target or confuser image chip and calculation of the correlation metric for the chip and the target library. The maximum correlation metric allows determination of object identity for each run. This allows the distribution of maximum correlation metric values to be derived and the algorithm true positive and false positive probabilities to be predicted over a large number of repeat runs.

A target or confuser object is represented as a simple piecewise constant pixel array, with appropriate object contrast, modified by inclusion of suitably distributed and correlated random fields for target and background. The fidelity of the Monte Carlo simulation is thus heavily reliant on the ability to generate correlated and un-correlated random fields, with the appropriate statistical properties, to allow the variability of target and clutter background images to be represented with the necessary fidelity.

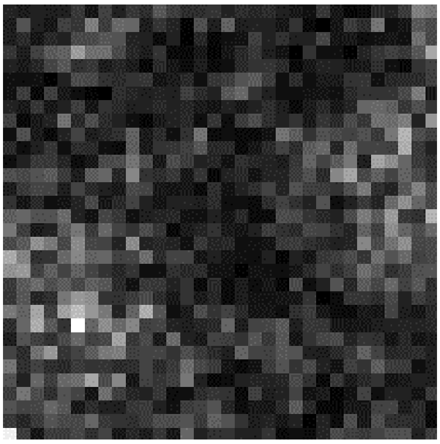
Whilst random number generators for un-correlated exponential and gamma distributed variates are readily available, generation of correlated K-distributed variates relies on the following proprietary approach. A K-distributed random variate Z_K is the product of 2 independent gamma distributed random variates: Y_{RCS} representing terrain radar cross section fluctuation and $\mathrm{Y}_{\mathrm{Speckle}}$ representing speckle variation. Hence, the spatial auto-correlation ρ_K of Z_K , is related to the spatial auto-correlation, ρ_{RCS} and $\rho_{Speckle}$, of Y_{RCS} and $\mathrm{Y}_{\mathrm{Speckle}}$:

$$\rho_K = f\left(\rho_{RCS}, \rho_{speckle}\right) \tag{4.1.2}$$

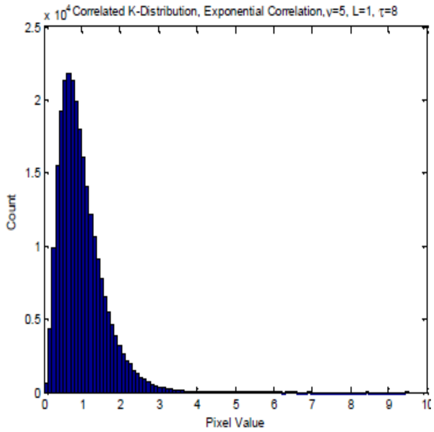
Following Raghavan (1991) and Marier (1995) the function $f\left(\rho_{RCS}, \rho_{speckle}\right)$ can be derived for K distributed intensity. Thus, given a requirement for ρ_K and a measured or assumed $\rho_{speckle}$, ρ_{RCS} can be determined, which together with the number of looks, L, shape parameter ν and mean μ allows Y_{RCS} and $\mathrm{Y}_{\mathrm{Speckle}}$ to be generated, using the method of Armstrong & Griffiths (1991). Z_K is then generated from:

$$Z_K = Y_{RCS} Y_{Speckle} \tag{4.1.3}$$

An example correlated K distributed random field and the associated histogram are shown in Figure 3 below:



Reproduced by kind permission BAE Systems AI Labs



Reproduced by kind permission BAE Systems AI Labs

Figure 3 : Correlated K-distributed pixels with shape *v* = 5, *L* = 1 and exponential correlation τ = 8 (left hand image). Corresponding histogram (right hand plot).

4.2. Tracker Performance

Kalman filter based tracking systems are widely used in military aircraft mission systems. A key design issue is to understand the relationship between individual sensor performance and overall system track quality; so that the sensors can be specified correctly to ensure that overall system performance is acceptable. In particular, given a sensor's detection performance: in terms of a detection probability and position/velocity covariance, how does this effect system track quality in terms of:

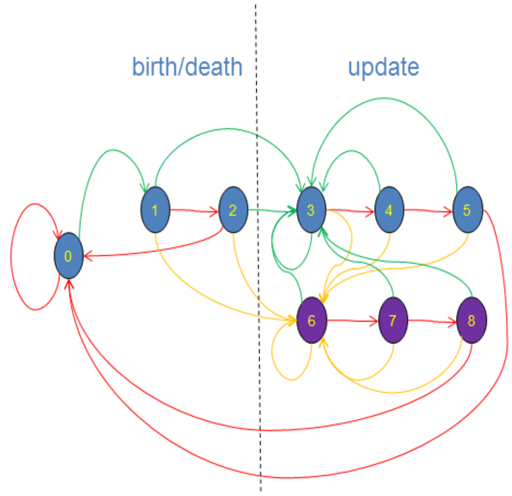
- Probability of true track declaration?
- Probability of track declaration on clutter features?
- Probability of track update without corruption and related expected track life?
- Probability of track corruption?

A schematic of the Markov random chain used by a tracker performance model is shown in Figure 4. Eight track acquisition states are represented in the model, together with the associated transitions between states. The transitions between states happen when detection, of target or clutter, occurs or if no detection occurs within a specified number of time steps. A state transition matrix ***M*** is determined from: the transition rules illustrated in Figure 4 and the sensor detection performance and related position/velocity covariance, for specified target and clutter objects, determined by an upstream sensor performance model. The state probability at time k is then estimated from that at time k-1 by:

$$\mathbf{s}_k = \mathbf{M}\mathbf{s}_{k-1} \tag{4.2.1}$$

System track performance is determined from the state probabilities at each time step as follows:

$$\Pr(\text{Clean track}) = P(3) + P(4) + P(5) \tag{4.2.2}$$
$$\Pr(\text{Clutter affected track}) = P(6) + P(7)+ P(8) \tag{4.2.3}$$
$$\Pr(\text{Any sort of track}) = P(3) + P(4) + P(5) + P(6) + P(7) + P(8) \tag{4.2.4}$$
$$\Pr(\text{No track}) = P(0) + P(1) + P(2) \tag{4.2.5}$$



States:
- 0 --- No track.
- 1 --- New tentative track.
- 2 --- Tentative track awaiting confirming detection.
- 3 or 6 --- Converged track.
- 4 or 7 --- Converged track – not updated on the last sensor cycle.
- 5 or 8 --- Converged track – not updated on the last two sensor cycles.
- Purple states (6 to 8) are clutter affected tracks.

Transitions:
Green --- detection. Red --- no detection Amber --- clutter detection

Figure 4 : Track Acquisition Model Markov Random Chain Schematic

A number of versions of the tracker model summarised above have been used on multiple aircraft programmes in recent years, most notably in support of recent developments of the Typhoon PIRATE infra-red search and track (IRST) sensor.

4.3. Classifier Performance

Conceptually, classifier development relies on selection of a feature space in which feature vectors corresponding to different object classes are well separated. Once a suitable feature space has been selected, a classification rule can be defined that separates the different classes. Unfortunately, in real target classification applications the optimum feature space is rarely easy to identify and the feature vectors corresponding to different classes are difficult to separate. Hence the development, in recent years, of deep learning techniques to autonomously select the optimum feature space and classification rule, through extensive search over large numbers of training data items. Despite successful application of these techniques in many domains, design of a military aircraft mission system including classifier technology still requires use of forward performance models for requirement validation, sensor specification and design trade-off. A number of techniques for classifier performance prediction, avoiding Monte Carlo simulation, are being investigated, one of which is described below.

Assume an N-dimensional feature space in which feature vectors corresponding to different object classes occupy separate points in the space. Measurement noise and intra-class variation result in regions of uncertainty around the central point for each feature vector. For ease of representation, these regions of uncertainty are assumed to be ellipsoidal and adequately represented by covariance matrices. If two adjacent object classes have over-lapping covariance matrices, as shown in Figure 5 for a 2-dimensional feature space, then there is a fundamental classification performance limit imposed by the choice of feature space, the measurement noise, intra class variation and inter class similarity. This limit can be determined from the geometric separation of the central points for the two class feature vectors and the covariance for each feature vector. Assuming that system and measurement noise are the dominant sources of feature vector variation, then given a feature vector definition, the feature vector covariance can be estimated using a first order perturbation approach, if the derivatives in (4.3.1) can be determined.

$$\Sigma_{Fvec} = \frac{d\overline{Fvec}}{d\overline{P}} \Sigma_{\overline{P}} \frac{d\overline{Fvec}}{d\overline{P}}^{T} \tag{4.3.1}$$

Where:

- $\overline{Fvec}$ is the feature vector.
- $\overline{P}$ is the vector of independent parameters whose variation causes variation of the feature vector values.
- $\Sigma_{\overline{P}}$ is the covariance matrix for $\overline{P}$.
- $\frac{d\overline{Fvec}}{d\overline{P}}$ is the matrix of first derivatives of $\overline{Fvec}$ with respect to $\overline{P}$.

The overlap between adjacent multi-normal variate class feature vectors, with centre points μ_p and μ_q and covariance matrices Σ_p and Σ_q, can then be estimated using the Bhattacharyya Coefficient D_C derived from the multi-variate Bhattacharyya Distance D_B, Bhattacharyya (1943).

$$D_C = e^{-D_B} \tag{4.3.2}$$

$$D_B = \frac{1}{8}(\mu_p - \mu_q)^T \Sigma^{-1} (\mu_p - \mu_q) + \frac{1}{2} ln\left(\frac{|\Sigma|}{\sqrt{|\Sigma_p| + |\Sigma_q|}}\right) \tag{4.3.3}$$

Where:

$$\Sigma = \frac{\Sigma_p + \Sigma_q}{2}$$

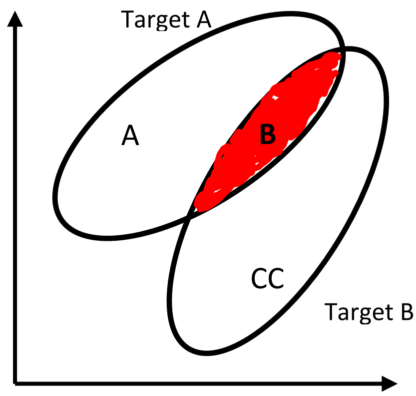


Figure 5 : Example Feature Vector Covariance

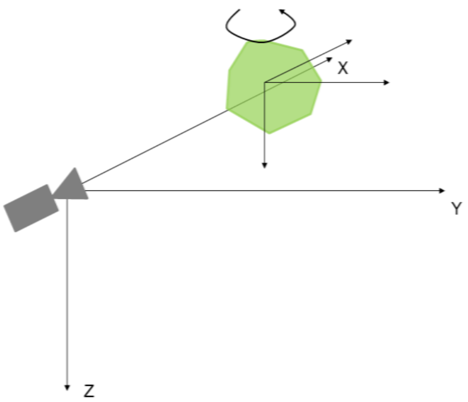


Figure 6 : Geometry for image based classifier example.

The percentage overlap between the two covariance matrices corresponds directly to the maximum limit on the mis-classification probability P_{FP}. This analysis can be repeated for all class pairs to determine the limits on mis-classification probability for each pair. For 2 classes, the percentage of each feature vector uncertainty region with no overlap corresponds to the minimum limit on true positive probability P_{TP} for each class. Thus, assuming Σ_{Fvec} can be derived, an analytic approach can be used to make an initial prediction of the limits on classifier performance, without the expense of large scale trials or Monte Carlo simulation.

Consider the situation in Figure 6 for example. An object is at (X_o, 0, 0) in the axis set of an imaging sensor. The object can rotate about its local Z-axis and the image captured by the sensor is corrupted by electronic noise. A feature vector comprising contrast, aspect ratio and

Proceedings - 5th IMA Conference on Mathematics in Defence
23 November 2017, The Royal Military Academy Sandhurst, Camberley, UK

object horizontal image size is used to discriminate between different objects imaged by the sensor. Feature vector variation occurs due to image noise and perturbation of object orientation. The above performance prediction process is applied to determine the limits on classifier performance, given the feature vector and sources of uncertainty. The performance limits shown in Figure 7 were generated for the following conditions:

- Clutter object is 18 pixels wide, 3 pixels high. Contrast = 38 wrt background.
- Target height = 3 pixels. Contrast = 40 wrt background. Target image width varies.
- Feature vector is contrast, aspect ratio and x-dimension.
- Image noise = 5 grey levels 1σ. Object heading variation 1^{o} 1σ.

The results in Figure 7 show that, when target and clutter sizes are similar, the minimum limit on P_{TP} approaches 0, whilst the maximum limit on P_{FP} approaches 1: indicating that the algorithm cannot reliably discriminate between the two objects for this condition. In other conditions, the minimum limit on P_{TP} approaches 1 and the maximum limit on P_{FP} approaches 0, indicating that satisfactory performance can be expected.

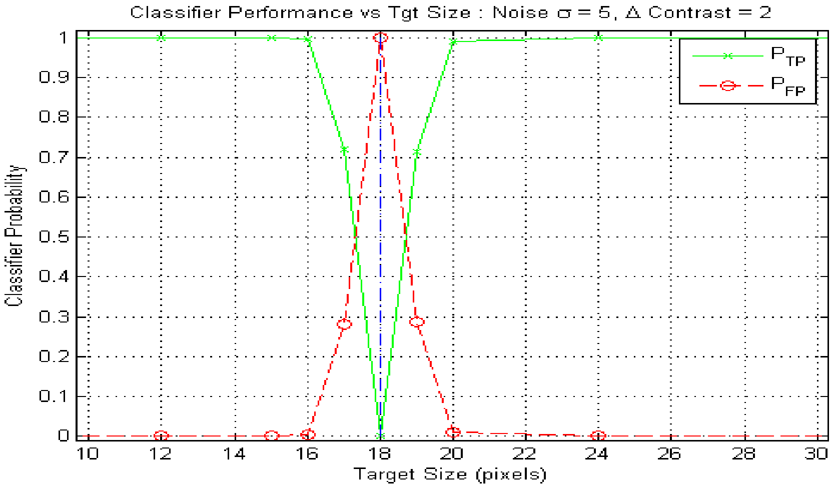


Figure 7 : Predicted limits on classification probability as a function of target size.

5. Conclusions

A key element of the military aircraft mission system development problem has been identified and the work undertaken by BAE Systems MAI to address this problem has been described. Development and integration of forward models for mission system component performance, in order to synthesise overall mission system performance, is a key enabler for system requirements verification, component specification, design trade-off studies and for bench-marking sub-system and system testing. The Cassandra tool has been developed to support these mission system development activities by facilitating multiple component model integration, allowing re-use of component models across many programmes and enabling sharing of models with partner companies, customers and suppliers.

A number of example applications have been described that include use of Monte Carlo simulation, Markov random chains and first order perturbation. These methods are used to propagate input measurement or environmental uncertainty through the mission system processing chain, to predict the uncertainty in the mission system output information and hence system performance. Thus the effect of varying sensor specification, operating conditions and algorithmic approaches on system performance can be investigated.

Work is currently underway to validate existing models: both individual component models and full system models, by comparison with actual system performance data generated during aircraft flight trials. Additional model development is also being undertaken to integrate

legacy models into the Cassandra framework, to enhance existing models and to develop models for sensing and signal processing modes that are not currently addressed. Future work will also aim to reduce the time penalty associated with Monte Carlo simulation.
BAE Systems MAI are actively seeking customer, industrial and academic partners for component performance model development.

6. Acknowledgements

The Cassandra performance modelling tool has required many man-hours of effort, from a significant number of people over a number of years, to achieve its current state of maturity. The author wishes to acknowledge the contribution of the following current and former BAE Systems MAI employees:

Andy Sheard	Simon Smith	Gary Bishop	Dr Colin Noonan.
Chris Leavitt	Brian Gibson	Adrian Blagg	Harry Finn

The author would also like to acknowledge the help and assistance of Dr Chris Willis and his colleagues at BAE Systems AI Labs for their work developing SAR sensor and associated detection and recognition performance models.

REFERENCES

Armstrong, B. C., Griffiths, H. D.; "**Modelling spatially correlated K-distributed clutter**"; Electronics Letters; Vol. 27; No. 15; pp. 1355-1356, 1991.

Bhattacharyya A., "**On a measure of divergence between two statistical populations defined by probability distributions**" *Bull. Calcutta Math. Soc.* , **35** (1943) pp. 99–109

Marier, L. J.; "**Correlated K-Distributed Clutter Generation for Radar Detection and Track**"; IEEE Transactions on Aerospace and Electronic Systems; Vol. 31, No. 2; pp. 568-580, April 1995.

C.A. Noonan, K.J. Orford, "**Entropy Measures of Multi-Sensor Fusion Performance**", IEE Colloquium on Target Tracking and Data Fusion (Digest No:1996/253) pp 15/1-15/5, 1996.

J. P. Oakley, "**Statistical properties of the local extrema of two-dimensional Gaussian Random Fields**". *IEEE Trans Signal Processing*, *46*(1), 130-140. (1998)

C. J. Oliver, S. Quegan, "**Understanding Synthetic Aperture Radar Images**", SciTech Publishing, Inc., 2004.

David R. Parker, "**Model, Predict and Test: Towards a Rigorous Process For Acquisition of Object Detection and Recognition Algorithms For Un-manned Air Vehicle Applications**", In Transactions IET Conference on Image Processing (IPR 2012), July 2012.

Raghavan, R. S.; "**A model for spatially correlated radar clutter**"; IEEE Transactions on Aerospace and Electronic Systems; Vol. 27; No. 2; pp. 268-275; March 1991.

James A. Ratches, C.P. Walters, Rudolf G. Buser and B.D. Guenther, "**Aided and Automatic Target Recognition Based upon Sensory Inputs from Image Forming Systems"**, *IEEE Pattern Analysis and Machine Intelligence* Vol 19, No 9, pp1004-1019, September 1997.

N.A. Thacker, P. Courtney, A. Clark, **"Algorithmic Modelling for Performance Evaluation"**, Tina Memo No 1996-001, 1st December 1998.

Chris J. Willis, "**Understanding target delineation using simple probabilistic modelling",** In proceedings SPIE Remote Sensing Volume 9643, Image and Signal Processing for Remote Sensing XXI, 2015.